Resource-efficient quantum cryptography

Cameron Foreman

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

June 26, 2025

I, Cameron Foreman, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Quantum cryptography leverages quantum mechanics to perform cryptographic tasks and can, in principle, offer security guarantees impossible with classical methods. However, a significant gap exists between current protocols and those required for the practical realisation of this security advantage. This thesis presents several results aimed at bridging this gap, by reducing the required amount and quality of initial resources and improving the computational efficiency of key processing steps – enabling secure resource-efficient quantum cryptography protocols that can be implemented in the near-term.

Part I focuses on randomness extractors, classical algorithms that distil uniform and secret randomness from partially random sources, which are essential for quantum cryptography. We introduce seeded randomness extractors with improved finite-size performance and present several two-source extractors that reduce the quality requirements of the seed. For all extractors, we provide efficient, information-theoretically secure, implementations utilising the convolution theorem to achieve quasi-linear computation time where applicable. Next, we introduce a class of extractors specifically designed for quantum protocols which use input sources that violate a Bell inequality. We show that these extractors can be implemented deterministically and provide several efficient and explicit constructions by exploiting a connection with error-correcting codes.

Part II explores randomness amplification and privatisation, a quantum process that converts weak (non-uniform and public) randomness into uniform and private randomness. We present a device-independent protocol for randomness amplification and privatisation, optimised for practical use, with generation rates linear in the

quantum device's repetition rate and quasi-linear computation time classical post-processing. We then demonstrate that this protocol can be experimentally realised in a semi-device-independent manner using current hardware, implementing it on several different quantum computers.

Impact Statement

The development of quantum technologies is poised to revolutionise numerous fields, such as cryptography and computing, offering substantial societal and commercial benefits. Quantum cryptography, in particular, has made remarkable advancements over recent years, progressing from theoretical models to proof-of-concept experiments to early commercial products. This technology provides higher levels of security for sensitive data than classical methods by utilising the principles of quantum mechanics.

Despite these exciting developments, significant barriers remain to the widespread adoption of quantum cryptographic systems. Key among these is the challenge of ensuring that these systems are robust and scalable enough for practical, real-world applications. To be useful outside of controlled lab settings, quantum cryptography protocols must maintain the theoretical security advantages even in environments where variables are far less controllable. Current protocols and systems are limited by the need for high-quality quantum devices, low-loss communication channels, and stable (often purpose built) infrastructure, all of which are expensive, complex, and technically challenging.

This thesis seeks to address these challenges by developing new methods and protocols for quantum cryptographic systems that maintain high security standards while also improving resource efficiency. In particular, the presented techniques aim to lower demands on quantum device quality, reduce computational requirements, and minimise the initial resource needs. These advancements bring us closer to realising quantum-secured communication, the ultimate goal of quantum cryptography.

The findings presented in this thesis are expected to be of considerable value

to researchers and engineers working in the quantum cryptography field, offering new insights and tools to enhance system robustness and scalability. Our results help understand the fundamental limits in resource requirements for quantum cryptography and provide open-to-academia tools such as Cryptomite, which has facilitated (and will continue to facilitate) important new contributions, with over 60000 downloads. From a commercial perspective, our results have contributed to the development of Quantum Origin, an advanced cryptography platform that utilises quantum mechanics to verify randomness, which is already being used by market leaders such as Honeywell and Thales. Quantum Origin exemplifies the transition from theory to application, showcasing quantum cryptography's potential to protect critical data across sectors such as finance and government.

While the primary focus of this thesis is on quantum cryptographic protocols, the developed techniques have applications across a wide range of (quantum) technologies, e.g. (quantum) error-correction processes and (quantum) algorithm development.

Acknowledgements

I am incredibly fortunate to be surrounded by many wonderful people who inspire and support me. To all of them, I am eternally grateful and it is a privilege to acknowledge those who have accompanied me on this journey, though words can only begin to express my appreciation.

My deepest gratitude goes to my supervisor, Prof. Lluís Masanes, whose mentorship and unwavering support has been invaluable. His expertise in mathematics and physics has significantly deepened my knowledge, while his guidance and encouragement has been instrumental in my development as an academic. I am grateful for his role in making my PhD journey fulfilling by fostering a nurturing environment where I could thrive.

In a similar vein, I extend my heartfelt thanks to Dr. Florian Curchod, my mentor, line manager, and friend. His encouragement has been pivotal to this thesis – initiating my research journey and igniting my passion for tackling complex, thought-provoking questions. His continued support has been a constant source of motivation throughout this journey and I will forever be grateful.

I am deeply thankful to my past and present colleagues at Quantinuum – Sherilyn Wright, Dr. Henry Semenenko, Dr. Hailey Jee, Dr. Karan Khathuria, Dr. Mafalda Almeida, Dr. Erik Woodhead, Dr. Kevin Milner, Dr. Victoria Wright, Prof. Matty Hoban, Dr. Kieran Wilkinson, Richie Yeung and many more – as well as the wonderful group at UCL, including Tom Holden-Dye, Anastasia Moroz, James Purcell, Simone Lin, Prof. Toby Cubitt and Dr. Harriet Apel. Together, they have enriched this journey with pub trips, useful discussions, and many other shared experiences. I thank all those with whom I had the privilege of collaborating –

especially Lewis Wooltorton, Prof. Mario Berta, Dr. Jaskaran Singh, Dr. Kishor Bharti and Prof. Adán Cabello – whose insights have deeply enriched my knowledge. I also thank my thesis examiners, Prof. Roger Colbeck and Prof. Mehrnoosh Sadrzadeh, for their time, thoughtful feedback, and engaging discussion during my viva.[1]

Finally, above all, I owe my deepest gratitude to my family and friends. First and foremost, to my grandparents Sylvia and Philip, who sacrificed so much to enable my success; their unwavering support, from my earliest school days to the completion of this thesis, has been a constant foundation. To my fiancé, Jess, thank you for your endless patience and compassionate understanding, especially with the abnormal work schedule that I have followed in recent years. To my son, Philip, thank you for the cuddles, the gummy smiles, and for perfectly timing your arrival so that I can blame any remaining typos on sleep deprivation. To my close family, Natasha, Mike, Eli, Thea, Gillian, Kay, Sharon, thank you for the love and encouragement, and to all my friends – especially Ben, Lydia, Christina, Ryan, Lewis, Jack, Dan, Sam, Ringo, Rhys, Kiyah, Joe, Ollie, Sean and Josh – thank you for the joy and happiness you bring to my life.

[1] A special (and well-deserved!) highlight is owed to Prof. Lluís Masanes, Dr. Florian Curchod, Sherilyn Wright, Dr. Karan Khathuria, Simone Lin and Dr. Jaskaran Singh for providing useful comments on early versions of this thesis.

Publications & Pre-prints

This thesis contains results from the following publications and pre-prints:

- Foreman, C., Wright, S., Edgington, A., Berta, M. and Curchod, F.J., 2023. Practical randomness amplification and privatisation with implementations on quantum computers. *Quantum, 7, p.969, DOI: 10.22331/q-2023-03-30-969*.

- Foreman, C., Yeung, R., Edgington, A., and Curchod, F.J., 2025. Cryptomite: A versatile and user-friendly library of randomness extractors. *Quantum 9, p.1584, DOI: 10.22331/q-2025-01-08-1584*.

- Foreman, C. and Masanes, L., 2025. Seedless extractors for device-independent quantum cryptography. *Quantum 9, p.1654, DOI: 10.22331/q-2025-03-06-1654*.

- Foreman, C., Wooltorton, L., Milner, K. and Curchod, F.J., 2025. An efficient construction of Raz's two-source randomness extractor with improved parameters. *arXiv preprint: arXiv:2506.15547*.

The following related publications and pre-prints are not included in this thesis:

- Foreman, C., Yeung, R. and Curchod, F.J., 2024. Statistical testing of random number generators and their improvement using randomness extraction. *Entropy, 26(12), p.1053, DOI: 10.3390/e26121053*.

- Singh, J., Foreman, C., Bharti, K. and Cabello, A., 2024. Local contextuality-based self-tests allow for randomness expansion secure against quantum adversaries. *arXiv preprint: arXiv:2409.20082*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and preliminaries

Cryptography, the mathematical science of secret communication, has a long and distinguished history, with military and diplomatic uses dating back to before 1900 BC[1]. Today, the ability to keep information secret remains as vital as ever, if not more so, with cryptography becoming increasingly important in everyday life: as more confidential information is shared online, the need for data protection from unauthorised access grows.

Traditionally, cryptography has relied on mathematical problems that are considered computationally difficult to solve for anyone without the appropriate credentials (for example, a *secret key*). However, this approach has been historically problematic, with many examples of new cryptographic systems being developed only for cryptanalysts to later find efficient techniques to break them. Arguably the best-known example of this is the Enigma machine from World War II, which was once thought to be unbreakable in practise.

In recent times, quantum mechanics, a physical theory describing the behaviour of particles such as atoms and molecules, has proven fruitful for cryptanalysts in tackling modern cryptographic algorithms. Shor's algorithm [7], a quantum algorithm for the hidden subgroup problem, can be used to find the prime factors of an integer and has a variant that can efficiently solve the discrete logarithm problem.

[1]The ancient Egyptians were known to use unusual hieroglyphs in some inscriptions to obscure the meaning of the text. Later, the ancient Greek Spartans used *scytale's*, a wooden baton of a particular shape and size, such that any separated parties wrapping the same spiralled strip of parchment with jumbled letters around it would recover the same message.

This, in principle, can be used to help break modern cryptographic schemes based on such problems, for example, the Rivest, Shamir and Adleman (RSA) public-key cryptosystem. Although quantum computers capable of realising this attack have yet to be built, it highlights the vulnerability of current cryptographic methods and gives rise to the *harvest now, decrypt later* strategy – an attack that involves collecting encrypted data now for decryption in the future, once quantum computers become capable of breaking such encryption.

Interestingly, quantum mechanics can also offer a solution – and one that could stop this frightful game of cat and mouse between cryptographers and cryptanalysts once and for all. Quantum mechanics enables the development of *unconditionally secure* protocols whose security rely on physical principles rather than on the computational difficulty of solving a mathematical problem. In 1970, Stephen Wiesner introduced a concept he termed *quantum conjugate coding* [8], where information is encoded in a quantum state chosen from pairs of complementary quantum states, selected such that measurement in the correct basis for one state inherently disturbs the other. In 1984, building on this work, Bennett and Brassard proposed a method for secure communication, now known as BB84 [9]. This sparked a surge of research interest, with numerous quantum protocols being developed for different primitives, for example, key distribution [9, 10], random number generation [11], oblivious transfer [12] and more [13, 14, 15, 16].

Although quantum cryptography is a relatively mature quantum technology, much work remains. Further development of protocols, robust security proofs, and solutions to practical challenges are needed to make quantum cryptography viable for widespread use. A key challenge is ensuring that protocols require minimal resources and computational power for implementation. In this thesis, we address this challenge by developing new methods, tools, and protocols that enhance the resource-efficiency of quantum cryptography. Through these contributions, we aim to advance practical quantum cryptography whilst retaining a meaningful security advantage over purely classical methods.

1.1 Thesis overview

The remainder of this chapter sets the scene for the rest of this thesis. It provides relevant preliminaries and notation, in particular, describing what it means to achieve secrecy in worlds where an adversary may be classical or quantum. After this introduction, the remainder of this thesis is organised into two parts.

In Part I, we explore the theory and construction of randomness extractors. Chapter 2 introduces the key concepts and challenges involved in extracting uniform randomness from weakly random sources. Chapter 3 presents new results on randomness extractors for conditional min-entropy sources, including discussions on construction techniques, seeded extractors, and two-source extractors. Chapter 4 introduces randomness extractors for sources that violate Bell inequalities, presenting several explicit extractors and applying them to cryptographic protocols to analyse their performance.

Part II focuses on randomness amplification and privatisation. Chapter 5 introduces the topic and discusses the need for such protocols. Chapter 6 presents a device-independent protocol for randomness amplification and privatisation, with a focus on being realisable on real-world devices. Chapter 7 implements the randomness amplification protocol from Chapter 6, in a semi-device-independent manner, on various quantum computers.

1.1.1 Authorship contribution statement

All original work in this thesis was carried out under the supervision of Lluís Masanes and/or in collaboration with researchers at Quantinuum. The four papers [17, 2, 3, 18] on which this thesis is based are joint work. Some parts of the papers are used almost verbatim; however, this is only done for sections that were written by me. Any parts of the papers written by another author have been rewritten before being included. All figures and tables were made by me.

- Chapter 1 contains no new results and summarises the background material that is the context for the whole thesis.
- Chapter 2 provides the background and important theorems that are used throughout Part I. It contains contributions from [17, 2], which is joint work

Sherilyn Wright, Alec Edgington, Mario Berta and Florian Curchod and with Richie Yeung, Alec Edgington and Florian Curchod, respectively. Some of the background material in this section is unique to this thesis, e.g. Section 2.1.

- Chapter 3 contains results from [17, 2, 3], which are joint works with Sherilyn Wright, Alec Edgington, Mario Berta and Florian Curchod, with Richie Yeung, Alec Edgington and Florian Curchod and with Lewis Wooltorton, Kevin Milner and Florian Curchod, respectively.
- Chapter 4 is based largely on [18], which is joint work with Lluís Masanes, and contains some unpublished work, Section 4.6, which is also joint work with Lluís Masanes.
- Chapter 5 provides the background for the second part of the thesis, based strongly on [17], which is joint work with Sherilyn Wright, Alec Edgington, Mario Berta and Florian Curchod.
- Chapter 6 is based on the first half of [17], a joint work with Sherilyn Wright, Alec Edgington, Mario Berta, and Florian Curchod. However, many results have been significantly improved using contributions from Chapter 3.
- Chapter 7 is based on the second half of [17], which is joint work with Sherilyn Wright, Alec Edgington, Mario Berta and Florian Curchod.

1.1.2 Abbreviations and notation

The abbreviations and notation used in this thesis are summarised in Table 1.1 and Table 1.2, respectively. Any deviations from these standards are highlighted in the text.

Abbreviation	Description
AQT	Alpine Quantum Technologies
BCH	Bose–Chaudhuri–Hocquenghem
ccq-state	Classical-classical-quantum state
CHSH	Clauser–Horne–Shimony–Holt
cq-state	Classical-quantum state
CRNG	Classical random number generator
DFT	Discrete Fourier transform
EAT	Entropy accumulation theorem
FFT	Fast Fourier transform
GB	Gigabytes
IBM	International Business Machines Corporation
IID	Independent and identically distributed
MBQA	Memory based quantum attacks
Mb	Megabits
MDL	Measurement dependent locality
NIST	National Institute of Standards and Technology
NPA	Navascués, Pironio, and Acín
NTT	Number-theoretic transform
PRNG	Psuedo random number generator
QKD	Quantum key distribution
QRNG	Quantum random number generator
RNG	Random number generator
RSH	Reed-Solomon and Hadamard
SHELA	Somewhere honest entropic look ahead
SV	Santha-Vazirani
TSSR	Tomamichel-Schaffner-Smith-Renner
UIBK	University of Innsbruck
XOR	Exclusive or

Table 1.1: Abbreviations and their descriptions.

1.2 Preliminaries

In this thesis, we explore scenarios where honest users generate cryptographic resources in the presence of an adversary. Specifically, we focus on two main tasks:

- **Quantum key distribution (QKD):** A process in which two users use quantum systems to generate a shared sequence of random bits (a *secret key*) that is *private* from and *unpredictable* to the adversary.
- **Quantum random number generation (QRNG):** A process in which a single user generates random bits using quantum systems that are private from

Notation	Description
exp(1)	The exponential constant, ≈ 2.718.
$\ln(\cdot)$	Logarithm base exp(1).
$\log(\cdot)$	Logarithm base 2.
$O(\cdot)$	The asymptotic upper bound, or *order*, of a function.
$\mathrm{tr}(\cdot)$	The trace operation.
$\|\cdot\|_1$	The trace norm.
$\langle\cdot\rangle$	The expectation of an observable $\cdot$.
$\dagger$	The Hermitian adjoint.
$\circ$	The concatenation of random variables.
$\mathbb{N}$	The set of all natural numbers.
$\mathbb{Z}$	The set of all integers.
$\mathbb{Z}_q$	The set of integers modulo q.
$\mathbb{Z}_q^n$	The Cartesian product of n sets of integers modulo q.
$\mathbb{P}$	The set of all prime numbers.
$\mathbb{N}_A$	The set of all prime numbers with primitive root two.
$\mathbb{C}$	The set of all complex numbers.
$\mathbb{F}(\cdot)$	Finite field of dimension $\cdot$.
$\mathbb{GF}(\cdot)$	Galois field of dimension $\cdot$.

Table 1.2: Notations and their descriptions.

and unpredictable to the adversary.

Moreover, for the task of QRNG, we consider two distinct approaches:

- **Randomness expansion:** An approach to generating random numbers that requires the user to have a short string of essentially perfectly private and unpredictable random bits.
- **Randomness amplification (and privatisation):** An approach to generating random numbers requiring the user to have an initial string of private and *somewhat unpredictable* (i.e. not fully predictable) random bits, or, in the case of both amplification and privatisation, *public* and somewhat unpredictable bits.

In both tasks (and for QRNG, both approaches), the final resource that the user(s) generates is a classical bit string. By convention, the honest parties are called Alice and Bob (sometimes including a third party, Charlie), while the adversary is called Eve. Depending on the protocol, a single user may represent multiple parties (e.g., in QRNG) or individual parties (e.g., in QKD). While we focus on these two tasks,

the methods and tools developed are broadly applicable to other cryptographic protocols. A natural question arises;

What does it mean for the users' resource to be private and unpredictable?

A resource is *private* if it remains entirely unknown to the adversary within the considered security model and *unpredictable* if it is uniformly distributed, meaning all possible outcomes are equally likely. Together, these properties make the resource *secret*. Perfect secrecy is a theoretical ideal that cannot be guaranteed; in practice, the goal is to make the resource effectively *indistinguishable* from one that is perfectly secret.

Abstract cryptography formalises this concept using the ideal/real paradigm. The ideal scenario represents the desired resource (a perfectly secret output) while the real scenario reflects the actual resource available to the user. A resource is considered secure if it is indistinguishable from the ideal resource [19].

This indistinguishability is captured by considering a hypothetical game: an adversary is randomly given either the real or ideal resource and must guess which one they received using all available information. The trivial strategy is to guess at random, yielding a success probability of $1/2$. For a resource to be secure, the adversary's success probability must be at most $\frac{1}{2} + p_{\text{adv}}$, where p_{adv}, the distinguishing advantage, is small (i.e., $p_{\text{adv}} \ll 1/2$). Informally, this condition means that the adversary's best strategy performs only marginally better than random guessing.

A key advantage of this framework is its *universal composability* [20], allowing any real resource that satisfies the security condition can be safely substituted for the ideal resource in any context. In this thesis, we focus on *information-theoretic* security, which requires a guarantee of secrecy regardless of the adversary's computational power or system size.

1.2.1 Representation of classical and quantum systems

The protocols in this thesis produce bit strings, which can be understood as realisations of classical random variables. Classical random variables are denoted using

upper case letters, e.g., X, which take values x in a finite alphabet $\mathcal{X}$ with probability $\Pr(X = x) = p_X(x)$. Given two random variables, X and Y, over alphabets $\mathcal{X}$ and $\mathcal{Y}$ with distributions $p_X(x)$ and $p_Y(y)$, respectively, we label $X \circ Y$ the joint random variable (the concatenation of X and Y) over $\mathcal{X} \times \mathcal{Y}$ distributed as $p_{XY}(x, y)$ with marginals $p_X(x)$ and $p_Y(y)$. If X and Y are independent, their joint distribution factorises, $p_{XY}(x, y) = p_X(x)p_Y(y)$. We label the distribution of X conditioned on Y by $p_{X|Y}(x|y) = \Pr(X = x|Y = y)$. We focus on the case where the classical random variables are bit strings, e.g. $X \in \mathbb{Z}_2^n$, and $X = x$ is its realisation. Since the realisations x are bit strings, we denote the whole sequence in bold and the individual bits with a subscript, e.g. $\mathbf{x} = (x_0, \ldots, x_{n-1})$ where $x_i \in \mathbb{Z}_2$ for all $i \in \mathbb{Z}_n$. The realisations $\mathbf{x}$ will be interpreted as bit strings, sequences, or (column) vectors, depending on the context.

In many cases, the user(s) and/or the adversary can have access to quantum systems. Quantum systems are represented by density matrices ρ, which are normalised ($\mathrm{tr}(\rho) = 1$) and positive semi-definite ($\rho \geqslant 0$). We use subscripts to associate states to particular systems, for example, if a quantum adversary has system E, this can be understood as the quantum state ρ_E on Hilbert space $\mathcal{H}_E$. Superscript is used to denote conditioning, for example, an adversary's state may depend on some information h, which would then be denoted ρ_E^h. It is sometimes convenient to represent classical systems as quantum states. For example, by representing the random variable $X \in \mathbb{Z}_2^n$ with distribution $p_X(\mathbf{x})$ as the quantum state $\rho_X = \sum_{\mathbf{x} \in \mathbb{Z}_2^n} p_X(\mathbf{x}) |\mathbf{x}\rangle\langle\mathbf{x}|$ where $\{|\mathbf{x}\rangle\}_{\mathbf{x} \in \mathbb{Z}_2^n}$ denotes the standard computational basis of the Hilbert space $\mathcal{H}_X = \mathbb{C}^{2^n}$.

1.2.2 Secrecy in the presence of a classical adversary

The distinguishability of two classical random variables is quantified by the *statistical distance* (also known as the total variation distance or statistical difference).

Definition 1 (Statistical distance)**.** The statistical distance, $\mathrm{SD}(\cdot)$, between two ran-

dom variables X and Y, with $X, Y \in \mathbb{Z}_2^n$, is

$$\mathrm{SD}(X,Y) := \frac{1}{2} \sum_{\mathbf{x} \in \mathbb{Z}_2^n} |p_X(\mathbf{x}) - p_Y(\mathbf{x})| \,. \tag{1.1}$$

At first glance, one might think that a classical system $X \in \mathbb{Z}_2^n$ that is close in statistical distance to a uniform distribution, U_n, would be maximally unpredictable and thus secret. However, in cryptographic settings, the adversary may possess *side information*, which, if not independent of the resource available to the user, could assist the adversary to distinguish the real resource from an ideal one. Consequently, to guarantee secrecy, the joint system of the user and adversary must be analysed, and the closeness of the real and ideal joint systems evaluated.

For a computationally unbounded classical adversary, this side information can be modelled as a random variable E over an arbitrary but finite alphabet. Using this model, the additional predictive power is captured by the *conditional statistical distance*, where the statistical distance is conditioned on an additional random variable.

Definition 2 (Conditional statistical distance)**.** The conditional statistical distance between two random variables X and Y, with $X, Y \in \mathbb{Z}_2^n$, conditioned on information in a random variable E with (finite) alphabet $\mathcal{E}$, is

$$\mathrm{SD}(X,Y|E) := \frac{1}{2} \sum_{\mathbf{e} \in \mathcal{E}} p_E(\mathbf{e}) \left(\sum_{\mathbf{x} \in \mathbb{Z}_2^n} |p_{X|E}(\mathbf{x}|\mathbf{e}) - p_{Y|E}(\mathbf{x}|\mathbf{e})| \right) . \tag{1.2}$$

If the conditional statistical distance between the real and ideal protocol outputs, given E, is bounded by ϵ, we say that they are *ϵ-close given E*. Operationally, this means that an adversary with side information E can distinguish between them with a distinguishing advantage $p_{\mathrm{adv}} \leq \frac{\epsilon}{2}$. Therefore, we can formalise the concept of secrecy against classical adversaries by defining *classical-proof (ϵ-)perfect randomness*.

Definition 3 (Classical-proof (ϵ-)perfect randomness)**.** The random variable $Z \in \mathbb{Z}_2^m$

is (ϵ-)perfectly random, given E, for some value $\epsilon \in [0,1]$, if

$$\mathrm{SD}(Z, U_m|E) \leq \epsilon \,, \tag{1.3}$$

where U_m is the uniform distribution on $\mathbb{Z}_2^m$.

In practice, ϵ is a small fixed security parameter, for example 2^{-32}, and when $\epsilon \ll 1$, we informally call such random variables (near-)perfectly random. An output that meets this criterion is called *classically secure*.

If the adversary can access quantum states to store information, secrecy cannot be guaranteed by Definition 3 alone (see, for example, [21]). Therefore, we next consider secrecy in the presence of quantum adversaries.

1.2.3 Secrecy in the presence of a quantum adversary

As in the previous section, proving secrecy requires analysing the composite system of the user and the adversary, rather than the user's system alone (see the detailed discussions in [22, Section 2] and [19]). For a quantum adversary (with quantum side information), the classical variable X generated by the user can be represented as a quantum state in the Hilbert space $\mathcal{H}_X$ and treated as part of a composite system $\mathcal{H}_X \otimes \mathcal{H}_E$, where $\mathcal{H}_E$ denotes the adversary's Hilbert space of arbitrary but finite dimension. This composite system can be written as the classical-quantum state (cq-state)

$$\rho_{XE} = \sum_{\mathbf{x}\in\mathbb{Z}_2^n} p_X(\mathbf{x}) |\mathbf{x}\rangle\langle\mathbf{x}| \otimes \rho_E^{\mathbf{x}} \,. \tag{1.4}$$

In Equation (1.4) the adversary's part of the system is the state $\rho_E^{\mathbf{x}}$, which is conditioned on the realisation $X = \mathbf{x}$. Let ω_X denote the maximally mixed state on the Hilbert space $\mathcal{H}_X$, i.e. $\omega_X = \mathbb{1}/|\mathcal{H}_X|$, where $|\mathcal{H}_X|$ denotes the size of the Hilbert space $\mathcal{H}_X$. Similarly to the classical case, the ideal maximally secret state is the

cq-state $\omega_X \otimes \rho_E$, defined by

$$\omega_X \otimes \rho_E = 2^{-n} \sum_{\mathbf{x} \in \mathbb{Z}_2^n} |\mathbf{x}\rangle\langle\mathbf{x}| \otimes \rho_E \ , \tag{1.5}$$

where, in particular, the state of the adversary does not depend on $\mathbf{x}$. Operationally, this ideal state ensures that any adversarial measurement strategy using the state ρ_E does not provide any advantage in predicting the output $\mathbf{x}$.

Analogously to the classical case, the quantum-proof secrecy criterion is that the real state (1.4) is essentially indistinguishable from the ideal state (1.5). The trace distance provides a measure of how indistinguishable two quantum states are.

Definition 4 (Trace distance)**.** The trace distance between two quantum states ρ and σ on $\mathcal{H}$ is defined by

$$\frac{1}{2}\|\rho - \sigma\|_1 := \frac{1}{2}\mathrm{tr}\left(\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}\right) . \tag{1.6}$$

Specifically, the trace distance quantifies the maximum distinguishing advantage for discriminating between two quantum states, as established by the Helstrom bound [23]. Using this measure, we formalise the concept of secrecy against an adversary with access to quantum side information by defining *quantum-proof (ϵ-)perfect randomness*.

Definition 5 (Quantum-proof (ϵ-)perfect randomness)**.** The classical random variable $Z \in \mathbb{Z}_2^m$ is considered quantum-proof (ϵ-)perfectly random (given E), for some value $\epsilon \in [0,1]$, if, for the cq-state

$$\rho_{ZE} = \sum_{\mathbf{z} \in \mathbb{Z}_2^m} p_Z(\mathbf{z}) |\mathbf{z}\rangle\langle\mathbf{z}| \otimes \rho_E^{\mathbf{z}} \ , \tag{1.7}$$

the following condition holds:

$$\frac{1}{2}\|\rho_{ZE} - \omega_Z \otimes \rho_E\|_1 \leq \epsilon \ , \tag{1.8}$$

where ω_Z is the maximally mixed state of dimension 2^m.

Again, this can be understood as a bound on the distinguishing advantage, where $p_{\text{adv}} \leq \frac{\epsilon}{2}$ and if $\epsilon \ll 1$, we informally call such random variables (near-)perfectly random. Any resource satisfying this criterion is said to be *quantum secure* in the sense that the adversary's information is quantum.

Part I

Randomness extractors

Chapter 2

Introduction

A randomness extractor is a function that distils a weakly random input, known as a *source*, into an output that is nearly indistinguishable from the uniform distribution. These functions, first introduced by von Neumann [24], can be generally classified into three types:

- **Deterministic extractors**: These process the source alone, without any additional randomness. They require specific types of sources, such as those that consist of independent and identically distributed (IID) bits.
- **Seeded extractors**: These require an additional input of uniform and independent randomness, called a *seed*. By leveraging this additional randomness, these extractors can extract from sources that cannot be extracted from deterministically.
- **Multi-source extractors**: Also known as blenders, these generalise seeded extractors by using multiple independent weak sources instead of a single uniform and independent seed. In this thesis, we focus on **two-source** extractors, which require a single additional weak source only. This additional weak source is called the *weak seed*.

In this thesis, we consider extractors that take as input a weakly random source of n_1 bits and produce m output bits. The indistinguishability of the output from uniform is quantified by the *extractor error*, ϵ, which bounds the secrecy against classical (Definition 3) or quantum adversaries (Definition 5). This general setup is

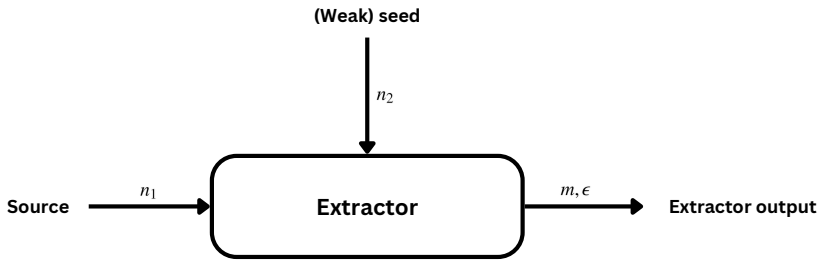

Figure 2.1: The general setup for randomness extractors. If $n_2 = 0$, the extractor is deterministic, if $n_2 > 0$, the extractor is seeded or two-source.

illustrated in Figure 2.1. We consider information-theoretically secure extractors, that is, extractors that are secure against adversaries with unlimited computational resources and time. Such extractors secure against classical adversaries are referred to as *classical-proof*, and those secure against quantum adversaries as *quantum-proof*.

Extractors are essential in numerous cryptographic applications, for example, as exposure-resilient functions, for privacy amplification in QKD, and distilling cryptographic randomness from entropy sources. Beyond cryptography, they have applications in de-randomising algorithms and constructing list-decodable error-correcting codes. However, certain challenges in the current state-of-the-art limit their usability in real-world protocols, particularly for use in quantum cryptography.

In the remainder of this part, we examine these limitations in detail and propose several solutions. This chapter explains the concept of weakly random sources, surveys existing results, explores key features of practical extractors and introduces the convolution theorem and number-theoretic transform (NTT), which will be important in constructing efficient extractors later. Chapter 3 focuses on extractors for min-entropy sources, which are the most general form of weakly random source. We present new extractor constructions with better performance, extend existing extractors to reduce resource requirements, and develop new efficient extractor implementations. Chapter 4 discusses extractors for weakly random sources characterised by the violation of a Bell inequality. We show that these extractors can be deterministic and provide a range of results, including efficient and explicit extractor

constructions.

2.1 Weakly random sources and extractability

A source is considered weakly random if it is not (ϵ-)perfectly random (see Definition 3 and Definition 5). In the 1960s, von Neumann [24] demonstrated that randomness could be extracted from sources modelled as independent tosses of a biased coin with unknown bias. These sources are known as *von Neumann sources* (sometimes also called *binary IID sources*).

Definition 6 (von Neumann source, adapted from [25])**.** A random variable $X = (X_0 \circ \ldots \circ X_{n-1}) \in \mathbb{Z}_2^n$, where $X_i \in \mathbb{Z}_2$ for all $i \in \mathbb{Z}_n$, is called a von Neumann source if $X_0, \ldots, X_{n-1}$ are independent and there exists $\delta \in [0,1]$ such that, for all i, $p_{X_i}(1) = \delta$.

Following von Neumann's work, efforts were made to relax the requirement of independence between bits. Santha and Vazirani [26] introduced a generalisation, known as a δ *Santha-Vazirani source*, which describes a scenario where each bit's outcome has a bounded probability conditioned on all previous bits.

Definition 7 (δ Santha-Vazirani (SV) source [26])**.** Given $\delta \in [0, 0.5]$, a random variable $X = X_0 \circ \ldots \circ X_{n-1} \in \mathbb{Z}_2^n$, where $X_i \in \mathbb{Z}_2$ for all $i \in \mathbb{Z}_n$, is a δ Santha-Vazirani (SV) source if, for all $j \in \mathbb{N}$,

$$\frac{1}{2} - \delta \leq \Pr\left(X_j = 1 | X_0, \ldots, X_{j-1}\right) \leq \frac{1}{2} + \delta \,. \tag{2.1}$$

A central result of [26] showed that no deterministic extractor can extract even a single bit from δ SV sources, for any $\delta > 0$. This revealed that even highly structured weakly random sources do not allow for deterministic extraction.

These results motivated a long line of research into which realistic sources of randomness can be extracted from and for each, what additional resources are required. Initially, deterministic extractors were shown to work for von Neumann sources with unknown bias [24, 27, 28, 29], sources that form a *finite Markov chain* [30], and sources in which some bits are fixed (both adversarially or not) while the remaining ones are random (called *non-oblivious bit-fixing* and *oblivious*

bit-fixing respectively) [31, 32, 33, 34, 35]. Since then, several other classes of sources have been shown to be deterministically extractable, including those with efficient sampling procedures [36, 37], sources sampled in *small space* [38], sources over *affine* subspaces [39, 40, 41, 42, 43, 44, 45], polynomial sources (distributions sampled by low degree polynomials over finite fields) [46], sources sampled by Turing machines [47], small circuits of bounded depth [48], generalised Santha-Vazirani sources [49, 50], *interleaved* sources [51], *sumset* sources [52] and our work: sources that violate a Bell inequality [18].

We summarise the results applicable to binary sources in Figure 2.2. Nodes with solid borders represent sources for which deterministic extractors are known, while dashed borders indicate sources for which deterministic extraction is impossible. Directed arrows illustrate that any source at the base of an arrow can be understood as a special case of the source at the head of the arrow. The relationships between various classes of weakly random sources and sumset sources were established in [53], while the connection between small space and interleaved sources was explored in [54]. Additionally, the relationship between δ-imbalanced sources and strong Santha-Vazirani sources was discussed in [55], the classification of SHELA (somewhere honest entropic look ahead) sources [56] as p-resettable sources [57] was demonstrated in [56], and the connection between p-resettable and Santha-Vazirani sources was shown in [57].

The remainder of this part focuses on min-entropy sources, as all other sources are special cases of these (i.e., min-entropy sources with additional guarantees), and on sources that violate Bell inequalities, as these extractors are a novel contribution of our work and are highly relevant to quantum cryptography.

2.2 Crucial extractor features

Despite the importance of extractors, selecting the right randomness extractor and its parameters for a specific task is challenging. When selecting a randomness extraction algorithm for a protocol, the following features are crucial.

- **Implementability and efficiency:** Some extractors are purely theoretical,

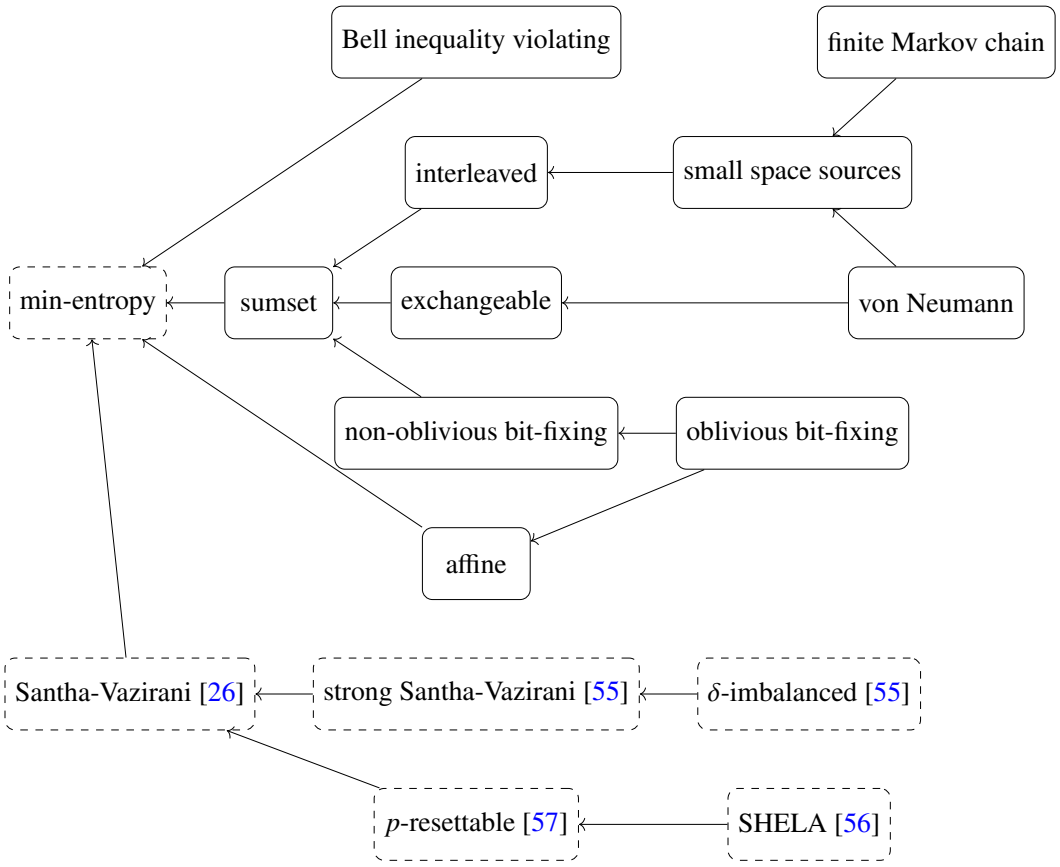

Figure 2.2: A directed graph illustrating relationships between classifications of weakly random sources and their extractability via deterministic functions. An arrow indicates that the source at its tail is a special case of the source at its head. Solid borders denote sources allowing deterministic extraction, while dashed borders require probabilistic extraction. References within nodes indicate impossibility proofs for deterministic extraction.

with non-constructive proofs, while others, though explicit, may have a computation time unsuitable for the application at hand. The majority of applications require at most $O(n_1^2)$ computation time, with many requiring quasi-linear computation time $O(n_1 \log(n_1))$, such as QKD (see [5, Appendix E, Section C] for an excellent discussion).

- **Additional resources:** Seeded and two-source extractors require a seed that is independent and either uniformly distributed (seeded extractors) or weakly random (two-source extractors). This adds complexity, as users need to trust the generation or acquisition of an independent (weak) seed. Minimising resource requirements is essential, such as using a short uniform seed [58, 5]

or reducing the entropy requirements on the weak seed [54, 59]. Certain extractors are *strong* extractors, meaning that the output is independent of the (weak) seed, allowing the seed to be reused or revealed without compromising security. For strong seeded extractors, this property alternatively allows the output length to be increased by concatenating the seed and the extractor output.

- **Security options:** Some extractors fail against quantum adversaries capable of storing side information about the source in quantum states [60]. For instance, in QKD, quantum-proof extractors are essential because the adversary has access to the quantum channel. Furthermore, if the input and (weak) seed are correlated, extractors that relax the independence requirement are vital e.g. [61, 62, 63].[1]

- **Entropy loss:** Entropy loss is the difference between the entropy of the input source and the length of the extractor's output. Seeded and two-source extractors have a minimum entropy loss that is logarithmic in the inverse of the extractor error [65], while deterministic extractors can, in principle, have no entropy loss. Some extractors, including those presented in this work, are known to achieve these fundamental bounds.

2.3 The number-theoretic transform and the convolution theorem

In what follows, we will use the convolution theorem to derive efficient randomness extractor implementations, leveraging the number-theoretic transform (NTT). When employing the convolution theorem, one can use the fast Fourier transform (FFT) or the NTT. Since we are focused on cryptographic applications, the FFT is unsuitable because it may introduce rounding errors from floating-point arithmetic

[1] Extractors may also have other security properties, like *non-malleability* [64], which ensures security even if the adversary has access to the extractors output using different seeds. However, none of the extractors in this thesis are known to have such properties, so they are not discussed here.

when computing with complex numbers. In contrast, the NTT ensures information-theoretic security and avoids these issues by performing computations in integer rings of bounded size. Before presenting the details, we first define a convolution.

Definition 8 (Convolution)**.** Let $n \in \mathbb{N}$ and $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^n$. The convolution $\mathbf{x} * \mathbf{y}$ is a vector of the same length, defined element-wise as

$$(\mathbf{x} * \mathbf{y})_i := \sum_{j \oplus k = i} x_j y_k \,, \tag{2.2}$$

where $\oplus$ denotes addition modulo n and $i \in \mathbb{Z}_n$.

This operation can be computed naively in $O(n^2)$ time, where n is the length of each input vector. However, the convolution theorem states that the discrete Fourier transform (DFT) of a convolution is the product of the DFTs of the vectors. Thus, by translating the vectors to Fourier space, computing a convolution reduces to vector multiplication. Furthermore, efficient DFT algorithms, such as the Cooley-Tukey FFT [66], reduce the computation time to $O(n \log n)$. We now present these details, as they are crucial for deriving our later results.

Definition 9 (Discrete Fourier transform (DFT))**.** Given a vector $\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{C}^n$, the DFT is a function $\mathsf{DFT} : \mathbb{C}^n \to \mathbb{C}^n$, defined element-wise as

$$\mathsf{DFT}(\mathbf{x})_j := \sum_{k=0}^{n-1} x_k \exp\left(-i \frac{2\pi}{n} k j\right), \tag{2.3}$$

with its inverse, $\mathsf{DFT}^{-1} : \mathbb{C}^n \to \mathbb{C}^n$, given by

$$\mathsf{DFT}^{-1}(\mathbf{x})_j := \frac{1}{n} \sum_{k=0}^{n-1} x_k \exp\left(i \frac{2\pi}{n} k j\right) \tag{2.4}$$

for $j \in \mathbb{Z}_n$, where i is the imaginary unit.

As noted above, the DFT (and its inverse) can be computed in $O(n \log n)$ using FFT algorithms. However, since the DFT of a vector $\mathbf{x} \in \mathbb{Z}^n$ generally produces complex values that cannot be exactly represented in finite-precision arithmetic, implementing a fast convolution via the DFT may introduce rounding errors. This can

be problematic in cryptographic applications, where exact correctness is required. The convolution theorem can be applied using the NTT, a specialisation of the DFT to finite fields, which is computed using the ring of integers modulo q, $\mathbb{Z}_q$, and a primitive n-th root of unity g (instead of complex numbers $\mathbb{C}$ with primitive n-th root of unity $\exp\left(-\frac{i2\pi}{n}\right)$).

Definition 10 (Primitive n-th root of unity)**.** An element $g \in \mathbb{Z}_q$ is a primitive n-th root of unity in $\mathbb{Z}_q$ if $g^n = 1 \mod q$ and $g^k \neq 1 \mod q$ for all $k \in \{1, \ldots, n-1\}$.

Definition 11 (Number-theoretic transform (NTT))**.** Given $\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_q^n$, the NTT, $\mathsf{NTT} : \mathbb{Z}_q^n \to \mathbb{Z}_q^n$, for an integer ring $\mathbb{Z}_q$ and a primitive n-th root of unity g is defined element-wise as

$$\mathsf{NTT}(\mathbf{x})_j := \sum_{k=0}^{n-1} x_k g^{jk} \mod q \,, \tag{2.5}$$

with its inverse given by

$$\mathsf{NTT}^{-1}(\mathbf{x})_j := \frac{1}{n} \sum_{k=0}^{n-1} x_k g^{-jk} \mod q \,, \tag{2.6}$$

for $j \in \mathbb{Z}_n$.

The convolution of two equal length vectors can be computed as the element-wise product of their NTTs. This result is the essence of the convolution theorem.

Theorem 12 (Convolution theorem using the NTT)**.** Let $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^n$, and let $\mathsf{NTT} : \mathbb{Z}_q^n \to \mathbb{Z}_q^n$ be the NTT for the integer ring $\mathbb{Z}_q$ and g a primitive n-th root of unity. Then,

$$\mathsf{NTT}(\mathbf{x} * \mathbf{y})_i = \mathsf{NTT}(\mathbf{x})_i \mathsf{NTT}(\mathbf{y})_i \tag{2.7}$$

for all $i \in \mathbb{Z}_n$, where $\mathbf{x} * \mathbf{y}$ denotes the convolution of $\mathbf{x}$ and $\mathbf{y}$.

Proof. Using the definition of the NTT (Definition 11) and a convolution (Defini-

tion 8), we can rewrite the convolution as

$$(\mathbf{x} * \mathbf{y})_i = \sum_{j \oplus k = i} x_j y_k = \sum_{j=0}^{n-1} x_j y_{i-j} \,, \tag{2.8}$$

where all subscripts are computed modulo n. Then, we have

$$\begin{aligned} \mathsf{NTT}(\mathbf{x} * \mathbf{y})_i &= \sum_{k=0}^{n-1} \left(\sum_{j=0}^{n-1} x_j y_{k-j} \right) g^{ik} \\ &= \sum_{j=0}^{n-1} x_j g^{ij} \sum_{k=0}^{n-1} y_{k-j} g^{i(k-j)} \\ &= \left(\sum_{j=0}^{n-1} x_j g^{ij} \right) \left(\sum_{t=0}^{n-1} y_t g^{it} \right) \\ &= \mathsf{NTT}(\mathbf{x})_i \mathsf{NTT}(\mathbf{y})_i \,, \end{aligned} \tag{2.9}$$

where in the second line, we haved used that $g^{ik} = g^{ij} g^{i(k-j)}$ for any j, and in the third line, we introduced the substitution $t = k - j$. □

Therefore, by combining the above, we arrive at the following corollary.

Corollary 13. Let $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$, then

$$\mathbf{x} * \mathbf{y} = \mathsf{NTT}^{-1} \left(\mathsf{NTT}(\mathbf{x}) \cdot \mathsf{NTT}(\mathbf{y}) \right) \,, \tag{2.10}$$

where $\cdot$ denotes the element-wise product of the vectors.

This reduces the problem of computing a convolution of length-n vectors to computing the NTT (and its inverse) and multiplication in an integer ring. However, computing the NTT of a vector is not always efficient; in general, it still requires $O(n^2)$ computation time for length-n vectors. Similarly to the FFT, achieving efficiency requires a specific parametrisation to enable, for example, the use of divide-and-conquer techniques [67].

Remark 14 (Efficient parametrisation for NTT implementation [68])**.** Choose L as a power of 2 such that $L \geq 2n - 1$ and q prime such that $q > n$ and $q = 1 \bmod L$. The

NTT for vectors in $\mathbb{Z}_q^n$ can be computed in $O(n \log n)$ time using divide-and-conquer techniques, similar to the FFT, with a computation time of $O(L \log L)$.

Selecting L as a power of 2 and $q = 1 \bmod L$ ensures the existence of a primitive L-th root of unity and enables the use of divide-and-conquer techniques. The parametrisation $L \geq 2n - 1$ and the choice of a prime q ensure that all operations are performed in $\mathbb{Z}_q$, without overflow issues.

A particularly useful choice, which we will use later, is $q = 3 \times 2^{30} + 1$ and $L = 2^{30}$. This allows $n \leq 2^{29} \approx 5 \times 10^8$.

Chapter 3

Randomness extractors for min-entropy sources

3.1 Introduction

In this chapter, we consider randomness extractors for sources characterised by their min-entropy. Min-entropy is the most conservative measure of unpredictability, making these extractors applicable in a wide range of protocols. Min-entropy, with respect to any side information an adversary has, quantifies the minimum unpredictability of a source. To formalise this, we first introduce the *guessing probability*, which quantifies the maximum probability that an adversary can correctly guess the outcome of a random variable given their side information.

Definition 15 (Guessing probability)**.** For a random variable $X \in \mathbb{Z}_2^n$, the classical maximum guessing probability, given side information E taking values $\mathbf{e}$ in some finite alphabet $\mathcal{E}$, is

$$p_{\text{guess}}(X|E)_C := \sum_{\mathbf{e}\in\mathcal{E}} p_E(\mathbf{e}) \max_{\mathbf{x}\in\mathbb{Z}_2^n} p_{X|E}(\mathbf{x}|\mathbf{e}) \,. \tag{3.1}$$

The quantum maximum guessing probability of a classical random variable $X \in \mathbb{Z}_2^n$ represented by the classical portion of the cq-state

$$\rho_{XE} = \sum_{\mathbf{x}\in\mathbb{Z}_2^n} p_X(\mathbf{x}) |\mathbf{x}\rangle\langle\mathbf{x}| \otimes \rho_E^{\mathbf{x}}, \tag{3.2}$$

given the side information E, is

$$p_{\text{guess}}(X|E)_Q := \max_{\{M_{\mathbf{x}}\}_{\mathbf{x}}} \sum_{\mathbf{x}} p_X(\mathbf{x}) \operatorname{tr}\left(M_{\mathbf{x}} \rho_E^{\mathbf{x}}\right), \tag{3.3}$$

where $\{M_{\mathbf{x}}\}_{\mathbf{x}}$ denotes all possible measurement strategies on the state $\rho_E^{\mathbf{x}}$.

Using this operational notion of unpredictability, we define the conditional min-entropy.

Definition 16 (Conditional min-entropy)**.** The conditional min-entropy of a random variable $X \in \mathbb{Z}_2^n$, denoted $\mathrm{H}_\infty(X|E)_i$, is defined as

$$\mathrm{H}_\infty(X|E)_i := -\log\left(p_{\text{guess}}(X|E)_i\right), \tag{3.4}$$

where $i \in \{C, Q\}$ indicates whether the side information E is classical (C) or quantum (Q), and $p_{\text{guess}}(X|E)_i$ is the adversary's maximum guessing probability given their system E.

A classical random variable is called an (n,k) source if it has length n (i.e., it belongs to $\mathbb{Z}_2^n$) and conditional min-entropy k. Its (conditional) *min-entropy rate*, denoted by α, is defined as $\alpha = k/n$. Extractors for min-entropy sources are called classical-proof or quantum-proof $(n_1, k_1, n_2, k_2, m, \epsilon)$ extractors, where the source is an (n_1, k_1) source, the (weak) seed is an (n_2, k_2) source, m is the output length, and ϵ quantifies the output's distance from perfect randomness (Equation (3) and Equation (5)). This setup is illustrated in Figure 3.1.

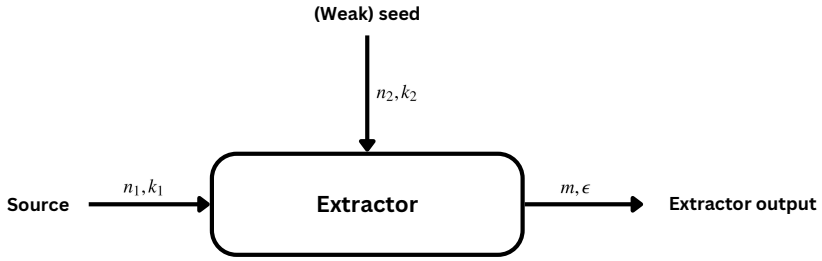

Figure 3.1: The setup for a $(n_1, k_1, n_2, k_2, m, \epsilon)$ extractor.

3.1.1 Challenge of extracting randomness in this scenario

It is known that deterministic extraction from min-entropy sources is impossible, originally shown in [26, 69]. We provide a simple proof of this fact.

Lemma 17 (Impossibility of deterministic extraction from min-entropy sources)**.** There is no function $f : \mathbb{Z}_2^{n_1} \to \mathbb{Z}_2$ such that, for all $(n_1, k_1 \leq n_1 - 1)$ sources X, the statistical distance between $f(X)$ and the uniform distribution on $\mathbb{Z}_2$, denoted U_1, satisfies $\mathrm{SD}(f(X), U_1) < \frac{1}{2}$.

Proof. Let $k_1 = n_1 - 1$ and suppose that there exists a function $f : \mathbb{Z}_2^{n_1} \to \mathbb{Z}_2$ such that $\mathrm{SD}(f(X), U_1) < \frac{1}{2}$ for all (n_1, k_1) sources X. This function can be understood as dividing the inputs into two sets: those mapped to 0 and those mapped to 1. Since there are 2^{n_1} possible inputs, at least one of these sets must contain 2^{n_1-1} or more inputs – we call this set I_b, where b denotes the output bit associated with that set. Now, consider an (n_1, k_1) source X uniformly distributed over I_b – that is, $p_X(\mathbf{x}_{I_b}) = 1/|I_b| \leq 2^{-n_1+1}$ for all $\mathbf{x}_{I_b} \in I_b$ and $p_X(\mathbf{x}_{I_{\bar{b}}}) = 0$ for all $\mathbf{x}_{I_{\bar{b}}} \notin I_b$. For this source, $f(X)$ always outputs the fixed bit b, resulting in $\mathrm{SD}(f(X), U_1) = \frac{1}{2}$, contradicting the assumption. The cases $k_1 \in \mathbb{Z}_{n_1-1}$ follow by noting that any (n_1, k_1) source is also a $(n_1, k_1 - c)$ source for any integer $c \leq k_1$. □

Due to this impossibility, randomness extractors for min-entropy sources must be constructed using probabilistic methods, i.e. they must be seeded or multi-source. Therefore, any $(n_1, k_1, n_2, k_2, m, \epsilon)$ extractors require $n_2 > 0$ and $k_2 > 0$. The first of these to be considered were seeded extractors, where $k_2 = n_2$.

3.2 Seeded extractors for min-entropy sources

Seeded extractors were first introduced by Nisan and Zuckermann in [70] and assume the extractor has access to an independent and uniform seed, i.e. an independent $(n_2, k_2 = n_2)$ source.

Definition 18 (Classical-proof seeded extractor)**.** A function $\mathrm{Ext} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^d \to \mathbb{Z}_2^m$ is a classical-proof $(n_1, k_1, n_2 = d, k_2 = d, \epsilon)$ seeded extractor if, for classical side

information E, any independent (n_1, k_1) source X and (d, d) source S, we have

$$\mathrm{SD}(\mathrm{Ext}(X,S), U_m|E) \leq \epsilon \,. \tag{3.5}$$

Moreover, Ext is called a *strong* classical-proof seeded extractor if

$$\mathrm{SD}(\mathrm{Ext}(X,S)\circ S, U_m \circ S|E) \leq \epsilon \,. \tag{3.6}$$

For quantum-proof seeded extractors, the source and seed are classical constituents of a composite system in which the adversary has quantum side information, called a classical-classical-quantum state (ccq-state).

Definition 19 (Classical-classical-quantum state (ccq-state))**.** A quantum state ρ_{XYE} is a ccq-state in the Hilbert space $\mathcal{H}_X \otimes \mathcal{H}_Y \otimes \mathcal{H}_E$ if it is classical on $\mathcal{H}_X$ and $\mathcal{H}_Y$ and quantum on $\mathcal{H}_E$. For classical random variables $X \in \mathbb{Z}_2^{n_1}$ and $Y \in \mathbb{Z}_2^{n_2}$ with joint distribution $p_{XY}(\mathbf{x}, \mathbf{y})$, ρ_{XYE} can be written as

$$\rho_{XYE} := \sum_{\mathbf{x}\in\mathbb{Z}_2^{n_1}} \sum_{\mathbf{y}\in\mathbb{Z}_2^{n_2}} p_{XY}(\mathbf{x},\mathbf{y}) \, |\mathbf{x}\rangle\langle\mathbf{x}| \otimes |\mathbf{y}\rangle\langle\mathbf{y}| \otimes \rho_E^{\mathbf{x},\mathbf{y}} \,, \tag{3.7}$$

where $\rho_E^{\mathbf{x},\mathbf{y}}$ is the reduced state on $\mathcal{H}_E$ conditioned on $\mathbf{x}$ and $\mathbf{y}$, and $\{|\mathbf{x}\rangle\}_{\mathbf{x}}$ and $\{|\mathbf{y}\rangle\}_{\mathbf{y}}$ are orthonormal bases for $\mathcal{H}_X$ and $\mathcal{H}_Y$, respectively.

If the classical random variables X and Y in the above definition are independent, then $p_{XY}(\mathbf{x}, \mathbf{y}) = p_X(\mathbf{x}) p_Y(\mathbf{y})$.

Applying an extractor $\mathrm{Ext} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \to \mathbb{Z}_2^m$ to a ccq-state can be described by the action of a completely positive trace-preserving map $\mathrm{Ext}_{XY} \otimes \mathbb{1}_E$ on ρ_{XYE}, where $(\mathrm{Ext}_{XY} \otimes \mathbb{1}_E)\rho_{XYE} = \rho_{\mathrm{Ext}(X,Y)E}$,

$$\rho_{\mathrm{Ext}(X,Y)E} = \sum_{\mathbf{z}\in\mathbb{Z}_2^m} p_{\mathrm{Ext}(X,Y)}(\mathbf{z}) \, |\mathbf{z}\rangle\langle\mathbf{z}| \otimes \rho_E^{\mathbf{z}} \,, \tag{3.8}$$

and

$$p_{\mathrm{Ext}(X,Y)}(\mathbf{z})\rho_E^{\mathbf{z}} = \sum_{\mathbf{x},\mathbf{y}|\mathrm{Ext}(\mathbf{x},\mathbf{y})=\mathbf{z}} p_{XY}(\mathbf{x},\mathbf{y})\rho_E^{\mathbf{x},\mathbf{y}} \,. \tag{3.9}$$

Using this notation, a quantum-proof seeded extractor is defined as follows.

Definition 20 (Quantum-proof seeded extractor)**.** A function $\mathrm{Ext} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^d \to \mathbb{Z}_2^m$ is a quantum-proof $(n_1, k_1, n_2 = d, k_2 = d, m, \epsilon)$ seeded extractor if, for any ccq-state ρ_{XSE} on Hilbert space $\mathcal{H}_X \otimes \mathcal{H}_S \otimes \mathcal{H}_E$ such that $\rho_{XSE} = \rho_{XE} \otimes \omega_S$, the system X on $\mathcal{H}_X$ is an (n_1, k_1) source and S on $\mathcal{H}_S$ is a (d, d) source, we have

$$\frac{1}{2}\|\rho_{\mathrm{Ext}(X,S)E} - \omega_{\mathrm{Ext}(X,S)} \otimes \rho_E\|_1 \leq \epsilon \,, \tag{3.10}$$

where $\rho_{\mathrm{Ext}(X,S)E} = (\mathrm{Ext}_{XS} \otimes \mathbb{1}_E)\rho_{XSE}$ and $\omega_{\mathrm{Ext}(X,S)}$ is the maximally mixed state on a system of dimension 2^m. Additionally, Ext is called a *strong* quantum-proof seeded extractor if

$$\frac{1}{2}\|\rho_{\mathrm{Ext}(X,S)SE} - \omega_{\mathrm{Ext}(X,S)} \otimes \rho_{SE}\|_1 \leq \epsilon \,, \tag{3.11}$$

where $\rho_{\mathrm{Ext}(X,S)SE} = \sum_{\mathbf{s}} p_S(\mathbf{s})|\mathbf{s}\rangle\langle\mathbf{s}| \otimes \rho_{\mathrm{Ext}(X,S)E}$.

Seeded extractors have been extensively studied, introduced in [70] and continued in works such as [58], [71], and [72]. In [65] it was shown using probabilistic methods that for every (n_1, k_1) source there exists an $(n_1, k_1, n_2 = d, k_2 = d, m, \epsilon)$ extractor with $d = \log(n_1 - k_1) + 2\log(1/\epsilon) + O(1)$ and output length $m = k_1 + d - 2\log(1/\epsilon) - O(1)$. Moreover, it was shown that these parameters are optimal, indicating that seeded extractors have a minimum entropy loss of $2\log(1/\epsilon) - O(1)$. Seeded extractors that asymptotically achieve these lower bounds (up to constants) have been developed. For example, the breakthrough construction of Trevisan [58] achieves optimal seed length, along with works such as [71] and [73].

Although many constructions exist, few are suitable for practical quantum cryptographic protocols. The reason for this is highlighted in [5, Appendix E, Section C], where the need for efficiently implementable extractors in quantum cryp-

tography is discussed. They state "for quantum key distribution (QKD), the usual notion of efficiency (i.e., with polynomial complexity) is not sufficient. Rather, we should restrict ourselves to algorithms with complexity $O(n\log n)$", where n here denotes the length of the source in bits. Their claim is supported by an example which demonstrates that QKD protocols, due to often requiring input lengths of $n \geq 10^6$, encounter performance limitations using extraction algorithms with $O(n^2)$ computation time.

In Table 3.1, we survey constructions with explicit implementations that have a total computation time (including any pre-processing steps) of at most $O(mn_1^2)$, with m denoting the output length and n_1 the source length. We note that only near-minimal entropy loss versions of the Trevisan extractors [58] from [6] are included[1]. A table that includes general polynomial computation time constructions can be found in [5].

Extractor	**Output length** m	**Seed length** d	**Computation time**
Circulant [2]	$m \leq k_1 + 2\log(\epsilon)$	$n_1 + 1 \in \mathbb{P}$	$O(n_1 \log n_1)$
Dodis et al. 'Right Cyclic Shift' [74]	$m \leq k_1 + 2\log(\epsilon) + 1$	$n_1 \in \mathbb{N}_A$	$O(n_1 \log n_1)$ [17]
Hayashi-Tsurumaru $f_{F1,R}$ [5]	$m \leq k_1 + 2\log(\epsilon)$	$n_1 - m$	$O(n_1 \log n_1)$ if $m+1 \in \mathbb{N}_A$
Hayashi-Tsurumaru $f_{F2,R}$ [5]	$m \leq k_1 + 2\log(\epsilon) - \log(c_{F2,R} - 1)$	$d = n_1 - m$	$O(n_1 \log n_1)$ if $n_1 - m + 1 \in \mathbb{N}_A$
Hayashi-Tsurumaru $f_{F3,R}$ [5]	$m \leq k_1 + 2\log(\epsilon) - \log\left(\lceil \frac{m}{n_1-m} \rceil \lceil \frac{k_1}{n_1-k_1} \rceil\right)$	$2k_1 - m$	$O(n_1 \log n_1)$
Hayashi-Tsuruamru $f_{F4,R}$ [5]	$m \leq k_1 + 4\log(\epsilon) - 4\log(e)$	k_1	$O(n_1 \log n_1)$
Toeplitz [75]	$m \leq k_1 + 2\log(\epsilon)$	$n_1 + m - 1$	$O(n_1 \log n_1)$
Modified Toeplitz [5, 76]	$m \leq k_1 + 2\log(\epsilon)$	$n_1 - 1$	$O(n_1 \log n_1)$
Trevisan [58], 'XOR code' [6]	$m \leq k_1 - \gamma n - 6\log\left(\frac{m(2+\sqrt{2})}{\epsilon}\right) + \log\left(\frac{4}{3}\right)$	$a(t_1)t_1^2$	$O(n_1 \log n \log \frac{n_1 m}{\epsilon})$
Trevisan [58], 'Polynomial hashing' [6]	$m \leq k_1 - 4\log\frac{m}{\epsilon} - 6$	$a(t_2)t_2^2$	$O(mn_1 \log^2 \frac{n_1 m^2}{\epsilon^2})$ [2]
TSSR, Theorem 10 [77]	$m \leq k_1 + 2\log(2\epsilon)$	$2\lceil m + \log\frac{n}{2m} \rceil$	$O(n_1 \log n_1)$ (claimed in [5])

Table 3.1: The parameters for $(n_1, k_1, n_2 = d, k_2 = d, m, \epsilon)$ seeded extractors constructions with computation time at most $O(mn_1^2)$. The output and seed lengths, m and d, are in bits. $\mathbb{P}$ denotes the set of primes and $\mathbb{N}_A$ denotes the set of primes with primitive root 2. For the Hayashi-Tsurumaru $f_{F1,R}$ and $f_{F2,R}$ constructions, the input length is $n_1 = c_i m$ for $i \in \{(F1,R),(F2,R)\}$ respectively, where c_i are natural numbers for all i and $c_{F2,R} \leq 1 + \lceil \frac{m}{n-m} \rceil$. For the Hayashi-Tsurumaru $f_{F4,R}$, the parameter $e \approx O(2^{(m-k_1)/4})$ (see [5] for the explicit term). For the Trevisan constructions, the function $a(t_i) \approx O(\log(m/t_i))$ for $i \in \{1,2\}$ where $t_1 \approx O(\log(n)\log(m/\epsilon))$ and $t_2 \approx O(\log\left(nm^2/\epsilon^2\right)$ (see [6] for explicit terms).

All seeded extractors in Table 3.1 are strong, and except Dodis et al., are

[1] The 'Lu's construction' from [6] is omitted due to its high computation time (rather than due to its entropy loss). This construction involves a free parameter ν, which must be made small for extraction to be possible, but the computation steps increase significantly as ν decreases.

all quantum-proof with the same parameters. The Circulant, Toeplitz, Modified Toeplitz, and TSSR (Tomamichel-Schaffner-Smith-Renner) constructions are quantum-proof by the Quantum Leftover Hash Lemma (Q-LHL) [22, 77]. The quantum-proof security of the Hayashi-Tsurumaru constructions is proved in [5], and the Trevisan-based constructions in [72].

In the remainder of this section, we present several seeded extractors well-suited to quantum cryptography, offering security against quantum adversaries and efficient computation: our Circulant extractor, and enhanced implementations of the Toeplitz and Trevisan extractors. First, we define what it means for a family of hash functions to be two-universal and state the Q-LHL, both of which are important to the following discussion.

Definition 21 (Two-universal hash functions)**.** A family $\mathcal{F}$ of functions from $\mathbb{Z}_2^n \to \mathbb{Z}_2^m$ is said to be *two-universal* if, for any $\mathbf{x} \neq \tilde{\mathbf{x}} \in \mathbb{Z}_2^n$ and f chosen uniformly from $\mathcal{F}$, we have

$$\Pr(f(\mathbf{x}) = f(\tilde{\mathbf{x}})) \leq 2^{-m} . \tag{3.12}$$

Lemma 22 (Quantum Leftover Hash Lemma (Q-LHL), rephrased from Corollary 5.6.1 in [22])**.** Consider the cq-state ρ_{XE} on the Hilbert space $\mathcal{H}_X \otimes \mathcal{H}_E$. Let $\mathcal{F}$ be a two-universal family of hash functions from $\mathbb{Z}_2^{n_1} \to \mathbb{Z}_2^m$. If X is an (n_1, k_1) source, then

$$\frac{1}{2}\|\rho_{F(X)FE} - \omega_{F(X)} \otimes \rho_{FE}\|_1 \leq 2^{(m-k_1)/2} , \tag{3.13}$$

where

$$\rho_{F(X)FE} = \sum_{f \in \mathcal{F}} p_F(f) \rho_{f(X)E} \otimes |f\rangle\langle f| , \tag{3.14}$$

$$\rho_{f(X)E} = \sum_{\mathbf{z} \in \mathbb{Z}_2^m} |\mathbf{z}\rangle\langle\mathbf{z}| \otimes \rho_E^{\mathbf{z}} , \tag{3.15}$$

and

$$\rho_E^{\mathbf{z}} = \sum_{\mathbf{x}\in f^{-1}(\mathbf{z})} \rho_E^{\mathbf{x}} \,. \tag{3.16}$$

Essentially, the Q-LHL says that any family of two-universal hash functions constitute a strong quantum-proof seeded extractor, where the seed is used to select the hash function f from the family $\mathcal{F}$. We note that there are extensions of the Q-LHL that, for example, allow *almost two-universal hash functions* [77]. However, Lemma 22 is sufficient for our purposes.

3.2.1 The Circulant extractor

We now present our Circulant extractor, which is based on matrix-vector multiplication using a circulant matrix generated from the source, applied to the vector associated with the seed.

Definition 23 (Circulant extractor)**.** Let $\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_2^n$ and $\mathbf{y} = (y_0, \ldots, y_n) \in \mathbb{Z}_2^{n+1}$. The function $\mathsf{Circulant} : \mathbb{Z}_2^n \times \mathbb{Z}_2^{n+1} \to \mathbb{Z}_2^m$ is implemented by:

1. Set $\mathbf{x}' = (\mathbf{x}, 0) = (x_0, \ldots, x_{n-1}, 0) \in \mathbb{Z}_2^{n+1}$.
2. Compute

$$\mathsf{Circulant}(\mathbf{x}, \mathbf{y}) := (\mathrm{circ}(\mathbf{x}')\mathbf{y})_{0:m-1} \,, \tag{3.17}$$

where the matrix-vector multiplication $\mathrm{circ}(\mathbf{x}')\mathbf{y}$ is performed modulo 2 (component-wise) and the subscript $0 : m-1$ denotes the first m elements of the result. The term $\mathrm{circ}(\mathbf{x}')$ refers to the $(n+1)\times(n+1)$ circulant matrix generated by $\mathbf{x}'$;

$$\mathrm{circ}(\mathbf{x}') := \begin{bmatrix} x'_0 & x'_1 & x'_2 & \ldots & x'_{n-1} & x'_n \\ x'_n & x'_0 & x'_1 & \ldots & x'_{n-2} & x'_{n-1} \\ & \ddots & \ddots & \ddots & \ddots & \\ x'_1 & x'_2 & x'_3 & \ldots & x'_n & x'_0 \end{bmatrix} . \tag{3.18}$$

This function is a strong quantum-proof $(n_1, k_1, n_2 = n_1 + 1, k_2 = n_1 + 1, m, \epsilon)$ seeded

extractor for prime $n_1 + 1$, with output length $m \leq k_1 + 2\log\epsilon$.[2] To prove this, we first show that, for any fixed m, the functions from Definition 23 indexed by different $\mathbf{y}$ form a two-universal family of hash functions and then apply the Q-LHL (Lemma 22). We begin with some lemmas crucial to our proof.

Lemma 24. For any prime n and $\mathbf{x}' = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_2^n$ such that $\mathbf{x}' \neq \{0\}^n, \{1\}^n$, the circulant matrix generated by $\mathbf{x}'$, $\mathrm{circ}(\mathbf{x}')$, is a bijection.

Proof. From Proposition 23 in [78], if $\mathbf{x}' \neq \{0\}^n, \{1\}^n$, the matrix $\mathrm{circ}(\mathbf{x}')$ is non-singular and thus has full rank. Since $\mathrm{circ}(\mathbf{x}')$ is square and full rank, it is both injective and surjective. Therefore, $\mathrm{circ}(\mathbf{x}')$ is a bijection whenever n is prime and $\mathbf{x}' \neq \{0\}^n, \{1\}^n$. □

Lemma 25. Let a random variable $Y \in \mathbb{Z}_2^n$ be such that $p_Y(\mathbf{y}) = 2^{-n}$ for all $\mathbf{y}$. Then, for any $n \times n$ matrix D that is a bijection, $p_Y(D\mathbf{y}) = 2^{-n}$ for all $\mathbf{y}$.

Proof. For each $\mathbf{y} \in \mathbb{Z}_2^n$, there is a unique value $Y = \mathbf{y}$ such that $D\mathbf{y} = \mathbf{y}^*$, as D is a bijection. Therefore, if $p_Y(\mathbf{y}^*) = 2^{-n}$ for all $\mathbf{y}^*$, it follows that $p_Y(D\mathbf{y}) = p_Y(\mathbf{y}^*) = 2^{-n}$ for all $\mathbf{y}^*$, completing the proof. □

Lemma 26. Given a random variable $Y \in \mathbb{Z}_2^n$ such that $p_Y(\mathbf{y}) = 2^{-n}$ for all $\mathbf{y} \in \mathbb{Z}_2^n$, the first m bits of Y, denoted $Y_{0:m-1}$, satisfy

$$p_{Y_{0:m-1}}(\mathbf{u}) = 2^{-m} \text{ for all } \mathbf{u} \in \mathbb{Z}_2^m \,. \tag{3.19}$$

Proof. We use that the distribution $p_{Y_{0:m-1}}(\mathbf{u})$ is the marginal distribution of $p_Y(\mathbf{y})$, and so

$$p_{Y_{0:m-1}}(\mathbf{u}) = \sum_{\mathbf{v} \in \mathbb{Z}_2^{n-m}} p_{Y_{0:m-1}Y_{m:n-1}}(\mathbf{u}, \mathbf{v}) \tag{3.20}$$

$$= \sum_{\mathbf{v} \in \mathbb{Z}_2^{n-m}} p_Y(\mathbf{u}, \mathbf{v}) = 2^{-n}2^{n-m} = 2^{-m} \,, \tag{3.21}$$

completing the proof. □

[2] Since we require $n_1 + 1$ to be prime, in Appendix A we list the closest primes for powers of 10 up to 10^{12} and powers of 2 up to 2^{40}.

Using the above three lemmas, we now prove that the function in Definition 23 constitutes a two-universal family of hash functions.

Theorem 27. The function Circulant in Definition 23 is a two-universal family of hash functions.

Proof. The function Circulant in Definition 23 is two-universal if, for all $\mathbf{x} \neq \tilde{\mathbf{x}} \in \mathbb{Z}_2^n$ and for any random variable $Y \in \mathbb{Z}_2^{n+1}$ such that $p_Y(\mathbf{y}) = 2^{-(n+1)}$ for all $\mathbf{y}$, the following holds:

$$\Pr(\mathsf{Circulant}(\mathbf{x}, Y) = \mathsf{Circulant}(\tilde{\mathbf{x}}, Y)) \leq 2^{-m} . \tag{3.22}$$

Let $\mathbf{x}' = (\mathbf{x}, 0)$ and $\tilde{\mathbf{x}}' = (\tilde{\mathbf{x}}, 0)$, and define $\mathbf{d}' = \mathbf{x}' \oplus \tilde{\mathbf{x}}'$, where $\oplus$ denotes bitwise addition modulo 2. The collision probability in Equation (3.22) can then be expressed as

$$\Pr(\mathsf{Circulant}(\mathbf{x}, Y) = \mathsf{Circulant}(\tilde{\mathbf{x}}, Y)) = \Pr((\mathrm{circ}(\mathbf{d}')Y)_{0:m-1} = \{0\}^m) . \tag{3.23}$$

Now, $\mathbf{d}' \neq \mathbf{0}$ (since $\mathbf{x} \neq \tilde{\mathbf{x}}$), and thus $\mathrm{circ}(\mathbf{d}')$ is a bijective linear transformation on $\mathbb{Z}_2^{n+1}$. By Lemma 25, since Y is uniform over $\mathbb{Z}_2^{n+1}$, the distribution of $\mathrm{circ}(\mathbf{d}')Y$ is also uniform over $\mathbb{Z}_2^{n+1}$. Then, using Lemma 26, the marginal distribution of $(\mathrm{circ}(\mathbf{d}')Y)_{0:m-1}$ is uniform over $\mathbb{Z}_2^m$, i.e.,

$$\Pr((\mathrm{circ}(\mathbf{d}')Y)_{0:m-1} = \{0\}^m) = 2^{-m} , \tag{3.24}$$

completing the proof. □

Putting everything together, we arrive at the following theorem.

Theorem 28. The function $\mathsf{Circulant} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_1+1} \to \mathbb{Z}_2^m$ from Definition 23 is a strong quantum-proof (and classical-proof) $(n_1, k_1, n_2 = n_1 + 1, k_2 = n_1 + 1, m, \epsilon)$ seeded extractor for prime $n_1 + 1$, with

$$m \leq k_1 + 2\log\epsilon . \tag{3.25}$$

Proof. This theorem follows from combining Theorem 27 and the Q-LHL, Lemma 22. □

3.2.1.1 Efficient implementation

We now prove that Circulant can be implemented in $O(n_1 \log n_1)$ computation time using the NTT. To do this, we need to rewrite the extractor as a convolution and adjust the input and output to match the efficient NTT parametrisation presented in Section 2.3. Note that the first step of Circulant (appending the bit 0 to the source) is a constant-time operation and can be neglected. To rewrite the extractor as a convolution, we introduce a reversal function.

Definition 29 (Reversal function)**.** The reversal function $R : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ for $\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_2^n$ is defined as

$$R(x_0, x_1, \ldots, x_{n-1}) := (x_0, x_{n-1}, x_{n-2}, \ldots, x_1) \,. \tag{3.26}$$

The Circulant extractor function (3.17) can be rewritten element-wise as, for $i \in \mathbb{Z}_m$,

$$\mathsf{Circulant}(\mathbf{x}, \mathbf{y})_i = (R(\mathbf{x}') * \mathbf{y})_i \mod 2 \,. \tag{3.27}$$

Thus, it is sufficient to provide an algorithm to compute the convolution $R(\mathbf{x}) * \mathbf{y}$ in $O(n_1 \log n_1)$ time. Although there are algorithms based on the FFT for this purpose, they are not information-theoretically secure due to potential rounding errors from floating-point arithmetic. Using the NTT presented in Section 2.3, we can achieve an information-theoretically secure algorithm with the same computational time as FFT-based implementations, provided the inputs can be correctly transformed and the outputs properly recovered. For this, we define an embedding and uprooting function.

Definition 30 (Embedding and uprooting functions)**.** For integers n, q and $L > n$, the embedding function, $\eta : \mathbb{Z}_2^n \to \mathbb{Z}_q^L$, for $\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_2^n$ is defined element-wise

by

$$\eta(\mathbf{x})_i := \begin{cases} 1 & \text{if } i < n \text{ and } x_i = 1 \,, \\ 0 & \text{otherwise} \,, \end{cases} \tag{3.28}$$

for $i \in \mathbb{Z}_L$. The uprooting function, $\phi : \mathbb{Z}_q^L \to \mathbb{Z}_2^n$, for $\tilde{\mathbf{x}} = (\tilde{x}_0, \ldots, \tilde{x}_{L-1}) \in \mathbb{Z}_q^L$ is defined element-wise by

$$\phi(\tilde{\mathbf{x}})_i := \sum_{j \equiv i \pmod{n}} \tilde{x}_j \mod 2 \,, \tag{3.29}$$

for $i \in \mathbb{Z}_n$.

Then, for $\mathbf{x}', \mathbf{y} \in \mathbb{Z}_2^{n_2}$, the convolution $R(\mathbf{x}') * \mathbf{y}$ can be computed as

$$R(\mathbf{x}') * \mathbf{y} = \phi\left(\mathsf{NTT}^{-1}\left(\mathsf{NTT}(\eta(R(\mathbf{x}'))) \cdot \mathsf{NTT}(\eta(\mathbf{y}))\right)\right) . \tag{3.30}$$

Choosing L as a power of two such that $L \geq 2n_2 - 1$ and q as a prime satisfying $q > n_2$ and $q \equiv 1 \pmod{L}$ aligns with the efficient NTT parametrisation in Remark 14, resulting in a computation time of $O(L \log L)$. Since $L = O(n_2)$ and $n_2 = n_1 + 1$, this provides an implementation of the Circulant extractor with a computation time of $O(n_1 \log n_1)$.

3.2.2 The Toeplitz extractor

The Toeplitz extractor [75] is a function $\mathsf{Toeplitz} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_1+m-1} \to \mathbb{Z}_2^m$, constructed using Toeplitz matrices. Given $\mathbf{x} = (x_0, x_1, \ldots, x_{n_1-1}) \in \mathbb{Z}_2^{n_1}$ and $\mathbf{y} = (y_0, y_1, \ldots, y_{n_1+m-2}) \in \mathbb{Z}_2^{n_1+m-1}$, the output of the extractor is computed as the matrix-vector multiplication

$$\mathsf{Toeplitz}(\mathbf{x}, \mathbf{y}) := \mathrm{toep}(\mathbf{y})\mathbf{x} \,, \tag{3.31}$$

where

$$\text{toep}(\mathbf{y}) := \begin{bmatrix} y_0 & y_{n_1+m-2} & \cdots & \cdots & y_m \\ y_1 & y_0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ y_{m-1} & y_{m-2} & \cdots & \cdots & . \end{bmatrix} \tag{3.32}$$

is the $m \times n_1$ Toeplitz matrix generated from $\mathbf{y}$. This function is known to constitute a two-universal family of hash functions [75], and therefore by the Q-LHL the Toeplitz extractor is a strong quantum-proof (and classical-proof) $(n_1, k_1, n_2 = n_1 + m - 1, k_2 = n_1 + m - 1, m, \epsilon)$ seeded extractor with $m \leq k_1 + 2\log(\epsilon)$.

3.2.2.1 Efficient implementation

The Toeplitz extractor can be implemented efficiently by embedding the computation into a larger circulant matrix-vector multiplication and utilising the convolution theorem. Specifically, the function

$$\mathsf{Toeplitz}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} y_0 & y_{n_1+m-2} & \cdots & \cdots & y_m \\ y_1 & y_0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ y_{m-1} & y_{m-2} & \cdots & \cdots & . \end{bmatrix} \mathbf{x} \tag{3.33}$$

can be rewritten as

$$\mathsf{Toeplitz}(\mathbf{x}, \mathbf{y}) = (\text{circ}(y_0, y_{n_1+m-2}, \ldots, y_1)\mathbf{x}')_{0:m-1} = (\text{circ}(R(\mathbf{y}))\mathbf{x}')_{0:m-1}\,, \tag{3.34}$$

where $\mathbf{x}' = (\mathbf{x}, \{0\}^{m-1}) \in \mathbb{Z}_2^{n_1+m-1}$. This embeds the Toeplitz matrix in the upper left quadrant of a larger circulant matrix. The output can be defined element-wise by the convolution

$$\mathsf{Toeplitz}(\mathbf{x}, \mathbf{y})_i = (\mathbf{y} * \mathbf{x}')_i \mod 2 \tag{3.35}$$

for $i \in \mathbb{Z}_m$. Using the embedding and uprooting functions from Definition 30, as for the Circulant extractor in Section 3.2.1.1, we can employ the NTT from Remark 14 to achieve an implementation with $O(n_1 \log n_1)$ computation time.

3.2.3 The Trevisan extractor

The Trevisan extractor [58] is built from two components: (1) a *weak design* partitions the seed into m substrings (*chunks*) by assigning each chunk a subset of seed-bit indices, carefully chosen so that any two chunks share only a limited number of indices, and (2) iteratively a *1-bit extractor* (a strong seeded extractor that outputs 1 bit) uses each chunk as a seed to extract from the source, producing a total of m bits that are (ϵ-)perfect. This breakthrough construction was able to saturate the lower bound on seed length, achieving $d \approx O(\log(n_1))$, where n_1 is the length of the source. However, a notable drawback of this approach is the computation time. For instance, if each 1-bit extractor has a computation time of $O(f(n_1))$, the total computation time scales to at least $O(mf(n_1))$. Even if $f(n_1)$ is linear, as $m \to n_1$ (a desirable property for practical applications), the computation time grows to $O(n_1^2)$.

The near-minimal entropy loss construction from [6, 'Polynomial hashing'] uses a seed length $d = a(t)t^2$, where

$$t \in \mathbb{P}_{\geq q}, \quad q = 2\lceil \log(n_1) + 2\log(2m/\epsilon) \rceil \,, \tag{3.36}$$

$$a(t) = \max\left\{1, \left\lceil \frac{\log(m - 2\exp(1)) - \log(t - 2\exp(1))}{\log(2\exp(1)) - \log(2\exp(1) - 1)} \right\rceil \right\} \,, \tag{3.37}$$

$\mathbb{P}_{\geq q}$ denotes the set of primes larger than or equal to q, and $\exp(1) \approx 2.718$ is the base of the natural logarithm. Similarly to the Circulant and Toeplitz extractors, Trevisan's extractor is directly quantum-proof, proved in [72]. Therefore, the Trevisan extractor [6, 'Polynomial hashing'] is a strong quantum-proof (and classical-proof) $(n_1, k_1, d, d, m, \epsilon)$ seeded extractor, with

$$m \leq k_1 + 4\log(\epsilon) - 4\log(m) - 6 \,. \tag{3.38}$$

We note that both Equation (3.36) and Equation (3.38) include a dependence

on m that does not appear in [6]. This difference arises from how ϵ is defined: in [6], ϵ denotes the extractor error per output bit, whereas here it denotes the total extractor error.

Trevisan with shorter seed length. The parameters above use the 'block weak design' from [6], which iteratively applies the weak design of [79] to generate the required chunks. Using the weak design from [79] without this iterative process results in a shorter seed length, but at the expense of increased entropy loss. Specifically, it achieves a seed length $d = t^2$ whilst reducing m to $m/2\exp(1) \approx m/5.43656$.

3.2.3.1 Implementation

We implement the Trevisan extractor following the 'Polynomial hashing' construction of [6], which uses a block weak design constructed by iteratively calling the Hartman and Raz [79] weak design $a(t)$ times, and the Reed-Solomon and Hadamard (RSH) 1-bit extractor (see [6, Section III.C.3]).

Let $\mathbf{x} \in \mathbb{Z}_2^{n_1}$ be the source and $\mathbf{y} \in \mathbb{Z}_2^d$ be the seed. The block weak design generates m sets of indices $S_i \subset \mathbb{Z}_d$, for $i \in \mathbb{Z}_m$ satisfying the conditions $|S_i| = t$ and

$$\sum_{j=0}^{i-1} 2^{|S_j \cap S_i|} \leq m \tag{3.39}$$

for all i. The block weak design iteratively uses the Hartman and Raz weak design, which requires t to be prime (the condition in Equation (3.36)). The full details and proofs can be found directly in [6, Section III. B]. The generation of the block weak design can be pre-computed and saved to memory, so the key computation step is performing the RSH 1-bit extractor.

Using the sets $S_0, \ldots, S_{m-1}$, we generate the substrings $\mathbf{y}_{S_i} \in \mathbb{Z}_2^t$ for $i \in \mathbb{Z}_m$, which denote the bits of $\mathbf{y}$ at the indices in S_i. Each sub-string $\mathbf{y}_{S_i}$ is then split into two parts for use in the Reed-Solomon and Hadamard steps, respectively, where each step requires a seed of length

$$l = \lceil \log(n_1) + 2\log(2m/\epsilon) \rceil \leq t/2 \,. \tag{3.40}$$

For each $i \in \mathbb{Z}_m$, we apply the RSH extractor $\mathsf{RSH} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{2l} \to \mathbb{Z}_2$ to the input source $\mathbf{x}$ and seed $\mathbf{y}_{S_i}$, where the extractor is defined by the following two steps:

1. **Reed-Solomon step:** Split the source $\mathbf{x}$ into $s = \lceil n_1/l \rceil$ chunks of length l (padding the final chunk with 0's if l is not a divisor of n_1), treating each chunk as an element in $\mathbb{GF}(2^l)$, which we denote ν_j for $j \in \mathbb{Z}_s$. Then, evaluate the polynomial

$$p_{\mu_1}(\mathbf{x}) = \sum_{j=0}^{s-1} \nu_j \mu_1^{s-j+1} , \tag{3.41}$$

   where μ_1 is the polynomial associated to the first l bits of $\mathbf{y}_{S_i}$.

2. **Hadamard step:** Let $(\mathbf{y}_{S_i})_{l:2l-1} = y_l, \ldots, y_{2l-1} \in \mathbb{Z}_2^l$ denote the second l seed bits of $\mathbf{y}_{S_i}$ and $p_{\mu_1}(\mathbf{x})_j$ denote the j-th bit of the polynomial $p_{\mu_1}(\mathbf{x})$. Then, compute the final output of the RSH extractor as

$$\mathsf{RSH}(\mathbf{x}, \mathbf{y}_{S_i}) = \bigoplus_{j=0}^{l-1} y_{l+j} p_{\mu_1}(\mathbf{x})_j , \tag{3.42}$$

   where $\bigoplus$ denotes addition modulo 2.

Therefore, the i-th output bit of the Trevisan extractor is given by

$$\mathsf{Trevisan}(\mathbf{x}, \mathbf{y})_i := \mathsf{RSH}(\mathbf{x}, \mathbf{y}_{S_i}) . \tag{3.43}$$

Computation time. The block weak design step is a pre-computation and can be ignored in the overall computation time. Even if not treated as a pre-computation, it does not constitute the leading-order term, as the computation time is dominated by the m RSH extractions. The computation time for the m RSH extractions is $O(msl\log(l))$. Using the relations $l \leq \frac{t}{2} = O(\log \frac{n_1 m^2}{\epsilon^2})$ and $s = \lceil n_1/l \rceil = O(n_1/\log \frac{n_1 m^2}{\epsilon^2})$, we obtain an overall computation time of $O(mn_1 \log\log\left(\frac{n_1 m^2}{\epsilon^2}\right))$.

3.3 Two-source extractors for min-entropy sources

Two-source extractors allow the seed's min-entropy to be less than its length. They are important in a variety of protocols, for example quantum randomness

amplification e.g. [80, 81, 82, 17]. The study of these extractors was initiated in [69, 83], where it was shown that extraction was possible from two independent $(n, k \geq n/2)$ sources. Later improvements [84, 74] relaxed this to $k_1 + k_2 > n$ for independent (n, k_1) and (n, k_2) sources. Raz's breakthrough [1] further reduced the requirement to $k_1 = n/2$ and $k_2 = O(\log n)$. This opened the question of whether explicit two-source extractors could be found for two independent $(n_i, k_i = \log n_i + O(1))$ sources for $i \in \{1, 2\}$.[3] After a long line of research [86, 87, 88, 89, 90, 44, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103], such extractors were found [59].

Two-source extractors for classical adversaries are defined as follows.

Definition 31 (Classical-proof two-source extractor)**.** Let X and Y be any independent (n_1, k_1) and (n_2, k_2) sources with classical side information E. A function $\text{Ext} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \to \mathbb{Z}_2^m$ is a classical-proof $(n_1, k_1, n_2, k_2, m, \epsilon)$ two-source extractor if

$$\text{SD}(\text{Ext}(X, Y), U_m | E) \leq \epsilon \,, \tag{3.44}$$

where U_m is the uniform distribution on $\mathbb{Z}_2^m$. It is *strong* in X if $\text{SD}(\text{Ext}(X, Y) \circ X, U_m \circ X | E) \leq \epsilon$ and strong in Y if $\text{SD}(\text{Ext}(X, Y) \circ Y, U_m \circ Y) \leq \epsilon$.

A natural quantum analogue to this classical security definition is the *product source* model, first introduced in [104], where an adversary holds independent quantum side information about the source and weak seed. This notion is captured by extractors that work for a particular type of ccq-states called product sources.

Definition 32 (Product source)**.** A ccq-state ρ_{XYE} on Hilbert space $\mathcal{H}_X \otimes \mathcal{H}_Y \otimes \mathcal{H}_E$ is a $[(n_1, k_1), (n_2, k_2)]$ product source if it decomposes as

$$\rho_{XYE} = \rho_{XE_1} \otimes \rho_{YE_2} \,, \tag{3.45}$$

[3] Such extractors imply explicit Ramsey graphs on n vertices with no clique or independent set of size $O(\log n)$, solving a long-standing problem proposed by Erdös [85].

where

$$\rho_{XE_1} = \sum_{\mathbf{x}\in\mathbb{Z}_2^{n_1}} p_X(\mathbf{x})|\mathbf{x}\rangle\langle\mathbf{x}|\otimes\rho_{E_1}^{\mathbf{x}}\,, \quad \rho_{YE_2} = \sum_{\mathbf{y}\in\mathbb{Z}_2^{n_2}} p_Y(\mathbf{y})|\mathbf{y}\rangle\langle\mathbf{y}|\otimes\rho_{E_2}^{\mathbf{y}}\,, \tag{3.46}$$

for $\mathcal{H}_E = \mathcal{H}_{E_1}\otimes\mathcal{H}_{E_2}$ with $\rho_{E_1}^{\mathbf{x}}$ on $\mathcal{H}_{E_1}$, $\rho_{E_2}^{\mathbf{y}}$ on $\mathcal{H}_{E_2}$, and the cq-states ρ_{XE_1} and ρ_{YE_2} are (n_1,k_1) and (n_2,k_2) sources, respectively.

Definition 33 (Quantum-proof two-source extractor in the product source model [62])**.** A function $\mathrm{Ext}:\mathbb{Z}_2^{n_1}\times\mathbb{Z}_2^{n_2}\to\mathbb{Z}_2^m$ is a quantum-proof $(n_1,k_1,n_2,k_2,m,\epsilon)$ two-source extractor in the product source model if, for all $[(n_1,k_1),(n_2,k_2)]$ product sources ρ_{XYE},

$$\frac{1}{2}\left\|\rho_{\mathrm{Ext}(X,Y)E}-\omega_{\mathrm{Ext}(X,Y)}\otimes\rho_E\right\|_1\leq\epsilon\,, \tag{3.47}$$

where $\rho_{\mathrm{Ext}(X,Y)E} = (\mathrm{Ext}_{XY}\otimes\mathbb{1}_E)\rho_{XYE}$ and $\omega_{\mathrm{Ext}(X,Y)}$ is the maximally mixed state of dimension 2^m. The extractor is *strong in Y* if

$$\frac{1}{2}\left\|\rho_{\mathrm{Ext}(X,Y)YE}-\omega_{\mathrm{Ext}(X,Y)}\otimes\rho_{YE}\right\|_1\leq\epsilon\,, \tag{3.48}$$

where $\rho_{\mathrm{Ext}(X,Y)YE} = \sum_{\mathbf{y}} p_Y(\mathbf{y})|\mathbf{y}\rangle\langle\mathbf{y}|\otimes\rho_{\mathrm{Ext}(X,Y)E}^{\mathbf{y}}$ (and similarly for X).

We note that in the seeded case, the product source model is sufficient, since the ccq-state of the source, seed and adversary's side information can always be decomposed as $\rho_{XYE} = \rho_{XE}\otimes\omega_Y$. Two-source extractors require additional care, as the adversary may possess side information about both sources which potentially correlates them.

The Markov model, introduced in [62], relaxes the independence requirement between the input and weak seed to conditional independence, given the adversary's side information. This model is particularly useful in randomness amplification [82, 17] and QKD. In QKD, seeded randomness extractors are commonly used to post-process the raw key material generated by the protocol (a process known as *privacy amplification*). If the adversary has predictive power over the seed before

all raw key material is produced, their actions on the quantum channel could create correlations between the raw key material and the seed. This may happen for various reasons, such as if the privacy amplification seed is made public or transmitted between parties before the full raw key material is generated, or if the seed is reused across multiple instances of a protocol. These scenarios, where the source becomes correlated with the seed through a third system (e.g. a common cause), are effectively captured by the Markov model.

Similarly to the product source model, extractors secure in the Markov model extract randomness from inputs known as Markov sources.

Definition 34 (Markov source)**.** The classical random variables $X \in \mathbb{Z}_2^{n_1}$ and $Y \in \mathbb{Z}_2^{n_2}$ form a $[(n_1,k_1),(n_2,k_2)]$ classical Markov source if, given the classical adversary's side information $E \in \mathcal{E}$, X is a (n_1,k_1) source, Y is a (n_2,k_2) source, and

$$\mathrm{I}(X:Y|E) = 0\,, \tag{3.49}$$

where $\mathrm{I}(X:Y|E)$ is the mutual information[4] between X and Y conditioned on E. Similarly, the ccq-state ρ_{XYE}, representing classical random variables X, Y and quantum side information E, is a $[(n_1,k_1),(n_2,k_2)]$ quantum Markov source if X is a (n_1,k_1) source, Y is a (n_2,k_2) source, and $\mathrm{I}(X:Y|E) = 0$.

Definition 35 (Classical-proof two-source extractor in the Markov model)**.** Let $X \in \mathbb{Z}_2^{n_1}$, $Y \in \mathbb{Z}_2^{n_2}$ and E be any $[(n_1,k_1),(n_2,k_2)]$ classical Markov source. A function $\mathrm{Ext} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \to \mathbb{Z}_2^{m}$ is a classical-proof $(n_1,k_1,n_2,k_2,m,\epsilon)$ two-source extractor in the Markov model if

$$\mathrm{SD}(\mathrm{Ext}(X,Y),U_m|E) \leq \epsilon\,, \tag{3.50}$$

where U_m is the uniform distribution on $\mathbb{Z}_2^m$. It is *strong* in X if $\mathrm{SD}(\mathrm{Ext}(X,Y)\circ X, U_m \circ X|E) \leq \epsilon$ and strong in Y if $\mathrm{SD}(\mathrm{Ext}(X,Y)\circ Y, U_m \circ Y) \leq \epsilon$.

[4]The mutual information $\mathrm{I}(X:Y|E)$ is given by $\mathrm{I}(X:Y|E) = H(XE) + H(YE) - H(E) - H(XYE)$, where $H(\cdot)$ represents Shannon entropy for classical systems and von Neumann entropy for quantum systems.

Definition 36 (Quantum-proof two-source extractor in the Markov model [62])**.** A function $\text{Ext} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \to \mathbb{Z}_2^m$ is a quantum-proof $(n_1, k_1, n_2, k_2, m, \epsilon)$ two-source extractor in the Markov model if, for all $[(n_1, k_1), (n_2, k_2)]$ quantum Markov sources ρ_{XYE},

$$\frac{1}{2} \left\| \rho_{\text{Ext}(X,Y)E} - \omega_{\text{Ext}(X,Y)} \otimes \rho_E \right\|_1 \leq \epsilon \,, \tag{3.51}$$

where $\rho_{\text{Ext}(X,Y)E} = (\text{Ext}_{XY} \otimes \mathbb{1}_E)\rho_{XYE}$ and $\omega_{\text{Ext}(X,Y)}$ is the maximally mixed state of dimension 2^m. The extractor is *strong in Y* if

$$\frac{1}{2} \left\| \rho_{\text{Ext}(X,Y)YE} - \omega_{\text{Ext}(X,Y)} \otimes \rho_{YE} \right\|_1 \leq \epsilon \,, \tag{3.52}$$

where $\rho_{\text{Ext}(X,Y)YE} = \sum_{\mathbf{y}} p_Y(\mathbf{y}) |\mathbf{y}\rangle\langle\mathbf{y}| \otimes \rho^{\mathbf{y}}_{\text{Ext}(X,Y)E}$ (and similarly for X).

Kasher and Kempe [104] initiated the study of two-source extractors in the presence of side information, showing that any classical 1-bit two-source extractor (i.e., with output length $m = 1$) and the Dodis et al. extractor [74] remain secure against quantum adversaries in the product source model, albeit with increased error. Chung et al. [105] explored a more general form of side information, where the side information is generated through a specific "leaking operation". This operation allows the adversary's side information to be equivalent to providing information about only one source and they prove the security of certain multi-source extractors within this framework. In the classical case, [61] explores what happens to the security of two-source extractors when the extractor input and the weak seed can be correlated in specific models, for example, when they have bounded mutual information $\text{I}(X : Y) \leq t$, for a constant t. In [62], it was shown that any two-source extractor can be made secure in the classical and quantum Markov model by taking a penalty on the error. The results in [62] are proven in the general case of multi-source extractors, so we restate their theorems for the specific case of two-source extractors.

Lemma 37 (Classical-proof in the Markov model, adapted from Theorem 1 in [62])**.** Any (strong) classical-proof $(n_1, k_1, n_2, k_2, m, \epsilon)$ two-source extractor is a

(strong) classical-proof $(n_1, k_1 - \log(\epsilon), n_2, k_2 - \log(\epsilon), m, 3\epsilon)$ two-source extractor in the Markov model.

Lemma 38 (Quantum-proof in the Markov model, adapted from Theorem 2 in [62])**.** Any (strong) quantum-proof $(n_1, k_1, n_2, k_2, m, \epsilon)$ two-source extractor is a (strong) quantum-proof $(n_1, k_1 - \log(\epsilon), n_2, k_2 - \log(\epsilon), m, \sqrt{3\epsilon 2^{m-2}})$ two-source extractor in the Markov model.

We note that "strong" is in parentheses because strong extractors remain strong, while extractors that are not strong do not become strong. The quantum Markov model implies security in the product source model, as the product source model can be seen as a special case of the Markov model. However, quantum-proof security in the product source model also implies security in the Markov model. We summarise this observation in the following lemma.

Lemma 39. Any (strong) quantum-proof $(n_1, k_1, n_2, k_2, m, \epsilon)$ extractor in the product source model is also a (strong) quantum-proof $(n_1, k_1, n_2, k_2, m, \epsilon)$ extractor in the Markov model.

Proof. According to [106], any $[(n_1, k_1), (n_2, k_2)]$ Markov source ρ_{XYE} can be decomposed into product states as

$$\rho_{XYE} = \bigoplus_{t \in \mathcal{T}} p_T(t) \rho^t_{XE^t_X} \otimes \rho^t_{YE^t_Y} \tag{3.53}$$

where T takes values t over a finite alphabet $\mathcal{T}$ with probability distribution $p_T(t)$, and $\mathcal{H}_E = \bigoplus_t \mathcal{H}_{E^t_X} \otimes \mathcal{H}_{E^t_Y}$ is the Hilbert space of the quantum system held by the adversary. Using this decomposition, we can write the extractor security condition

as

$$\begin{aligned}
&\frac{1}{2}\left\|\rho_{\mathrm{Ext}(X,Y)E}-\omega_{\mathrm{Ext}(X,Y)}\otimes\rho_E\right\|_1\\
&=\frac{1}{2}\left\|(\mathrm{Ext}_{XY}\otimes\mathbb{1}_E)\rho_{XYE}-\omega_{\mathrm{Ext}(X,Y)}\otimes\rho_E\right\|_1\\
&=\frac{1}{2}\left\|(\mathrm{Ext}_{XY}\otimes\mathbb{1}_E)\left(\bigoplus_{t\in\mathcal{T}}p_T(t)\rho^t_{XE^t_X}\otimes\rho^t_{YE^t_Y}\right)-\omega_{\mathrm{Ext}(X,Y)}\otimes\left(\bigoplus_{t\in\mathcal{T}}\rho^t_{E^t_X}\otimes\rho^t_{E^t_Y}\right)\right\|_1\\
&=\sum_{t\in\mathcal{T}}p_T(t)\frac{1}{2}\left\|\left(\mathrm{Ext}_{XY}\otimes\mathbb{1}_{E^t_XE^t_Y}\right)\left(\rho^t_{XE^t_X}\otimes\rho^t_{YE^t_Y}\right)-\omega_{\mathrm{Ext}(X,Y)}\otimes\rho^t_{E^t_X}\otimes\rho^t_{E^t_Y}\right\|_1\\
&\leq\sum_{t\in\mathcal{T}}p_T(t)\epsilon=\epsilon
\end{aligned}\tag{3.54}$$

where $\mathrm{Ext}(X,Y)$ denotes the application of the extractor on the classical systems X,Y. In the last inequality, we have used the fact that the extractor is already quantum-proof in the product source model. Specifically, that $\left\|\left(\mathrm{Ext}_{XY}\otimes\mathbb{1}_{E^t_XE^t_Y}\right)\left(\rho^t_{XE^t_X}\otimes\rho^t_{YE^t_Y}\right)-\omega_{\mathrm{Ext}(X,Y)}\otimes\rho^t_{E^t_X}\otimes\rho^t_{E^t_Y}\right\|_1\leq\epsilon$ holds for all t, because the ccq-states $\rho^t_{XE^t_X}\otimes\rho^t_{YE^t_Y}$ are (at least) $[(n_1,k_1),(n_2,k_2)]$ product sources for all t. □

Consequently, a quantum-proof extractor in the product source model is also quantum-proof in the Markov model with the same parameters, so proving security in either model ensures security in both.

3.3.1 Extending seeded extractors to two-source extractors

Hayashi and Tsurumaru [5] prove that strong seeded extractors can be extended to two-source extractors, allowing for a weak seed at the cost of increased error (or shorter output length). Furthermore, they provided an improved extension for extractors based on two-universal families of hash functions.

Theorem 40 (Classical-proof two-source extension, Theorems 6 and 7 in [5])**.** Any strong classical-proof $(n_1,k_1,n_2,k_2=n_2,m,\epsilon)$ seeded extractor can be extended to a strong classical-proof $(n_1,k_1,n_2,k_2,m,2^{n_2-k_2}\epsilon)$ two-source extractor, strong in the (now) weak seed. If the extractor is constructed from a two-universal family of hash functions, it becomes a strong classical-proof $(n_1,k_1,n_2,k_2,m,2^{(n_2-k_2)/2}\epsilon)$ two-

source extractor.

Moreover, if the strong seeded extractor is initially quantum-proof, the resulting two-source extractor is secure against quantum side information on the weak source. However, it does not account for quantum side information on the (now weak) seed. Using the proof techniques of [76, Proposition 1], Theorem 40 can be generalised to show that any two-universal hashing-based extractor is quantum-proof in the product source model.

Theorem 41 (Quantum-proof two-source extension in the product sources model, generalisation of 'Proposition 1' in [76])**.** Any strong quantum-proof $(n_1, k_1, n_2, k_2 = n_2, m, \epsilon)$ seeded extractor based on two-universal hash functions is a strong quantum-proof $(n_1, k_2, n_2, k_2, m, 2^{(n_2-k_2)/2}\epsilon)$ two-source extractor in the product sources model, strong in the (now weak) seed.

Proof. The proof follows from [76, Proof of Proposition 1], noting that $n-1$ in the original proof can be replaced generically with n_2 (the length of the (weak) seed) and the bound on term labelled $(*)$ in the proof is derived by bounding the collision entropy of the seeded extractor, with the same bound applying to all extractors based on two-universal families of hash functions. □

Combining Theorem 41 and Lemma 39 leads to the following corollary.

Corollary 42. Any quantum-proof $(n_1, k_1, n_2, k_2 = n_2, m, \epsilon)$ seeded extractor based on two-universal hash families is a strong, in the (now weak) seed, quantum-proof $(n_1, k_1, n_2, k_2, m, 2^{(n_2-k_2)/2}\epsilon)$ two-source extractor in the Markov model.

Together, these results allow us to derive two-source extensions for any seeded extractors and we compute these parameters explicitly for our particular implementations.

Corollary 43 (Circulant two-source extractor)**.** The Circulant extractor is a strong (in the weak seed) quantum-proof (and classical-proof) $(n_1, k_1, n_2 = n_1 + 1, k_2, m, \epsilon)$ two-source extractor, for prime n_2, with

$$m \leq k_1 + (k_2 - n_2) + 2\log(\epsilon) \tag{3.55}$$

in the product source and Markov model (via Corollary 42).

Corollary 44 (Toeplitz two-source extractor)**.** The Toeplitz extractor is a strong (in the weak seed) quantum-proof (and classical-proof) $(n_1, k_1, n_2 = n_1 + m - 1, k_2, m, \epsilon)$ two-source extractor, with

$$m \leq \frac{1}{2}(k_1 + (k_2 - n_1) + 1 + 2\log(\epsilon)) \tag{3.56}$$

in the product source model and Markov model (via Corollary 42).

Note that k_2 can be larger than n_1, since the length of the (weak) seed is $n_2 = n_1 + m - 1$.

Corollary 45 (Trevisan two-source extractor)**.** The Trevisan extractor is a strong (in the weak seed) classical-proof $(n_1, k_1, n_2, k_2, m, \epsilon)$ two-source extractor, with

$$m \leq k_1 + 4(k_2 - n_2) + 4\log(\epsilon) - 4\log(m) - 6 \tag{3.57}$$

(via Theorem 40) and a strong (in the weak seed) quantum-proof $(n_1, k_1, n_2, k_2, m, \epsilon)$ two-source extractor with

$$m \leq \frac{1}{10}\left(k_1 + 4(k_2 - n_2) - 4\log(m) + 18\log(\epsilon) + 9\log\left(\frac{4}{3}\right) - 6\right) \tag{3.58}$$

in the product source and Markov model (via Lemma 38).

3.3.2 The Dodis et al. two-source extractor

The Dodis et al. extractor [74] is a function $\mathsf{DEOR} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \to \mathbb{Z}_2^m$ that requires two inputs of equal length $n = n_1 = n_2$ and uses a set of m $n \times n$ matrices, which we label $A_0, ..., A_{m-1}$, with entries in $\mathbb{Z}_2$. The matrices $A_0, ..., A_{m-1}$ must be chosen such that, for any subset $B \subseteq A_0, ..., A_{m-1}$, the sum of the matrices in the subset B has rank at least $n - r$ for a small *rank deficiency* r (which will act as a penalty to the output length). The extractor output of m bits is given by

$$\mathsf{DEOR}(\mathbf{x}, \mathbf{y}) := (A_0\mathbf{x}) \cdot \mathbf{y}, (A_1\mathbf{x}) \cdot \mathbf{y}, ..., (A_{m-1}\mathbf{x}) \cdot \mathbf{y}\ , \tag{3.59}$$

where $\cdot$ denotes the inner product modulo 2.

In [74], the authors show that this construction gives a strong (in either input) classical-proof $(n, k_1, n, k_2, m, \epsilon)$ two-source extractor with $m \leq k_1 + k_2 - n + (2 - r) + 2\log(\epsilon)$. They provide several explicit constructions, based on "cyclic shift matrices" (attributed to [83]) achieving $r = 1$, "(non-cyclic) right shift matrices" achieving $r = m - 1$, "matrices from a general error-correcting code" achieving $r = n - d$, where d is the *distance* of the selected code, and "own construction", achieving $r = 0$.

The general family of Dodis et al. extractors is not known to be quantum-proof with the same parameters, even in the seeded ($n = k_2$) setting.[5] However, we can make it quantum-proof by applying Lemma 38 which reduces the output length, leading to the following corollary.

Corollary 46 (Quantum-proof Dodis et al. extractor)**.** The Dodis et al. extractor is a strong quantum-proof $(n, k_1, n, k_2, m, \epsilon)$ two-source extractor with

$$m \leq \frac{1}{5}\left(k_1 + k_2 - n + 8\log\epsilon + 4\log\left(\frac{4}{3}\right) + (2 - r)\right), \tag{3.60}$$

in the product source and Markov model.

The construction based on cyclic shift matrices is almost optimal, achieving $r = 1$ and in what follows, we find that this construction can be implemented in computation time $O(n_1 \log n_1)$. This construction requires $n = n_1 = n_2$ to be prime with 2 as a primitive root and that the first input is not all 0 or all 1. This may seem restrictive at first, but Artin's conjecture suggests that the primes for which 2 is a primitive root have an asymptotic density in the primes equal to Artin's constant (i.e., about 37.3% of all primes). In Appendix A, we list the closest primes and primes with 2 as a primitive root for powers of 10 up to 10^{12} and powers of 2 up to 2^{40}.

[5] In the special case of $r = 0$, the extractor forms a two-universal family of hash functions and is therefore quantum-proof via the Q-LHL with the same parameters.

3.3.2.1 Efficient implementation

Consider the matrices $A_0, A_1, \ldots, A_{n-1}$ which are the so-called right cyclic shift matrices [107], defined as the $n \times n$ matrices

$$A_0 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}, \ldots . \tag{3.61}$$

As shown in [83], the sum of the matrices in any non-empty subset $B \subset \{A_i\}_{i \in \mathbb{Z}_n}$ has rank at least $n-1$ if n is prime with primitive root 2. In this form, the Dodis et al. extractor based on cyclic shift matrices, $\mathsf{DEOR} : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \to \mathbb{Z}_2^m$, can be rewritten as the matrix-vector multiplication

$$\mathsf{DEOR}(\mathbf{x}, \mathbf{y}) = (A_0\mathbf{x}) \cdot \mathbf{y}, (A_2\mathbf{x}) \cdot \mathbf{y}, ..., (A_{m-1}\mathbf{x}) \cdot \mathbf{y} \tag{3.62}$$

$$= \left(\begin{bmatrix} x_0 & x_1 & \cdots & x_{n-1} \\ x_{n-1} & x_0 & \cdots & x_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & \cdots & x_0 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} \right)_{0:m-1} \tag{3.63}$$

$$= (\mathrm{circ}(\mathbf{x})\mathbf{y})_{0:m-1} , \tag{3.64}$$

where $\mathbf{x} = (x_0, x_1, \ldots x_{n-1})$, $\mathbf{y} = (y_0, y_1, \ldots y_{n-1})$ and the subscript $0 : m-1$ denotes the first m elements (bits) of the matrix-vector multiplication. Although this construction loses one output bit compared to the optimal version, it enables the function to be rewritten as a convolution. Specifically, the extractor can be defined element-wise as

$$\mathsf{DEOR}(\mathbf{x}, \mathbf{y})_i = \Big(R(\mathbf{x}) * \mathbf{y} \Big)_i \tag{3.65}$$

for $i \in \mathbb{Z}_m$, where R is the reversal function from Definition 29. As shown in the efficient implementation of the Circulant extractor (see 3.2.1.1), being able to express the function in this form allows the Dodis et al. extractor to be implemented with $O(n \log n)$ computation time.

3.3.3 The Raz two-source extractor

In [1], Raz presents an explicit two-source extractor that is optionally strong in either input (with a penalty to the output length). The construction is based on (p', ζ)-biased generators producing outputs that are *ζ-biased for linear tests of size p'*.

Definition 47 (**(p', ζ)-biased generator**)**.** A function $G : \mathbb{Z}_2^n \to \mathbb{Z}_2^N$ is a (p', ζ)-biased generator if, for any (n, n) source X, the output $G(X) = G(X)_0, \ldots, G(X)_{N-1}$ is ζ-biased for linear tests of size p'. This means that, for any non-empty subset $\tau \subseteq \mathbb{Z}_N$ with $|\tau| \leq p'$, the variable $G(X)_\tau := \bigoplus_{i \in \tau} G(X)_i$ satisfies

$$\mathrm{SD}(G(X)_\tau, U_1) \leq \frac{\zeta}{2} . \tag{3.66}$$

Using (p', ζ)-biased generators, Raz proves the following lemma.

Lemma 48 ([1], Lemma 3.3 and Lemma 3.4)**.** Let $N = m2^{n_2}$ and consider N random variables $G_{(i,\mathbf{y})} \in \mathbb{Z}_2$ for $i \in \mathbb{Z}_m$ and $\mathbf{y} \in \mathbb{Z}_2^{n_2}$, that are ζ-biased for linear tests of size p' and can be constructed using n_1 random bits. Define $\mathsf{Raz} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \to \mathbb{Z}_2^m$ by $\mathsf{Raz}(\mathbf{x}, \mathbf{y})_i = G(\mathbf{x})_{(i,\mathbf{y})}$. Then, for any even integer $p \leq p'/m$ and any independent (n_1, k_1) and (n_2, k_2) sources, the function Raz is a classical-proof $(n_1, k_1, n_2, k_2, m, \epsilon = 2^{m/2}\gamma)$ two-source extractor with

$$\gamma \geq 2^{(n_1 - k_1)/p} \left[\zeta^{1/p} + p 2^{-k_2/2} \right] . \tag{3.67}$$

It is also a strong (in either input) classical-proof $(n_1, k_1', n_2, k_2', m, \epsilon')$ two-source

extractor for any independent (n_1, k_1') and (n_2, k_2') sources with

$$k_1' = k_1 + m/2 + 2 + \log(1/\gamma)\,, \tag{3.68}$$

$$k_2' = k_2 + m/2 + 2 + \log(1/\gamma)\,, \tag{3.69}$$

$$\epsilon' = \gamma 2^{m/2+1}\,. \tag{3.70}$$

By the appropriate choice of p and p' in Lemma 48, and using the explicit (p', ζ)-biased generator in [108, Lemma 4.1], the following analytic version of the Raz extractor is recovered.

Lemma 49 ([1], Theorem 1)**.** For any independent (n_1, k_1') and (n_2, k_2') sources and any $0 < \delta' < 1/2$, such that

$$\begin{aligned} n_1 &\geq 6\log(n_1) + 2\log(n_2)\,, \\ k_1' &\geq (1/2 + \delta')n_1 + 3\log(n_1) + \log(n_2)\,, \\ k_2' &\geq 5\log\left(n_1 - k_1'\right), \\ m &\leq \delta' \min[n_1/8, k_2'/40] - 1\,, \end{aligned} \tag{3.71}$$

there exists an explicit $(n_1, k_1', n_2, k_2', m, \epsilon')$ strong two-source extractor with $\epsilon' = 2^{-3m/2}$, that can be computed by a circuit with size polynomial in n_1 and n_2.

The condition on k_1' implies $k_1 \geq (1/2 + \delta')n_1$, while the condition on k_2' allows k_2 to be logarithmic in n_1. This enables Raz's extractor to surpass the $k_1/n_1 + k_2/n_2 > 1$ barrier required by Dodis et al. [74] and others, e.g., [76]. In this section, we present an improved construction which is efficiently implementable and has improved parameters.

Raz's extractor requires generating ζ-biased variables for linear tests of size p' and the presentation in [1] uses [108, Lemma 4.1] for this purpose. This generator uses two steps: (*i*) generate ζ-biased strings for linear tests [109, Proposition 3], and (*ii*) generate p'-wise independent strings, i.e., $(\zeta = 0)$-biased for linear tests of size p' [110, Proposition 6.5]. However, as we noted in [17, Remark 14] this approach means Raz's extractor would have a computation time of at least $O(n_1^4)$,

making it unsuitable for many applications. To address the computational inefficiency of Raz's extractor, we propose using the fast $(p', 2\zeta)$-biased generator by Meka et al. [111], which avoids multiple steps. This generator function is defined as follows.

Lemma 50 (Fast $(p', 2\zeta)$-biased generator, Section 1.1 in [111])**.** Let n and p' be positive integers such that $p' \leq n$, $\zeta > 0$, and let $\mathbb{F}$ be a finite field satisfying $|\mathbb{F}| \geq \max\{n, p'/\zeta\}$. Define two subsets $A, B \subseteq \mathbb{F}$, where $|A| = n$ and $|B| = p'/\zeta$. A function $G : B \times \mathbb{F} \to \mathbb{F}^{|A|}$ is given by:

$$G(\beta, \nu)_\upsilon := \nu \sum_{i=0}^{p'-1} (\upsilon\beta)^i, \tag{3.72}$$

for $\beta \in B$, $\nu \in \mathbb{F}$, and each $\upsilon \in A$. Fix p' be a power of two, $r := \log(p'/\zeta)$, and t a positive integer such that $2^t \geq \max\{n, 2^r\}$. Then the generator $G : \mathbb{Z}_2^r \times \mathbb{Z}_2^t \to \mathbb{Z}_2^{nt}$ is a $(p', 2\zeta)$-biased generator when the input $(\beta, \nu) \in \mathbb{Z}_2^r \times \mathbb{Z}_2^t$ is a uniformly random seed.

Notably, for a single υ (which we call a single output *block* of the generator), the generator from Lemma 50 can be computed efficiently in just $O(\log(p'))$ finite field operations. This fact is crucial for the efficient implementation we present later – since the output bits always constitute a subset of the bits from a single block.

To apply this generator to Raz's extractor, we set the seed of G as the first source of length n_1, imposing $n_1 = r + t$. The output length of the generator function (i.e. all blocks) must satisfy $nt \geq m2^{n_2}$, and the weak seed, of length n_2, is used to select which output block constitutes the final extractor output bits, i.e. which $\upsilon \in A$, imposing $n = 2^{n_2}$. Combining these constraints, we must set r and t such that (i) $n_1 = r + t$, (ii) $t \geq m$, and (iii) $2^t \geq \max\{2^{n_2}, 2^r\}$, which leads to $t \geq n_1/2$ and $r \leq n_1/2$. For simplicity, we set $r = t = n_1/2$, so $n_1/2 = \log(p'/\zeta)$ which is equivalent to $\zeta = p'2^{-n_1/2}$, where p' a power of 2 (required by Lemma 50). This parameter matching is summarised as the following lemma.

Lemma 51 (Efficient Raz extractor)**.** Using the $(p', 2\zeta)$-biased generator in Lemma 50, for any independent (n_1, k_1') and (n_2, k_2') sources such that n_1 is even and

$n_2 \leq n_1/2$, the function $\mathsf{Raz} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \to \mathbb{Z}_2^m$ defined by $\mathsf{Raz}(\mathbf{x},\mathbf{y})_i = G(\mathbf{x})_{(i,\mathbf{y})}$ for $\mathbf{x} \in \mathbb{Z}_2^{n_1}$, $\mathbf{y} \in \mathbb{Z}_2^{n_2}$ and $i \in \mathbb{Z}_m$ is a strong $(n_1, k_1', n_2, k_2', m, \gamma')$ two-source extractor with

$$k_1' = k_1 + m/2 + 2 + \log(1/\gamma)\,, \tag{3.73}$$

$$k_2' = k_1 + m/2 + 2 + \log(1/\gamma)\,, \tag{3.74}$$

$$\gamma' = \gamma 2^{m/2+1}\,, \tag{3.75}$$

where

$$\gamma \geq 2^{(n_1-k_1)/p}\left[(2\zeta)^{1/p} + p2^{-k_2/2}\right]\,, \tag{3.76}$$

for $m \leq n_1/2$, $l \leq n_2 + \log(n_1/2)$, and $p \leq 2^l/m$ where $\zeta = 2^{l-n_1/2}$ and $p' = 2^l$.

Before describing the efficient implementation, we present an improved analytic version of the Raz extractor using this construction. In particular, this can be seen as an improved version of [1, Theorem 1], presented here as Lemma 49.

Theorem 52 (Improved Raz extractor)**.** Consider independent (n_1, k_1) and (n_2, k_2) sources such that $n_2 \leq n_1/2$,

$$k_1 \geq \left(\frac{1}{2} + \delta\right) n_1 + 2\log(n_1) + 1\,, \tag{3.77}$$

$$k_2 \geq \max\left[3.2\log\left(\frac{8n_1}{k_2}\right), 40\right]\,, \tag{3.78}$$

$$m \leq \frac{1}{\lambda}\left(\frac{\delta k_2}{16} - 1\right)\,, \tag{3.79}$$

$0 < \delta < 1/2$ and $0.25 < \lambda < (\delta k_2/16 - 1)$. Then there exists an explicit classical-proof $(n_1, k_1, n_2, k_2, m, \epsilon \leq 2^{(1-4\lambda)m/2-1})$ two-source extractor, and for independent (n_1, k_1') and (n_2, k_2') sources, an explicit strong (in either input) classical-proof $(n_1, k_1', n_2, k_2', m, \epsilon' \leq 2^{(1-4\lambda)m/2})$ two-source extractor, with

$$k_1' = k_1 + 3(m+1)\,, \tag{3.80}$$

$$k_2' = k_2 + 3(m+1)\,. \tag{3.81}$$

Proof. Our proof follows the overall structure of Raz's [1] proof of Theorem 1, incorporating several improvements to achieve better parameters. We choose $N = (n_1/2)2^{n_2}$, $p' = 2^l$ where $l = \lfloor \log(m(n_1 - k_1)) \rfloor$, and $\zeta = 2^{l-n_1/2}$. Thus, $\log(p') = O(\log(n_1))$ and $p' \leq m(n_1 - k_1)$. Next, we set $r = \log(p'/\zeta) = n_1/2$ and $t = n_1/2$. By Lemma 50, the variables $G_0, \ldots, G_{N-1}$, which are 2ζ-biased for linear tests of size p', can be generated by n_1 random bits. For an even integer $p \leq p'$, let

$$\log(\gamma_1) = \frac{1}{p}\Big(n_1 - k_1 + \log(2\zeta)\Big) , \tag{3.82}$$

$$\log(\gamma_2) = (n_1 - k_1)/p + \log(p) - k_2/2 , \tag{3.83}$$

and

$$\gamma_1 + \gamma_2 = 2^{(n_1-k_1)/p}[\zeta^{1/p} + p2^{-k_2/2}] . \tag{3.84}$$

We now consider two cases.

Case 1: $k_2 < 4(n_1 - k_1)$**.** Set p to the smallest even integer larger than $8(n_1 - k_1)/k_2$. Then,

$$8(n_1 - k_1)/k_2 \leq p \leq 8n_1/k_2 . \tag{3.85}$$

Next, by inserting $\zeta = p'2^{-n_1/2} = 2^{l-n_1/2}$ and $l = \lfloor \log m(n_1 - k_1) \rfloor$, we get

$$\begin{aligned} -\log(\gamma_1) &= \frac{-1}{p}(n_1 - k_1 + l - n_1/2) \\ &= \frac{1}{p}(k_1 - n_1/2 - \lfloor \log(m(n_1 - k_1)) \rfloor - 1) . \end{aligned} \tag{3.86}$$

To lower bound the above, we use the largest value of p from Equation (3.85), since constraint 3.77 implies that $k_1 - \frac{n_1}{2} - \lfloor \log(m(n_1 - k_1)) \rfloor \geq 0$. This follows from the

fact that

$$\begin{aligned} k_1 - \frac{n_1}{2} - \lfloor \log(m(n_1-k_1)) \rfloor - 1 &> k_1 - \frac{n_1}{2} - \log\left(\frac{k_2}{\lambda 32}(n_1-k_1)\right) - 1 \\ &> k_1 - \frac{n_1}{2} - \log\left(\frac{k_2}{8}(n_1-k_1)\right) - 1 \\ &> k_1 - \frac{n_1}{2} - 2\log(n_1) - 1 \geq 0\,, \end{aligned} \tag{3.87}$$

where the first inequality uses constraint (3.79) followed by the bound $\delta < 1/2$, the second follows from the bound $\lambda > 1/4$, the third uses $k_2(n_1-k_1) < n_1^2$, and the last follows from (3.77) and the fact that $\delta > 0$. Therefore

$$\begin{aligned} -\log(\gamma_1) &= \frac{1}{p}\left(k_1 - \frac{n_1}{2} - \lfloor \log(m(n_1-k_1)) \rfloor - 1\right) \\ &\geq \frac{k_2}{8n_1}\left(k_1 - \frac{n_1}{2} - \lfloor \log(m(n_1-k_1)) \rfloor - 1\right) \\ &\geq \frac{k_2}{8n_1}\left(k_1 - \frac{n_1}{2} - 2\log(n_1) - 1\right) \\ &\geq \frac{k_2}{8n_1}\delta n_1 \\ &\geq 2(\lambda m + 1)\,, \end{aligned} \tag{3.88}$$

where the penultimate inequality uses the bound on k_1 from constraint (3.77) and the final inequality uses the bound on m from constraint (3.79). Next, we bound γ_2 using the restrictions on p in (3.85),

$$\begin{aligned} -\log(\gamma_2) &= \frac{k_1 - n_1}{p} - \log(p) + \frac{k_2}{2} \\ &\geq \frac{k_2(k_1-n_1)}{8(n_1-k_1)} - \log\left(\frac{8n_1}{k_2}\right) + \frac{k_2}{2} \\ &= \frac{3k_2}{8} - \log\left(\frac{8n_1}{k_2}\right). \end{aligned} \tag{3.89}$$

Noting that $\log(8n_1/k_2) \leq 5k_2/16$ by (3.78) and $\delta < 1/2$,

$$\frac{3k_2}{8} - \log\left(\frac{8n_1}{k_2}\right) \geq \frac{3k_2}{8} - \frac{5k_2}{16} > \frac{3k_2}{8} - \frac{3-\delta}{8}k_2 = \frac{k_2\delta}{8} \geq 2(\lambda m + 1), \tag{3.90}$$

where the final inequality comes from (3.79). Therefore, combining the bounds on γ_1 and γ_2, we get that

$$\gamma_1 + \gamma_2 \leq 2^{-2\lambda m-2} + 2^{-2\lambda m-2} = 2^{-2\lambda m-1} \,. \tag{3.91}$$

Case 2: $k_2 \geq 4(n_1 - k_1)$**.** Set $p = 2$. We find

$$\begin{aligned} -\log(\gamma_1) &= \frac{1}{p}\left(k_1 - \frac{n_1}{2} - \lfloor \log\left(m(n_1 - k_1)\right)\rfloor - 1\right) \\ &= \frac{1}{2}\left(k_1 - \frac{n_1}{2} - \lfloor \log\left(m(n_1 - k_1)\right)\rfloor - 1\right) \\ &\geq \frac{k_2}{8n_1}\left(k_1 - \frac{n_1}{2} - \lfloor \log\left(m(n_1 - k_1)\right)\rfloor - 1\right) \\ &\geq \frac{k_2}{8n_1}\delta n_1 \\ &\geq 2(\lambda m + 1) \,, \end{aligned} \tag{3.92}$$

where the first inequality follows from the constraint $k_2 \leq n_2 \leq n_1/2$, and the remainder follows the steps in Case 1. Now, we consider γ_2:

$$\begin{aligned} -\log(\gamma_2) &= \frac{k_1 - n_1}{p} - \log(p) + \frac{k_2}{2} \\ &= \frac{k_1 - n_1}{2} - \log(2) + \frac{k_2}{2} \\ &= \frac{k_1 - n_1}{2} + \frac{k_2}{2} - 1 \\ &\geq \frac{3k_2}{8} - 1 \\ &\geq 2(\lambda m + 1) \,, \end{aligned} \tag{3.93}$$

where the penultimate inequality comes from the fact that the Case 2 condition implies $n_1 - k_1 \leq k_2/4$, so $(k_1 - n_1)/2 + k_2/2 - 1 \geq -k_2/8 + k_2/2 - 1 = 3k_2/8 - 1$. The final inequality comes from the fact that $2(\lambda m + 1) \leq \frac{\delta k_2}{8} \leq \frac{3k_2}{8} - 1$, where we used (3.79) followed by (3.78). Therefore, combining the bounds on γ_1 and γ_2, we get

that

$$\gamma_1 + \gamma_2 \leq 2^{-2\lambda m-2} + 2^{-2\lambda m-2} = 2^{-2\lambda m-1} \,. \tag{3.94}$$

By Lemma 51, we obtain a $(n_1, k_1, n_2, k_2, m, \epsilon = 2^{m/2}\gamma \leq 2^{m/2}2^{-2\lambda m-1} = 2^{(1-4\lambda)m/2-1})$ two-source extractor and a strong $(n_1, k_1', n_2, k_2', m, \epsilon' \leq 2^{(1-4\lambda)m/2})$ two-source extractor, where

$$k_1' = k_1 + 3(m+1) \,, \tag{3.95}$$

$$k_2' = k_2 + 3(m+1) \,, \tag{3.96}$$

concluding the proof. □

Selecting $\lambda = 1$ in Theorem 52 (for the case $\delta k_2/16 - 1 > 1$) yields equivalent constraints for both the output length and error, as in the original Raz extractor [1, Lemma 3.6]. For this choice of λ, our constraints on k_1 and k_2 are strictly weaker than those in [1, Lemma 3.6] for any valid n_1, k_1, n_2, k_2 and m such that $n_2 \leq n_1/2$ and $k_2 \geq 8/(1 - k_1/n_1)$. The first restriction arises because, if $n_1 > n_2/2$, the fast generator from Lemma 50 does not exist. The second restriction ensures $3.2\log(8n_1/k_2) \leq 4\log(n_1 - k_1)$ (required in the proof), a constraint that is generally satisfied unless $k_1 \approx n_1$. Using the improved Raz extractor from Theorem 52, we now apply Lemma 38 to derive a quantum-proof version.

Corollary 53 (Quantum-proof Raz extractor in the Markov model)**.** For any ccq-state that is a $[(n_1, k_1), (n_2, k_2)]$ Markov source, $0 < \delta < 1/2$, and $0.75 < \lambda < (\delta k_2/16 - 1)$, such that $n_2 \leq n_1/2$, and

$$k_1 \geq \left(\frac{1}{2} + \delta\right) n_1 + 2\log n_1 + 1 \,, \tag{3.97}$$

$$k_2 \geq \max\left[3.2\log\left(\frac{8n_1}{k_2}\right), 40\right] \,, \tag{3.98}$$

$$m \leq \frac{1}{\lambda}\left(\frac{\delta k_2}{16} - 1\right) \,, \tag{3.99}$$

the improved Raz extractor from Theorem 52 is a strong quantum-proof

$(n_1, k_1', n_2, k_2', m, \epsilon \leq \sqrt{3}\,2^{(3/4-\lambda)m-1})$ two-source extractor in the Markov model, with

$$k_1' = k_1 + (2\lambda + 5/2)m + 3\,, \tag{3.100}$$

$$k_2' = k_1 + (2\lambda + 5/2)m + 3\,. \tag{3.101}$$

We note that Lemma 38 can be applied directly to the efficient Raz extractor from Lemma 51, with the resulting parameters less constrained than those in Corollary 53 due to not fixing p and p'. However, numerical optimisation is required to determine the optimal values. As a contribution, we provide code for this optimisation and analyse its performance compared to the analytic theorems (see Section 3.4 and Figure 3.4).

3.3.3.1 Efficient implementation

The improved Raz extractor can be efficiently implemented using the efficient generator from [111], described in Lemma 50. Let $p' = 2^l$, where l is a positive integer. Then,

$$G(\beta, \nu)_\upsilon = \nu \sum_{i=0}^{p'-1} (\upsilon\beta)^i = \nu \prod_{j=0}^{\log(p')-1} (1 + (\upsilon\beta)^{2^j})\,. \tag{3.102}$$

Since υ, β, ν can be viewed as elements in $\mathbb{GF}(2^t)$, the right-hand side of (3.102) (for a specific value of υ) can be computed in $O(\log(p'))$ finite field operations. We emphasise that the generator G is efficient only for computing a single (or constant number of) blocks, i.e. for a constant number of different υ's, rather than the entire output. This is sufficient for our purpose, as the $m \leq n_2 \leq n_1/2$ output bits are taken from a single block (they are the bits $G(\mathbf{x})_{(i,\mathbf{y})}$ for $i \in \mathbb{Z}_m$, for a given $\mathbf{y}$).

By using, for example, the fast finite field arithmetic based on circulant matrices [5, Appendix D], the overall computation time of Raz's extractor to $O(n_1 \log(n_1) \log(p'))$. Restricting $p' = O(n_1)$ results in a computation time of $O(n_1 \log^2 n_1)$, and we find that this restriction does not constrain the performance of the analytic Theorem 52 or the numerical approach to parameter optimisation (see

[3]).

3.4 Code implementations

We implement the Circulant, Dodis et al., Toeplitz, and Trevisan extractors in the software library `Cryptomite`, available at https://github.com/CQCL/cryptomite, or via the terminal command `pip install cryptomite`. The library is written in Python for usability and ease of installation, with performance-critical components implemented in C++ and accessible through Python. Full details are available in [2]. Example Python code showing how to perform extraction using the Circulant extractor with `Cryptomite` can be found in Figure 3.2.

```
import cryptomite
circulant = cryptomite.Circulant(n, m)
circulant.extract(input_bits, seed_bits)
```

Figure 3.2: Example of Circulant extraction using `Cryptomite`.

When implementing the convolution-based extractors (Circulant, Dodis et al. and Toeplitz), we choose a prime q over which to implement the NTT in advance and leverage compiler optimisation or custom code for computing remainders with a fixed modulus to speed up modular multiplications [112]. Specifically, we use Proth primes (primes of the form $q = a2^b + 1$, where a, b are positive integers, a is odd, and $2^b > a$), which have simple binary representations, enabling fast modular reduction without pre-computation [113]. We implement two versions of the NTT, `NTT` with $q = 3 \times 2^{30} + 1$ for values of $n_1 \leq 2^{29}$, and `bigNTT` with $q = 9 \times 2^{40} + 1$ for values of $n_1 \leq 2^{39}$.

We also implement parameter calculation modules for the Circulant, Dodis et al., Toeplitz, and Raz extractors. The parameter calculation module for the Raz extractor is of independent interest, as it performs numerical optimisation to yield significantly better extractor parameters than the analytic results of [1, Theorem 1] and our improved version, Theorem 52 (see Figure 3.4).

To the best of our knowledge, existing software implementations achieve only a small subset of the features discussed in this chapter. There are no alternative implementations of the Dodis et al. or Circulant extractors. Several implementations

of the Toeplitz extractor exist, such as [114, 115, 116] (Python) and [117] (C++ and Verilog for FPGA), but all fail to use the convolution theorem for efficient computation. There are implementations of the Trevisan extractor, for instance in [118] and [119], both in C++, with [118] offering various flexible instantiations for experts.[6] In [120], a modified Toeplitz extractor based on [5] is implemented using the FFT, allowing $n_1 \leq 10^7$. However, unlike the use of the NTT, using the FFT may introduce rounding errors, making it unsuitable for cryptographic applications.

3.5 Performance analysis

In this section we consider the performance of the extractors already described in this chapter, including some specific analysis related to applications in quantum cryptography.

3.5.1 Seeded extractors for quantum cryptography

A wide variety of tasks in quantum cryptography require seeded extractors, for example, QKD [22, 121] and quantum randomness expansion [122, 123]. In this section, we present a concrete comparison of the extractors presented in this work, which represent the state-of-the-art in terms of finite-size performance.

We compare the quantum-proof seeded extractors implemented in `Cryptomite` with near-minimal entropy loss: Circulant, Toeplitz and Trevisan, comparing both the seed length and *throughput* (number of bits produced divided by the computation time). To perform this comparison, we vary the length of the source n_1 and fix $m = n_1/2$.[7] The throughput is computed on a Apple M2 Max with 64GB processor and is the average of 5 instances. The results are shown in Figure 3.3.

We find that the Circulant extractor has the highest throughput and the shortest seed length for $n_1 \leq 10^6$. The Circulant and Toeplitz extractors achieve a throughput of $\approx$ 1Mb per second, even for large input lengths. The slight gradual decline in throughput as n_1 increases is expected to be due to inefficiencies in the central

[6] Recently, a bug was identified in the code of [118], which, at the time of writing this thesis, remains unresolved; see https://github.com/wolfgangmauerer/libtrevisan/pull/2.

[7] We note that the behaviour of the extractors varies under different parameter regimes. For example, if $m \ll n$, the throughput and seed length of the Trevisan extractor improve, as its computation time and seed length depend on both m and n_1.

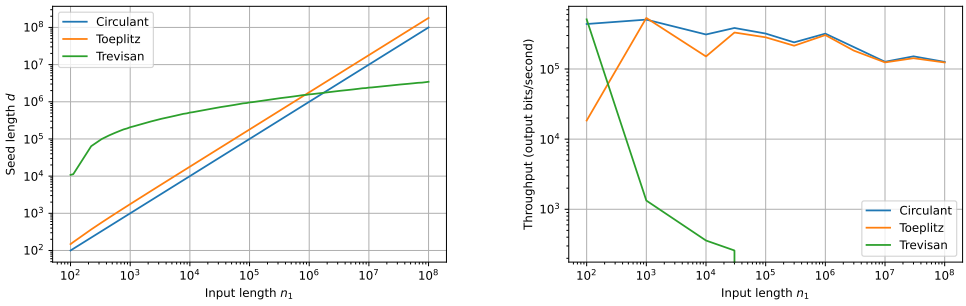

Figure 3.3: Comparison of seed lengths (left) and throughput (right) for different extractors varying the input lengths n_1 with $m = n_1/2$. The throughput is computed on a Apple M2 Max with 64GB processor and is the average of 5 instances.

processing units' data handling, such as cache hit rate and branch prediction. The Trevisan extractor achieves output speeds comparable to Circulant and Toeplitz extractors when n_1 is very small. However, for $n_1 \approx 10^4$ or larger, and still before reaching the point of minimal seed length, it fails to produce a non-trivial throughput on the laptop used for benchmarking. Overall, the Circulant extractor from `Cryptomite` currently offers the best performance for quantum cryptography protocols that require a strong quantum-proof seeded extractor. That said, future improvements to the Trevisan extractor or the use of different hardware may change this (see Open problem 1).

Further, we note that the run-time of the Circulant and Toeplitz extractors is independent of the output length (m), due to the NTT implementation handling the convolution of n_2 length vectors before outputting the first m bits. This means that, for those extractors, the throughput would double if $m \approx n_1$.

3.5.2 Two-source extractors

We analyse the performance of the two source extractors presented in this work. Among the important features of two-source extractors, a key aspect is the minimum quality of the source that can be amplified, for varying qualities of weak seed. For this analysis, we fix $n_1 = 2^{15}$, with $n_2 = n_1/2$ for the Raz extractor and $n_2 \approx n_1$ for the Dodis et al. and Circulant extractors. We set $\epsilon \leq 1/2$ and compute the classical-proof extractor parameters, since these provide the upper limits. Varying $\alpha_1 = k_1/n_1$, we compute, for each value, the minimum $\alpha_2 = k_2/n_2$ for which $m \geq 1$. The results are

presented in Figure 3.4.

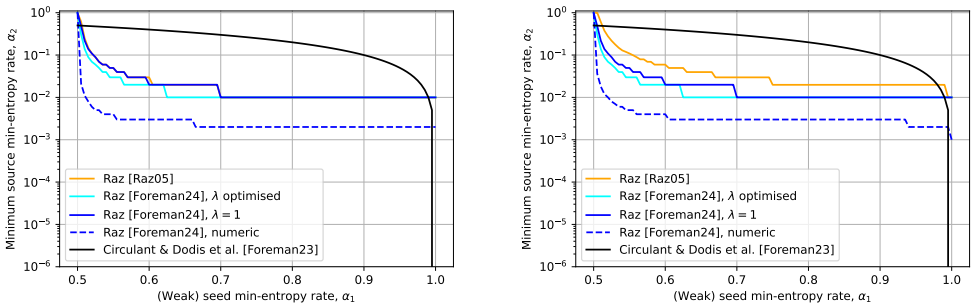

Figure 3.4: Comparison of the minimum min-entropy rate of the source, α_2, required for min-entropy rates of the (weak) seed, α_1. Left: Weak classical-proof extractors. Right: Strong classical-proof extractors. The label Raz05 refers to [1], whilst the other labels relate to those presented in this thesis, based on [2, 3].

The results show that the Raz extractor is able to amplify much weaker sources than the Circulant or Dodis et al. extractors. We find that our improved analytic theorem for the Raz extractor Theorem 52 outperforms the original theorem of Raz [1], particularly in the case of considering strong extractors. Furthermore, our results show that the numerical approach to parameter calculation for the Raz extractor provides a significant improvement over all of the analytic theorems. We note that the extractors we compare are all implemented in quasi-linear computation time.

3.5.3 Improving recent experimental demonstrations

In this section, we demonstrate how our extractor improvements can enhance recent experimental demonstrations of quantum cryptography protocols, by reducing the required seed length and quality. This analysis is applied to recent results for QKD and QRNG.

- In QKD, randomness extractors are used for *privacy amplification*. After completing other subroutines, Alice and Bob share a *raw key* that is identical but may be partially known to an adversary. Privacy amplification transforms this raw key into a near-perfect *secret key*, unknown to the adversary.
- In QRNG, randomness extractors distil near-perfect randomness from the raw output of an entropy source.

We consider the QKD experiment from [124] and the QRNG experiment based on randomness expansion from [125]. While we focus on these specific examples, the analysis can be easily adapted to other scenarios. Note that a particularly useful application for our extractors is randomness amplification and privatisation; however, as this is discussed in detail in Part II, we do not include examples here.

3.5.3.1 Privacy amplification in QKD

In the continuous variable QKD demonstration in [124], Alice and Bob obtain $n_1 = 1.738 \times 10^9$ bits of shared raw key on which privacy amplification will be performed. Privacy amplification is performed using a quantum-proof seeded extractor and the authors use the Toeplitz extractor, computing an output final secret key length of $m = 41378264$ with an extraction error $\epsilon = 10^{-10}$. For this extraction, a seed of length $d = m + n_1 - 1 = 1.738 \times 10^9 + 41378263$ bits is required.

Reduced seed length. Considering the same scenario but replacing the Toeplitz extractor with the Circulant extractor, the same amount of secret key can be generated while significantly reducing the seed length: saving 41378200 seed bits. Concretely, the Circulant extractor requires a seed length $d = n_1 + 1$ such that $n_1 + 1$ is prime, so we must perform minor manipulation of the input. Note that any (n_1, k_1) source is also an $(n_1 + c, k_1)$ source for any $c > 0$ by padding the input with c fixed bits. Using this fact, and, since the closest prime above 1.738×10^9 is $1.738 \times 10^9 + 63$, one can use the Circulant extractor with a seed length of $d' = 1.738 \times 10^9 + 63$.

Reduced seed length (inefficient computation). Considering the same scenario but replacing the Toeplitz extractor with the Trevisan extractor, almost the same amount of secret key can be generated whilst saving approximately 1.732×10^9 seed bits. Unfortunately, this reduction is interesting theoretically only, since the throughput of Trevisan's extractor tends to 0 for large input and output lengths due to its computation time scaling with both m and n_1 (see Figure 3.3).

Reduced seed entropy requirements. This experiment relies on a quantum random number generator to generate the seed [126]. For the seed to be truly uniform and independent, all QRNG components must be correctly characterised. However, such guarantees have been questioned, for instance, in [127] and if this does not hold, the

security of the protocol is compromised. To account for potential imperfections in the seed, we can consider the seed to have a min-entropy rate of $\alpha_2 = k_2/n_2 = 0.99$ (i.e. instead of 1) and adjust the output length of the extractor accordingly, using either the quantum product source or the Markov model. For the Circulant extractor, this gives an output length of $m \leq k_1 + (k_2 - n_2) + 2\log(\epsilon)$, from Corollary 43. Taking $n_2 = 1.738 \times 10^9 + 63$ as above (padding the input $n_1 \to n_1 + 62$) and adjusting $m \to \lfloor m - 0.01 n_2 \rfloor = m - 17380001$. This means that a reduction in the entropy requirements (and thus the protocol assumptions) associated with the seed for privacy amplification can be performed, at the cost of reducing the final secret key length.

One could consider using the two-source extension of the Toeplitz extractor (Corollary 44) instead of modifying the extractor. However, this approach would result in a greater reduction in the final secret key length, as the penalty term $(1 - \alpha_2)n_2$ is the same for both the Toeplitz and Circulant extractors, but the seed length n_2 is smaller for the Circulant extractor than for the Toeplitz extractor.

3.5.3.2 Randomness extraction in (Q)RNG

Randomness extraction is used in RNGs to process the raw output from an entropy source into ϵ-perfect randomness. Often, especially for QRNGs, this is performed using strong seeded extractors. The semi-device-independent QRNG experiment in [125] uses the Toeplitz extractor to extract $m = 581294933$ output bits from an input of length $n_1 = 6.5 \times 10^9$ with $k_1 = \lfloor 0.08943 \times n_1 \rfloor = 581295000$ and an extractor error $\epsilon = 10^{-10}$. This extraction requires a uniform seed of length $n_2 = n_1 + m - 1 = 7081294932$ bits.

Reduced seed length. Similarly to the previous example, using the Circulant extractor, the seed length can be reduced. In this case, it can be reduced by 581294925 bits, noting that $n_2 = 6.5 \times 10^9 + 9$. However, this requires adjusting the input length n_1 to $n_1' = 6.5 \times 10^9 + 8$, padding the original input with 8 fixed bits.

Reduced seed length (inefficient computation). Considering the same scenario but replacing the Toeplitz extractor with the Trevisan extractor, almost the same amount of secret key can be generated while saving approximately 6.49×10^9 seed bits. Unfortunately, this reduction is interesting theoretically only, since the throughput

of Trevisan's extractor because of the computation time scaling polynomially if m is linear in n_1, as noted in the previous example.

Reduced seed entropy requirements. Using a seeded extractor for RNG requires a perfect seed, which introduces circularity: uniform randomness is needed to generate more uniform randomness. Alternatively, one can assume the seed has a min-entropy of $\alpha_2 = 0.99$ by adjusting the output length under the quantum-proof product source or the Markov model, reducing protocol assumptions. For the Circulant extractor, this results in an output length of $m \leq k_1 + (k_2 - n_2) + 2\log(\epsilon)$, as shown in Corollary 43. Taking $n_2 = 6.5 \times 10^9 + 9$ (as above) and padding the input as $n_1 \to n_1 + 8$, the output length becomes $m \to \lfloor m - 0.01 n_2 \rfloor = m - 65000009$. Thus, for a reduction in the final secret key length by 65000009 bits, one can relax the entropy requirements (and hence protocol assumptions) for extraction.

3.6 Conclusion and discussion

In this chapter, we explored randomness extractors for min-entropy sources, with a focus on application to quantum cryptography. We presented constructions that improve resource efficiency, particularly regarding computation time and the quality and length of additional randomness. These extractors rely on a sufficiently independent random string, and we considered two main types: seeded and two-source extractors.

We began by investigating seeded extractors, presenting several constructions: Circulant [2], Toeplitz [75], and Trevisan [58]. We calculated their parameters and analysed their security against both classical and quantum adversaries. Efficient implementations for the Circulant and Toeplitz extractors were derived using the NTT, ensuring computational efficiency and information-theoretical security by avoiding floating-point errors. For Trevisan's extractor, we implemented a modular construction based on [6], incorporating several code-based performance improvements.

We then studied two-source extractors, where only a weak seed is available. In this context, we analysed two models: the product source model, where the adversary's information about the source and seed is independent, and the Markov

model, where the source and seed are independent conditioned on the adversary's information. From our seeded extractors, we derived two-source extractors and efficient implementations for the two-source extractors by Dodis et al. [74] and Raz [1], achieving a computation time of $O(n_1 \log n_1)$. Notably, an efficient implementation of the Dodis et al. extractor was previously considered "very unlikely" (see [5, Appendix E, Section D]). For Raz's extractor, our improved construction using the generator function from [111] represents a significant improvement, reducing the computation time from (at least) $O(n_1^4)$ to $O(n_1 \log n_1)$. Additionally, we derived improved analytic parameters for Raz's extractor.

We implemented these extractors and their parameter calculation modules in the `Cryptomite` library [2], providing complete code. Our implementations are designed for practical use in quantum cryptographic protocols, where efficiency and security are critical. To meet these demands, our implementations are numerically-precise (using the NTT) and have at most $O(mn_1 \log\log\left(\frac{n_1 m^2}{\epsilon^2}\right))$ computation time.

We then analysed the performance of our extractors. We demonstrated how these extractors can improve resource efficiency in recent QKD and QRNG experiments, reducing seed lengths and improving overall performance. Our improved analytic result for Raz's two-source extractor significantly outperforms the original construction [1], and our numerical optimisations further enhances output length and reduces entropy requirements.

In the context of resource-efficient quantum cryptography, our work presents both theoretical and practical advancements in randomness extraction. By optimising computation time and reducing resource demands (such as seed length and entropy requirements) we make quantum cryptography protocols that rely on extractors for min-entropy sources more experimentally feasible. Notably, we find that our extractors significantly reduce resource overhead in QKD and QRNG, while ensuring computation times that do not constrain state-of-the-art hardware.

Several open problems and potential future directions remain:

Open Problem 1. The Trevisan extractor [58] with near-minimal entropy loss from [6, 'Polynomial hashing'] (which we implement) boasts an asymptotically small

seed length. However, its computation time often prohibits its use for sufficiently long input lengths that would benefit from this small seed property. In particular, the computationally intensive step is to perform the m 1-bit extractors. This opens several possibilities;

a. Can the computation time of this step be reduced, for example, by adapting Trevisan's approach to perform the entire extraction in a single computation rather than iteratively?
b. The 1-bit extractors can be executed in parallel, raising the possibility of making the extractor fast in practice. Could this be implemented, for example on graphics processing unit, to achieve performance comparable with the quasi-linear computation time extractors presented in this thesis?

Open Problem 2. In this chapter, we prove that any $(n_1, k_1, n_2, k_2 = n_2, m, \epsilon)$ seeded extractor based on two-universal hash functions is a strong, in the (now weak) seed, quantum-proof $(n_1, k_2, n_2, k_2, m, 2^{(n_2-k_2)/2}\epsilon)$ two-source extractor in both the product source and Markov models. Can a similar result be achieved for other families of seeded extractors without incurring the significant parameter penalties associated with combining the classical-proof reduction of Lemma 40 and the generic quantum-proof reduction in the Markov model from Lemma 38? For example, if such a result could be obtained for Trevisan-based extractors, it could lead to a class of very good two-source extractors asymptotically, thanks to their short (asymptotic) seed length.

Open Problem 3. It remains an open problem whether the family of extractors by Dodis et al. [74] is quantum-proof with better parameters than those obtained using the generic quantum-proofing technique in Lemma 38, for any rank deficiency $r > 0$ (defined in Section 3.3.2).

Open Problem 4. In this chapter, we explored several security models for two-source extractors, specifically the Markov and product source models. However, other security models may be more relevant to practical cryptographic protocols, such as those addressing leakage or specific dependencies between sources. Inves-

tigating these models and deriving the related extractor parameters in each setting would be an interesting direction for future work.

Open Problem 5. In Section 3.3, we present an efficient implementation of Raz's extractor [1] using the generator function from [111]. However, this generator function requires that $n_2 \leq n_1/2$. It remains an open question whether:

a. An efficient implementation of Raz's extractor can be found for $n_2 > n_1/2$, enabling extraction from a significantly broader class of weakly random sources.

b. Raz's extractor is inherently strong or quantum-proof, or whether such guarantees can be established with better parameters than those provided by the generic strong reduction in [1] and the quantum-proof technique in Lemma 38.

Such a construction would be particularly useful for the task of randomness amplification (and privatisation), described in detail in Part II.

Chapter 4

Randomness extractors for Bell inequality violating sources

In this chapter, we consider randomness extractors for sources that originate from a process that violates a Bell inequality. We demonstrate that such sources can be extracted using deterministic functions and present explicit extractors. We then show that the properties required for deterministic extraction in this context align closely with the desirable properties of error-correcting codes. As a result, we find that good error-correcting codes also serve as effective extractors.

The intuition behind our results is that the violation of a Bell inequality not only guarantees a lower bound on the min-entropy of the outcomes but also ensures a degree of statistical independence between the outcomes of different rounds in the experiment. This idea is supported by the results from *self-testing*, such as the finding that maximal violation of the Clauser–Horne–Shimony–Holt (CHSH) inequality [128] implies the measured bipartite state is maximally entangled and the performed measurements are projective measurements in mutually unbiased bases [129]. Our approach consists of designing extractors that take advantage of the promise of Bell inequality violation.

The results in this chapter are derived in the scenario where the measurement devices have no memory, or equivalently, where each round of the protocol uses independent, non-communicating measurement devices. We also restrict ourselves to the CHSH Bell inequality [128] due to its simple experimental setup (although

the approach can be generalised to other Bell inequalities, see 'Conclusion and discussion' 4.8). While not fully general, this marks a significant step towards a new *seedless* approach to randomness extraction in quantum cryptography protocols that test Bell inequality violations, leaving many open questions for future exploration. From a fundamental perspective, by leveraging Bell inequality violations, we identify a new class of distributions that can be deterministically extracted from and generated by feasible experimental processes. This contributes to ongoing research in computer science on deterministic extraction from realistically generated sources [24, 39, 33, 130, 50, 96, 52] (see Section 'Weakly random sources and extractability' 2.1 for an overview). Moreover, we observe that any violation of the CHSH Bell inequality is sufficient for seedless extraction, highlighting the power of Bell non-locality as a resource [131, 132, 80, 133, 134].

It is important to note that quantum cryptography protocols relying on Bell inequalities require randomness to perform steps other than extraction. Thus, these extractors do not entirely eliminate the need for initial randomness in such protocols. However, we are optimistic that future developments using these extractors will enhance the capabilities of quantum cryptography, particularly in the context of randomness amplification [80, 135, 81, 136, 82, 17, 137].

4.1 Bell inequalities and randomness

Bell inequalities place constraints on probability distributions that classical theories must satisfy. Specifically, they apply to scenarios involving at least two non-communicating *(no-signalling)* devices, each receiving inputs and producing outputs. If the probability distribution generated by these devices violates a Bell inequality, it indicates that no locally causal model can explain the observed correlations and implies that the devices share entangled systems. The CHSH inequality [128] is one of the simplest and most studied Bell inequalities, involving two devices with binary inputs and outputs. Therefore, in this chapter we focus on the CHSH Bell inequality, although our proof techniques can be straightforwardly applied to others.

To understand the CHSH inequality, consider two devices (which we sometimes call parties), Alice and Bob, and a *verifier* who evaluates the inequality. Alice and Bob receive randomly chosen binary inputs x and y, respectively, and produce binary outputs a and b, as illustrated in Figure 4.1.

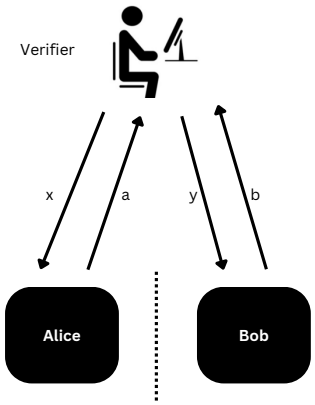

Figure 4.1: A verifier interacts with the Alice and Bob device, each receiving inputs x or y and generating outputs a and b.

After many rounds, the verifier computes the probability distribution $\{\Pr(a,b|x,y)\}_{a,b}^{x,y}$ based on the input-output combinations they obtained. Using this distribution, the CHSH inequality [128] can be written as

$$\text{CHSH} := \sum_{a,b,x,y} (-1)^{a+b+xy} \Pr(a,b|x,y) \leq 2 \,, \tag{4.1}$$

where $a,b,x,y \in \mathbb{Z}_2$. This inequality constrains the joint probability distribution of Alice and Bob, specifically to those compatible with locally causal models for their input-output relations. Importantly, it can be violated if Alice and Bob share entangled quantum systems.

This inequality can be equivalently expressed using quantum states and measurements, since classical systems can always be represented as quantum states. Assume that Alice holds a quantum system with Hilbert space $\mathcal{H}_A$ and measures it using one of two observables, labelled by $x \in \mathbb{Z}_2$, with outcomes $a \in \mathbb{Z}_2$. These

measurements are represented by the POVM elements $A(a|x)$. Similarly, assume Bob has a system on Hilbert space $\mathcal{H}_B$ and measures with one of two observables labelled $y \in \mathbb{Z}_2$ and outcomes $b \in \mathbb{Z}_2$, represented by the POVMs $B(b|y)$. The joint state on $\mathcal{H}_A \otimes \mathcal{H}_B$ is denoted by ρ_{AB}. With this notation, the CHSH inequality can be expressed as

$$\text{CHSH} := \sum_{x,y,a,b} \text{tr}\left[\rho_{AB} A(a|x) B(b|y)\right] (-1)^{a+b+xy} \leq 2 \,, \tag{4.2}$$

and we will use this formulation for the remainder of this chapter. Note that, in our notation, $A(a|x)$ is understood to only act non-trivially on $\mathcal{H}_A$, so we write $A(a|x)B(b|y)$ instead of $A(a|x) \otimes B(b|y)$. The predictability of the outcome a when measuring x can be quantified by the bias in the probability distribution of a, given by

$$\left|\Pr(a=0|x) - \Pr(a=1|x)\right| = \left|\text{tr}\left(\rho_{AB}\left[A(0|x) - A(1|x)\right]\right)\right| . \tag{4.3}$$

Next, we relate the CHSH violation to the predictability of the outcome a. To achieve this, we first define the *shifted CHSH operator* and present a theorem, proven in [138], that establishes a useful pair of semi-definite inequalities.

Definition 54 (Shifted CHSH operator)**.** For any $s \in [2, 2\sqrt{2})$, any pair of Hilbert spaces $\mathcal{H}_A, \mathcal{H}_B$, and measurements $\{A(a|x) : a = 0,1\}$ on $\mathcal{H}_A$ and $\{B(b|y) : b = 0,1\}$ on $\mathcal{H}_B$, for $x, y \in \mathbb{Z}_2$, the shifted CHSH operator is defined as

$$S := \mu_s \mathbb{1} - \nu_s \sum_{x,y,a,b=0}^{1} A(a|x) B(b|y) (-1)^{a+b+xy} \,, \tag{4.4}$$

where the coefficients are given by

$$\mu_s := 2\left(2 - \frac{s^2}{4}\right)^{-1/2} , \qquad \nu_s := \frac{s}{4}\left(2 - \frac{s^2}{4}\right)^{-1/2} . \tag{4.5}$$

Theorem 55. For any pair of Hilbert spaces $\mathcal{H}_A, \mathcal{H}_B$ and measurements $\{A(a|x) : a = 0,1\}$ on $\mathcal{H}_A$ and $\{B(b|y) : b = 0,1\}$ on $\mathcal{H}_B$, the following two semi-definite in-

equalities hold:

$$\pm\left[A(0|0)-A(1|0)\right]\leq S\ , \tag{4.6}$$

where S is the shifted CHSH operator from Definition 54.

We note that Equation (4.6) is known to be tight for all values of the CHSH inequality; that is, for any CHSH value, there exists an s such that the equation holds with equality.

The shifted CHSH operator (4.4), contains the CHSH expression (4.2), with a negative coefficient. Therefore, inequality (4.6) implies that greater CHSH violations correspond to lower predictability, as expressed by the relation

$$\left|\mathrm{tr}\left(\rho_{AB}\left[A(0|0)-A(1|0)\right]\right)\right|\leq \mathrm{tr}\left(\rho_{AB}S\right)\ . \tag{4.7}$$

This fact and its generalisations are the essence of quantum cryptography protocols that exploit Bell inequality violations, and it is crucial for the results of this chapter. For what follows, we need a generalisation, introduced in the following lemma.

Lemma 56. Consider the composite Hilbert space $\mathcal{H}_A\otimes\mathcal{H}_B$ where $\mathcal{H}_A=\bigotimes_{i=0}^{n-1}\mathcal{H}_{A_i}$ and $\mathcal{H}_B=\bigotimes_{i=0}^{n-1}\mathcal{H}_{B_i}$. For each pair of Hilbert spaces $\mathcal{H}_{A_i}$ and $\mathcal{H}_{B_i}$, and measurements $\{A_i(a_i|x_i):a_i=0,1\}$ on $\mathcal{H}_{A_i}$ and $\{B_i(b_i|y_i):b_i=0,1\}$ on $\mathcal{H}_{B_i}$, with $x_i,y_i\in\mathbb{Z}_2$, the shifted CHSH operator on $\mathcal{H}_{A_i}\otimes\mathcal{H}_{B_i}$ is defined as

$$S_i=\mu_{s_i}\mathbb{1}-\nu_{s_i}\sum_{x_i,y_i,a_i,b_i=0}^{1}A_i(a_i|x_i)B_i(b_i|y_i)(-1)^{a_i+b_i+x_iy_i}\ , \tag{4.8}$$

with coefficients

$$\mu_{s_i}=2\left(2-\frac{s_i^2}{4}\right)^{-1/2},\qquad \nu_{s_i}=\frac{s_i}{4}\left(2-\frac{s_i^2}{4}\right)^{-1/2}\ . \tag{4.9}$$

Define $C_i:=[A_i(0|0)-A_i(1|0)]$. The following two semi-definite inequalities hold

for all $s_i \in (2, 2\sqrt{2})$:

$$\pm \prod_{i=0}^{n-1} C_i \leq \prod_{i=0}^{n-1} S_i \,. \tag{4.10}$$

We note that C_i only acts non-trivially in Hilbert space $\mathcal{H}_{A_i}$ and similarly, S_i on $\mathcal{H}_{A_i} \otimes \mathcal{H}_{B_i}$. Defining S_i in this way (with the dependence on i) allows us to choose a different value of s for each index i, enabling an operator equality on each subsystem of the composite Hilbert space even when Alice's measurements vary across rounds.

Proof. For any assignment $\xi_i = \pm 1$ for all $i \in \mathbb{Z}_n$ we have $\prod_i (S_i + \xi_i C_i) \geq 0$. Averaging this product over all configurations $\{\xi_i\}$ such that $\prod_i \xi_i = 1$ gives the positive semi-definite operator

$$0 \leq \mathop{\mathbb{E}}_{\{\xi_i\}} \prod_{i=0}^{n-1} (S_i + \xi_i C_i) = \prod_{i=0}^{n-1} S_i + \prod_{i=0}^{n-1} C_i \,. \tag{4.11}$$

Similarly, averaging over all configurations $\{\xi_i\}$ such that $\prod_i \xi_i = -1$ gives

$$0 \leq \mathop{\mathbb{E}}_{\{\xi_i\}} \prod_{i=0}^{n-1} (S_i + \xi_i C_i) = \prod_{i=0}^{n-1} S_i - \prod_{i=0}^{n-1} C_i \,. \tag{4.12}$$

Since both expressions are positive semi-definite, this completes the proof. □

4.2 Deterministic extraction of Bell inequality violating sources

Having discussed Bell inequalities and their connection to randomness, we now describe how the specific Bell inequality violating sources considered in this Chapter can be generated, along with the resulting real and ideal extractor outputs.

Setup. Alice, Bob, and Eve share an arbitrary n-round state denoted ρ_{ABE} on the Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_E$. The factorisation of their Hilbert spaces enforces the no-signalling condition between Alice, Bob, and Eve. Assuming that Alice's and Bob's measurement devices are memoryless, their Hilbert spaces can be further decomposed as $\mathcal{H}_A = \bigotimes_{i=0}^{n-1} \mathcal{H}_{A_i}$ and $\mathcal{H}_B = \bigotimes_{i=0}^{n-1} \mathcal{H}_{B_i}$. Consequently, each round $i \in$

$\mathbb{Z}_n$ can be modelled independently using separate Hilbert spaces $\mathcal{H}_{A_i} \otimes \mathcal{H}_{B_i}$. Alice and Bob are assumed to have the capability to perform the measurements $\{A_i(a_i|x_i) : a_i = 0,1\}$ on $\mathcal{H}_{A_i}$ and $\{B_i(b_i|y_i) : b_i = 0,1\}$ on $\mathcal{H}_{B_i}$, respectively, where $x_i, y_i \in \mathbb{Z}_2$. These measurements can differ across rounds (as indicated by the indexing) but are causally independent, as they act on distinct Hilbert spaces.

Generating the extractor input. The process of generating the extractor input, $\mathbf{a} = (a_0, \ldots, a_{n-1}) \in \mathbb{Z}_2^n$, involves Alice sequentially performing measurements on her share of the n-round state. In particular, in each round i Alice performs the measurement $\{A_i(a_i|0) : a_i = 0,1\}$, corresponding to the input $x_i = 0$ for all i (i.e., without requiring randomness to select the measurement, unlike in Bell inequality tests). Bob, whose system resides in $\mathcal{H}_{B_i}$, does not need to perform any measurements to generate the extractor input. It is worth noting that the choice of $x = 0$ is arbitrary; it could equally be $x = 1$, or the analysis could instead be based on Bob's measurements.

We note that this generation procedure does not directly test the CHSH inequality; however, a guarantee on the CHSH violation associated with the state and measurements used during this procedure is required to (later) compute the extractor error. This guarantee can be obtained through various methods, and in Section 4.5, we present a protocol for establishing it.

Generating the extractor output. The extractor output, $\mathbf{k} = (k_0, \ldots, k_{m-1}) \in \mathbb{Z}_2^m$, is generated by applying the deterministic function $\mathrm{Ext} : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ to the extractor input, $\mathbf{a}$, such that $\mathbf{k} = \mathrm{Ext}(\mathbf{a})$. The specific functions used for this process are described in the following sections. Similarly to the previous chapter, this can be understood as applying the completely positive trace-preserving map $\mathrm{Ext}_A \otimes \mathbb{1}_B \otimes \mathbb{1}_E$ to ρ_{ABE}. Although the extractor output $\mathbf{k}$ is a classical system, we associate it with a Hilbert space $\mathcal{H}_K = \mathbb{C}^{2^m}$ and represent its values using an orthonormal basis $\{|\mathbf{k}\rangle\}_{\mathbf{k}} \in \mathcal{H}_K$. Once Alice generates the extractor output $\mathbf{k}$, after measuring her system on $\mathcal{H}_A$ and performing deterministic extraction, the joint state of the systems $\mathcal{H}_K \otimes \mathcal{H}_E$

is given by

$$\rho_{KE} = \sum_{\mathbf{k}\in\mathbb{Z}_2^m} \sum_{\mathbf{a}\in\mathbb{Z}_2^n} |\mathbf{k}\rangle\langle\mathbf{k}| \, \delta^{\mathbf{k}}_{\mathrm{Ext}(\mathbf{a})} \, \mathrm{tr}_{AB}\left[\rho_{ABE} \prod_{i=0}^{n-1} A_i(a_i|0)\right] , \tag{4.13}$$

where $\delta^{\mathbf{k}}_{\mathrm{Ext}(\mathbf{a})}$ is the Kronecker delta function.

The goal of this process is to produce a state ρ_{KE} that is indistinguishable from an ideal extractor output $\omega_K \otimes \rho_E$, satisfying

$$\frac{1}{2}\|\rho_{KE} - \omega_K \otimes \rho_E\|_1 \leq \epsilon \, , \tag{4.14}$$

where ω_K is the maximally mixed state on $\mathcal{H}_K$. This bound corresponds to the definition of quantum-proof (ϵ-)perfect randomness (Definition 5), implying that the real extractor output ρ_{KE} and the ideal extractor output $\omega_K \otimes \rho_E$ are nearly indistinguishable, with distinguishing advantage $p_{\mathrm{adv}} \leq \epsilon$.

4.3 The XOR extractor

The XOR function extracts a single bit $k \in \mathbb{Z}_2$ with an error that can be made exponentially small in n (the number of extractor input bits). This function can be computed in $O(n)$ time, demonstrating that deterministic extractors for Bell inequality violating sources need not be computationally expensive to implement.

Theorem 57 (XOR extractor)**.** After measuring the n-round state ρ_{ABE} with the observables $\{A_i(a_i|0) : a_i = 0, 1\}$ for all rounds $i = \mathbb{Z}_n$ obtaining outcomes $\mathbf{a} = (a_0, \ldots, a_{n-1}) \in \mathbb{Z}_2^n$ and applying the XOR function

$$\mathsf{XOR}(\mathbf{a}) := \bigoplus_{i=0}^{n-1} a_i \, , \tag{4.15}$$

where $\oplus$ denotes addition modulo 2, to the outcomes $k = \mathsf{XOR}(\mathbf{a})$. The resulting state ρ_{KE} as written in Equation (4.13) satisfies

$$\frac{1}{2}\|\rho_{KE} - \omega_K \otimes \rho_E\|_1 \leq \frac{1}{2}\mathrm{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} S_i\right] , \tag{4.16}$$

for all $s \in [2, 2\sqrt{2})$.

This result shows that the smaller the expectation of $\prod_i S_i$ (i.e., the larger the CHSH violation), the less distinguishable the real and ideal extractor outputs.

Proof. We start by recalling that

$$C_i := A_i(0|0) - A_i(1|0) \tag{4.17}$$

and noting that

$$A_i(a_i|0) = \frac{1}{2}(\mathbb{1} + (-1)^{a_i} C_i) \ . \tag{4.18}$$

As proven in [139], there is no loss of generality in assuming that the operators $A_i(a_i|0)$ are projectors, which implies that C_i is full-rank. Next, we substitute Equation (4.18) into the joint state after Alice generates the extractor output (4.13) and expand the product $\prod_i (\mathbb{1} + (-1)^{a_i} C_i)$ into 2^n terms labelled by the vectors $\mathbf{r} = (r_0, \ldots, r_{n-1}) \in \mathbb{Z}_2^n$, noting $k \in \mathbb{Z}_2$, $\mathbf{a} \in \mathbb{Z}_2^n$:

$$\begin{aligned} \rho_{KE} &= \sum_{k,\mathbf{a}} |k\rangle\langle k| \, \delta^k_{\mathsf{XOR}(\mathbf{a})} \, \mathrm{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} \frac{1}{2} (\mathbb{1} + (-1)^{a_i} C_i) \right] \\ &= \sum_{k,\mathbf{a}} |k\rangle\langle k| \, \delta^k_{\mathsf{XOR}(\mathbf{a})} \, \mathrm{tr}_A \left[\rho_{AE} \, 2^{-n} \sum_{\mathbf{r}} \prod_{i=1}^{n-1} (-1)^{a_i r_i} C_i^{r_i} \right] , \end{aligned} \tag{4.19}$$

where we used the power identities $C_i^0 = \mathbb{1}$ and $C_i^1 = C_i$ for full-rank operators. We then write the XOR function as a dot product modulo 2, $\mathsf{XOR}(\mathbf{a}) = \mathbf{a} \cdot \mathbf{1} \bmod 2$, where $\mathbf{1} = (1, \ldots, 1) \in \mathbb{Z}_2^n$. This allows us to express the Kronecker delta as

$$\delta^k_{\mathsf{XOR}(\mathbf{a})} = \frac{1}{2}\left(1 + (-1)^{\mathbf{a}\cdot\mathbf{1}+k}\right) . \tag{4.20}$$

Now, performing the summation

$$2^{-n}\sum_{\mathbf{a}}\delta^{k}_{\mathsf{XOR}(\mathbf{a})}(-1)^{\mathbf{a}\cdot\mathbf{r}} = 2^{-n-1}\sum_{\mathbf{a}}\left((-1)^{\mathbf{a}\cdot\mathbf{r}}+(-1)^{\mathbf{a}\cdot(\mathbf{r}\oplus\mathbf{1})+k}\right)$$
$$= 2^{-1}\left(\delta^{\mathbf{0}}_{\mathbf{r}}+(-1)^{k}\delta^{\mathbf{1}}_{\mathbf{r}}\right), \tag{4.21}$$

where $\mathbf{0} = (0,\ldots,0)$ and $\oplus$ denotes element-wise addition module 2. Substituting Equation (4.21) into Equation (4.19) gives

$$\rho_{KE} = \sum_{k}\frac{1}{2}|k\rangle\langle k|\left(\rho_E + (-1)^k \mathrm{tr}_A\left[\rho_{AE}\prod_{i=0}^{n-1}C_i\right]\right), \tag{4.22}$$

which leads to the trace norm

$$\|\rho_{KE}-\omega_K\otimes\rho_E\|_1 = \sum_{k}\frac{1}{2}\left\|(-1)^k \mathrm{tr}_A\left[\rho_{AE}\prod_{i=0}^{n-1}C_i\right]\right\|_1$$
$$= \left\|\mathrm{tr}_A\left[\rho_{AE}\prod_{i=0}^{n-1}C_i\right]\right\|_1. \tag{4.23}$$

For any Hermitian operator X and Hermitian operator H with eigenvalues ±1, the trace norm $\|X\|_1 = \max_H \mathrm{tr}[HX]$. Therefore, there exists an Hermitian operator H acting on $\mathcal{H}_E$ with spectrum ±1, such that

$$\left\|\mathrm{tr}_A\left[\rho_{AE}\prod_{i=0}^{n-1}C_i\right]\right\|_1 = \mathrm{tr}\left[\rho_{AE}\left(\prod_{i=0}^{n-1}C_i H\right)\right]. \tag{4.24}$$

Using the spectral decomposition $H = H_+ - H_-$, where $H_\pm$ are projectors on $\mathcal{H}_E$, we have

$$\left\|\mathrm{tr}_A\left[\rho_{AE}\prod_{i=0}^{n-1}C_i\right]\right\|_1 = \mathrm{tr}\left[\rho_{AE}\left(\prod_{i=0}^{n-1}C_i\right)H_+\right] - \mathrm{tr}\left[\rho_{AE}\left(\prod_{i=0}^{n-1}C_i\right)H_-\right]. \tag{4.25}$$

At this point, recall Lemma 56, which proves $\pm\prod_i C_i \le \prod_i S_i$. This implies that

$\pm(\prod_i C_i)H_\pm \leq (\prod_i S_i)H_\pm$ and therefore that

$$\pm \operatorname{tr}\left[\rho_{AE}\left(\prod_{i=0}^{n-1} C_i\right)H_\pm\right] \leq \operatorname{tr}\left[\rho_{ABE}\left(\prod_{i=0}^{n-1} S_i\right)H_\pm\right] . \tag{4.26}$$

Finally, using Equation (4.26) and the fact that $H_+ + H_- = \mathbb{1}$, we obtain

$$\left\| \operatorname{tr}_A\left[\rho_{AE}\prod_{i=0}^{n-1} C_i\right]\right\|_1 \leq \operatorname{tr}\left[\rho_{AB}\prod_{i=0}^{n-1} S_i\right] , \tag{4.27}$$

which concludes the proof. □

We remark that, if the state ρ_{AB} is initially a product state across different rounds, the bound in Theorem 57 becomes $\prod_i \operatorname{tr}(\rho_{A_iB_i}S_i)$, which is the product of the expectation of the shifted CHSH operator at each round. This means that if the states and measurements in any round maximally violate the CHSH inequality, the extractor error becomes 0. Moreover, if the state and measurements are identical at every round, the error decreases exponentially with n.

4.4 Extractors with arbitrary output length

In this section, we analyse seedless extractors that generate an extractor output $\mathbf{k}$ of arbitrary length m. The proof here relies on randomised methods, which means that we do not obtain explicit constructions, and the resulting extractors are likely to have a high computation time. However, in the next section, we present explicit functions that generate arbitrary output lengths implementable in $O(n \log n)$ time, where n is the extractor input length.

First, we prove the existence of functions, called *m-bit extractor functions*, with specific properties that are essential for deriving useful bounds in our later proofs.

Lemma 58 (m-bit extractor functions)**.** There exist functions $g : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$, for $n > 5$ and $n - m > 0$, satisfying

$$\left|\sum_{\mathbf{a}} \left(\delta^{\mathbf{k}}_{g(\mathbf{a})} - 2^{-m}\right)(-1)^{\mathbf{a}\cdot\mathbf{r}}\right| \leq n^2\, \sqrt{2}^{n-m} , \tag{4.28}$$

for all $\mathbf{k} \in \mathbb{Z}_2^m$ and all $\mathbf{r} \in \mathbb{Z}_2^n$. We call such functions *m-bit extractor functions*.

Proof. Consider a random function $G : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$, which assigns each $\mathbf{a} \in \mathbb{Z}_2^n$ to a uniformly random and independent element $\mathbf{k} \in \mathbb{Z}_2^m$. Define 2^n random variables indexed by $\mathbf{a} \in \{0,1\}^n$ as

$$X_{\mathbf{a}}(\mathbf{k}, \mathbf{r}) = \delta^{\mathbf{k}}_{G(\mathbf{a})}(-1)^{\mathbf{a}\cdot\mathbf{r}} \,. \tag{4.29}$$

For any particular value $\mathbf{k}^*$ and $\mathbf{r}^*$, these random variables are independent (since G is a random function) and satisfy the probability assignment

$$\Pr(X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^*) = 0) = (1 - 2^{-m}) \,, \tag{4.30}$$

$$\Pr\Big(X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^*) = (-1)^{\mathbf{a}\cdot\mathbf{r}^*}\Big) = 2^{-m} \,. \tag{4.31}$$

Therefore, each $X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^*)$ has mean $\mathbb{E}(X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^*)) = (-1)^{\mathbf{a}\cdot\mathbf{r}^*}2^{-m}$ and second moment $\mathbb{E}(X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^*)^2) = 2^{-m}$.

Bernstein's inequality states; for any $C > 0$ and independent random variables $X_1, \ldots X_n$,

$$\Pr\left(|\sum_{i=1}^{n} X_i - \sum_{i=1}^{n} \mathbb{E}(X_i)| \geq C\right) \leq 2\exp\left(\frac{-C^2}{2(\sum_{i=1}^{n} \mathbb{E}(X_i^2) + \frac{1}{3}\max_i |X_i| C)}\right) . \tag{4.32}$$

Using our variables $X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^*)$, we obtain

$$\begin{aligned}\Pr\left(|\sum_{\mathbf{a}} X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^*) - \sum_{\mathbf{a}} (-1)^{\mathbf{a}\cdot\mathbf{r}^*}2^{-m}| \geq C\right) &\leq 2\exp\left(\frac{-C^2}{2(\sum_{\mathbf{a}} 2^{-m} + \frac{1}{3}C)}\right) \\ &= 2\exp\left(\frac{-C^2}{2(2^{n-m} + \frac{1}{3}C)}\right) . \end{aligned} \tag{4.33}$$

When $\mathbf{r}^* = \mathbf{0}$, $\sum_{\mathbf{a}}(-1)^{\mathbf{a}\cdot\mathbf{r}^*}2^{-m} = \sum_{\mathbf{a}} 2^{-m} = 2^{n-m}$, which leads to

$$\mathrm{P}(|\sum_{\mathbf{a}} \Big(X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^* = \mathbf{0}) - 2^{-m}\Big)| \geq C) \leq 2\exp\left(\frac{-C^2}{2(2^{n-m} + \frac{1}{3}C)}\right) . \tag{4.34}$$

When $\mathbf{r}^* \neq \mathbf{0}$, $\sum_{\mathbf{a}}(-1)^{\mathbf{a}\cdot\mathbf{r}^*}2^{-m} = 0$, which leads to

$$\Pr\left(|\sum_{\mathbf{a}} X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^* \neq \mathbf{0})| \geq C\right) \leq 2\exp\left(\frac{-C^2}{2(2^{n-m} + \frac{1}{3}C)}\right). \tag{4.35}$$

Together, this can be expressed, for any $\mathbf{r}^* \in \mathbb{Z}_2^n$, as

$$\Pr\left(|\sum_{\mathbf{a}} \left(X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^*) - (-1)^{\mathbf{a}\cdot\mathbf{r}^*}2^{-m}\right)| \geq C\right) \leq 2\exp\left(\frac{-C^2}{2(2^{n-m} + \frac{1}{3}C)}\right), \tag{4.36}$$

and, setting $C = n^2\sqrt{2}^{n-m}$, leads to

$$\begin{aligned}\Pr&\left(|\sum_{\mathbf{a}} \left(X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^*) - (-1)^{\mathbf{a}\cdot\mathbf{r}^*}2^{-m}\right)| \geq n^2\sqrt{2}^{n-m}\right) \\ &\leq 2\exp\left(\frac{-n^4 2^{n-m}}{2(2^{n-m} + \frac{1}{3}n^2\sqrt{2}^{n-m})}\right) =: p^*\,.\end{aligned} \tag{4.37}$$

Next, we apply the union bound to bound the probability that Equation (4.28) holds for all $\mathbf{r}$ and $\mathbf{k}$, as

$$\begin{aligned}&\bigcap_{\mathbf{k},\mathbf{r}} \Pr\left(|\sum_{\mathbf{a}} \left(X_{\mathbf{a}}(\mathbf{k}, \mathbf{r}) - (-1)^{\mathbf{a}\cdot\mathbf{r}}2^{-m}\right)| \leq C\right) \\ &= 1 - \bigcup_{\mathbf{k},\mathbf{r}} \Pr\left(|\sum_{\mathbf{a}} \left(X_{\mathbf{a}}(\mathbf{k}, \mathbf{r}) - (-1)^{\mathbf{a}\cdot\mathbf{r}}2^{-m}\right)| \geq C\right) \\ &\geq 1 - \sum_{\mathbf{k},\mathbf{r}} \Pr\left(|\sum_{\mathbf{a}} \left(X_{\mathbf{a}}(\mathbf{k}, \mathbf{r}) - (-1)^{\mathbf{a}\cdot\mathbf{r}}2^{-m}\right)| \geq C\right) \\ &= 1 - 2^{n+m}\Pr\left(|\sum_{\mathbf{a}} \left(X_{\mathbf{a}}(\mathbf{k}^*, \mathbf{r}^*) - (-1)^{\mathbf{a}\cdot\mathbf{r}^*}2^{-m}\right)| \geq C\right) \\ &\geq 1 - 2^{n+m}p^*\,.\end{aligned} \tag{4.38}$$

Finally, to prove the existence of functions that satisfy Equation (4.28) for all $\mathbf{k}$ and $\mathbf{r}$, it suffices to show that $1 - 2^{n+m}p^* > 0$, i.e., $2^{n+m}p^* < 1$. Given $n - m > 0$ and $n > 5$;

we have that

$$\begin{aligned} n^4 2^{n-m} &> 4n2^{n-m}(1+\frac{1}{3}n^2) \\ &> 4n(2^{n-m}+\frac{1}{3}n^2\sqrt{2}^{n-m}) \\ &\geq \ln(2)(n+m)2(2^{n-m}+\frac{1}{3}n^2\sqrt{2}^{n-m})+1\,, \end{aligned} \tag{4.39}$$

which ensures $2^{n+m}p^* < 1$ and completes the proof. □

This property is crucial for proving the following theorem, as it enables us to bound each coefficient after bounding the trace distance between the real and ideal output using the triangle inequality. Specifically: (1) For $\mathbf{r} = \mathbf{0}$, this property ensures that the sizes of the sets $\{\mathbf{a} \mid g(\mathbf{a}) = \mathbf{k}\}$ are similar for all $\mathbf{k}$. (2) For $\mathbf{r} \neq \mathbf{0}$, it ensures that the sizes of the sets $\{\mathbf{a} \mid g(\mathbf{a}) = \mathbf{k} \text{ and } \mathbf{a}\cdot\mathbf{r} = 0\}$ and $\{\mathbf{a} \mid g(\mathbf{a}) = \mathbf{k}\} \text{ and } \mathbf{a}\cdot\mathbf{r} = 1\}$ are similar for all $\mathbf{k}$.

Theorem 59. Let $g : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ be a function that satisfies condition (4.28). After measuring the n-round state ρ_{ABE} with the observables $\{A_i(a_i|0) : a_i = 0,1\}$ for all $i \in \mathbb{Z}_n$, and applying the function $k = g(\mathbf{a})$ to the outcomes $\mathbf{a} = (a_0,\ldots,a_{n-1})$, the resulting state ρ_{KE} as defined in (4.13) satisfies

$$\frac{1}{2}\|\rho_{KE} - \omega_K \otimes \rho_E\|_1 \leq \frac{1}{2}n^2\sqrt{2}^{m-n}\,\mathrm{tr}\left[\rho_{AB}\prod_{i=0}^{n-1}(\mathbb{1}+S_i)\right]. \tag{4.40}$$

This result shows that the smaller the expectation of $\prod_i(\mathbb{1}+S_i)$ (i.e. the larger the violation of CHSH) the smaller the distance between the real extractor output and ideal output. Alternatively, if this distance is fixed to a specific value, such as $\|\rho_{KE} - \omega_K \otimes \rho_E\|_1 = \epsilon$, then a larger CHSH violation allows for a longer extractor output length m.

Proof. We begin by substituting Equation (4.18) into the joint state after Alice generates the extractor output (4.13), expanding the product $\prod_i(\mathbb{1}+(-1)^{a_i}C_i)$ into 2^n

terms labelled by vectors $\mathbf{r} \in \mathbb{Z}_2^n$, noting $\mathbf{k} \in \mathbb{Z}_2^m, \mathbf{a} \in \mathbb{Z}_2^n$,

$$\begin{aligned}\rho_{KE} &= \sum_{\mathbf{k},\mathbf{a}} |\mathbf{k}\rangle\langle\mathbf{k}| \delta^{\mathbf{k}}_{g(\mathbf{a})} \operatorname{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} \frac{1}{2} \left(\mathbb{1} + (-1)^{a_i} C_i \right) \right] \\ &= \sum_{\mathbf{k},\mathbf{a}} |\mathbf{k}\rangle\langle\mathbf{k}| \delta^{\mathbf{k}}_{g(\mathbf{a})} \operatorname{tr}_A \left[\rho_{AE} \, 2^{-n} \sum_{\mathbf{r}} \prod_{i=0}^{n-1} (-1)^{a_i r_i} C_i^{r_i} \right], \end{aligned} \tag{4.41}$$

where we have used the identities $C_i^0 = \mathbb{1}$ and $C_i^1 = C_i$ for full-rank operators. After substituting this in the left-hand side of Equation (4.40), applying the triangle inequality and using promise (4.28), we obtain

$$\begin{aligned}\| \rho_{KE} - \omega_K \otimes \rho_E \|_1 &= 2^{-n} \sum_{\mathbf{k}} \left\| \sum_{\mathbf{r}} \left(\sum_{\mathbf{a}} \left(\delta^{\mathbf{k}}_{g(\mathbf{a})} - 2^{-m} \right) (-1)^{\mathbf{a}\cdot\mathbf{r}} \right) \operatorname{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 \\ &\leq 2^{-n} \sum_{\mathbf{k}} \sum_{\mathbf{r}} \left| \sum_{\mathbf{a}} \left(\delta^{\mathbf{k}}_{g(\mathbf{a})} - 2^{-m} \right) (-1)^{\mathbf{a}\cdot\mathbf{r}} \right| \left\| \operatorname{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 \\ &\leq 2^{-n} \sum_{\mathbf{k}} \sum_{\mathbf{r}} n^2 \sqrt{2}^{n-m} \left\| \operatorname{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 \\ &= n^2 \sqrt{2}^{m-n} \sum_{\mathbf{r}} \left\| \operatorname{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 . \end{aligned} \tag{4.42}$$

Following the same steps as in the proof of Theorem 57, there exist two complementary projectors $H_\pm$ acting on $\mathcal{H}_E$ such that

$$\left\| \operatorname{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 = \operatorname{tr} \left[\rho_{AE} \left(\prod_{i=0}^{n-1} C_i^{r_i} \right) H_+ \right] - \operatorname{tr} \left[\rho_{AE} \left(\prod_{i=0}^{n-1} C_i^{r_i} \right) H_- \right] . \tag{4.43}$$

By applying Lemma 56, which states $\pm \prod_i C_i \leq \prod_i S_i$, we have

$$\pm \prod_i C_i^{r_i} \leq \prod_i S_i^{r_i} \tag{4.44}$$

for any $\mathbf{r}$, which implies

$$\pm \operatorname{tr} \left[\rho_{AE} \left(\prod_{i=0}^{n-1} C_i^{r_i} \right) H_\pm \right] \leq \operatorname{tr} \left[\rho_{ABE} \left(\prod_{i=0}^{n-1} S_i^{r_i} \right) H_\pm \right] . \tag{4.45}$$

Substituting this back into Equation (4.43) and noting that $H_+ + H_- = \mathbb{1}$, we get

$$\left\| \mathrm{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 \leq \mathrm{tr} \left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i} \right] . \tag{4.46}$$

Finally, substituting this result into Equation (4.42), we obtain

$$\begin{aligned} \|\rho_{KE} - \omega_K \otimes \rho_E\|_1 &\leq n^2 \sqrt{2}^{m-n} \sum_{\mathbf{r}} \mathrm{tr} \left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i} \right] \\ &= n^2 \sqrt{2}^{m-n} \, \mathrm{tr} \left[\rho_{AB} \prod_{i=0}^{n-1} (\mathbb{1} + S_i) \right] , \end{aligned} \tag{4.47}$$

which concludes the proof. □

4.5 Application to quantum cryptography

In this section, we construct a toy protocol to demonstrate that the relevant quantities can be estimated experimentally and thus, our extractors can be used in quantum cryptography protocols. Specifically, we focus on Theorems 57 and 59, which establish a relationship between the error ϵ, the extractor output length m, and the Bell inequality violation quantified by $\langle \prod_i S_i \rangle$ or $\langle \prod_i (\mathbb{1} + S_i) \rangle$.

This Bell-violation quantifier differs from the one commonly used in Bell inequality-based quantum cryptography protocols with standard seeded extractors, namely $\langle \sum_i S_i \rangle$. For large n, the statistical fluctuations of $\sum_i S_i$ are small, allowing us to relate the expectation $\langle \sum_i S_i \rangle$ to the specific value of $\sum_i S_i$. However, the quantities $\prod_i S_i$ and $\prod_i (\mathbb{1} + S_i)$ in our bounds exhibit strong fluctuations and cannot be constrained by standard techniques.

In this section, we introduce a proof technique for bounding $\langle \prod_i (\mathbb{1} + S_i) \rangle$ and $\langle \prod_i S_i \rangle$ using estimation data. Our approach employs the *spot-checking* procedure commonly used in quantum cryptography protocols, making it broadly applicable. In this procedure, rounds are randomly selected (with some bias) to be used either for estimating the Bell violation or for generating the extractor input. This random selection limits potential malicious behaviour by the devices. Concretely, to implement this procedure and prove security, the required protocol assumptions are:

- The devices and adversary operate according to quantum theory.
- The classical computer used for processing and statistical analysis is trusted and functions correctly.
- The quantum device comprises two isolated parts that do not exchange information during each round of the experiment (i.e., they are no-signalling).
- The quantum device does not signal to the adversary.
- The measurement devices are memoryless, meaning the measurements in each round act non-trivially on distinct Hilbert spaces and are independent of previous rounds.

Importantly, (1) we make no assumptions about the state, which can be arbitrary and may exhibit correlations between rounds, and (2) although the measurements are modelled to act on a separate Hilbert space in each round, they do not need to be identical. While our protocol demonstrates that the relevant quantities needed to quantify the extractor error can be estimated, it is likely far from optimal, as discussed in more detail in 'Conclusion and discussion' 4.8.

4.5.1 Spot-checking protocol for 1-bit extraction

The spot-checking protocol for 1-bit extraction using the XOR extractor is outlined in Figure 4.2. While the expressions for the extractor output length (4.48) and subsequent maximisation constraints may appear unintuitive, they are the most general form that allows us to prove the security condition in Theorem 60 later in this section.

4.5.2 Spot-checking protocol for m-bit extraction

The spot-checking protocol for m-bit extraction using m-bit extractor functions follows the same structure as the XOR extractor in Figure 4.2, with the Data Processing step replaced by the procedure in Figure 4.3. This protocol also meets the security condition Theorem 60, as shown below.

4.5.3 Security proof

We now prove the security of the 1-bit and m-bit spot-checking protocols described above. The security condition we establish is a variable output length statement, en-

SPOT-CHECKING PROTOCOL FOR XOR EXTRACTION

1. Set parameters: Define n (total rounds), $p_{\mathsf{e}} \in (0,1)$ (estimation probability), and $\epsilon_{\text{sec}} > 0$ (tolerable error).

2. Data generation: For each round $l \in \mathbb{Z}_n$:

2a. Generate $t_l \in \{\texttt{estimation}, \texttt{rawbit}\}$ with probabilities p_{e} and $p_{\mathsf{r}} = 1 - p_{\mathsf{e}}$.

2b. If $t_l = \texttt{estimation}$:
- Generate random variables $x_l, y_l \in \mathbb{Z}_2$ with $\Pr(x_l, y_l) = 1/4$.
- Perform local measurements $A_l(a_l|x_l)$ and $B_l(b_l|y_l)$ with outcomes $a_l, b_l \in \mathbb{Z}_2$.
- Record $z_l = a_l + b_l + x_l y_l \bmod 2$, for CHSH inequality evaluation.

2c. If $t_l = \texttt{rawbit}$:
- Perform local measurement $A_l(a_l|0)$.
- Record a_l as part of the extractor input.

3. Data processing:

3a. Compute n_{e}, the number of rounds with $t_l = \texttt{estimation}$, and compile the *estimated data* by relabelling as $\mathbf{z} = (z_0, \ldots, z_{n_{\mathsf{e}}-1})$.

3b. Compute $n_{\mathsf{r}} = n - n_{\mathsf{e}}$, and compile the extractor input by relabelling as $\mathbf{a} = (a_0, \ldots, a_{n_{\mathsf{r}}-1})$.

3c. Calculate the extractor output length $m(\mathbf{t}, \mathbf{z})$ using:

$$m(\mathbf{t},\mathbf{z}) = \begin{cases} 1, & \text{if } \max\limits_{s,\alpha_0,\alpha_1,\beta} \sum_{j=0}^{n_{\mathsf{e}}-1} \alpha_{z_j} + (\beta-1) n_{\mathsf{r}} + 2\log(\epsilon_{\text{sec}}) \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4.48}$$

The maximisation is over $s \in [2, 2\sqrt{2})$, and $\alpha_0, \alpha_1, \beta \in \mathbb{R}$, subject to:

$$p_{\mathsf{r}} \sqrt{2}^{\beta-1} (\mu_s - 4\nu_s) + p_{\mathsf{e}} \sqrt{2}^{\alpha_0} = 1, \tag{4.49}$$

$$p_{\mathsf{r}} \sqrt{2}^{\beta-1} (\mu_s + 4\nu_s) + p_{\mathsf{e}} \sqrt{2}^{\alpha_1} = 1. \tag{4.50}$$

3d. If $m(\mathbf{t}, \mathbf{z}) > 0$: Generate the extractor output $k = \mathsf{XOR}(\mathbf{a}) \in \mathbb{Z}_2^{m(\mathbf{t},\mathbf{z})}$ by applying the XOR function $\mathsf{XOR} : \mathbb{Z}_2^{n_{\mathsf{r}}} \to \mathbb{Z}_2^{m(\mathbf{t},\mathbf{z})}$ (as defined in 4.15).

Figure 4.2: Spot-checking protocol for XOR extraction.

suring that the protocol output is indistinguishable from an ideal extractor output for any output length produced. Typically, protocols produce a fixed output of length m or abort. While allowing variable output lengths is less significant for the 1-bit extractor, it is valuable for the m-bit extractor, as it allows for a larger acceptable set of estimation data, making a protocol (in principle) more robust to fluctuating

SPOT-CHECKING PROTOCOL FOR m-bit EXTRACTION

3. Data processing: Define ϵ_{sec}, $\mathbf{t}$, n_{e}, n_{r}, j, i, $\mathbf{z}$, $\mathbf{a}$ as in Figure 4.2.

3a. Calculate the length of the extractor output as a function of $\mathbf{t}$, $\mathbf{z}$ with

$$m(\mathbf{t},\mathbf{z}) = \left\lfloor \max_{s,\alpha_0,\alpha_1,\beta} \sum_{j=0}^{n_{\mathsf{e}}-1} \alpha_{z_j} + \beta n_{\mathsf{r}} + 2\log(\epsilon_{\text{sec}}) - 4\log(n_{\mathsf{r}}) \right\rfloor . \tag{4.51}$$

where the maximisation over the parameters $s \in [2, 2\sqrt{2})$ and $\alpha_0, \alpha_1, \beta \in \mathbb{R}$ is constrained by

$$p_{\mathsf{r}}\sqrt{2}^{\beta-1}(1+\mu_s-4\nu_s) + p_{\mathsf{e}}\sqrt{2}^{\alpha_0} = 1 , \tag{4.52}$$

$$p_{\mathsf{r}}\sqrt{2}^{\beta-1}(1+\mu_s+4\nu_s) + p_{\mathsf{e}}\sqrt{2}^{\alpha_1} = 1 . \tag{4.53}$$

3b. If $m(\mathbf{t},\mathbf{z}) > 0$: Generate the extractor output $\mathbf{k} = g(\mathbf{a}) \in \mathbb{Z}_2^{m(\mathbf{t},\mathbf{z})}$ by applying to the extractor input a function $g : \mathbb{Z}_2^{n_{\mathsf{r}}} \to \mathbb{Z}_2^{m(\mathbf{t},\mathbf{z})}$ satisfying (4.28).

Figure 4.3: Spot-checking protocol for m-bit extraction.

device performance. For further discussion, see [140].

Theorem 60. The protocols for 1-bit and m-bit extraction (Figure 4.2 and Figure 4.3) generate an extractor output $\rho_{KE}^{\mathbf{t},\mathbf{z}}$ satisfying the following security condition

$$\sum_{\mathbf{t}} \sum_{\mathbf{z}} \Pr(\mathbf{t},\mathbf{z}) \left\| \rho_{KE}^{\mathbf{t},\mathbf{z}} - \omega_K^{\mathbf{t},\mathbf{z}} \otimes \rho_E^{\mathbf{t},\mathbf{z}} \right\|_1 \leq \epsilon_{\text{sec}} , \tag{4.54}$$

where $\omega_K^{\mathbf{t},\mathbf{z}} \otimes \rho_E^{\mathbf{t},\mathbf{z}}$ denotes the ideal key $\omega_K \otimes \rho_E$ conditioned on the estimated data $\mathbf{t}$ and $\mathbf{z}$, which, for example, impacts the dimension of ω_K.

Proof. The random variables $\mathbf{t} \in \{\texttt{estimation}, \texttt{rawbit}\}^n$ are independent and identically distributed according to the probabilities p_{e} and p_{r} respectively. If in round l we have $t_l = \texttt{estimation}$ then the systems $\mathcal{H}_{A_l}$ and $\mathcal{H}_{B_l}$ are included in $\mathcal{H}_{A_{\mathsf{e}}} = \bigotimes_{j=0}^{n_{\mathsf{e}}-1} \mathcal{H}_{A_j}$ and $\mathcal{H}_{B_{\mathsf{e}}} = \bigotimes_{j=0}^{n_{\mathsf{e}}-1} \mathcal{H}_{B_j}$, and used for estimation. If $t_l = \texttt{rawbit}$ then the systems $\mathcal{H}_{A_l}$ and $\mathcal{H}_{B_l}$ are included in $\mathcal{H}_{A_{\mathsf{r}}} = \bigotimes_{i=0}^{n_{\mathsf{r}}-1} \mathcal{H}_{A_i}$ and $\mathcal{H}_{B_{\mathsf{r}}} = \bigotimes_{i=0}^{n_{\mathsf{r}}-1} \mathcal{H}_{B_i}$, and used for generating the extractor input. Without loss of generality, we can assume that $\mathbf{t}$ is initially generated before any measurement (and kept private from the devices), and right after, we can re-order the rounds and write the global state as

$\rho^{\mathbf{t}}_{A_\mathsf{r}B_\mathsf{r}A_\mathsf{e}B_\mathsf{e}E}$, i.e. a state on the system $\mathcal{H}_{A_\mathsf{e}} \otimes \mathcal{H}_{B_\mathsf{e}} \otimes \mathcal{H}_{A_\mathsf{r}} \otimes \mathcal{H}_{B_\mathsf{r}} \otimes \mathcal{H}_E$ conditioned on $\mathbf{t}$.

In each estimation round $j \in \mathbb{Z}_{n_\mathsf{e}}$ the state on system $\mathcal{H}_{A_j} \otimes \mathcal{H}_{B_j}$ is measured with

$$Q_{z_j} = \sum_{a,b,x,y} \frac{1}{4} A_j(a|x) B_j(b|y) \delta^{z_j}_{a+b+xy \bmod 2} , \tag{4.55}$$

to obtain outcome $z_j \in \mathbb{Z}_2$. This produces the estimation data $\mathbf{z} = (z_0, \ldots, z_{n_\mathsf{e}-1})$, distributed according to

$$\Pr(\mathbf{z}|\mathbf{t}) = \mathrm{tr}\left[\rho^{\mathbf{t}}_{A_\mathsf{e}B_\mathsf{e}} \prod_{j=0}^{n_\mathsf{e}-1} Q_{z_j}\right] . \tag{4.56}$$

The global state conditioned on a particular value of the estimation data $\mathbf{z}$ is

$$\rho^{\mathbf{t},\mathbf{z}}_{A_\mathsf{r}B_\mathsf{r}E} = \frac{1}{\Pr(\mathbf{z}|\mathbf{t})} \mathrm{tr}_{A_\mathsf{e}B_\mathsf{e}}\left[\rho^{\mathbf{t}}_{A_\mathsf{r}B_\mathsf{r}A_\mathsf{e}B_\mathsf{e}E}\left(\prod_{j=0}^{n_\mathsf{e}-1} Q_{z_j}\right)\right] . \tag{4.57}$$

We start by proving the case when using the XOR extractor, as described in Figure 4.2. By using Theorem 57 and the function which defines the output length of the XOR extractor in our spot checking protocol, Equation (4.48), the left-hand side of (4.54) can be upper bound by

$$\begin{aligned} &\sum_{\mathbf{t}} \sum_{\mathbf{z}} \Pr(\mathbf{t},\mathbf{z}) \left\| \rho^{\mathbf{t},\mathbf{z}}_{KE} - \omega^{\mathbf{t},\mathbf{z}}_{K} \otimes \rho^{\mathbf{t},\mathbf{z}}_{E} \right\|_1 \\ \leq &\sum_{\mathbf{t}} \sum_{\mathbf{z}} \Pr(\mathbf{t},\mathbf{z})\, m(\mathbf{t},\mathbf{z})\, \mathrm{tr}\left[\rho^{\mathbf{t},\mathbf{z}}_{A_\mathsf{r}B_\mathsf{r}} \prod_{i=0}^{n_\mathsf{r}-1} S_i\right] , \end{aligned} \tag{4.58}$$

since, in the case $m(\mathbf{t},\mathbf{z}) = 0$, the term inside the trace norm is 0 and in the case $m(\mathbf{t},\mathbf{z}) = 1$, the error can be bound using Theorem 57. Now, using the global state conditioned on a particular value of the estimation data $\mathbf{z}$ (4.57) and the facts that

$\Pr(\mathbf{t},\mathbf{z}) = \Pr(\mathbf{t})\Pr(\mathbf{z}|\mathbf{t}) = p_{\mathsf{e}}^{n_{\mathsf{e}}}\, p_{\mathsf{r}}^{n_{\mathsf{r}}} \Pr(\mathbf{z}|\mathbf{t})$ and

$$m(\mathbf{t},\mathbf{z}) = \begin{cases} 1 & \text{if} \quad \max\limits_{\substack{s\in[2,2\sqrt{2}) \\ \alpha_0,\alpha_1,\beta\in\mathbb{R}}} \sum_{j=1}^{n_{\mathsf{e}}} \alpha_{z_j} + (\beta-1)n_{\mathsf{r}} + 2\log(\epsilon_{\text{sec}}) \geq 0 \\ 0 & \text{otherwise} \end{cases} , \tag{4.59}$$

is upper bound by $\sqrt{2}^{\sum_{j=1}^{n_{\mathsf{e}}} \alpha_{z_j} + (\beta-1)n_{\mathsf{r}} + 2\log(\epsilon_{\text{sec}})}$, we bound (4.58)

$$\begin{aligned} &\leq \sum_{\mathbf{t}} \sum_{\mathbf{z}} p_{\mathsf{e}}^{n_{\mathsf{e}}}\, p_{\mathsf{r}}^{n_{\mathsf{r}}}\, \sqrt{2}^{\sum_{j=0}^{n_{\mathsf{e}}-1} \alpha_{z_j} + (\beta-1)n_{\mathsf{r}}}\, \epsilon_{\text{sec}} \operatorname{tr}\left[\rho^{\mathbf{t}}_{A_{\mathsf{r}}B_{\mathsf{r}}A_{\mathsf{e}}B_{\mathsf{e}}} \prod_{i=0}^{n_{\mathsf{r}}-1} S_i \prod_{j=0}^{n_{\mathsf{e}}-1} Q_{z_j}\right] \\ &= \epsilon_{\text{sec}} \sum_{\mathbf{t}} \operatorname{tr}\left[\rho^{\mathbf{t}}_{A_{\mathsf{r}}B_{\mathsf{r}}A_{\mathsf{e}}B_{\mathsf{e}}} \prod_{i=0}^{n_{\mathsf{r}}-1} \left(p_{\mathsf{r}}\sqrt{2}^{\beta-1} S_i\right) \prod_{j=0}^{n_{\mathsf{e}}-1} \left(p_{\mathsf{e}}\left[\sqrt{2}^{\alpha_0} Q_{0_j} + \sqrt{2}^{\alpha_1} Q_{1_j}\right]\right)\right] \\ &= \epsilon_{\text{sec}} \operatorname{tr}\left[\rho_{A_{\mathsf{g}}B_{\mathsf{g}}} \prod_{l=0}^{n-1} \left(p_{\mathsf{r}}\sqrt{2}^{\beta-1} S_l + p_{\mathsf{e}}\left[\sqrt{2}^{\alpha_0} Q_{0_l} + \sqrt{2}^{\alpha_1} Q_{1_l}\right]\right)\right] , \end{aligned} \tag{4.60}$$

where we denote the global Hilbert spaces of Alice and Bob $\rho_{A_{\mathsf{g}}B_{\mathsf{g}}}$ on system $\mathcal{H}_{A_{\mathsf{g}}} \otimes \mathcal{H}_{B_{\mathsf{g}}} = \mathcal{H}_{A_{\mathsf{r}}} \otimes \mathcal{H}_{A_{\mathsf{e}}} \otimes \mathcal{H}_{B_{\mathsf{r}}} \otimes \mathcal{H}_{B_{\mathsf{e}}}$. Finally, using the identities

$$S = \mu_s \mathbb{1} - 4\nu_s (Q_0 - Q_1) , \tag{4.61}$$

$$\mathbb{1} = Q_0 + Q_1 , \tag{4.62}$$

we can write each of the factors in Equation (4.60) as

$$\begin{aligned} &p_{\mathsf{r}}\sqrt{2}^{\beta-1} S + p_{\mathsf{e}}\left[\sqrt{2}^{\alpha_0} Q_0 + \sqrt{2}^{\alpha_1} Q_1\right] \\ &= p_{\mathsf{r}}\sqrt{2}^{\beta-1}\left[\mu_s(Q_0+Q_1) - 4\nu_s(Q_0-Q_1)\right] + p_{\mathsf{e}}\left[\sqrt{2}^{\alpha_0} Q_0 + \sqrt{2}^{\alpha_1} Q_1\right] \\ &= \left[p_{\mathsf{r}}\sqrt{2}^{\beta-1}(\mu_s - 4\nu_s) + p_{\mathsf{e}}\sqrt{2}^{\alpha_0}\right] Q_0 + \left[p_{\mathsf{r}}\sqrt{2}^{\beta-1}(\mu_s + 4\nu_s) + p_{\mathsf{e}}\sqrt{2}^{\alpha_1}\right] Q_1 \\ &= Q_0 + Q_1 = \mathbb{1} , \end{aligned} \tag{4.63}$$

where the penultimate equality follows from imposing conditions

$$p_{\mathsf{r}}\sqrt{2}^{\beta-1}(\mu_s - 4\nu_s) + p_{\mathsf{e}}\sqrt{2}^{\alpha_0} = 1 , \tag{4.64}$$

$$p_{\mathsf{r}}\sqrt{2}^{\beta-1}(\mu_s + 4\nu_s) + p_{\mathsf{e}}\sqrt{2}^{\alpha_1} = 1 , \tag{4.65}$$

expressed in (4.49) and (4.50). Substituting Equation (4.63) back into (4.60) gives us the bound (4.54) and completes the proof. □

Similarly, we now complete the same proof in the m-bit extractor function case, as described in Section 4.5.2. In this case, we introduce an indicator function

$$I(m(\mathbf{t},\mathbf{z})) = \begin{cases} 1 & \text{if} \quad m(\mathbf{t},\mathbf{z}) > 0 \\ 0 & \text{otherwise} , \end{cases} \tag{4.66}$$

to encode the case when no output is produced and the extractor error is 0. By using Theorem 59 and substituting the global state conditioned on the estimation data (4.57), the output length (4.51) and the indicator function (4.66), we can write the left-hand side of (4.54) as follows

$$\begin{aligned}
&\sum_{\mathbf{t}} \sum_{\mathbf{z}} \Pr(\mathbf{t},\mathbf{z}) \left\| \rho_{KE}^{\mathbf{t},\mathbf{z}} - \omega_{K}^{\mathbf{t},\mathbf{z}} \otimes \rho_{E}^{\mathbf{t},\mathbf{z}} \right\|_1 \\
\leq &\sum_{\mathbf{t}} \sum_{\mathbf{z}} \Pr(\mathbf{t},\mathbf{z})\, I(m(\mathbf{t},\mathbf{z}))\, \sqrt{2}^{m(\mathbf{t},\mathbf{z}) - n_{\mathsf{r}} + 4\log(n_{\mathsf{r}})} \operatorname{tr}\left[\rho_{A_{\mathsf{r}}B_{\mathsf{r}}}^{\mathbf{t},\mathbf{z}} \prod_{i=0}^{n_{\mathsf{r}}-1} (\mathbb{1} + S_i) \right] \\
\leq &\sum_{\mathbf{t}} \sum_{\mathbf{z}} \Pr(\mathbf{t},\mathbf{z})\, \sqrt{2}^{m(\mathbf{t},\mathbf{z}) - n_{\mathsf{r}} + 4\log(n_{\mathsf{r}})} \operatorname{tr}\left[\rho_{A_{\mathsf{r}}B_{\mathsf{r}}}^{\mathbf{t},\mathbf{z}} \prod_{i=0}^{n_{\mathsf{r}}-1} (\mathbb{1} + S_i) \right] \\
= &\sum_{\mathbf{t}} \sum_{\mathbf{z}} p_{\mathsf{e}}^{n_{\mathsf{e}}}\, p_{\mathsf{r}}^{n_{\mathsf{r}}}\, \sqrt{2}^{\sum_{j=0}^{n_{\mathsf{e}}-1} \alpha_{z_j} + (\beta-1) n_{\mathsf{r}}} \epsilon_{\mathrm{sec}} \operatorname{tr}\left[\rho_{A_{\mathsf{r}}B_{\mathsf{r}}A_{\mathsf{e}}B_{\mathsf{e}}}^{\mathbf{t}} \prod_{i=0}^{n_{\mathsf{r}}-1} (\mathbb{1} + S_i) \prod_{j=0}^{n_{\mathsf{e}}-1} Q_{z_j} \right] \\
= &\epsilon_{\mathrm{sec}} \sum_{\mathbf{t}} \operatorname{tr}\left[\rho_{A_{\mathsf{r}}B_{\mathsf{r}}A_{\mathsf{e}}B_{\mathsf{e}}}^{\mathbf{t}} \prod_{i=0}^{n-1} \left(p\, \sqrt{2}^{\beta-1} [\mathbb{1} + S_i] \right) \prod_{j=0}^{n_{\mathsf{e}}-1} \left(p_{\mathsf{e}} \left[\sqrt{2}^{\alpha_0} Q_{0_j} + \sqrt{2}^{\alpha_1} Q_{1_j} \right] \right) \right] \\
= &\epsilon_{\mathrm{sec}} \operatorname{tr}\left[\rho_{A_{\mathsf{g}}B_{\mathsf{g}}} \prod_{l=0}^{n-1} \left(p\, \sqrt{2}^{\beta-1} [\mathbb{1} + S_l] + p_{\mathsf{e}} \left[\sqrt{2}^{\alpha_0} Q_{0_l} + \sqrt{2}^{\alpha_1} Q_{1_l} \right] \right) \right] ,
\end{aligned} \tag{4.67}$$

where we again denote the state on the global Hilbert spaces of Alice and Bob by $\rho_{A_{\mathsf{g}}B_{\mathsf{g}}}$ on system $\mathcal{H}_{A_{\mathsf{g}}} \otimes \mathcal{H}_{B_{\mathsf{g}}} = \mathcal{H}_{A_{\mathsf{r}}} \otimes \mathcal{H}_{A_{\mathsf{e}}} \otimes \mathcal{H}_{B_{\mathsf{r}}} \otimes \mathcal{H}_{B_{\mathsf{e}}}$. Using the identities for S and $\mathbb{1}$ from Equations (4.61) and (4.62), we can write each of the factors in

Equation (4.67) as

$$\begin{aligned}
&p\,\sqrt{2}^{\beta-1}\,[\mathbb{1}+S]+p_{\mathsf{e}}\Big[\,\sqrt{2}^{\alpha_0}Q_0+\,\sqrt{2}^{\alpha_1}Q_1\Big]\\
&=p_{\mathsf{r}}\,\sqrt{2}^{\beta-1}\,[(1+\mu_s)(Q_0+Q_1)-4\nu_s\,(Q_0-Q_1)]+p_{\mathsf{e}}\Big[\,\sqrt{2}^{\alpha_0}Q_0+\,\sqrt{2}^{\alpha_1}Q_1\Big]\\
&=\Big[p_{\mathsf{r}}\,\sqrt{2}^{\beta-1}(1+\mu_s-4\nu_s)+p_{\mathsf{e}}\,\sqrt{2}^{\alpha_0}\Big]Q_0+\Big[p_{\mathsf{r}}\,\sqrt{2}^{\beta-1}(1+\mu_s+4\nu_s)+p_{\mathsf{e}}\,\sqrt{2}^{\alpha_1}\Big]Q_1\\
&=Q_0+Q_1=\mathbb{1}\,,
\end{aligned}\tag{4.68}$$

where the penultimate equality follows from the conditions expressed in (4.52) and (4.53). Substituting this back into Equation (4.67) gives us the bound (4.54) and completes the proof. □

Notably, the above proof provides context to the earlier choice of output length in each protocol, as expressed in Equation (4.48) and Equation (4.51). These choices enable the distribution of free terms (which constitute the free terms in the maximisation) across relevant factors in the proof. While this general form adds flexibility, certain identities must still hold, and these are enforced by the maximisation constraints; see Equations (4.63) and (4.68) in the proof.

Moreover, the output length $m(\mathbf{t},\mathbf{z})$ for the XOR extractor is upper bound by an exponential in the proof. This bound is tight for large n, as it approximates a step function with an exponential, ensuring no asymptotic impact on the proof. However, for m-bit extractor functions, a loose upper bound is used and tightening this bound could potentially improve the results.

4.5.4 Performance

We now evaluate the performance of the spot-checking protocols introduced in this section. First, consider the XOR extractor, where we define the relative frequency of the estimation outcomes as

$$q_z=\frac{|\{z_j=z:j\in\mathbb{Z}_{n_{\mathsf{e}}}\}|}{n_{\mathsf{e}}}\,,\tag{4.69}$$

for $z \in \mathbb{Z}_2$. This is related to the CHSH expression (4.2) by

$$\text{CHSH} = 4(q_0 - q_1) = 8q_0 - 4 \,. \tag{4.70}$$

For any permutation σ of $\mathbf{z} = (z_0, z_1, \ldots, z_{n_e-1})$, we have $m(\mathbf{t}, \sigma(\mathbf{z})) = m(\mathbf{t}, \mathbf{z})$, so m depends on $\mathbf{z}$ only through the relative frequency q_0. Similarly, it also depends on $\mathbf{t}$ via its relative frequency n_e/n which, in the large-n limit, tends to p_e. Assuming a constant error $\epsilon_{\text{sec}} > 0$, the condition for positive yield (i.e., $m = 1$) in the large-n limit is

$$\max_{s,\alpha_0,\alpha_1,\beta} p_e(\alpha_0 q_0 + \alpha_1 q_1) + \beta p_r \geq 0 \,, \tag{4.71}$$

subject to the constraints (4.49) and (4.50). Figure 4.4 shows the minimum value of CHSH required for a positive yield as a function of p_e. Interestingly, for a sufficiently high estimation probability ($p_e \geq 0.74$), an extractor output bit can be generated with arbitrarily small violation of the CHSH inequality for any constant error. However, Figure 4.4 shows that a necessary requirement for extracting a single bit is $p_e > 0.5$. This contrasts with protocols that use seeded extractors, where only a small proportion of rounds are needed for estimation (e.g. [141]), highlighting a limitation of our estimation method (see 'Conclusion and discussion' 4.8 for further comment).

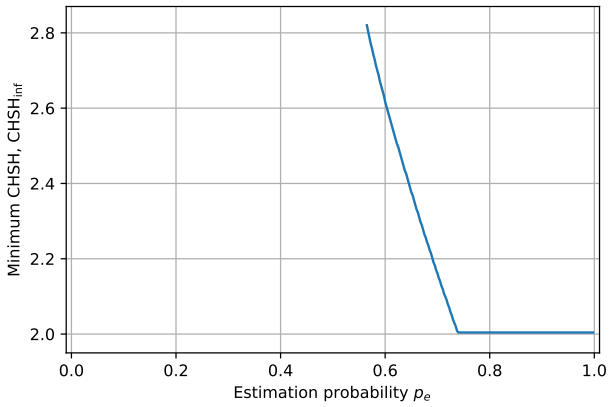

Figure 4.4: The minimum CHSH value (4.71) that our XOR extractor (presented in 57) can produce a single bit with arbitrarily small error in the large-n regime.

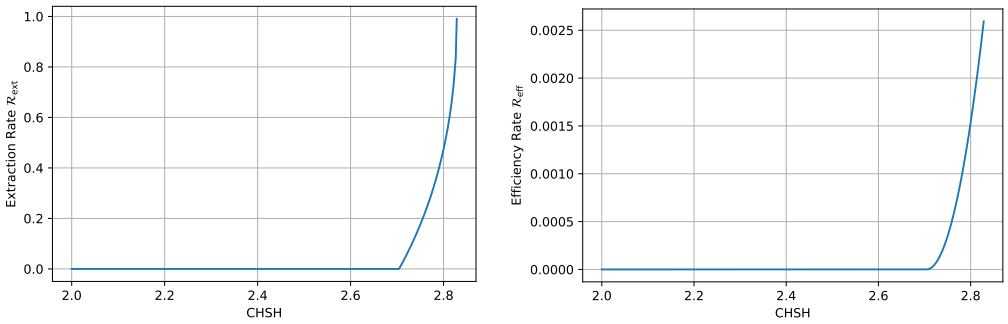

Figure 4.5: Left: The maximum extraction rate, $\mathcal{R}_{\text{ext}}$, Equation (4.72), for different values of CHSH. Right: The maximum efficiency rate, $\mathcal{R}_{\text{eff}}$, Equation (4.73), for our m-bit extractor functions, for different values of CHSH.

For the m-bit protocol, using the expression for the extractor output length, Equation (4.51), we derive the efficiency rate $\mathcal{R}_{\text{eff}}$ and the extraction rate $\mathcal{R}_{\text{ext}}$. The extraction rate, indicating the number of extractor output bits per extractor input bit, in the asymptotic limit as $n \to \infty$ is given by

$$\mathcal{R}_{\text{ext}} = \lim_{n\to\infty} \frac{m(\mathbf{t},\mathbf{z})}{n_{\text{r}}} = \frac{p_{\text{e}}}{p_{\text{r}}}(\alpha_0 q_0 + \alpha_1 q_1) + \beta \,. \tag{4.72}$$

The efficiency rate, representing the number of output bits per round, in the asymptotic limit as $n \to \infty$ is given by

$$\mathcal{R}_{\text{eff}} = \lim_{n\to\infty} \frac{m(\mathbf{t},\mathbf{z})}{n} = p_{\text{e}}(\alpha_0 q_0 + \alpha_1 q_1) + \beta p_{\text{r}} \,. \tag{4.73}$$

Figure 4.5 shows the maximum values of $\mathcal{R}_{\text{eff}}$ and $\mathcal{R}_{\text{ext}}$ as a function of the CHSH parameter, optimised over $p_{\text{e}}, \alpha_0, \alpha_1$ and β. The extraction rate approaches 1, demonstrating that our m-bit extractor functions are near-optimal in the large-n regime with high CHSH violations. However, the efficiency rate is low. This occurs because a significant proportion of rounds is required for estimation, similar to the case with our XOR extractor. Therefore, improving the estimation techniques for the Bell value quantifier would significantly enhance the performance of this protocol.

4.6 Extractors from error-correcting codes

In this section, we consider seedless extractors $G : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ defined by an $m \times n$ matrix G with entries in $\mathbb{Z}_2$, such that

$$\mathbf{k} = G\mathbf{a} \mod 2 \,. \tag{4.74}$$

It is important to highlight that the linear function $\mathbf{k} = G\mathbf{a}$ can be evaluated in at most $O(mn)$ computation time. While the results in this section are partial, they provide a valuable foundation for further study. We begin by establishing a useful relationship between linear functions and extractors for CHSH-violating sources.

Lemma 61 (Linear functions as seedless extractors)**.** For any full row rank $m \times n$ matrix G with entries in $\mathbb{Z}_2$, we define the indicator function

$$I_G(\mathbf{r}) := \begin{cases} 0 & \text{if} \quad \mathbf{r} \notin \text{span}(G) \\ \\ 1 & \text{if} \quad \mathbf{r} \in \text{span}(G) \end{cases} , \tag{4.75}$$

where $\mathbf{r} \in \mathbb{Z}_2^n$ and $\text{span}(G) \subseteq \mathbb{Z}_2^n$ denotes is the subspace spanned by the rows of G. After measuring the n-round state ρ_{ABE} using the observables $\{A_i(a_i|0) : a_i = 0, 1\}$ for all $i \in \mathbb{Z}_n$, and applying the linear function $G\mathbf{a} = \mathbf{k}$ to the outcomes $\mathbf{a} = (a_0, \ldots, a_{n-1})$, the resulting state ρ_{KE} (as described in (4.13)) satisfies

$$\frac{1}{2}\|\rho_{KE} - \omega_K \otimes \rho_E\|_1 \leq \frac{1}{2} \sum_{\mathbf{r} \neq \mathbf{0}} I_G(\mathbf{r}) \, \text{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i}\right] . \tag{4.76}$$

Proof. We begin by substituting Equation (4.18) into the state Equation (4.13) and expanding the product $\prod_i (\mathbb{1}_i + (-1)^{a_i} C_i)$ into 2^n terms indexed by vectors $\mathbf{r} \in \mathbb{Z}_2^n$,

noting $\mathbf{k} \in \mathbb{Z}_2^m, \mathbf{a} \in \mathbb{Z}_2^n$,

$$\begin{aligned}\rho_{KE} &= \sum_{\mathbf{k},\mathbf{a}} |\mathbf{k}\rangle\langle\mathbf{k}| \, \delta_{G\mathbf{a}}^{\mathbf{k}} \operatorname{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} \frac{1}{2} \left(\mathbb{1}_i + (-1)^{a_i} C_i \right) \right] \\ &= \sum_{\mathbf{k},\mathbf{a}} |\mathbf{k}\rangle\langle\mathbf{k}| \, \delta_{G\mathbf{a}}^{\mathbf{k}} \operatorname{tr}_A \left[\rho_{AE} \sum_{\mathbf{r}} 2^{-n} \prod_{i=0}^{n-1} (-1)^{a_i r_i} (C_i)^{r_i} \right] \\ &= \sum_{\mathbf{k}} |\mathbf{k}\rangle\langle\mathbf{k}| \operatorname{tr}_A \left[\rho_{AE} \sum_{\mathbf{r}} 2^{-n} \sum_{\mathbf{a}} \delta_{G\mathbf{a}}^{\mathbf{k}} (-1)^{\mathbf{a}\cdot\mathbf{r}} \prod_{i=0}^{n-1} (C_i)^{r_i} \right] \end{aligned} \tag{4.77}$$

where we have used the identities $C_i^0 = \mathbb{1}$ and $C_i^1 = C_i$, since these are full-rank operators. Next, since G has full row rank, for all $\mathbf{k} \in \mathbb{Z}_2^m$, there exists at least one $\mathbf{a}_\mathbf{k} \in \mathbb{Z}_2^n$ such that $\mathbf{k} = G\mathbf{a}_\mathbf{k}$. This allows us to rewrite the Kronecker delta as

$$\delta_{G\mathbf{a}}^{\mathbf{k}} = \prod_{j=0}^{m-1} \left(G_j \cdot (\mathbf{a} \oplus \mathbf{a}_\mathbf{k}) + 1 \bmod 2 \right) , \tag{4.78}$$

where G_j denotes the $(j+1)$-th row of G and $\oplus$ denotes element-wise addition modulo two. In this form, we see that the Kronecker delta is 1 only if $\mathbf{a} \oplus \mathbf{a}_\mathbf{k}$ is in the kernel of G. Evaluating the summation over $\mathbf{a}$, we get

$$\begin{aligned} 2^{-n} \sum_{\mathbf{a}} \delta_{G\mathbf{a}}^{\mathbf{k}} (-1)^{\mathbf{a}\cdot\mathbf{r}} &= 2^{-n} \sum_{\mathbf{a} | \mathbf{a}\oplus\mathbf{a}_\mathbf{k} \in \ker(G)} (-1)^{\mathbf{a}\cdot\mathbf{r}} \\ &= 2^{-n} (-1)^{\mathbf{a}_\mathbf{k}\cdot\mathbf{r}} \sum_{\mathbf{v}\in\ker(G)} (-1)^{\mathbf{v}\cdot\mathbf{r}} \\ &= \begin{cases} 2^{-m} (-1)^{\mathbf{a}_\mathbf{k}\cdot\mathbf{r}} & \text{if} \quad \mathbf{v}\cdot\mathbf{r} = 0 \quad \forall\, \mathbf{v} \in \ker(G) \\ 0 & \text{otherwise} , \end{cases} \end{aligned} \tag{4.79}$$

where $\ker(G)$ denotes the kernel of G and contains 2^{n-m} elements (by the rank-nullity theorem, and noting the rank of G is m). The set of all $\mathbf{r}$ satisfying $\mathbf{v}\cdot\mathbf{r} = 0$, for every $\mathbf{v} \in \ker(G)$, corresponds to the row space of G, denoted $\operatorname{span}(G)$. Therefore, substituting Equation (4.79) into Equation (4.77), and using the indicator function

$I_G(\cdot)$ (which indicates whether or not $\cdot$ is in span(G)) from Equation (4.75), we have

$$\rho_{KE} = \sum_{\mathbf{k}} |\mathbf{k}\rangle\langle\mathbf{k}| \left(\rho_E + \mathrm{tr}_A \left[\rho_{AE} \sum_{\mathbf{r}\neq\mathbf{0}} 2^{-m} (-1)^{\mathbf{a}_\mathbf{k}\cdot\mathbf{r}} I_G(\mathbf{r}) \prod_{i=0}^{n-1} C_i^{r_i} \right] \right) . \tag{4.80}$$

Substituting this into the left-hand side of Equation (4.76) and applying the triangle inequality, we get

$$\begin{aligned} \|\rho_{KE} - \omega_K \otimes \rho_E\|_1 &= \sum_{\mathbf{k}} \left\| \mathrm{tr}_A \left[\rho_{AE} \sum_{\mathbf{r}\neq\mathbf{0}} 2^{-m} (-1)^{\mathbf{a}_\mathbf{k}\cdot\mathbf{r}} I_G(\mathbf{r}) \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 \\ &\leq \sum_{\mathbf{k}} \sum_{\mathbf{r}\neq\mathbf{0}} \left| 2^{-m} (-1)^{\mathbf{a}_\mathbf{k}\cdot\mathbf{r}} I_G(\mathbf{r}) \right| \left\| \mathrm{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 \\ &\leq \sum_{\mathbf{k}} \sum_{\mathbf{r}\neq\mathbf{0}} 2^{-m} I_G(\mathbf{r}) \left\| \mathrm{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 \\ &\leq \sum_{\mathbf{r}\neq\mathbf{0}} I_G(\mathbf{r}) \left\| \mathrm{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 . \end{aligned} \tag{4.81}$$

Now, following the same steps as in the previous proofs, there exist two complementary projectors $H_\pm$ acting on $\mathcal{H}_E$ such that

$$\left\| \mathrm{tr} \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 = \mathrm{tr} \left[\rho_{AE} \left(\prod_{i=0}^{n-1} C_i^{r_i} \right) H_+ \right] - \mathrm{tr} \left[\rho_{AE} \left(\prod_{i=0}^{n-1} C_i^{r_i} \right) H_- \right] . \tag{4.82}$$

By applying Lemma 56, which states $\pm \prod_i C_i \leq \prod_i S_i$, we have

$$\pm \prod_i C_i^{r_i} \leq \prod_i S_i^{r_i} \tag{4.83}$$

for any $\mathbf{r}$, which implies

$$\pm \mathrm{tr} \left[\rho_{AE} \left(\prod_{i=0}^{n-1} C_i^{r_i} \right) H_\pm \right] \leq \mathrm{tr} \left[\rho_{ABE} \left(\prod_{i=0}^{n-1} S_i^{r_i} \right) H_\pm \right] . \tag{4.84}$$

Substituting this back into Equation (4.82) and noting that $H_+ + H_- = \mathbb{1}$, we get

$$\left\| \mathrm{tr}_A \left[\rho_{AE} \prod_{i=0}^{n-1} C_i^{r_i} \right] \right\|_1 \leq \mathrm{tr} \left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i} \right] . \tag{4.85}$$

Finally, substituting this result into Equation (4.81), we obtain

$$\|\rho_{KE} - \omega_K \otimes \rho_E\|_1 \leq \sum_{\mathbf{r} \neq \mathbf{0}} I_G(\mathbf{r}) \, \mathrm{tr} \left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i} \right] , \tag{4.86}$$

which concludes the proof. □

To understand the implications of this lemma, we consider an idealised scenario where the state and measurements are product (i.e., the state and measurements for a particular round have support on distinct factors of the full Hilbert space) and identical in every round. Furthermore, we consider that this state and measurements violate the CHSH inequality. In this scenario, we can write $\mathrm{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i}\right] = \prod_{i=0}^{n-1} \langle S_1^{r_i} \rangle$, with $\langle S_1 \rangle < 1$ and $\langle S_1^0 \rangle = 1$. Define the *Hamming weight* function $w : \mathbb{Z}_2^n \to \mathbb{Z}_{n+1}$, which counts the number of non-zero elements in the input. Then, the expectation of the shifted CHSH operator term for a specific $\mathbf{r}$ is given by

$$\prod_{i=0}^{n-1} \mathrm{tr}\left[\rho_{AB} S_i^{r_i}\right] = \langle S_1 \rangle^{w(\mathbf{r})} . \tag{4.87}$$

In this form, we see that the Hamming weight of $\mathbf{r}$ determines the exponent and, since $\langle S_1 \rangle < 1$, larger exponents lead to smaller contributions to the sum. Thus, it is preferable for all elements of span(G) (ignoring the all zero vector $\mathbf{0}$) to have a high Hamming weight. Furthermore, it is necessary to be able to compute and analyse the row-space of G, to compute the error. However, for a general matrix G, enumerating all elements in span(G) and understanding their properties is computationally intractable.

Binary linear error-correcting codes are defined by a $k \times n$ generator matrix G with entries in $\mathbb{Z}_2$, where the rows of G form a basis of the code space, meaning the codewords are the elements of span(G). These codes are characterised by the

parameters (n,k,d), where n is the length of the codewords, k is the length of the message and d is the minimum *Hamming distance* between distinct codewords. The minimum distance d guarantees that

$$d \leq \min_{\mathbf{r}\neq\mathbf{r}'\in \mathrm{Span}(G)} w(\mathbf{r}\oplus\mathbf{r}') \,, \tag{4.88}$$

where $\oplus$ denotes element-wise addition modulo 2. The code space (span(G)) of binary linear error-correcting codes is well-explored, and the guarantee on the minimum distance provides additional structure that helps analyse its elements. Notably, using a generator matrix G from an (n,k,d) code in Equation (4.76) ensures that all terms in the sum with $w(\mathbf{r}) < d$ vanish. These observations motivate the exploration of binary linear error-correcting codes as seedless extractors for Bell inequality violating sources. In what follows, we will denote codes as (n,m,d) codes instead of the standard (n,k,d) as, in our use case, k will denote the extractor output length which we typically denote by m. We now present several explicit extractor constructions based on binary linear error-correcting codes.

4.6.1 Repetition code

The first family of explicit extractors we consider generates an extractor output of 1 bit with minimal error. The repetition code is a simple error-correcting code in which each bit of the original message is repeated multiple times to introduce redundancy. Specifically, it is an $(n,1,n)$ code that encodes a single bit into n bits by repeating it n times. The generator matrix for the repetition code is the $1\times n$ matrix

$$G := \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} . \tag{4.89}$$

This defines the XOR function, which, for an input $\mathbf{a} = (a_0,\dots,a_{n-1}) \in \mathbb{Z}_2^n$, is

$$\mathsf{XOR}(\mathbf{a}) = \bigoplus_{i=0}^{n-1} a_i \,, \tag{4.90}$$

where $\oplus$ denotes addition modulo 2, and is computable in $O(n)$ time. Inspecting the generator matrix G, we find that $\mathrm{span}(G) = \{\mathbf{0}, \mathbf{1}\}$ (noting that $\mathbf{1}$ denotes $\{1\}^n$). Applying this in Lemma 61 provides an alternative proof for Theorem 57.

However, Lemma 61 allows us to generalise this result by using an m-bit repetition code. The repetition code on m bits is, for a divisor m of n, an $(n, m, d = n/m)$ code with the generator matrix

$$G := \begin{bmatrix} \mathbf{1} & & \\ & \ddots & \\ & & \mathbf{1} \end{bmatrix}, \tag{4.91}$$

where $\mathbf{1} = \{1\}^{n/m}$, and blank entries are understood to be zeros. For this error-correcting code, $\mathrm{span}(G)$ is straightforward to compute and consists of 2^m elements. Specifically, the codewords are of the form $\{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{m-1}\}$, where each $\mathbf{c}_i \in \{\{0\}^{n/m}, \{1\}^{n/m}\}$. This allows direct application of Lemma 61, due to the simplicity of the code space.

Efficient implementation. The generator matrix of the m-bit repetition code can be viewed as a combination of the im-th rows, for all $i \in \mathbb{Z}_{n/m}$, of the $n \times n$ circulant matrix $\mathrm{circ}(\{1\}^{n/m}, \{0\}^{n-n/m})$. This structure enables efficient implementation of the generator matrix's action on the extractor input by performing the matrix-vector multiplication $\mathrm{circ}(\{1\}^{n/m}, \{0\}^{n-n/m})\mathbf{a}$ and retaining every m-th element. Specifically, we can write the extractor output element-wise as $(G\mathbf{a})_i = (\mathrm{circ}(\{1\}^{n/m}, \{0\}^{n-n/m})\mathbf{a})_{im}$ for $i \in \mathbb{Z}_m$. Therefore, since the function $\mathrm{circ}(\{1\}^{n/m}, \{0\}^{n-n/m})\mathbf{a}$ can be computed in $O(n \log n)$ time using the NTT methods described in Section 2.3, so can extraction with the m-bit repetition code.

4.6.2 Bose–Chaudhuri–Hocquenghem (BCH) codes

The second family of explicit seedless extractors use the generator matrix of *primitive binary BCH codes* to produce an output of m bits, where m depends on the length of the extractor input, n, and the tolerable error, ϵ. These (n, m, d) codes are parametrised by two integers l and t, and satisfy the relationships $n = 2^l - 1$,

$m \geq n - lt$, and $d \geq 2t + 1$ (for further details, see [142, Chapter 9, Section 1]). With this parametrisation, for a fixed n, there is flexibility to vary m (corresponding to the output length) and d (relating to the final extractor error). The generator matrix G of the primitive binary BCH codes is constructed by performing m right cyclic shifts of specific coefficients, $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{Z}_2^n$, which are computed based on the particular code instance. Concretely, this $m \times n$ generator matrix is

$$G := \begin{bmatrix} c_0 & c_1 & \ldots & c_{n-2} & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \ldots & c_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}. \tag{4.92}$$

Efficient implementation. The generator matrix G of a BCH code consists of the first m rows of the circulant matrix generated by $\mathbf{c}$, circ($\mathbf{c}$). This structure allows the function $\mathbf{k} = G\mathbf{a}$, for any primitive binary BCH code, to be computed in quasi-linear computation time using methods outlined in Section 2.3. Specifically, $G\mathbf{a}$ can be represented as the first m elements of a convolution, which can then be implemented in $O(n \log n)$ computation time via the NTT. In fact, this function is identical to the Circulant extractor from Section 3, replacing the randomly generated source $\mathbf{x}$ with the deterministic coefficients $\mathbf{c}$, and the (weak) seed $\mathbf{y}$ with the extractor input $\mathbf{a}$.

The derivation of the coefficients $\mathbf{c}$ for G is described in [142, Chapter 9], but we provide example coefficients for the $n = 15$ BCH codes for varying minimum distances in Example 62.

Example 62 (Explicit $\mathbf{c}$ for the $n = 15$ BCH codes)**.** Consider the $(n = 15, m \geq 15 - 4t, d \geq 2t + 1)$ BCH code with specific configurations based on minimum distance d:

- For $d \geq 3$ (i.e., $t = 1$): $c_0 = c_1 = c_4 = 1$, all other $c_i = 0$.
- For $d \geq 5$ (i.e., $t = 2$): $c_0 = c_4 = c_6 = c_7 = c_8 = 1$, all other $c_i = 0$.
- For $d \geq 7$ (i.e., $t = 3$): $c_0 = c_1 = c_2 = c_4 = c_5 = c_8 = c_{10} = 1$, all other $c_i = 0$.
- For $d \geq 9$ (i.e., $t = 4$): $c_i = 1$ for all $i \in \mathbb{Z}_{15}$.

Interestingly, the parametrisation based on l and t for these codes is known to be not tight in some cases (see, for example, the BCH code tables in [142, Chapter

9, Section 4]). Even in Example 62, the lower bounds defining the code are not always tight. For instance, when $t = 4$, the actual minimum distance is $d = 15$, and $m \geq -1$, yet one can choose $m = 1$. Similarly, when $t = 3$ with $m \geq 3$, it is possible to choose $m = 5$. Regardless, the fact that primitive binary BCH codes form a family of $(n = 2^l - 1, m \geq n - lt, d \geq 2t + 1)$ codes for any integers t and l enables us to derive the following theorem.

Theorem 63 (Primitive binary BCH codes)**.** Let G be an $m \times n$ generator matrix of a $(n = 2^l - 1, m \geq n - lt, d \geq 2t + 1)$ BCH code for any positive integers l and t. After measuring the n-round state ρ_{ABE} with the observables $\{A_i(a_i|0) : a_i = 0, 1\}$ for all $i \in \mathbb{Z}_n$ and applying the matrix G to the outcomes $\mathbf{k} = G\mathbf{a}$, the resulting state ρ_{KE} written in Equation (4.13) satisfies

$$\frac{1}{2}\|\rho_{KE} - \omega_K \otimes \rho_E\|_1 \leq \frac{1}{2} \sum_{\mathbf{r}:\, w(\mathbf{r}) \geq d} \operatorname{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i}\right], \tag{4.93}$$

where $w(\cdot)$ is the Hamming weight.

Proof. We start by recalling Lemma 61, which bounds the relevant distance as

$$\|\rho_{KE} - \omega_K \otimes \rho_E\|_1 \leq \sum_{\mathbf{r} \neq \mathbf{0}} I_G(\mathbf{r}) \operatorname{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i}\right], \tag{4.94}$$

with

$$I_G(\mathbf{r}) = \begin{cases} 0 & \text{if} \quad \mathbf{r} \notin \operatorname{span}(G) \\ 1 & \text{if} \quad \mathbf{r} \in \operatorname{span}(G) \end{cases}, \tag{4.95}$$

where $\mathbf{r} \in \mathbb{Z}_2^n$ and $\operatorname{span}(G) \subseteq \mathbb{Z}_2^n$ denotes is the subspace spanned by the rows of G. Let G be the generator matrix of the primitive binary BCH codes as defined in (4.92) and define a new indicator function

$$\tilde{I}_G(\mathbf{r}) := \begin{cases} 0 & \text{if} \quad w(\mathbf{r}) < d \\ 1 & \text{if} \quad \text{otherwise} \end{cases}. \tag{4.96}$$

For all $\mathbf{r} \in \text{span}(G)$ such that $\mathbf{r} \neq \mathbf{0}$, $w(\mathbf{r}) \geq d$ by definition of an (n, m, d) code. Then, $\tilde{I_G}(\mathbf{r}) \leq I_G(\mathbf{r})$ for all $\mathbf{r} \in \mathbb{Z}_2^n$ and therefore,

$$\begin{aligned} \|\rho_{KE} - \omega_K \otimes \rho_E\|_1 &\leq \sum_{\mathbf{r}\neq\mathbf{0}} I_G(\mathbf{r}) \operatorname{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i}\right] \\ &\leq \sum_{\mathbf{r}\neq\mathbf{0}} \tilde{I_G}(\mathbf{r}) \operatorname{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i}\right] \\ &= \sum_{\mathbf{r}:w(\mathbf{r})\geq d} \operatorname{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i}\right], \end{aligned} \tag{4.97}$$

and completes the proof. □

Note that Equation (4.93) is not a tight bound. It can be substantially improved by applying Lemma 61 directly to the generator matrix G and summing over only the elements in $\text{span}(G)$. This could be done numerically; however, the number of elements in $\text{span}(G)$ is exponential in m and the problem of enumerating all codewords for a general code is considered computationally intractable. In the IID case, some simplifications can be applied, which are summarised in the following remark.

Remark 64. Consider the IID scenario where, for a generator matrix G from a binary linear error-correcting code, each $\mathbf{r} \in \text{span}(G)$ gives an additive error contribution of $\operatorname{tr}\left[\rho_{AB} \prod_i S_i^{r_i}\right] = \langle S_1 \rangle^{w(\mathbf{r})}$. Here, we can rewrite the right-hand side error bound of Lemma 61 as $\sum_{i=1}^{n} C_i \langle S_1 \rangle^i$, where C_i denotes the number of codewords with Hamming weight i. With this formulation, the individual codewords $\mathbf{r}$ no longer matter. Instead, only the number of codewords of each weight is relevant, reducing the problem from enumerating codewords to computing the weight distribution.

Following this remark, it is important to note when considering BCH codes that: (1) The weight distribution problem for BCH codes has been extensively studied (e.g., [143, 144]), allowing the use of already established results to estimate C_i for each i, resulting in a much tighter bound on the error than Equation (4.93). (2) In the IID case, low-weight codewords are the primary contributors to error, as the

weight determines the exponential decay factor. It may be computationally tractable to calculate the exact number of codewords with specific weights close to the minimum distance, e.g., the number of codewords with weight $d+i$ for all $i \in \mathbb{Z}_c$ where c is some constant, since the BCH codes have a lot of structure [145, 146].

4.6.3 Concatenated codes

Concatenated error-correcting codes combine an *inner* and an *outer* code to produce a new error-correcting code that inherits properties from both. By concatenating error-correcting codes, we can construct codes that improve performance, interpolating between the strengths of two different codes. The core idea of concatenated codes is to first encode the information bits with the outer code and then encode subsets of bits in the resulting codeword with the inner code. We focus on binary linear codes, using an outer (n,m,d) code and an inner (N,M,D) code, where M is a divisor of n. For an inner code with $M \times N$ generator matrix G_{in} and an outer code with $m \times n$ generator matrix G_{out}, we define the concatenated generator matrix as $G = G_{\text{out}} G'_{\text{in}}$, where G'_{in} is an $n \times (nN/M)$ matrix constructed by placing n/M copies of G_{in} along the diagonal,

$$G'_{\text{in}} := \begin{bmatrix} G_{\text{in}} & & & & \\ & G_{\text{in}} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & G_{\text{in}} \end{bmatrix}. \tag{4.98}$$

We note that, since the generator matrix of concatenated codes can always be written in this form, the concatenation of two (binary) linear codes is itself a (binary) linear code. The minimum distance of a concatenated code combines the properties of its inner and outer codes, as detailed in Theorem 65.

Theorem 65. For an outer binary linear (n,m,d) code and an inner binary linear (N,M,D) code, where M divides n, the concatenated code is a $(Nn/M, m, D')$ code with $D' \geq dD/M$.

Proof. The concatenated code is constructed by first encoding an information vector with the outer code, then encoding each block of M elements of the codeword using the inner code. Since the generator matrix G of the concatenated code has dimensions $m \times (Nn/M)$, it defines an $(Nn/M, m)$ code. To determine the minimum distance of the concatenated code, consider two distinct codewords $\mathbf{r}$ and $\mathbf{r}'$ produced by the outer code, with a Hamming distance of at least d, meaning $w(\mathbf{r} \oplus \mathbf{r}') \geq d$. The codeword $\mathbf{r}$ (and similarly $\mathbf{r}'$) can be viewed as consisting of n/M blocks, each of length M. Since the outer code is an (n, m, d) code, at least d/M of these blocks differ between $\mathbf{r}$ and $\mathbf{r}'$. When these differing blocks are encoded by the inner code, each block of length M is mapped to a codeword of length N with minimum distance D. Thus, in the concatenated codeword, each of the d/M different blocks now contributes D different bits. Therefore, the final codeword has a Hamming distance of at least dD/M, meaning that the minimum distance of the concatenated code is $D' \geq dD/M$. □

We now use Theorem 65 to concatenate the repetition code and the primitive binary BCH code to generate a new family of codes.

Corollary 66. The concatenated code obtained by using the $(n = 2^l - 1, m \geq n - lt, d \geq 2t + 1)$ BCH code as the outer code and the $(N, 1, N)$ repetition code as the inner code is a (Nn, m, dN) code.

Notably, the code generated by Corollary 66 has a lot of structure and is computable in quasi-linear time. Its generator matrix is

$$G := \begin{bmatrix} \mathbf{c}_0 & \mathbf{c}_1 & \dots & \mathbf{c}_{n-2} & \mathbf{c}_{n-1} \\ \mathbf{c}_{n-1} & \mathbf{c}_0 & \mathbf{c}_1 & \dots & \mathbf{c}_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{4.99}$$

where $\mathbf{c}_i := \{c_i\}^N$, with $c_i \in \mathbb{Z}_2$ denoting the i-th element in the first row of the generator matrix of the BCH code, for $i \in \mathbb{Z}_n$.

Efficient implementation. This generator matrix G in Equation (4.99) can be viewed as a subset of rows from a circulant matrix, specifically every N-th row.

Its action on an Nn-length input corresponds to computing m bits of a convolution, enabling the use of the efficient NTT-based techniques discussed earlier, which results in a computation time of $O(nN \log nN)$.

Furthermore, for this concatenated code, we derive a theorem analogous to Theorem 63, leveraging the increased analytical tractability provided by concatenation with a repetition code.

Theorem 67 (Concatenated m-bit repetition and primitive binary BCH code)**.** Let G be an $m \times n'$ generator matrix of a $(n' = Nn, m \geq n - lt, d \geq N(2t+1))$ concatenated code from Corollary 66, for any positive integers l, t, and N, where $n = 2^l - 1$. After measuring the n'-round state ρ_{ABE} with the observables $\{A_i(a_i|0) : a_i = 0, 1\}$ for all $i \in \mathbb{Z}_{n'}$ and applying the matrix G to the outcomes $\mathbf{k} = G\mathbf{a}$, the resulting state ρ_{KE}, as written in Equation (4.13), satisfies

$$\frac{1}{2}\|\rho_{KE} - \omega_K \otimes \rho_E\|_1 \leq \frac{1}{2} \sum_{\mathbf{r}:\, w(\mathbf{r}) \geq 2t+1} \operatorname{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} \left(\prod_{j=0}^{N-1} S_{iN+j}^{r_i} \right)\right], \tag{4.100}$$

where $w(\cdot)$ denotes the Hamming weight and $\mathbf{r} \in \mathbb{Z}_2^n$.

Proof. The proof follows directly from Theorem 63, noting that the concatenated code includes the m-bit repetition code as the inner code, the codewords are the same as those of the outer code, with each element repeated N times. Therefore, we can write each element in the span(G) as $\mathbf{r}' = (\mathbf{r}'_0, \ldots, \mathbf{r}'_{2^l-2}) \in \mathbb{Z}_2^{n'}$ where $\mathbf{r}'_i \in \{\{0\}^N, \{1\}^N\}$ for all $i \in \mathbb{Z}_{2^l-1}$, and, in particular, we can write a compact representation of $\mathbf{r}'$ as $\mathbf{r} \in \mathbb{Z}_2^{2^l-1}$ where each element r_i is mapped to $\mathbf{r}'_i$ via the transform $r_i \to \{r_i\}^N$. □

Again, we note that this theorem is not tight due to the difficulty of enumerating the codewords of the primitive binary BCH codes. However, it improves on Theorem 63 by incorporating the analytic tractability of the m-bit repetition code. However, we emphasise that the concatenated code offers flexibility in how the codes are combined. For instance, one could restrict to BCH code parameters where the exact code space can be computed, enabling the direct use of Lemma 61.

4.7 Comparison with min-entropy extractors

In this section, we compare the extractors for Bell inequality violating sources presented in this chapter with those for min-entropy sources from the previous chapter. For simplicity, we assume that the state and measurements used to generate the extractor input are identical and independent in every round. This enables us to establish a straightforward connection between the expectation of the shifted CHSH operator and the min-entropy of the extractor input. Under this assumption, for any $\mathbf{r} \in \mathbb{Z}_2^n$, we can write

$$\mathrm{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i}\right] = \langle S_1 \rangle^{w(\mathbf{r})} , \tag{4.101}$$

where $\langle S_1 \rangle \in [0,1]$. Define the expected CHSH value $\mathrm{CHSH} \in (2, 2\sqrt{2})$. With the parametrisation of the shifted CHSH operator, setting $s = \mathrm{CHSH}$ in Theorem 55 relates the expected value of the shifted CHSH operator to the expected CHSH value via the relation

$$\langle S_1 \rangle = \sqrt{2 - \frac{\mathrm{CHSH}^2}{4}} . \tag{4.102}$$

To compare the extractors from this chapter to those of the previous, we need a method to evaluate the min-entropy of the extractor input. Fortunately, a tight analytic bound exists for the maximum single-round guessing probability for different expected CHSH values [147, 138]. Given the input x and the adversary's side information E, the maximum guessing probability of outcome a is given by

$$p_{\mathrm{guess}}(a|x,E)_Q \leq \frac{1}{2} + \frac{1}{2}\sqrt{2 - \frac{\mathrm{CHSH}^2}{4}} , \tag{4.103}$$

for any x and a. Therefore, the min-entropy of the extractor input $\mathbf{a} \in \mathbb{Z}_2^n$ is

$$k = -n \log\left(p_{\mathrm{guess}}(a|x,E)_Q\right) = -n \log\left(\frac{1}{2} + \frac{1}{2}\sqrt{2 - \frac{\mathrm{CHSH}^2}{4}}\right) . \tag{4.104}$$

These relations allow us to compute the extraction rate $\mathcal{R}_{\text{ext}}$, the extractor output length divided by its input length, for our constructions and those for min-entropy sources, as a function of the expected CHSH value.[1] Note that all the extractors we compare are computable in quasi-linear time.

The m-bit repetition code. The m-bit repetition code is a $(n, m, d = n/m)$ code for some integers n and d, where m is a divisor of n. Using the generator matrix G of the m-bit repetition code (Equation (4.91)), the extractor error can be bound by

$$\epsilon \leq \frac{1}{2} \sum_{\mathbf{r} \neq \mathbf{0}} I_G(\mathbf{r}) \operatorname{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} S_i^{r_i}\right] \tag{4.105}$$

from Lemma 61, where $I_G(\mathbf{r})$ is an indicator function that equals 1 if $\mathbf{r} \in \text{span}(G)$ and 0 otherwise. Using the relation (4.101), this becomes

$$\epsilon \leq \frac{1}{2} \sum_{\mathbf{r} \neq \mathbf{0}} I_G(\mathbf{r}) \langle S_1 \rangle^{w(\mathbf{r})} \,. \tag{4.106}$$

Given the simple structure of the codewords (the elements in span(G)), we expand this analytically by noting that the codewords are of the form $\{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{(n/d)-1}\}$, where each $\mathbf{c}_i \in \{\{0\}^d, \{1\}^d\}$ for $i \in \mathbb{Z}_{n/d}$. It is easy to see that the codewords have weights id for $i \in \mathbb{Z}_{m+1}$, with the number of codewords for each weight i given by $\binom{m}{i}$. Therefore, we can express ϵ as

$$\begin{aligned} \epsilon &\leq \frac{1}{2} \sum_{i=1}^{m} \binom{m}{i} \langle S_1 \rangle^{id} \\ &= \frac{1}{2} \left((1 + \langle S_1 \rangle^d)^m - 1 \right) \end{aligned} \tag{4.107}$$

where the equality comes from using the Binomial expansion. Then, we can compute the maximum output length m for a given error tolerance, $\epsilon_{\max}$ via the maximi-

[1] We note that extractors for min-entropy sources can be adapted to use *smooth min-entropy* (see, e.g., [62, Lemma 17]), with a corresponding adjustment to the error parameter. Asymptotically, smooth min-entropy approaches the von Neumann entropy, which is typically much larger than the min-entropy. It would be interesting to perform the same comparison using the appropriate adjustments and computing the smooth min-entropy.

sation problem;

$$\max_{d:d\backslash n} \quad m = \frac{n}{d} \tag{4.108}$$

$$\text{subject to:} \quad (1+\langle S_1\rangle^d)^m - 1 \le 2\epsilon_{\max} \tag{4.109}$$

and thus find the extraction rate $\mathcal{R}_{\text{ext}}^{m-\text{XOR}} = m/n$, which is evaluated to construct Figure 4.6.

The primitive binary BCH codes. The primitive binary BCH codes are $(n = 2^l - 1, m \ge n - lt, d \ge 2t+1)$ codes for positive integers l and t. Using their generator matrix as an extractor, we compute a conservative lower bound and an approximate extraction rate. To establish a lower bound on the extraction rate, we apply Theorem 63 to upper bound the error ϵ as

$$\begin{aligned}
\epsilon &\le \frac{1}{2}\sum_{\mathbf{r}:\, w(\mathbf{r})\ge d} \text{tr}\left[\rho_{AB}\prod_{i=0}^{n-1} S_i^{r_i}\right] \\
&= \frac{1}{2}\sum_{i=d}^{n}\binom{n}{i}\langle S_1\rangle^i \\
&= \frac{1}{2}(1-p)^{-n}\sum_{i=d}^{n}\binom{n}{i}p^i(1-p)^{n-i} \\
&= \frac{1}{2}(1-p)^{-n}\sum_{i=0}^{n-d}\binom{n}{i}p^{n-i}(1-p)^i\,,
\end{aligned} \tag{4.110}$$

where, for the first equality, we apply relation (4.101) and note the number of weight-i vectors of length n is $\binom{n}{i}$. In the second equality, we set $\langle S_1\rangle = p/(1-p)$ for some p. The summation on the right-hand side now follows a binomial distribution and can be upper bound using a multiplicative Chernoff bound [148], yielding

$$\begin{aligned}
\epsilon &\le \frac{1}{2}(1-p)^{-n}2^{-nD\left(\frac{n-d}{n}\|1-p\right)} \\
&\le \frac{1}{2}2^{n\left(\log(1+\langle S_1\rangle)-D\left(1-\frac{d}{n}\left\|\frac{1}{1+\langle S_1\rangle}\right.\right)\right)}\,,
\end{aligned} \tag{4.111}$$

where $D(\cdot\|\cdot)$ denotes the relative entropy (also known as the Kullback-Leibler divergence), defined by

$$D(q\|p) = q\log\frac{q}{p} + (1-q)\log\frac{1-q}{1-p}\,. \tag{4.112}$$

This enables us to compute a valid m for a given error tolerance $\epsilon_{\max}$ via the optimisation

$$\max_d\, m = n - \frac{\log(n+1)(d-1)}{2} \tag{4.113}$$

$$\text{subject to: } 2^{n\left(\log(1+\langle S_1\rangle) - D\left(1-\frac{d}{n}\middle\|\frac{1}{1+\langle S_1\rangle}\right)\right)} \leq 2\epsilon_{\max}\,. \tag{4.114}$$

The extraction rate lower bound is computed as $\mathcal{R}_{\text{ext}}^{\text{BCH}} = m/n$.

Next, we compute an approximate rate for the primitive binary BCH codes. According to [142], the weight distribution of BCH codes is well-approximated by a normal distribution; specifically, BCH codes have approximately $2^{m-n}\binom{n}{i}$ codewords of weight i for every $i \geq d$. Using this approximation in Lemma 61, the error can be approximately upper bound by

$$\begin{aligned}
\epsilon &\leq \frac{1}{2}\sum_{\mathbf{r}\neq\mathbf{0}} I_G(\mathbf{r})\,\mathrm{tr}\left[\rho_{AB}\prod_{i=0}^{n-1} S_i^{r_i}\right] \\
&\approx \frac{1}{2}2^{m-n}\sum_{i=d}^{n}\binom{n}{i}\langle S_1\rangle^i \\
&\leq \frac{1}{2}2^{m-n}\sum_{i=0}^{n}\binom{n}{i}\langle S_1\rangle^i \\
&\leq \frac{1}{2}2^{m-n}(1+\langle S_1\rangle)^n\,.
\end{aligned} \tag{4.115}$$

Rearranging this gives $m \geq n(1-\log(1+\langle S_1\rangle)) + \log(2\epsilon)$ and therefore an approximate rate of

$$\tilde{\mathcal{R}}_{\text{ext}}^{\text{BCH}} = 1 + \log(1-\langle S_1\rangle) - \frac{1}{n}\log(2\epsilon)\,. \tag{4.116}$$

The concatenated repetition and primitive binary BCH code. The concatenated code formed from an inner primitive binary BCH code and a outer m-bit repetition code is a $(n' = Nn, m \geq n - lt, d \geq N(2t+1))$ code for any positive integers l, t, and N, where $n = 2^l - 1$, as stated in Corollary 66. Due to the concatenation with the m-bit repetition code, the codewords exhibit additional structure, providing greater analytic tractability. Applying Theorem 67 for this family of codes with generator matrix G, we obtain

$$\begin{aligned}
\epsilon &\leq \frac{1}{2} \sum_{\mathbf{r}:\, w(\mathbf{r}) \geq d} \operatorname{tr}\left[\rho_{AB} \prod_{i=0}^{n-1} \left(\prod_{j=0}^{N-1} S^{r_i}_{iN+j} \right)\right] \\
&\leq \frac{1}{2} \sum_{i=d}^{n} \binom{n}{i} \langle S_1 \rangle^{iN} \\
&= \frac{1}{2} (1-p)^{-n} \sum_{i=d}^{n} \binom{n}{i} p^i (1-p)^{n-i} \\
&= \frac{1}{2} (1-p)^{-n} \sum_{i=0}^{n-d} \binom{n}{i} p^{n-i} (1-p)^i \,, \qquad (4.117)
\end{aligned}$$

where we have set $\langle S_1 \rangle^N = p/(1-p)$ for some value $p \in (0, 1/2)$ and note that $\mathbf{r} \in \mathbb{Z}_2^n$. This can again be upper bound using a multiplicative Chernoff bound, yielding

$$\begin{aligned}
\epsilon &\leq \frac{1}{2} (1 + \langle S_1 \rangle^N)^n 2^{-nD((n-d)/n \| (1-p))} \\
&\leq \frac{1}{2} 2^{n(\log(1+\langle S_1 \rangle^N) - D(1-d/n \| 1/(1+\langle S_1 \rangle^N)))} \,. \qquad (4.118)
\end{aligned}$$

This allows us to compute the maximum m for a given error tolerance $\epsilon_{\max}$ via the maximisation

$$\begin{aligned}
&\max_{d,n,N} m = n'/N - \log(n+1)(d-1)/2 \\
&\text{subject to: } 2^{n\left(\log(1+\langle S_1 \rangle^N) - D(1-\frac{d}{n} \| \frac{1}{1+\langle S_1 \rangle^N})\right)} \leq 2\epsilon_{\max} \,, \\
&\qquad\qquad n' = nN \,, \\
&\qquad\qquad n = 2^l + 1 \,, \qquad (4.119)
\end{aligned}$$

for m and l positive integers and obtain an extraction rate $\mathcal{R}_{\text{ext}}^{\text{concat}} = m/n'$.

Circulant extractor for min-entropy sources. The Circulant extractor from Chapter 3 is a quantum-proof $(n_1, k_1, n_2 = n_1 + 1, k_2, m, \epsilon)$ extractor, with

$$m \leq k_1 - (n_2 - k_2) + 2\log(\epsilon)\,, \tag{4.120}$$

in the product source and Markov model. This achieves the optimal output length in the seeded case, making it the ideal benchmark for comparison. For this construction to be valid, $n_2 = n_1 + 1$ must be prime. Under the IID assumption and taking the extractor input as the source, we set $n_1 = n$ and compute the min-entropy as

$$k_1 \geq -n\log\left(\frac{1}{2} + \frac{1}{2}\sqrt{2 - \frac{\text{CHSH}^2}{4}}\right). \tag{4.121}$$

Since we are comparing with deterministic extractors, we also consider the case where the seed is weakly random, setting $k_2 = \alpha_2 n_2$, where $\alpha_2 \in (0, 1]$. The extraction rate $\mathcal{R}_{\text{ext}}^{\text{circ}}$ is then computed as

$$\mathcal{R}_{\text{ext}}^{\text{circ}} = \frac{k_1}{n} - (1 - \alpha_2) + \frac{2}{n}\log(\epsilon)\,. \tag{4.122}$$

4.7.1 Results

To provide a concrete comparison, we calculate the extraction rates for each extractor using a extractor input length of $n = 2^{15}$ and various expected CHSH values $\text{CHSH} \in [2, 2\sqrt{2}]$. Due to varying constraints on the extractor input length for different extractors, we sometimes need to truncate the input to a shorter length $\tilde{n} < n$ to ensure validity (e.g., for the primitive binary BCH code-based constructions and the Circulant extractor), which is then accounted for in the extraction rate. The results are shown in Figure 4.6. Solid lines represent lower bounds, though some, like the BCH code, are not tight.[2] The approximate rate for the BCH codes is represented

[2]Because the bound for BCH codes is far from optimal, their extraction rate in Figure 4.6 appears almost like a delta function; yielding a rate of 0 except when the CHSH inequality is nearly maximally violated, at which point the rate jumps to 1.

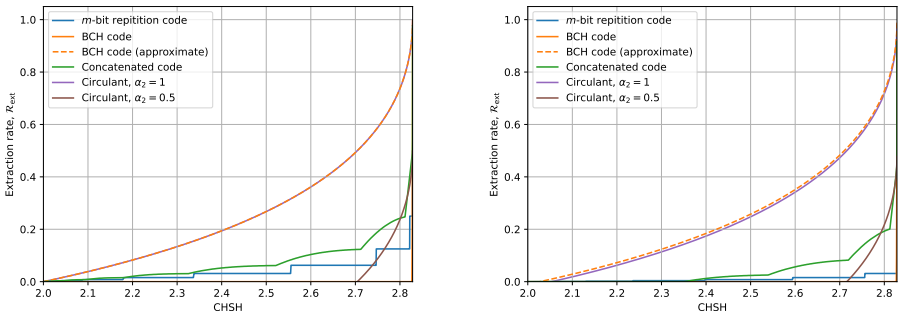

Figure 4.6: The extraction rate $\mathcal{R}_{\text{ext}}$ (the extractor output length divided by input length), for different values of CHSH with $n = 2^{15}$. Left: $\epsilon \leq 1$. Right: $\epsilon \leq 2^{-0.01n}$.

with a dashed line.

We find that the approximate extraction rate of BCH codes is comparable to that of seeded extractors. Moreover, in cases where the error scales as 2^{-cn} for a constant $c > 0$ (as shown in the right-hand plot of Figure 4.6), the approximate BCH codes outperform seeded extractors. This suggests that, in the IID setting, our seedless extractors may not incur the minimum entropy loss that is inherent to all seeded extractors.

Furthermore, when the Circulant min-entropy extractor uses a weak seed (with $\alpha_2 = 1/2$), explicit extractors based on error-correcting codes – particularly our construction combining concatenated repetition and BCH codes – exhibit lower entropy loss across a wide range of CHSH values. Notably, this is true even without analysing the BCH code weight distribution, which would significantly improve the extraction rate of these extractors. Moreover, in protocols for randomness amplification, this is exactly the scenario that occurs (see Part II, for more evidence on this). This observation reinforces our belief that seedless extractors for Bell inequality-violating sources are especially advantageous in such scenarios.

4.8 Conclusion and discussion

In this chapter, we demonstrated that sources which violate a Bell inequality can be extracted from deterministically, even in the presence of a quantum adversary and without assumptions about the dimension of the adversary's system.

We first showed that a single bit could be deterministically extracted with an

error rate that decreases exponentially with the length of the extractor input. We then introduced deterministic extractors for arbitrary output lengths, whose existence was proven using probabilistic methods. For these extractors, we proposed a toy spot-checking protocol and, using new proof techniques, demonstrated that the relevant Bell quantifiers required for these extractors could be estimated. We also showed that, with sufficiently many rounds, a single bit could be produced with arbitrarily small error for any CHSH inequality violation, and that the m-bit extractor functions achieve optimal extraction rates in the high-CHSH regime.

Since the m-bit extractor we present is non-constructive and relies on randomised methods, we next introduce a general lemma for linear functions that serve as extractors for Bell-violating sources. This lemma relates the extractor error to the row space of the linear function, which corresponds to the code space of a binary linear error-correcting code. Using this relationship, we explored several extractor designs based on binary linear error-correcting codes, including explicit constructions using m-bit repetition codes, primitive binary BCH codes, and their concatenations. For each extractor, we proposed an information-theoretically secure implementation with quasi-linear computation time, using the NTT methods detailed in Section 2.3. Finally, we analysed their performance in the IID setting, comparing them with extractors for min-entropy sources presented in Chapter 2. Notably, our new extractors outperformed those relying solely on a min-entropy promise in certain scenarios.

In the context of resource-efficient quantum cryptography, we introduced extractors for certain quantum protocols that can be implemented deterministically, without requiring additional randomness as is usually the case, and provided several explicit, computationally efficient, constructions. These results represent a significant step toward a new paradigm for quantum cryptographic protocols based on Bell inequality violations, such as device-independent protocols, with the potential to reduce resource requirements across various protocols and applications. Moreover, they open the door to entirely new protocols, potentially eliminating the need for input randomness. This paradigm contains numerous open problems, some of

which we summarise below.

Open Problem 6. The estimation protocol presented in Section 4.5 has low efficiency rates. A key challenge is to develop an estimation protocol with significantly higher efficiency, potentially by improving the procedure used to bound $\langle \prod_i S_i \rangle$, enhancing the security proof, or other methods. Improvements may be obtained for randomness generation by:

a. Using both Alice's and Bob's outcomes for extractor input generation (the current protocol only uses Alice's outcomes).
b. Reusing the randomness for estimation, as proposed in [149].
c. Using all rounds for both estimation and extractor input generation, instead of spot-checking.

Furthermore, we prove our security condition for variable output lengths, whereas such conditions are typically proved for fixed output lengths. Adapting the proof to consider a fixed output length would likely improve efficiency.

Open Problem 7. Analyse and generalise our spot-checking protocol into a device-independent protocol for randomness amplification. The seedless extraction protocol presented here requires initial uniform randomness to generate the variables t_l, x_l, and y_l for testing the Bell violation. However, the minimal statistical requirements for this initial randomness are not yet known (see [150] for necessary requirements). Determining these requirements, as well as using our deterministic extractors for Bell inequality violating sources, may enable protocols with weaker assumptions than those in existing literature which rely on multi-source randomness extractors.

Open Problem 8. The spot-checking protocol and analysis primarily addressed the task of QRNG, specifically randomness expansion. It would be interesting to use our extractors to design protocols for other quantum cryptographic tasks, for example, QKD. Unlike randomness expansion, QKD aims to generate shared randomness rather than minimising the use of local randomness. This distinction mitigates a drawback of our current protocol, which requires significant local randomness due

to the large proportion of rounds needed for estimation. Developing a QKD protocol using these extractors could lead to new approaches with potentially weaker assumptions or lower computational requirements than existing protocols. This could be especially advantageous in resource-constrained settings, such as satellites.

Open Problem 9. In this chapter, we have developed seedless extractors based on functions derived from linear error-correcting codes. These constructions are explicit, have quasi-linear computational complexity, and are likely to achieve near-optimal extraction rates if the weight distribution of the codewords could be computed exactly (see the BCH approximation in Section 4.7). Are there alternative constructions that offer similar performance while being more analytically tractable? Additionally, can the weight distributions for the codes presented here be computed exactly, or at least bounded tightly from above?

Open Problem 10. The results presented in this chapter focus on the CHSH Bell inequality, involving two parties with binary inputs and outputs. However, these results can be generalised to arbitrary scenarios by applying the NPA hierarchy [151] or by constructing analytical semi-definite inequalities similar to Theorem 55 for other Bell inequalities, which relate the predictability of measurement outcomes to the expected values of a (shifted) Bell operator. We expect that increasing the number of inputs will substantially improve efficiency rates, as observed in other scenarios [152, 132].

Open Problem 11. Currently, our theorems assume that the measurement devices used in each round have no internal memory (as in [138]), which is a restrictive assumption. Given the connection between Bell violation and the independence of rounds discussed in the introduction, we expect this assumption could potentially be relaxed. In certain cases, such as when the Bell violation is sufficiently high to directly apply self-testing results, this is known to be feasible. To address the general case, one could attempt to establish proofs equivalent to those presented here within the experimental setting of the entropy accumulation theorem [153, 141].

Open Problem 12. From a foundational perspective, the class of weakly random sources investigated in this chapter represents a new class of deterministically extractable sources.

a. Understanding how this class of sources relates to others could provide insight, potentially adding more directed arrows to Figure 2.2.
b. Identifying necessary and/or sufficient conditions for deterministic extraction would help understand the limits of what is achievable.

Part II

Randomness amplification and privatisation

Chapter 5

Introduction

Randomness is a crucial resource in various fields, such as cryptography, statistical sampling, and algorithm design. In cryptography, for example, keys used for encrypting data must be generated with (near-)perfect randomness (Definition 3 and Definition 5), otherwise the security of the encryption can be compromised. Consequently, a question of both fundamental and practical interest arises:

> *How can we ensure that the output of a RNG is truly near-perfectly random, in the presence of an adversary?*

One approach is to attempt to build a trustworthy RNG. Such a device must be correctly characterised to ensure reliable operation; otherwise, the security of the intended application may be compromised. Physical RNGs generate random numbers through chaotic [154] or quantum processes [155, 156].[1] The idea is that these physical processes yield outcomes that are difficult to predict, or, in some quantum cases, are inherently random. However, there are at least three key challenges with this approach:

1. An accurate model of the underlying physical process is essential but challenging to develop, as isolating the desired process from noise and environmental factors is difficult, hardware characterisation is prone to errors, and system characteristics may change over time. Moreover, relying on such

[1] Pseudo random number generators (PRNGs) expand a short, (near-)perfect random seed into a larger output using mathematical functions. These depend on placing computational assumptions on the adversary, making them outside the scope of interest here.

a model necessitates trusting the RNG provider or subjecting the device to thorough inspection.

2. Many RNGs require an initial (near-)perfectly random seed.[2] Ensuring access to such a seed is often difficult and leads to a circularity: one requires near-perfect randomness to generate near-perfect randomness.
3. Most RNGs lack security against quantum adversaries that might share quantum correlations with the device. Given the advancement in quantum technologies, this opens up possible vulnerabilities.

An alternative approach is to acknowledge that building a RNG capable of producing (near-)perfect randomness is extremely challenging, if not practically impossible. Instead, one can construct a scheme that *amplifies* an initial source of randomness such that the amplified output is provably near-perfectly random and private. This shifts the requirement on the initial RNG from generating near-perfect randomness to producing randomness that is only somewhat unpredictable; a significantly weaker and more realistic assumption. This idea is formalised through the closely related tasks of *randomness amplification* [157][3] and *randomness amplification and privatisation* [82]. In the former, the user has access to a private, imperfect source of randomness, aiming to generate near-perfect randomness. In the latter, the source is similar but also *public*: the adversary learns the output after generation but cannot fully predict it in advance.

Unfortunately, imperfect sources of randomness cannot be amplified using classical processes alone without strong assumptions on the source (see 'Weakly random sources and extractability' 2.1 for an exposition). However, this limitation can be overcome with the inclusion of quantum resources [157]. In fact, quantum devices enable *device-independent randomness amplification (and privatisation)* [157, 80, 158, 159, 160, 81, 161, 162, 82], where the randomness and privacy of the output are *certified* without modelling the device's internal workings (see [82,

[2]Exceptions include RNGs that directly output near-perfect randomness without post-processing or allow for deterministic randomness extraction (see Chapter 2.1). However, these RNGs must be highly characterised, which introduces the drawbacks mentioned in the first point.

[3]It is important to note that [157] achieves both randomness amplification and privatisation, though it does not explicitly distinguish between the two tasks as we do here.

Table II] for a comparative overview of these protocols). The device is treated as a black box, requiring minimal trust from the user, as discussed, for example, in the review [11].

5.1 Device-independence and its importance

Instead of building a trusted RNG directly, we adopt the approach outlined above: constructing a *device-independent* protocol for randomness certification that makes minimal assumptions about the quantum hardware. By treating devices as black boxes, we establish lower bounds on the unpredictability (or *entropy*) of the generated outcomes without requiring a detailed understanding of the devices' internal workings. This approach provides a high level of security largely independent of hardware assumptions, effectively addressing problem 1 discussed earlier. The security is derived from the violation of Bell inequalities, as detailed in Section 6.2.3.

To illustrate the advantages of device-independent certification, we highlight examples of known attacks on existing cryptographic systems. A notable case is the vulnerability in the Dual EC PRNG, favoured by the National Security Agency and standardised by the National Institute of Standards and Technology (NIST), which allowed future outputs to be predicted from a small amount of previous outputs [163, 164, 165]. Numerous other weaknesses and attacks on cryptographic PRNGs have been identified and executed [166, 167, 168]. Physical RNGs are also susceptible to attacks. Side-channel attacks have been found for those based on classical processes (e.g., chaotic sources) [169], exploiting device leakage or active implementation attacks, such as error injection [170]. Quantum hardware, too, is vulnerable; for example, popular QRNGs based on quadrature measurements in shot-noise limited states are attackable if the hardware is not fully trusted or well-characterised [127]. Additionally, poorly characterised quantum measurements can lead to inaccurate claims in state tomography and entanglement witnessing [171], which opens the door to systematic errors and potential exploitation. Even without active attacks, QRNGs that require trust in their components may fail advanced statistical tests, as shown in [172] and by us in [173].

5.2 The advantages of randomness amplification

The device-independent framework we follow offers a very high level of security, as described above; however, several device-independent tasks are possible and we discuss two related to randomness generation: randomness amplification and randomness expansion. We argue that, while randomness expansion is useful, randomness amplification is both strictly stronger and practically necessary.

In a randomness expansion protocol, an initial near-perfectly random seed is extended into a longer output. This assumes access to near-perfect randomness to begin the protocol – a challenging and often overlooked assumption. In contrast, randomness amplification begins with an initial weak source of randomness, which may be correlated with the quantum devices, and uses it to generate (near-perfect) randomness. This approach relaxes the need for independence between the initial RNG and the quantum device, which may share an environment – a scenario known to occur. For example, as noted in [174] "In classical ring oscillator (RO) based RNGs, the ring oscillators need to be located at a distance from each other to avoid coupling between adjacent ROs.". Once near-perfect randomness is generated from an amplification protocol, however, this can serve as a well-justified seed for an expansion protocol to increase generation rates; addressing problem 2 mentioned above.

5.3 Cryptography with weak randomness

Access to near-perfect randomness is vital for almost all cryptographic applications. However, as previously discussed, this assumption is difficult to justify in practice. This raises a crucial question: *What is the impact of using weak randomness in cryptography?* In other words, how does security change if the randomness is only partially unpredictable rather than indistinguishable from uniformly distributed and private?

In [175], the authors show that randomness lacking near-perfect unpredictability, i.e., where each bit is not nearly impossible to predict, is insufficient for encryption, bit commitment, secret sharing, zero-knowledge proofs (interactive or

not), and two-party computations. This vulnerability persists even against computationally bounded adversaries. In [176], the authors demonstrate that achieving unconditionally secure encryption requires the ability to deterministically extract a (near-)perfectly random string at least as long as the message from the available randomness – something that is unachievable with many weak randomness sources. They extended these findings to include privacy primitives that are perfectly or statistically binding, such as commitment schemes and computationally secure private and public-key cryptography. Other works also examine the impact of weak randomness, for example, [177, 178, 179], but these results generally fall within the scope of those previously mentioned. However, there are some positive outcomes, notably for tasks involving differential privacy [179] or authentication [175].

Although not all cryptographic tasks fail when using weak randomness, a substantial portion do, which emphasises the importance of generating (and being able to certify) near-perfect randomness. Moreover, cryptographic primitives are often combined for specific applications – meaning that if any individual underlying primitive is insecure, the security of the entire procedure is compromised.

Chapter 6

Practical device-independent randomness amplification and privatisation

In this chapter, we present an end-to-end protocol for device-independent randomness amplification and privatisation with a focus on practical implementation. This work builds on previous works on randomness amplification (and privatisation) [157, 81, 161, 82], and in particular, follows the statistical analysis techniques of [82]. Our main contribution is a practical protocol with high noise tolerance and generation rates, achieved through the optimisation of the Bell inequality, statistical analysis, and post-processing for real-world devices. The implementation is designed to ensure that the required states and measurements are easy to realise on real-world hardware, with tailored analysis and compilation. Moreover, all components of the protocol are fully implemented and analysed as a complete system. Together, these contributions allow us to implement our protocol no real-world quantum hardware, in a semi-device-independent manner (see Chapter 7).

The first to consider the task of randomness amplification were Colbeck and Renner, providing a proof-of-concept [157]. Later work aimed to achieve noise resistance and amplify imperfect sources with arbitrary bias [80], although it required a large number of devices and had low generation rates, making it unsuitable for implementation. Other works allowed for more general correlations between

the imperfect RNG and the quantum device [158, 159, 160, 180], but with significant trade-offs: amplification was only feasible for imperfect sources with very small bias [158], required many devices (polynomial or exponential in $1/\epsilon$, where ϵ is the protocol error [160, 159]), had low or no noise tolerance [158, 159, 160], and/or involved highly complex classical-processing steps that are unrealistic for implementation [159, 160, 180]. The only potentially implementable works are [81, 161, 162, 82]. However, our protocol is the only one that offers all the following features:

- **Efficiency:** Our protocol achieves randomness generation rates that scale linearly with the number of uses of the quantum device. Only [82] shares this property. The protocols in [81, 161, 162] yield, at best, output rates sub-linear in quantum devices' runtime.[1]
- **Quasi-linear computation time post-processing:** Our randomness post-processing is implemented using the NTT, which ensures both information-theoretic security and quasi-linear computation time. Other works typically use generic polynomial-time methods or the FFT, which may introduce rounding errors that could be exploited in attacks.
- **Amplification and privatisation:** Our protocol can optionally be used to achieve both randomness amplification and privatisation. The only other work with this feature is [82].
- **Optimised for real-world implementation:** While we rely on the statistical analysis of [82], our protocol offers greater noise tolerance, which is unachievable in the simpler experimental setup considered in their work. Further explanation is provided in Section 6.2.3.

This unique combination of features allowed us to implement our protocol in a semi-device-independent manner on real quantum computers (see Chapter 7), achieving randomness generation speeds of megabits per second.

[1] Note that this is because these works consider post-quantum adversaries, constrained only by the no-signalling principle. In this setting, there is a lack of secure extractors for such adversaries [181], leading to sub-linear randomness generation rates.

6.1 Idea of the protocol

The setup for device-independent randomness amplification (and privatisation) follows the same structure as in previous work [157, 81, 161, 82]. The user's facility is assumed to be secure and isolated from external influence once the protocol begins. To run the protocol, the user needs an imperfect RNG, a quantum device capable of running a Mermin Bell test [182] (made of three separate parts that cannot communicate during the protocol) and a classical computer for storing data, performing the verification and implementing the post-processing. The protocol can be understood as three subroutines; *data collection, randomness certification* and *randomness post-processing*.

Data collection. The first part of the protocol involves collecting data to analyse the behaviour of the quantum device. This is the only step of the protocol that requires quantum hardware. The quantum device is driven by varying inputs generated by the imperfect RNG, and its responses, or outputs, are recorded. Both inputs and outputs are saved for later analysis. After a sufficient number of interaction rounds, the conditional input-output probability distribution can be constructed for the device – this *observed behaviour* is then used to certify that the device generates truly unpredictable outputs.

Randomness certification. In the second step, the collected data is analysed to certify private randomness in the quantum device's output. Certain input-output statistics can only arise from specific quantum processes, so observing such statistics serves as evidence that the device's underlying process is quantum. This allows us to prove that the device's output contains private randomness and quantify its amount. Instead of assuming randomness is generated, the user *certifies* it based on the observed behaviour. Using the device-independent approach, this verification requires minimal modelling of the quantum hardware, treating it essentially as a black box. Thus, the protocol's security remains largely independent of the specifics of the hardware implementation.

Randomness post-processing. The third and final step is to *extract* certified private randomness from the quantum device's outcomes using randomness extraction

algorithms on a classical computer. These algorithms transform partially random and private data into a shorter, near-perfectly random output. To achieve this, the quantum device's outcomes are combined with a fresh string from the imperfect RNG and processed using a quantum-proof randomness extractors for min-entropy sources detailed in Chapter 3.

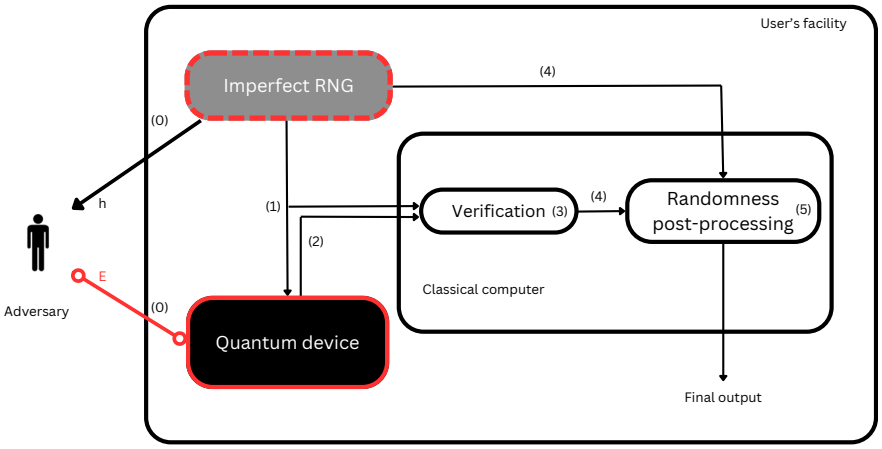

Figure 6.1: Our setup for device-independent randomness amplification (and privatisation), with the quantum hardware components highlighted in red, including the quantum device and, optionally, the imperfect RNG.

The full setup is depicted in Figure 6.1, where the numbered labels indicate the following:

(0) Before the protocol begins, the adversary has access to any previously generated output from the imperfect RNG and a general description of it, forming the (classical) side information h. The adversary may also have built the quantum device on which the protocol is performed, using information h, with which the adversary might still be correlated through their quantum systems E.

(1) The imperfect RNG provides inputs which are used to repeatedly challenge the quantum device.

(2) The quantum device (made of three separate parts) generates outputs for each set of inputs it receives.

(3) After many interaction rounds, the input-output statistics are computed and a verification is performed, which serves to certify the amount of private randomness in the outcomes.

(4) If verification is successful, a fresh sequence from the imperfect RNG is generated and sent, with the outputs of the quantum device, to the post-processing step.

(5) Classical algorithms process the outputs of the quantum device and the fresh sequence from the imperfect RNG to produce a near-perfectly random and private output string, which constitutes the final protocol output.

6.2 Main tools and ingredients

6.2.1 Security criteria

When considering the security of a protocol, following [19, 22], one can consider its *security* (sometimes called *soundness*). Informally, a protocol is secure if the protocol aborts with high probability, or the real and ideal protocol outputs are essentially indistinguishable. Mathematically, this is formalised as follows: A protocol is said to be ϵ_{sec}-secure, if

$$\frac{1}{2}\left\|\rho^h_{KE} - \omega_K \otimes \rho^h_E\right\|_1 (1 - p_{\text{abort}}) \leq \epsilon_{\text{sec}} \,, \tag{6.1}$$

where ρ^h_{KE} denotes the real cq-state shared by the user and the adversary generated by the protocol which is conditioned on the information h, $\omega_K \otimes \rho^h_E$ is the ideal cq-state shared by the user and the adversary and p_{abort} is the probability of the protocol aborting.

Altogether, the security condition (6.1) ensures that either the protocol aborts with high probability or the real and ideal systems are near-indistinguishable, i.e. the real and ideal outputs satisfy Equation (1.8). Therefore, the security parameter ϵ_{sec} quantifies both the probability of not aborting and the distinguishability between the actual joint state ρ^h_{KE} and the ideal state $\omega_K \otimes \rho^h_E$, even for a powerful adversary possessing information h and E about the quantum device. Here, the adversary is

assumed to obey the laws of quantum physics but is otherwise unbounded, potentially having access to a highly advanced quantum computer.

Importantly, this security definition is universally-composable [20], meaning the generated random numbers can be safely used in other protocols (see Chapter 1 for more information). We note that composability for device-independent protocols holds only when physical devices are not reused and are kept inaccessible to the adversary after the protocol ends. As found in [183], reusing devices could allow a malicious device to store information from one execution and leak it in subsequent ones, compromising security.

6.2.2 Imperfect random number generators

The protocol begins with an imperfect RNG that must be amplified to satisfying the security condition (6.1). We consider imperfect random number generators that sequentially output bits $r_i \in \mathbb{Z}_2$ such that bit r_i is produced before r_{i+1} for all non-negative integers i. Unlike other approaches, such as randomness expansion, randomness amplification does not assume that these bits are completely unpredictable nor fully independent of the quantum device. Instead, each bit is only *somewhat* unpredictable, conditioned on previously generated bits and any classical side information h' potentially available to the adversary (which could be, for example, a model of the RNG).

We consider the imperfect RNG to be a SV source, introduced in [26]. Recalling the definition from Part I and adapting to include the classical side information, the δ SV source satisfies

$$\frac{1}{2} - \delta \leq \Pr(r_i|\mathbf{r}_{i-1}, h') \leq \frac{1}{2} + \delta \tag{6.2}$$

for all i, where $\mathbf{r}_{i-1} = (r_0, \ldots, r_{i-2}, r_{i-1})$ represents all bits generated before bit r_i and $\Pr(r_i|\mathbf{r}_{i-1}, h')$ denotes the probability r_i given side information h' and prior generated bits $\mathbf{r}_{i-1}$. We collectively denote the adversary's side information about the imperfect RNG, h' and $\mathbf{r}_{i-1}$, as h (see Figure 6.1).

Throughout this part of the thesis, we use *imperfect RNG* synonymously with a

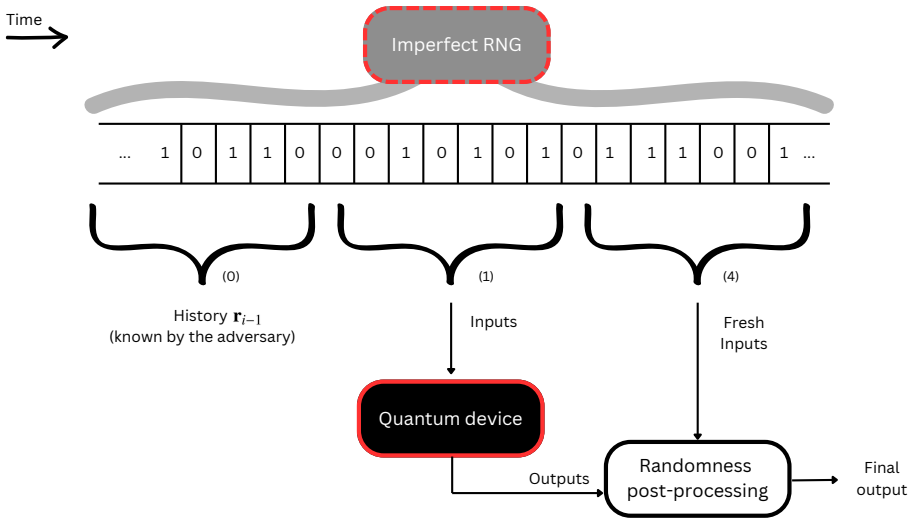

Figure 6.2: Schematic of the imperfect RNG's role in our randomness amplification and privatisation protocol. The labels (0), (1), and (4) correspond to labelled steps in Figure 6.1.

device sequentially producing bits that satisfy the δ SV condition in Equation (6.2). In our protocol, the SV source is optionally not private; such a *public* source satisfies Equation (6.2) before bits are generated but the bits may be known to an adversary afterward. A typical example of a public source is an internet randomness beacon accessible through the internet, which cannot be directly used for cryptographic applications requiring privacy. The goal of a randomness amplification and privatisation protocol is to process the outcomes of a public SV source with parameter $\delta \in [0, \frac{1}{2})$ into a final output that is provably near-perfectly random and private (i.e., satisfying criteria (6.1)). If $\delta = 0$, the SV source is already perfectly random, and no amplification is needed; however, privatisation of such a source can be desirable.

6.2.3 Quantum devices and the Mermin inequality

The central component of any device-independent protocol is the quantum device and its associated certification process, which is based on a Bell test. In our protocol, the quantum device consists of three parts, labelled Alice, Bob and Charlie, which are shielded or separated to prevent communication between them during the experiment. The certification process starts with a verifier (user) sending inputs to each part of the device, which generate and return outputs. This process is repeated

sequentially for many rounds. The objective of the interaction with these boxes is to verify that they are performing measurements on quantum states with specific properties, ruling out any classical (deterministic) explanation for the input-output combinations the user obtains. The inputs to Alice, Bob and Charlie are labelled x, y, and z, respectively, and the outputs are a, b, and c. For the Mermin inequality, the inputs and outputs in every round are bits, i.e. $x,y,z,a,b,c \in \mathbb{Z}_2$. This is summarised in Figure 6.3. After many rounds of interactions, the user can estimate the joint conditional probability distribution

$$\vec{P}_{\text{obs}} := \{\Pr(abc|xyz)\}_{x,y,z}^{a,b,c} , \tag{6.3}$$

referred to as the device's *observed behaviour*.

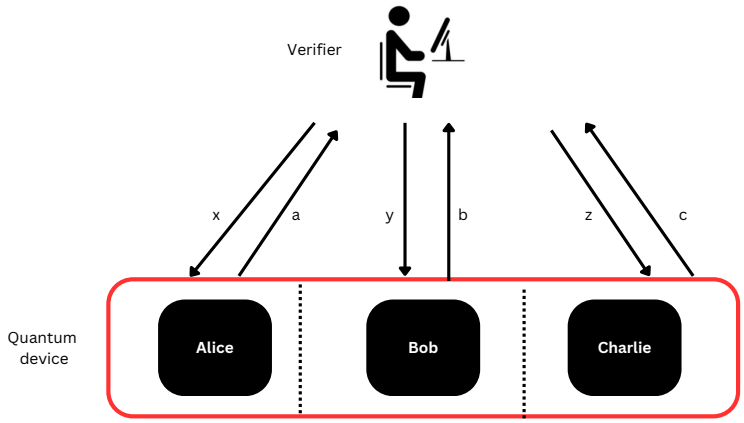

Figure 6.3: The verifier interacts with the quantum device, composed of three isolated parts Alice, Bob and Charlie, each receiving inputs x,y or z and generating outputs a,b and c.

The observed behaviour $\vec{P}_{\text{obs}}$ is used to quantify the unpredictability of the outcomes, performed by evaluating a Bell inequality. An ideal experimental setup required for testing a Bell inequality test minimises the risk of *loopholes* (see [184, Section VII B]), certifying that the only explanation for certain observed behaviours is non-classical.

In our protocol, we use the Mermin inequality [182], given by

$$M_{\text{obs}} := M(\vec{P}_{\text{obs}}) = \langle A_0 B_1 C_1 \rangle + \langle A_1 B_0 C_1 \rangle + \langle A_1 B_1 C_0 \rangle - \langle A_0 B_0 C_0 \rangle \leq 2 \,, \tag{6.4}$$

where $\langle A_x B_y C_z \rangle := \sum_{a,b,c=0,1} \left(\Pr(a \oplus b \oplus c = 0 | xyz) - \Pr(a \oplus b \oplus c = 1 | xyz) \right)$ and $\oplus$ denotes addition modulo 2. A violation of the Mermin inequality, with $M_{\text{obs}} > 2$, is possible only if the three boxes share entangled quantum states and perform quantum measurements, thereby certifying their quantum nature based solely on the observed behaviour. The Mermin inequality is advantageous for randomness amplification because, in the noiseless limit, a quantum device can reach the algebraic maximum $M_{\text{obs}} = 4$ (which is not the case, for example, with the CHSH inequality). This property enables our protocol to generate randomness from any SV source with $\delta \in [0, \frac{1}{2})$.[2]

6.2.4 Testing the Mermin inequality with weakly random inputs

In Equation 6.4, we implicitly assumed that the inputs x, y, z were chosen uniformly and independently of the device, implying no correlation between the input distribution and the device's behaviour. However, in our setting, we rely on an imperfect RNG to generate the inputs, which may be partially correlated with the quantum device through adversarial information h and quantum systems E (see Figure 6.1). Therefore, standard Bell inequalities like M_{obs} in Equation (6.4) cannot be used. Instead, we use a different type of inequalities that accommodates correlations between the inputs and the device, known as *measurement-dependent locality* (MDL) inequalities [186].

The Mermin inequality (6.4) can be written in the equivalent form $L^{\text{obs}} := L(\vec{P}_{\text{obs}})$ with

$$L(\vec{P}_{\text{obs}}) := \frac{1}{8} \sum_{\substack{abc \\ xyz}} l(a,b,c,x,y,z) \Pr(abc|xyz) \geq \frac{1}{8} \tag{6.5}$$

[2] Furthermore, using the Mermin inequality provides a practical advantage over the setup in [82], as the maximal quantum correlations are further from the classical boundary. Due to this feature, our protocol can tolerate higher noise levels, handling up to 50% white noise compared to $1 - \frac{1}{\sqrt{2}} \approx 30\%$ in [82] (see [185] for further discussion).

and

$$l(a,b,c,x,y,z) := \begin{cases} 1 & \text{if } a\oplus b\oplus c = 1 \text{ and } x,y,z \in \{011,101,110\}\,, \\ 1 & \text{if } a\oplus b\oplus c = 0 \text{ and } x,y,z = 000\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{6.6}$$

where $\frac{1}{8}$ is the local (or classical) bound. L^{obs} represents the *conditional losing probability* of the Mermin Bell game, while the function $l(\cdot)$ serves as an indicator function testing the game's *losing condition*. Using this equivalent form, we have the following relationship between L^{obs} and M_{obs},

$$L^{\text{obs}} = \frac{4 - M_{\text{obs}}}{16}\,. \tag{6.7}$$

To construct the MDL inequality [186] from the function L, we apply the mapping $\Pr(abc|xyz) \to \Pr(abcxyz)$ to convert from conditional distributions of inputs and outputs to their joint distributions. This yields $L^{\text{obs}} \to L^{\text{obs}}_{\text{MDL}} := L_{\text{MDL}}(\vec{P}_{\text{obs}})$ where

$$L_{\text{MDL}}(\vec{P}_{\text{obs}}) := \sum_{\substack{abc \\ xyz}} l(a,b,c,x,y,z)\Pr(abcxyz)\,. \tag{6.8}$$

The classical bound of $L^{\text{obs}}_{\text{MDL}}$ differs from L^{obs} and depends on the probability distribution $\Pr(xyz)$. Since the Mermin inequality is evaluated using statistics from the input settings $xyz \in \{000,110,101,011\}$, rather than all 3-bit combinations, it can be computed without the full joint probability distribution. Specifically, only 4 of the 8 settings need to be tested on the device. Therefore, the inputs x, y, and z can be generated by the imperfect RNG using two bits of randomness: generating x and y, and setting $z = x\oplus y$. If the observed probability of the input settings is uniform across the relevant settings, i.e., $\Pr(xyz) = \Pr(xy) = \sum_h \Pr(xy|h) = 1/4$ for all $xyz \in \{000,110,101,011\}$, then $2L(\vec{P}_{\text{obs}}) = L_{\text{MDL}}(\vec{P}_{\text{obs}})$, with the classical bound for $L_{\text{MDL}}(\vec{P}_{\text{obs}})$ being $1/4$. This is the case if the imperfect RNG is statistically random,

producing uniformly distributed outputs when ignoring the additional information h. For simplicity, we assume this condition in the remainder of our analysis. However, the results also hold without this assumption, in which case the bound for the MDL version of the Bell inequality is determined by the experimentally observed probabilities $\Pr(xyz)$.

Experimentally, $L_{\mathrm{MDL}}^{\mathrm{obs}}$ is calculated as the average of the losing condition results over all rounds. Specifically, for an n-round experiment, with $l_i = l(a_i, b_i, c_i, x_i, y_i, z_i)$ for all $i \in \mathbb{Z}_n$, we have $L_{\mathrm{MDL}}^{\mathrm{obs}} = \frac{1}{n}\sum_{i=0}^{n-1} l_i$.

6.2.5 Computing single-round min-entropy

We now relate the value of the MDL inequality, $L_{\mathrm{MDL}}^{\mathrm{obs}}$, to the single-round min-entropy, which will later be connected to the total accumulated entropy over the n rounds of the protocol. Without loss of generality, we assume that Alice, Bob, Charlie, and the adversary share a 4-partite quantum state ρ_{ABCE}^{h}, where Eve holds a purification of Alice, Bob, and Charlie's system. Furthermore, Alice, Bob, and Charlie perform local measurements $M_A^{a|x,h}$, $M_B^{b|y,h}$, and $M_C^{c|z,h}$, respectively. Unlike standard Bell experiments, these measurements and the state depend on the adversary's information h. The adversary makes a measurement $O_E^{e|x,y,z,h}$ on their system E, obtaining the outcome e. Since all measurements are considered to act locally on the state, so the user's outcome statistics are given by

$$\Pr(abc|xyz, Q_h) := \mathrm{tr}\left[\left(M_A^{a|x,h} \otimes M_B^{b|y,h} \otimes M_C^{c|z,h} \otimes \mathbb{1}_E\right)\rho_{ABCE}^{h}\right], \tag{6.9}$$

for unknown state and measurements, where the Hilbert space dimension is finite but arbitrary, and h inaccessible to the user. For simplicity, we denote this specific quantum realisation (the state ρ_{ABCE}^{h} and measurements $M_A^{a|x,h}, M_B^{b|y,h}, M_C^{c|z,h}$, $O_E^{e|x,y,z,h}$) by Q_h.

The adversary's ability to guess the outcomes is characterised by the maximum guessing probability $p_{\mathrm{guess}}(ABC|xyz, E, h)_Q$, where A, B, C are random variables corresponding to the outcomes a, b, c, respectively, $x, y, z = x \oplus y$ are the input bits, E is the adversary's quantum system, and h represents any additional side in-

formation. To simplify the analysis, we upper bound the adversary's ability to guess all three outcomes using the guessing probability of any two. This allows us to apply an analytical bound in later arguments, and moreover, when the Mermin inequality is maximally violated, the third outcome is fully determined by the other two and contributes no additional randomness. The resulting optimisation problem is:

$$\begin{aligned} p_{\text{guess}}(ABC|xyz,E,h)_Q &\leq p_{\text{guess}}(AB|xyz,E,h)_Q \\ &= \max_{Q_h} \sum_{a,b} \Pr(ab|xyz,Q_h)\Pr(e=(ab)|xyz,(ab),Q_h) \\ &\quad \text{subject to:} \quad L_{\text{MDL}}(\{\Pr(abcxyz,Q_h)\}_{x,y,z}^{a,b,c}) = L_{\text{MDL}}^{\text{obs}} \,. \end{aligned} \tag{6.10}$$

This optimisation problem allows the adversary to maximise their correlation with the outcomes i.e., to maximise $\Pr(e=(ab)|xyz,(ab),Q_h)$ for each outcome pair a,b, representing a worst-case bound. Besides requiring Q_h to be quantum (in the form of (6.9)), the solution must match the observed MDL inequality value (6.8). With only the MDL constraint rather than a Bell inequality constraint, directly solving this optimisation is challenging, so we instead constrain possible Bell inequality values compatible with the observed MDL violation. In particular, for all realisations Q_h (i.e., for any h), we have

$$\begin{aligned} L_{\text{MDL}}(\{\Pr(abcxyz|Q_h)\}_{x,y,z}^{a,b,c}) &= \sum_{\substack{abc\\xyz}} l(a,b,c,x,y,z)\Pr(abcxyz|Q_h) \\ &= \sum_{\substack{abc\\xyz}} l(a,b,c,x,y,z)\Pr(abc|xyz,Q_h)\Pr(xyz|h) \\ &\geq \left(\frac{1}{2}-\delta\right)^2 \sum_{\substack{abc\\xyz}} l(a,b,c,x,y,z)\Pr(abc|xyz,Q_h) \\ &= 8\left(\frac{1}{2}-\delta\right)^2 L(\{\Pr(abc|xyz,Q_h)\}_{x,y,z}^{a,b,c}) \,, \end{aligned} \tag{6.11}$$

where $l(a,b,c,x,y,z)$ is given in Equation (6.6) and the joint and conditional probabilities are related using $(\frac{1}{2}-\delta)^2 \leq \Pr(xyz|h) \leq (\frac{1}{2}+\delta)^2$ (from the δ SV assumption

in Equation (6.2), noting only 2 RNG bits are required, since $z = x \oplus y$).

Now, $L_{\text{MDL}}(\{\Pr(abcxyz|Q_h)\}_{x,y,z}^{a,b,c})$ is equal to $L_{\text{MDL}}^{\text{obs}}$ when there are no statistical fluctuations in the observed behaviour due to finite statistics. However, this is not always realistic, so we must introduce a parameter $\Delta_f(n) \geq 0$ which is a function that depends on the number of rounds, n, and allows us to relate the quantities via $L_{\text{MDL}}^{\text{obs}} + \Delta_f(n) \geq L_{\text{MDL}}(\{\Pr(abcxyz|Q_h)\}_{x,y,z}^{a,b,c})$. We note that $\Delta_f(n)$ can be computed using an appropriate concentration inequality. Combining all of the above discussion, we obtain the bound

$$L(\{\Pr(abc|xyz, Q_h)\}_{x,y,z}^{a,b,c}) \leq \frac{L_{\text{MDL}}(\vec{P}_{\text{obs}}) + \Delta_f(n)}{8(\frac{1}{2} - \delta)^2} =: L_b \,. \tag{6.12}$$

where we call L_b the *adjusted losing probability*. Thus, we have bound the required hypothetical Bell inequality value based on the observed MDL inequality, allowing us to replace the constraint $L_{\text{MDL}}(\{\Pr(abcxyz, Q_h)\}_{x,y,z}^{a,b,c}) = L_{\text{MDL}}^{\text{obs}}$ in the optimisation (6.10) by the bound from Equation (6.12). The resulting guessing probability (for two outcomes), derived in [187], is expressed as a function $p_{\text{g}}(\cdot)$ of L_b;

$$p_{\text{g}}(L_b) \leq \begin{cases} \frac{1}{4} + 2L_b + \sqrt{3L_b(1-4L_b)} & \text{if } L_b \leq \frac{1}{16} \,, \\ \frac{1}{2} + 4L_b & \text{if } \frac{1}{16} \leq L_b < \frac{1}{8} \,. \end{cases} \tag{6.13}$$

Due to setup symmetries, this guessing probability applies to all input triples xyz. The min-entropy of the outcomes is thus bounded as

$$\begin{aligned} \text{H}_\infty(ABC|xyz, E, h) &= -\log\Big(p_{\text{guess}}(ABC|xyz, E, h)_Q\Big) \\ &\geq -\log\left(p_{\text{g}}(L_b)\right) =: \text{H}_\infty(L_b) \,. \end{aligned} \tag{6.14}$$

We now evaluate how this randomness *accumulates* through multiple rounds of the data collection process.

6.2.6 Identical and independent rounds

The first scenario we consider assumes that each round of interaction with the quantum device is identical and independent of the others. This scenario helps explore

the protocol's ultimate limits, as results in more general settings often converge to those in the IID setting as $n \to \infty$ [141]. In this setting, the global quantum state describing the joint system of the adversary and the quantum device (see Figure 6.3) over n rounds is structured as

$$\rho^{n}_{\mathbf{ABCE}} = (\sigma_{ABCE})^{\otimes n} \,, \tag{6.15}$$

where $\otimes n$ denotes the tensor product of n copies, and the systems across n rounds are in bold. We assume no knowledge of σ_{ABCE}, only that such a state exists and this decomposition holds. Similarly, measurements on Alice's device are structured as $(M_A^{a|x})^{\otimes n}$, with the same assumption for Bob and Charlie's measurements. These conditions mean that the losing indicator results, $l_0, l_1, \ldots$, used to compute L_b and the outcomes of each party (i.e. $a_0, a_1, \ldots$ for Alice) are all IID. In the large-n limit, the probability $p_{\text{guess}}(\mathbf{ABC}|\mathbf{xyz}, E, h)_Q$ of guessing the outcomes $\mathbf{ABC}$ generated by n uses of the quantum device is simply the product of the guessing probabilities $p_{\text{guess}}(ABC|xyz, E, h)_Q$ of the outcomes generated at each round

$$p^{\text{IID}}_Q[n] := p_{\text{guess}}(\mathbf{ABC}|\mathbf{xyz}, E, h)_Q = \Big(p_{\text{guess}}(ABC|xyz, E, h)_Q\Big)^n \,. \tag{6.16}$$

Using the single-round bound from the previous section, we have that

$$p^{\text{IID}}_Q[n] = \Big(p_{\text{guess}}(ABC|xyz, E, h)_Q\Big)^n \leq \Big(p_{\text{guess}}(AB|xyz, E, h)_Q\Big)^n \leq \Big(p_{\text{g}}(L_b)\Big)^n \,. \tag{6.17}$$

In the large-n limit, this means we can express the total accumulated min-entropy, for two outcomes per round, as

$$\mathrm{H}_\infty(\mathbf{AB}|\mathbf{xyz}, E, h) \geq n\mathrm{H}_\infty(L_b) = -n\log\Big(p_{\text{g}}(L_b)\Big) \,. \tag{6.18}$$

Notably, this bound applies to two outcomes per round, and when $L_b = 0$, the min-entropy is $2n$. Thus, under this assumption and considering only two outcomes per

round gives a min-entropy rate

$$\alpha_{\mathrm{IID}} := \frac{\mathrm{H}_\infty(\mathbf{AB}|\mathbf{xyz}, E, h)}{2n} \geq -\frac{1}{2}\log\Big(p_{\mathrm{g}}(L_b)\Big)\,. \tag{6.19}$$

Assuming the quantum device operates identically and independently at every round may be reasonable in some cases, for example, if the device provider is trusted and the device operates at low speed. However, it is generally quite a restrictive assumption.

6.2.7 Memory based quantum attacks (MBQA)

To generalise the security proof, we use the framework from [188, 141, 82], and apply the *entropy accumulation theorem* (EAT) [153]. This approach relies on a reduction to the single-round quantities introduced in Section 6.2.5, where the total entropy accumulated over n rounds is derived from single-round quantities with an added penalty term. It is important to note that not all structure is lost. The interactions with the quantum device occur sequentially, and to apply the EAT, the experiment and protocol must satisfy certain conditions [188, 141]:

1. The outputs $\{A_iB_iC_i\}_{i\in\mathbb{Z}_n}$, the inputs $\{X_iY_iZ_i\}_{i\in\mathbb{Z}_n}$ (which are *leaked* to the adversary, since the SV source is understood to be public) and the losing condition evaluated at each round (6.6), $\{L_i\}_{i\in\mathbb{Z}_n}$ are all finite-dimensional classical random variables.
2. At each round, the losing condition l_i (6.6) can be evaluated from the classical outcomes $a_i, b_i, c_i, x_i, y_i, z_i$, without altering the underlying state.
3. For all i, the following Markov condition holds:

$$\mathrm{I}(\mathbf{A}_{i-1}, \mathbf{B}_{i-1}, \mathbf{C}_{i-1} : X_i, Y_i, Z_i | \mathbf{X}_{i-1}, \mathbf{Y}_{i-1}, \mathbf{Z}_{i-1}, E)\,, \tag{6.20}$$

where $\mathrm{I}(\cdot)$ is the mutual information function, $\mathbf{A}_{i-1}$ denotes the sequence of inputs $A_0, \ldots A_{i-1}$ and similarly for $\mathbf{B}_{i-1}, \mathbf{C}_{i-1}, \mathbf{X}_{i-1}, \mathbf{Y}_{i-1}$ and $\mathbf{Z}_{i-1}$. This condition implies that the inputs X_i, Y_i, Z_i in round i reveal no additional information about previous outcomes beyond what is already known with the past

inputs and the adversary's system E (conditioned on the information h). In other words, the inputs in round i are independent of all previous outcomes, given all prior inputs and the adversary's system E.

These conditions are reasonable in certain experimental scenarios, such as when the imperfect RNG cannot adapt its behaviour based on the outcomes of the quantum device. Assuming these conditions hold, the EAT can be applied, and so we now compute the necessary quantities. For details on these computations and their relevance, see [82, Theorem 33]. The derivative of the single round min-entropy $\mathrm{H}_\infty(L_b) = -\log(p_{\mathrm{g}}(L_b))$ with respect to the MDL inequality value $L_{\mathrm{MDL}}^{\mathrm{obs}}$ is

$$\begin{aligned}\frac{\mathrm{dH}_\infty(L_b)}{\mathrm{d}L_{\mathrm{MDL}}^{\mathrm{obs}}} &= \frac{\mathrm{dH}_\infty(L_b)}{\mathrm{d}L_b}\frac{\mathrm{d}L_b}{\mathrm{d}L_{\mathrm{MDL}}^{\mathrm{obs}}} \\ &= \begin{cases} \frac{-2-\frac{\sqrt{3}(1-8L_b)}{2\sqrt{L_b(1-4L_b)}}}{\ln(2)(\frac{1}{4}+2L_b+\sqrt{3L_b(1-4L_b)})}\frac{1}{8(\frac{1}{2}-\delta)^2} & \text{if } L_b \leq \frac{1}{16}\,, \\ \frac{-4}{\ln(2)(\frac{1}{2}+4L_b)}\frac{1}{8(\frac{1}{2}-\delta)^2} & \text{if } \frac{1}{16} \leq L_b < \frac{1}{8}\,, \end{cases}\end{aligned} \tag{6.21}$$

where L_b is given in Equation (6.12). Define the min-tradeoff function $f_{\min}$ as

$$f_{\min}(L_b, L_{\mathrm{cut}}) = \begin{cases} \mathrm{H}_\infty(L_b) & \text{if } L_b \geq L_{\mathrm{cut}}\,, \\ \mathrm{H}_\infty(L_{\mathrm{cut}}) + (L_b - L_{\mathrm{cut}})\frac{\mathrm{dH}_\infty}{\mathrm{d}L_{\mathrm{MDL}}^{\mathrm{obs}}}\Big|_{L_b=L_{\mathrm{cut}}} & \text{if } L_b < L_{\mathrm{cut}}\,, \end{cases} \tag{6.22}$$

with L_{cut} as a degree of freedom to be optimised. The total conditional *smooth min-entropy*[3] accumulated in outputs **ABC** over n rounds, following [82, Equation 33], is then

$$\mathrm{H}_\infty^\kappa(\mathbf{ABC}|(\mathbf{xyz})^*, E, h) \geq n \max_{L_{\mathrm{cut}}} \left(f_{\min}(L_b, L_{\mathrm{cut}}) - \frac{\nu(L_b, \kappa, \epsilon_{\mathrm{EAT}})}{\sqrt{n}} \right), \tag{6.23}$$

where κ is a smoothing parameter and $\epsilon_{\mathrm{EAT}} \in (0,1)$ is a security parameter related to the entropy accumulation step failing, which is discussed in more detail in Section 6.3.3. The penalty term $\nu(L_b, \kappa, \epsilon_{\mathrm{EAT}})$ is computed as (from [188, Lemma 10],

[3]The *smooth min-entropy* of a quantum state ρ_A, denoted $\mathrm{H}_\infty^\kappa(A)_\rho$ is defined as $\mathrm{H}_\infty^\kappa(A)_\rho := \max_{\tilde{\rho}\in\mathcal{B}^\kappa(\rho)} \mathrm{H}_\infty(A)_{\tilde{\rho}}$ where $\mathcal{B}^\kappa(\rho)$ denotes all density operators $\tilde{\rho}_A$ that are κ-close to ρ_A in terms of trace distance.

see also [82, Claim 1])

$$\nu(L_{\text{cut}},\kappa,\epsilon_{\text{EAT}}) := 2(\log(17)+\left\lceil\left|\frac{\mathrm{dH}_\infty(L_b)}{\mathrm{d}L^{\text{obs}}_{\text{MDL}}}\right|_{L_b=L_{\text{cut}}}\right\rceil)\sqrt{1-2\log(\kappa\epsilon_{\text{EAT}})}\,, \tag{6.24}$$

where $\left|\frac{\mathrm{dH}_\infty(L_b)}{\mathrm{d}L^{\text{obs}}_{\text{MDL}}}\right|_{L_b=L_{\text{cut}}}$ is the maximum value of $\left|\frac{\mathrm{dH}_\infty(L_b)}{\mathrm{d}L^{\text{obs}}_{\text{MDL}}}\right|$, achieved at L_b. The total conditional smooth min-entropy, Equation (6.23), can be loosely interpreted as the single round IID case, $f_{\min}(L_b,L_{\text{cut}})$, with a penalty term $\frac{\nu(L_b,\kappa,\epsilon_{\text{EAT}})}{\sqrt{n}}$ to account for general adversary attacks and memory effects in the device. We note that the min-tradeoff function can be based on the von Neumann entropy; however, we use min-entropy as a lower bound due to its analytic expression, which enables a closed-form derivative in Equation (6.21). Substantial improvements may be possible by working directly with the von Neumann entropy (see Open Problem 13).

The min-entropy rate, the average min-entropy per bit, of the $3n$-bit outcome string produced by the quantum device across the n rounds, is

$$\alpha_{\text{MBQA}} := \frac{\mathrm{H}^{\kappa}_{\infty}(\mathbf{ABC}|(\mathbf{xyz})^*,E,h)}{3n} > \frac{1}{3}\max_{L_{\text{cut}}}\left(f_{\min}(L_b,L_{\text{cut}})-\frac{\nu(L_b,\kappa,\epsilon_{\text{EAT}})}{\sqrt{n}}\right). \tag{6.25}$$

Due to using the analytic bound on the two-outcome guessing probability, a limitation of our protocol is $\alpha_{\text{MBQA}} \leq \frac{2}{3}$. This limitation could be removed if the EAT could be used to calculate the accumulated entropy when considering only two outcomes (of the three) per round, as is done in the IID case. However, this is currently not possible as such a protocol would fail to satisfy the three conditions required to apply the EAT (see the discussion in [189] following Lemma 11.3).

6.2.8 Post-processing randomness

When verification is successful, the final step is to process the raw output into near-perfect randomness, Equation (1.8). This is accomplished using randomness extractors for min-entropy sources, as described in Chapter 3. Here, we employ two types of extractors: two-source and seeded. To ensure security against quantum adversaries, it is crucial to use quantum-proof randomness extractors that can withstand potential quantum attacks. We distinguish between two tasks:

- **Randomness amplification:** Random number generation where the user has access to a *private* δ SV source, meaning the outcomes are not known to the adversary.
- **Randomness amplification and privatisation:** Random number generation in which the user has access to a *public* δ SV source, meaning the outcomes are known to the adversary after generation but cannot be fully predicted beforehand.

For both tasks, we describe the randomness post-processing setup and which randomness extractors we use.

In randomness amplification, illustrated in Figure 6.4, both the output of the imperfect RNG and the quantum device are assumed to be private. In this protocol, the outcomes of the quantum device (length n_1) and an additional bit string from the imperfect RNG (length n_2) are processed by a two-source randomness extractor. The resulting string of near-perfect and private random bits is then extended using a seeded randomness extractor, requiring an additional n_S bits from the imperfect RNG.[4]

For randomness amplification and privatisation, shown in Figure 6.5, the imperfect RNG is assumed to be public. In this protocol, the outcomes of the quantum device (length n_1) and an additional bit string from the imperfect RNG (length n_2) are processed by a two-source randomness extractor. Importantly, the two-source extractor must be strong (in the imperfect RNG) to ensure that the final output is independent from the public imperfect RNG data. Note that the final output length can be increased by generating an additional n_S bits from the quantum device using another n_S inputs from the imperfect RNG, followed by seeded extraction, as in the amplification-only scenario.

To ensure practical relevance, randomness post-processing must achieve throughput rates suitable for real-world applications. For realistic quantum hard-

[4] If the seeded extractor is strong, the second extraction step can be repeated as needed, allowing the imperfect RNG to be amplified almost indefinitely. The only restriction arises from an additive error contribution of each extraction, which must be taken into account when computing the final security parameter (see [2, Section 3.2] for a discussion on composing extractors).

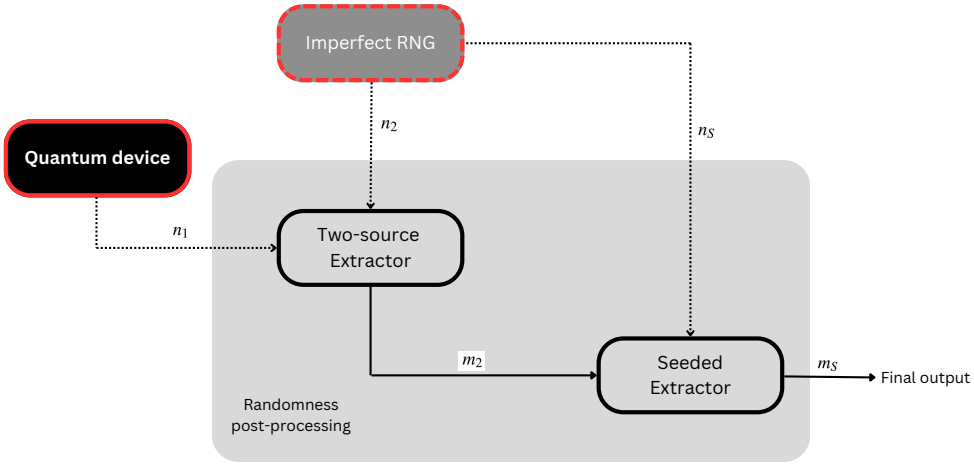

Figure 6.4: Randomness post-processing flow for randomness amplification only (label (5) in Figure 6.1).

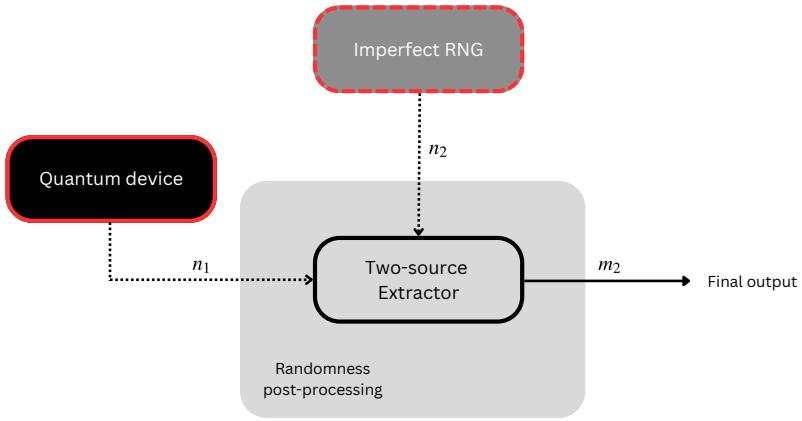

Figure 6.5: Randomness post-processing flow for randomness amplification and privatisation (label (5) in Figure 6.1).

ware, sensible security parameters typically require at least $n \approx 10^7$ rounds to derive meaningful bounds on the smooth min-entropy using MBQA analysis. Consequently, randomness extractor algorithms with $O(n \log n)$ computation time are essential. To meet this requirement, we use the extractor constructions from Chapter 3, which leverage the NTT for an information-theoretically secure implementation with $O(n \log n)$ computation time. These implementations achieve processing

speeds of several Mb/s on a standard laptop, even with input lengths exceeding $n > 10^8$ bits.

Santha-Vazirani source. As mentioned in Section 6.2.2, we model the imperfect RNG as a Santha-Vazirani source (6.2) with parameter $\delta > 0$. Hence, any n bits $\mathbf{r} \in \mathbb{Z}_2^n$ generated by the imperfect RNG can be guessed by the adversary with a probability of at most

$$p_{\mathrm{SV}}[n] \leq 2^{n \log(1/2+\delta)} \,. \tag{6.26}$$

Thus, the probability of guessing an n-bit string generated by a Santha-Vazirani source decreases exponentially with n. The entropy of n bits from a δ SV source is $\mathrm{H}_\infty(\mathbf{r}|h) \geq -n \log(1/2+\delta)$.

Two-source extractor. Our first extractor, for both tasks, is the Circulant extractor [2], presented in Section 3. For an n_1-bit input source and an $n_2 = n_1 + 1$-bit weak seed, with min-entropies k_1 and k_2 respectively, and $n_2 = n_1 + 1$ prime, the Circulant extractor is a strong quantum-proof $(n_1, k_1, n_2, k_2, m_2, \epsilon_{\mathrm{ext}})$ two-source extractor with

$$m_2 = \left\lfloor k_1 + (k_2 - n_2) + 2\log(\epsilon_{\mathrm{ext}}) \right\rfloor , \tag{6.27}$$

where $\epsilon_{\mathrm{ext}} > 0$ is the extractor error. We note that this extractor is computable in $O(n_1 \log n_1)$ time. For sufficiently large input lengths, n_1, n_2, this extractor allows to distil ϵ_{ext}-perfect randomness roughly when $k_1 + k_2 > n_2$.

Consider an example where the guessing probabilities for the quantum device and SV source are denoted as $p_Q[n_1]$ and $p_{SV}[n_2]$, respectively, and are defined as

$$p_Q[n_1] \leq 2^{-n_1 c_Q} \,, \qquad p_{SV}[n_2] \leq 2^{-n_2 c_{SV}} \,, \tag{6.28}$$

for $c_Q, c_{SV} \in [0,1]$. The min-entropy of each is therefore $k_1 = c_Q$ and $k_2 = c_{SV}$. Using the Circulant extractor, the output length m_2 is

$$m_2 \approx (c_Q + c_{SV} - 1) n_1 \,. \tag{6.29}$$

As a concrete example, if an experiment yields $c_Q = 0.35$, and an imperfect RNG has quality $\delta = 0.036$ (corresponding to $c_{SV} = 0.9$), the linear output length is

$$m_2 \approx (0.9 + 0.35 - 1)n_1 = 0.25n_1 \, . \tag{6.30}$$

We also consider our improved Raz extractor, presented in Chapter 3, which enables the amplification of lower-quality imperfect RNGs. However, it suffers significantly higher entropy loss compared to the Circulant extractor. Since the focus of this work is on practicality, we use the Raz extractor primarily for comparison and focus most of the analysis on the Circulant extractor.

Working with smooth min-entropy. When using the EAT, the quantum-proof two-source extractors must operate on sources with guaranteed smooth min-entropy, rather than just min-entropy. This is covered by the following lemma.

Lemma 68. Let $\mathrm{Ext} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \to \mathbb{Z}_2^m$ be a quantum-proof $(n_1, k_1, n_2, k_2, m, \epsilon_{\mathrm{ext}})$ two-source extractor, $\kappa \in (0, 1]$, and $\epsilon_1 \in (0, 1)$. Then, for any Markov source ρ_{XYE} with $\mathrm{H}_\infty^\kappa(X|E) \geq k_1 - \log(\epsilon_1)$ and $\mathrm{H}_\infty(Y|E) \geq k_2$,

$$\frac{1}{2} \left\| \rho_{\mathrm{Ext}(X,Y)E} - \omega_{\mathrm{Ext}(X,Y)} \otimes \rho_E \right\|_1 \leq 6\kappa + 2\epsilon_1 + 2\epsilon_{\mathrm{ext}} \, . \tag{6.31}$$

Similarly, for any strong quantum-proof $(n_1, k_1, n_2, k_2, m, \epsilon_{\mathrm{ext}})$ two-source extractor strong in Y, we have

$$\frac{1}{2} \left\| \rho_{\mathrm{Ext}(X,Y)YE} - \omega_{\mathrm{Ext}(X,Y)} \otimes \rho_{YE} \right\|_1 \leq 6\kappa + 2\epsilon_1 + 2\epsilon_{\mathrm{ext}} \tag{6.32}$$

and likewise for extractors strong in X.

Proof. The proof follows from [62, Lemma 17], relying on two adapted results: (i) the existence of a subnormalised Markov source σ_{XYE} with $\mathrm{H}_\infty(X|E) \geq k_1$ and $\mathrm{H}_\infty(Y|E) \geq k_2$, which is $3\kappa + \epsilon_1$-close (in trace distance) to the original state, which can be shown analogously to [62, Lemma 18]; and (ii) applying [62, Lemma 37] to bound the extractor's trace distance when a subnormalised state is used as input. □

Seeded extractor. Our second extractor is an explicit implementation of the quantum-proof extractor '$f_{F2,R}$' due to Hayashi and Tsurumaru [5, Section V.B]. This construction was originally developed for quantum key distribution protocols, but some adaptations make the work applicable (and very useful) in our settings. For an n_S-bit source and a seed of length $m_2 = n_S - m_S$, the Hayashi-Tsurumaru $f_{F2,R}$ extractor (which we will refer to as simply the Hayashi-Tsurumaru extractor) is a strong quantum-proof $(n_S, k_S, m_2, k_2 = m_2, m_S, \epsilon_S)$ seeded extractor, with

$$m_S = k_S + 2\log\epsilon_S - \log\left\lceil \frac{n_S - m_2}{m_2} \right\rceil , \tag{6.33}$$

where $\epsilon_S > 0$ denotes the extractor error. This leads to linear output rates as long as the guessing probability of the source is of the form

$$p_S[n_S] \leq 2^{-\alpha n_S} , \tag{6.34}$$

for some $\alpha \in (0, 1]$. Conveniently, this is essentially the form of the guessing probability of a δ SV source, given in Equation (6.26). For a source satisfying the condition in Equation (6.34), the extractor output m_S can be written as

$$m_S = (c-1)m_2 , \tag{6.35}$$

where c is an integer satisfying $1 \leq c \leq \lfloor \frac{1}{1-\alpha} \rfloor$, and the error is given by

$$\epsilon_S \leq 2^{-m_2 \frac{1+c(\alpha-1)}{2}} \sqrt{c-1} . \tag{6.36}$$

Note that condition (6.35) requires $\alpha > 1/2$ (which, for SV sources is roughly $\delta < 0.207$), so the extractor only works for sources that are already sufficiently unpredictable. This extractor can be implemented with quasi-linear computation time $O(n_S \log n_S)$ using the NTT procedure described in Section 2.3, as outlined in [17, Appendix D].

As a concrete example, if $m_2 = 10^4$ and $\alpha = 9/10$, then $c \leq 10$, and for $c = 9$, the output length is $m_S = 8m_2$ with an error $\epsilon_S \leq 10^{-150}$. While other seeded extractors

discussed in this work could be used, this extractor is particularly effective for high-quality imperfect RNGs (i.e., with δ close to 0), as the seed length is $m_2 = n_S/c$.

Working with near-perfect seeds. When performing seeded extraction using the output of a two-source extractor as a seed, the quantum-proof seeded extractor must operate with a seed that is ϵ_{ext}-close to perfectly random, rather than perfectly random. This scenario is encompassed by the following lemma.

Lemma 69. Let $\text{Ext} : \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \to \mathbb{Z}_2^m$ be a quantum-proof $(n_1, k_1, n_2, k_2 = n_2, m, \epsilon_S)$ seeded extractor. Then, for any source $\rho_{XYE} = \rho_{XE} \otimes \rho_Y$ with $\text{H}_\infty(X|E) \geq k_1$ and $\text{H}_\infty^{\epsilon_{\text{ext}}}(Y|E) = n_2$,

$$\frac{1}{2} \left\| \rho_{\text{Ext}(X,Y)E} - \omega_{\text{Ext}(X,Y)} \otimes \rho_E \right\|_1 \leq \epsilon_S + \epsilon_{\text{ext}} \,. \tag{6.37}$$

Proof. This is a special case of the result in [190, Appendix A], setting the number of seeded extractions (denoted t in [190]) to 1. □

6.3 Protocol and concrete numerical examples

This section presents the full protocol, integrating the steps from the previous section and illustrating the results achievable with our protocol through numerical examples. Our protocol can be used in two distinct ways:

(a) With a *public* imperfect source of randomness, generating a ϵ_{sec}-secure output – randomness amplification and privatisation.

(b) With a *private* imperfect source of randomness, generating a ϵ_{sec}-secure output – randomness amplification only.

6.3.1 Steps of the protocol

Our full protocol is summarised in Figure 6.6. Following [82], we consider an *expected adjusted losing probability*, denoted $\tilde{L}_b$, which is set by the user at the start of the protocol. If the input-output statistics produced by the quantum device during the protocol satisfy $L_b \leq \tilde{L}_b$, post-processing is performed and a final output is produced; otherwise, the protocol aborts. If the protocol does not abort, the min-

entropy of the quantum device outcomes is computed using $\tilde{L}_b$, resulting in a fixed output length corresponding to the chosen value of $\tilde{L}_b$.

RANDOMNESS AMPLIFICATION AND PRIVATISATION PROTOCOL

1. Set parameters: Define n (total rounds), $\tilde{L}_b$ (expected adjusted losing probability) and $\epsilon_{\text{sec}} > 0$ (tolerable error).

2. Data collection: For each of the n rounds:

2a. Generate 2 bits x, y with the imperfect RNG of quality δ as defined in (6.2).

2b. Drive the quantum device with settings $x, y, z = x \oplus y$ and collect the 3 output bits a, b, c. Record the 6 bits for each round.

3. Randomness certification:

3a. From the recorded data, compute the adjusted losing probability L_b with Equation (6.12).

3b. If the adjusted losing probability $L_b \leq \tilde{L}_b$, proceed to randomness post-processing; otherwise, abort.

4. Randomness post-processing:

4a. Collect outputs a, b, c for each of the n rounds (if the device is IID, collect only a, b each round).

4b. Send this bit string of length $3n$ ($2n$ if IID) to the strong Circulant two-source extractor along with a fresh $3n+1$-bit (or $2n+1$-bit) string from the imperfect RNG. The two-source extractor outputs an m_2-bit string of secure random numbers as defined by Equation (6.1).

4c. (Randomness amplification only) Extend the m_2-bit string using the quantum-proof Hayashi-Tsurumaru [5] seeded extractor, using a new string from the imperfect RNG. The seeded extractor output is an m_S-bit string ($m_S > m_2$) of secure random numbers.

Figure 6.6: Randomness amplification and privatisation protocol.

6.3.2 List of assumptions

For clarity, we collect a list of all the assumptions needed to run our device-independent randomness amplification and privatisation protocol.

1. The devices and adversary operate according to quantum theory.
2. The classical computer used (see Figure 6.1) is trusted and functions correctly.
3. The user's facility in which the protocol is run is shielded from the outside, in particular, it does not signal to Eve.

4. The quantum device is made of three separate parts that do not exchange information during a round of the experiment (see Figure 6.3).
5. The adversary only holds classical information h about the imperfect RNG, that is a δ SV source.
6. For randomness amplification only, the output of the imperfect RNG is assumed to be private. For randomness amplification and privatisation, the output of the imperfect RNG may be public (i.e. known to the adversary).
7. The conditions for the EAT, listed in Section 6.2.7, hold. In particular, for all i, the following Markov condition holds:

$$\mathrm{I}(\mathbf{A}_{i-1}, \mathbf{B}_{i-1}, \mathbf{C}_{i-1} : X_i, Y_i, Z_i | \mathbf{X}_{i-1}, \mathbf{Y}_{i-1}, \mathbf{Z}_{i-1}, E) \,, \tag{6.38}$$

   where $\mathrm{I}(\cdot)$ is the mutual information function, $\mathbf{A}_{i-1}$ denotes the sequence of inputs $A_0, \ldots A_{i-1}$ and similarly for $\mathbf{B}_{i-1}, \mathbf{C}_{i-1}, \mathbf{X}_{i-1}, \mathbf{Y}_{i-1}$ and $\mathbf{Z}_{i-1}$.
8. If the devices running the Bell test are later re-used, they do not leak any relevant information about previous protocols that were run on them.

We note that without Assumptions 2 and 3, no cryptography would be possible. Assumption 1 has been generalised in some works [81, 161, 162] to adversaries who are not necessarily constrained by the laws of quantum mechanics, at the cost of a substantial reduction in protocol efficiency. Assumptions 4, 5, 6, 7 and 8 are related to our specific setting and are necessary to obtain security.

6.3.3 Security analysis: putting everything together

We now combine the statistical analysis of the quantum component with the classical post-processing to present a complete security analysis. In the MBQA case, the n-round Bell test using $2n$ bits of imperfect randomness, as described in Section 6.2.4, guarantees that for any smoothing parameter $\kappa \in (0, 1)$ and entropy accumulation security parameter $\epsilon_{\mathrm{EAT}} \in (0, 1)$, the protocol either aborts with probability at least $1 - \epsilon_{\mathrm{EAT}}$ or the $3n$ outcomes from the quantum device have κ-smooth min-entropy at least $k_1^{\kappa} = 3\alpha_{\mathrm{MBQA}} n$ (see [82, Theorem 4]). If the protocol does not abort, the randomness post-processing in the MBQA setting is performed as follows.

- The quantum device outputs $3n$ bits with κ-smooth min-entropy rate α_{MBQA}, as given in Equation (6.25) and denoted $[3n, \kappa, k_1^{\kappa} = 3\alpha_{\mathrm{MBQA}} n]$.
- From the imperfect RNG, obtain $3n+1$ bits with a min-entropy rate α_{SV}, denoted by $[3n+1, 0, k_2 = \alpha_{\mathrm{SV}}(3n+1)]$. Here, $\alpha_{\mathrm{SV}} = -\log\left(\frac{1}{2}+\delta\right)$, where δ is the SV source parameter (Equation (6.2)).
- The strong quantum-proof $(3n, k_1, 3n+1, k_2, m_2, \epsilon_{\mathrm{ext}})$ two-source extractor Circulant, presented in Section 3.3, with output length m_2 (6.27), is applied to the source $[3n, \kappa, k_1^{\kappa}]$ (in the smooth min-entropy form of Lemma 68) and the (weak) seed $[3n+1, 0, k_2]$. Note that the extractor is strong with respect to the imperfect RNG, and $3n+1$ must be prime (a list of such numbers is provided in Appendix A).[5]
- The output consists of m_2 bits of randomness with security parameter ϵ_2, denoted $[m_2, \epsilon_2]$, where m_2 depends on min-entropy of the inputs and the tolerable error $\epsilon_{\mathrm{sec}} \geq \epsilon_2 = 6\kappa + 2\epsilon_1 + 2\epsilon_{\mathrm{ext}}$. We note that ϵ_1 and κ are free parameters that can be optimised over to maximise randomness generation rates.

Putting this all together, the two-source extractor output length m_2 is given by

$$m_2 = \lfloor k_1 + (k_2 - 3n - 1) + 2\log(\epsilon_{\mathrm{ext}}) \rfloor, \tag{6.39}$$

with total error $\epsilon_2 = 6\kappa + 2\epsilon_1 + 2\epsilon_{\mathrm{ext}}$ as in (6.31). Applying MBQA statistical analysis, we compute $k_1 = k_1^{\kappa} - \log(\epsilon_1)$ where $k_1^{\kappa} = 3n\alpha_{\mathrm{MBQA}}$ from Equation (6.25). From (6.2), $k_2 = \alpha_{\mathrm{SV}}(3n+1)$ with $\alpha_{\mathrm{SV}} = -\log\left(\frac{1}{2}+\delta\right)$. The optimal output length m_2 for a given security parameter ϵ_{sec}, δ, and expected adjusted losing probability $\tilde{L}_b$, is

$$m^{\mathrm{opt}}_{2,\mathrm{MBQA}}(\tilde{L}_b, \epsilon_{\mathrm{sec}}) = \max_{\substack{\kappa,\epsilon_1,\epsilon_{\mathrm{ext}} \\ \text{s.t. } 6\kappa+2\epsilon_1+2\epsilon_{\mathrm{ext}} \leq \epsilon_{\mathrm{sec}} \\ \text{and } \epsilon_{\mathrm{EAT}} \leq \epsilon_{\mathrm{sec}}}} \Bigg\lfloor (3n+1)(\alpha_{\mathrm{SV}} - 1) + \log(\epsilon_1) + 2\log(\epsilon_{\mathrm{ext}}) + n \max_{L_{\mathrm{cut}}} \left(f_{min}(\tilde{L}_b, L_{\mathrm{cut}}) - \frac{\nu(\tilde{L}_b, \kappa, \epsilon_{\mathrm{EAT}})}{\sqrt{n}} \right) \Bigg\rfloor , \tag{6.40}$$

[5] We note that $3n+1$ cannot be prime, as it is even by definition. Therefore, a slightly longer seed length, $3n+c$, must be used for an integer $c > 1$. The output from the quantum device should then be padded with $c-1$ fixed bits to match the required seed length.

with $\nu(L_b, \kappa, \epsilon_{\text{EAT}})$ as in Equation (6.24), $\tilde{L}_b \geq L_b$ and $f_{min}(\tilde{L}_b, L_{\text{cut}})$ from Equation (6.22).

To ensure the protocol satisfies the security condition in Equation (6.1), two cases must be considered: (1) when the protocol aborts with probability less than $1 - \epsilon_{\text{EAT}}$, and (2) when the protocol aborts with probability greater than $1 - \epsilon_{\text{EAT}}$. In case (1), the security condition in Equation (6.1) is trivially bounded by ϵ_{EAT}, as the trace distance is always at most 1. In case (2), the security condition in Equation (6.1) is bounded by ϵ_2, which limits the trace distance term while ensuring the probability of not aborting is at most 1. Since the conditions $\epsilon_{\text{EAT}} \leq \epsilon_{\text{sec}}$ and $\epsilon_2 \leq \epsilon_{\text{sec}}$ are incorporated into the output length expression in Equation (6.40), the protocol satisfies the security condition in Equation (6.1).

In the IID case the n-round Bell test using $2n$ bits of imperfect randomness (see Section 6.2.6), produces $2n$ bits (since only 2 outcomes per round are recorded), with at-least min-entropy $k_1 = 2\alpha_{\text{IID}} n$ from Equation (6.16). All subsequent steps for randomness post-processing are the same as in the MBQA case, except that $2n$ bits are used from the quantum device instead of $3n$, $k_1 = 2n\alpha_{\text{IID}}$ replaces $k_1^{\kappa} = 3n\alpha_{\text{MBQA}}$, and $\epsilon_1 = \kappa = 0$. This results in the optimised output of the quantum-proof Circulant extractor by setting $\epsilon_{\text{ext}} = \epsilon_{\text{sec}}$, giving

$$m^{\text{opt}}_{2,\text{IID}}(\tilde{L}_b, \epsilon_{\text{sec}}) = \lfloor (2n+1)(\alpha_{\text{SV}} - 1) + 2\log(\epsilon_{\text{sec}}) - n\log\left(p_{\text{g}}(\tilde{L}_b)\right) \rfloor . \tag{6.41}$$

Randomness amplification only. In the case of randomness amplification only, a seeded extractor is added, as illustrated in Figure 6.5, and is performed as follows:

- From the previous post-processing, we obtain m_2 bits of randomness with security parameter ϵ_2, denoted $[m_2, \epsilon_2]$, using a two-source extractor.
- An additional n_S bits are generated by the imperfect RNG, with min-entropy rate α_{SV}, denoted $[n_S, 0, k_S = \alpha_{\text{SV}} n_S]$.
- The Hayashi-Tsurumaru quantum-proof $(n_S, k_S, m_2, k_2 = m_2, m_S, \epsilon_S)$ seeded extractor is applied to the source $[n_S, 0, k_S = \alpha_{\text{SV}} n_S]$, using the output of the two-source extractor as a seed, $[m_2, \epsilon_2]$. This process produces m_S output

bits with a combined security parameter of $\epsilon_2 + \epsilon_S$, denoted as $[m_S, \epsilon_2 + \epsilon_S]$, where Lemma 69 is used to account for the ϵ_S-perfect seed. Recall that $m_S = (c-1)m_2$, where $1 \leq c \leq \lfloor \frac{1}{1-\alpha_{\text{SV}}} \rfloor$ is an integer, and the security parameter satisfies $\epsilon_S \leq 2^{-m_2 \frac{1+c(\alpha-1)}{2}} \sqrt{c-1}$.

This setup allows achieving $m_S \gg m_2$ in the final output $[m_S, \epsilon_2 + \epsilon_S]$, where m_S can be optimised such that $\epsilon_2 + \epsilon_S \leq \epsilon_{\text{sec}}$. This step can be repeated if the extractor is a strong seeded extractor, giving a final output of $[tm_S, \epsilon_2 + t\epsilon_S]$, where t is the number of repetitions (for a proof, see [190, Appendix A]).

6.3.4 Efficiency of the protocol

The protocol *efficiency*, $\mathcal{R}_{\text{eff}} = \frac{m_2}{n}$, represents the relative output length per use of the quantum device. The derivation and formula for m_2 are provided in Equation (6.40) in the case of MBQA, whilst the IID case follows from using Equation (6.41) with the min-entropy (6.18) for 2 outcomes per round. For simplicity, we set the security parameter $\epsilon_{\text{sec}} \leq 2^{-32} \approx 10^{-10}$ and penalty term $\Delta_f(n) = 0$, as $\Delta_f(n)$ decreases exponentially with n.

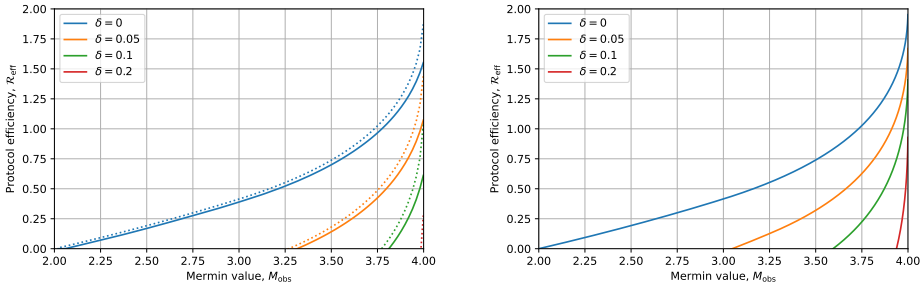

Figure 6.7: The protocol efficiency $\mathcal{R}_{\text{eff}} = \frac{m_2}{n}$ at the output of the Circulant two-source extractor as a function of the observed Mermin value M_{obs} for differing qualities of imperfect RNG (δ). Left: MBQA with $n = 10^8$ (solid lines) and $n = 10^{11}$ (dotted lines). Right: IID in the asymptotic limit ($n \to \infty$).

The efficiency $\mathcal{R}_{\text{eff}}$ as a function of M_{obs} is shown in Figure 6.7. To summarise the quality of randomness, $\delta = 0$ corresponds to a perfectly random source, while the values δ quantify the predictability of the RNG output as follows:

- $\delta = 0.05$: $\approx 86\%$ random,
- $\delta = 0.1$: $\approx 74\%$ random,

- $\delta = 0.2$: $\approx 51\%$ random.

Appending a seeded extractor. If the protocol is used for randomness amplification only, (i.e. the imperfect RNG is private), a seeded extractor can be used to extend the output length after two-source extraction. If the Hayashi-Tsurumaru extractor is appended, using Equation (6.35), one can obtain $m_S = (c-1)m_2$ bits where $1 \leq c \leq \lfloor \frac{1}{1-\alpha_{\mathrm{SV}}} \rfloor$ with error $\epsilon_S + \epsilon_2$, where $\epsilon_S \leq 2^{-m_2 \frac{1+c(\alpha_{\mathrm{SV}}-1)}{2}} \sqrt{c-1}$. These improved efficiency rates, denoted $\mathcal{R}^{\mathrm{s}}_{\mathrm{eff}}$ are presented in Figure 6.8.

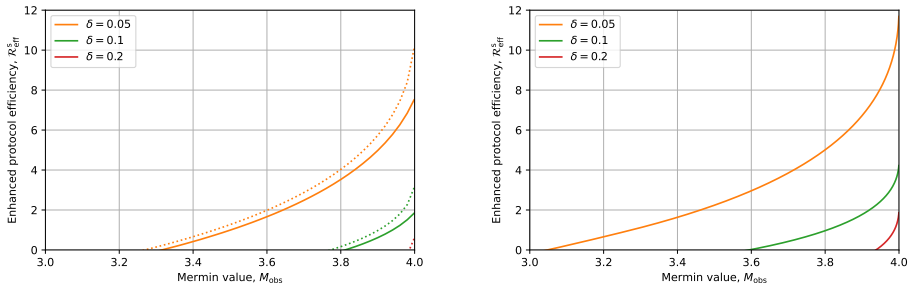

Figure 6.8: The enhanced protocol efficiency $\mathcal{R}^{\mathrm{s}}_{\mathrm{eff}} = \frac{m_S}{n}$ at the output of the Hayashi-Tsurumaru seeded extractor as a function of the observed Mermin value M_{obs} for differing qualities of imperfect RNG (δ). Left: MBQA with $n = 10^8$ (solid lines) and $n = 10^{11}$ (dotted lines). Right: IID in the asymptotic limit ($n \to \infty$).

6.3.5 Maximum δ that can be amplified

Figure 6.9 compares the maximum amplifiable δ as a function of the observed Mermin value for the Circulant two-source extractor (the one used in our protocol) and the improved Raz extractor [1] presented in Section 3.3. For simplicity, we set the security parameter $\epsilon_{\mathrm{sec}} \leq 2^{-32} \approx 10^{-10}$ and penalty term $\Delta_f(n) = 0$, as $\Delta_f(n)$ decreases exponentially with n.

In the IID case, full amplification and privatisation is achievable with the Circulant extractor, while in the MBQA case, amplification is capped at $\delta \approx 0.3$. This is because the min-entropy rate of the outputs from the quantum device in the MBQA analysis is bounded by $2/3$ (instead of 1). Raz's extractor shows interesting plateau behaviour, which happens as one source's min-entropy rate must always exceed 0.5. The imperfect RNG satisfies this criteria until $\delta \lesssim 0.207$, at which point the quantum output must replace the imperfect RNG as the source whose min-entropy rate

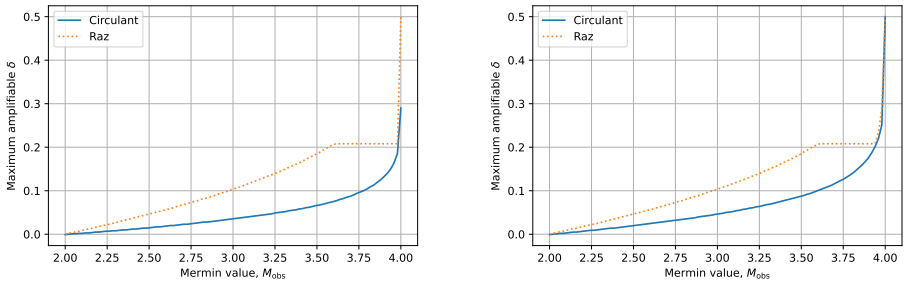

Figure 6.9: The maximum δ that can be amplified as a function of the observed Mermin value M_{obs} for Circulant (solid lines) and our improved Raz' construction [1] (dotted lines). Left: MBQA with $n = 10^{11}$. Right: IID rounds in the asymptotic limit $n \to \infty$.

exceeds 0.5 – which is achieved only when M_{obs} is close to its maximum value of 4. While the Raz extractor allows amplification of a lower maximum δ, its short output length (see Theorem 52) is the reason we focus our protocol on the Circulant extractor.

6.4 Conclusion and discussion

We have presented an end-to-end protocol for device-independent randomness amplification and privatisation, optimised for real-world quantum devices in terms of setup, parameters, randomness post-processing, and statistical analysis. The Bell inequality we use, although requiring an additional party compared to [82], provides increased noise tolerance. The randomness post-processing was specifically tailored for the task of randomness amplification (and privatisation), utilising the Circulant two-source extractor (described in Chapter 3) and the Hayashi-Tsurumaru extractor [5]. For these extractors, we introduced methods that enable quasi-linear computation time while preserving information-theoretic security, allowing the post-processing to run efficiently on a standard personal laptop – even for the large block sizes required for device-independent protocols secure against MBQA.

Our protocol achieves generation rates that are essentially linear in the number of rounds. Additionally, for randomness amplification, appending the Hayashi-Tsurumaru extractor significantly enhances the efficiency rate, which can become arbitrarily high for sufficiently high-quality imperfect RNGs. As a result, after an

initial latency, the protocol's output speed becomes linear in the speed of the imperfect RNG rather than that of the quantum device.

In the context of resource-efficient quantum cryptography, we presented a protocol for high-security randomness generation that can be implemented on near-term devices. By avoiding randomness expansion, the protocol requires only imperfect input randomness; where each bit is somewhat unpredictable and optionally public. Additionally, we derive and implement quasi-linear computation time classical post-processing, unlike existing protocols for randomness amplification (and privatisation). Following our work, a number of future research directions can be considered.

Open Problem 13. When computing the entropy accumulated by the protocol, we used the guessing probability for two outcomes for convenience. However, this likely incurs a significant penalty in the amount of certifiable entropy. It would be interesting to evaluate the rates without relying on this lower bound; for example, by solving the optimisation problem for the full three-outcome guessing probability directly, or by constructing a min-tradeoff function using the von Neumann entropy within the entropy accumulation framework.

Open Problem 14. It would be interesting to apply the generalised entropy accumulation theorem [191] to the setup for randomness amplification, as this could reduce the required assumptions. Similarly, it would be interesting to apply and compare with the quantum probability estimation framework [192], which may offer improved finite-size performance.

Open Problem 15. In this analysis, we consider using the Circulant extractor and improved Raz' extractor [1], both presented in Chapter 3. It would be interesting to perform this analysis using other extractors, for example, the asymptotically optimal (in terms of the required min-entropy of each input) two-source extractor from [59].

Open Problem 16. In our protocol, we assume that the imperfect RNG satisfies a Santha-Vazirani condition. This is a relatively strong assumption, as it requires every bit of the source to contain some randomness. It would be valuable to develop protocols that retain the practicality of the one presented here while relying

on sources less structured than Santha-Vazirani. We have results that present a practical protocol based on a *somewhere Santha-Vazirani* source, where only a subset of the output bits satisfy a Santha-Vazirani condition at unknown positions. However, exploring practical protocols that work with even weaker imperfect RNGs, such as min-entropy sources, remains an interesting direction for future work.

Open Problem 17. There are a number of assumptions required to perform the protocol, for example, those required to employ the entropy accumulation theorem or to use quantum-proof two-source extractors in the Markov model. It would be interesting to see if these assumptions could be relaxed, and what the limit of these assumptions could be whilst retaining a practical protocol; for example, by using the seedless extractors for Bell inequality violating sources presented in Chapter 4.

Chapter 7

Semi-device-independent implementation on quantum computers

In this chapter, we implement the randomness amplification and privatisation protocol from Chapter 6 on currently available ion-trap and superconducting quantum computers. We demonstrate that these devices, with additional assumptions, can achieve randomness amplification and privatisation in a semi-device-independent manner[1] – and achieve randomness generation speeds of up to megabits per second. Notably, the techniques presented in this chapter enable our protocol to be adapted for use on devices incapable of performing a loophole-free Bell test [184], thereby making the protocol more experimentally accessible.

To implement our protocol with an imperfect RNG with $\delta > 0$, a realistic number of experimental rounds and achieve high generation rates, a high Bell inequality violation relative to the algebraic maximum is required. Ideally, this would be achieved using a quantum device capable of performing a loophole-free Bell test [184]. However, such devices are challenging to construct and currently offer low randomness generation speeds and small Bell inequality violations. Quantum computers cannot perform a loophole-free Bell test due to the locality loophole, but

[1] This notion of semi-device-independence differs from the standard definition, which typically refers to a prepare-and-measure scenario.

they offer promising features, especially those based on superconducting circuits [193] and ion traps [194]. Both ion-trap and superconducting devices avoid the detection loophole [195], and ion-trap devices are noted for their minimal *crosstalk* (undesired signalling effects in which one circuit or channel affects another circuit or channel) [196].

If a quantum device is trusted but noisy and/or uncharacterised, we find that these devices can reliably perform Bell tests and thus run our protocol. Moreover, by optimising the circuit implementation for each specific device, we achieve high Bell inequality violations across all tested quantum computers. Notably, the Quantinuum and Alpine Quantum Technologies (AQT)/University of Innsbruck (UIBK) devices achieved the highest observed Mermin inequality value, $M_{\text{obs}} \approx 3.9$, while IBM's *ibmq_toronto* achieved $M_{\text{obs}} = 3.62$. All these values surpass the previously known maximum of $M_{\text{obs}} = 3.57$ reported in 2006 [197].

As a consistency check, we perform statistical testing on the randomness produced by our protocol implemented on quantum computers. From a statistical perspective, we show that our protocol successfully amplified randomness from a range of imperfect sources. Without amplification, some of these imperfect RNGs fail common statistical tests; however, after applying our protocol on quantum computers, they pass, highlighting the protocol's effectiveness in improving randomness quality.

Before presenting our results, we first analyse the performance of our protocol on an ideal device and on a device achieving the previous highest Mermin inequality violation of 3.57 from [197].

On an ideal quantum device. This scenario would involve achieving $M_{\text{obs}} = 4$, which is impossible in real-world conditions but provides valuable insight into the protocol's theoretical limits. Under these ideal conditions, amplification and privatisation could be performed with an imperfect RNG up to $\delta \to 0.5$ (IID setting) and $\delta \to 0.3$ (MBQA setting)[2], achieving protocol efficiencies of up to $\mathcal{R}_{\text{eff}} = 200\%$

[2]With our improved Raz extractor, achieving $\delta \to 0.5$ is possible in the MBQA setting (see Section 6.3.5). However, since this section focuses on achieving high efficiency rates, we limit our analysis to the protocol using the Circulant extractor, as it typically produces substantially longer output lengths.

(i.e. producing 2 output bits per use of the quantum device), depending on the number of rounds n and the value of δ. For the task of randomness amplification alone, the Hayashi-Tsurumaru extractor could be appended to achieve arbitrarily high efficiency rates $\mathcal{R}^{\mathrm{s}}_{\mathrm{eff}}$, provided the imperfect RNG is of sufficiently high quality (i.e. δ is sufficiently small).

The previous highest violation of the Mermin inequality. The highest Mermin value reported in the literature is $M_{\mathrm{obs}} = 3.57$, dating back to 2006 [197]. Using this value with our protocol, an imperfect RNG with $\delta \leq 0.096$ can be amplified. The protocol achieves an efficiency of up to $\mathcal{R}_{\mathrm{eff}} \approx 39\%$ for $\delta = 0.05$, depending on the number of rounds and whether the IID assumption is applied. For the task of randomness amplification alone, this Mermin value results in an enhanced efficiency rate of up to $\mathcal{R}^{\mathrm{s}}_{\mathrm{eff}} \approx 2.73\%$ with $\delta = 0.05$.

7.1 Implementation on quantum computers

In this section, we implement our protocol on quantum computers and present our experimental results. In order to do this, we first analyse the validity of using quantum computers for performing Bell tests.

7.1.1 Quantum computers for Bell experiments

Quantum computers are not specifically designed to run Bell tests and, in particular, permit the so-called locality loophole. In a loophole-free setup, qubits are isolated and the experiment is synchronised to prevent communication between parts of the device during a round. In contrast, quantum computers may not isolate qubits, allowing for a type of signalling known as crosstalk. This signalling, if sufficiently strong, can produce statistics that violate Bell inequalities without quantum resources [198, 199]. As signalling cannot be excluded, the following condition must be met:

> *The quantum computer must be trusted, meaning it is non-malicious and designed to faithfully attempt to implement any circuit it receives, albeit with noise or imperfections.*

If this condition is not satisfied, the user cannot guarantee that a quantum device performed the Bell test, as any input-output statistics can be replicated by a classical simulator with access to all three inputs.

Moreover, since signalling effects can increase the observed Bell inequality violation [198], the amount of signalling must be estimated and its potential impact addressed. Several methods have been developed to estimate and account for (limited) signalling effects in Bell experiments. For example, some approaches allow local measurements by each party to have a bounded effect on both systems [200], or permit limited classical communication between parties [198]. In [17], we propose a method to address specific signalling effects that occur with fixed probability or in a certain proportion of the experimental rounds – a setting that is particularly well-suited to ion-trap quantum computers (see [17, Appendix F]).

7.1.2 Implementing the Mermin Bell test

To use quantum computers for the Bell test used in our protocol (Equation (6.4)), we implemented circuits that prepare the three qubit Greenberger-Horne-Zeilinger state [201]

$$|\mathrm{GHZ}\rangle = \frac{1}{\sqrt{2}}\Big(|000\rangle + i|111\rangle\Big)\,, \tag{7.1}$$

where each qubit represents one of the parties Alice, Bob and Charlie. This state is measured using the Pauli X or Y operators on each qubit, determined by the inputs: input 0 corresponds to a Pauli X measurement, and input 1 to a Pauli Y measurement. These states and measurements enable a simple circuit implementation, allowing for high Bell inequality violations (see an example in Figure 7.1). On each device, we perform circuit and qubit optimisation to achieve the best possible results. Specifically, we convert the state preparation and measurements into the device's native gate set with a minimal number of single- and two-qubit gates and performed the experiment using the three qubits with the lowest error rates for our particular circuit (when the device offers more than three available qubits). It is important to note, however, that we do not perform full circuit optimisation. In-

stead, we optimise state preparation and measurement separately, as indicated by the barrier in Figure 7.1. This approach ensures a clear distinction between the state preparation and measurement steps. In each round of the experiment, the state preparation remains fixed while the measurement bases vary, determined by the inputs for that particular round.

Superconducting quantum computers. Superconducting quantum computers use qubits made from superconducting materials cooled to extremely low temperatures. These materials make up circuits that allow for the control of qubits and the fast manipulation of quantum information (see, e.g., [202] and [193]). These qubits are attractive because of their speed, scalability, and compatibility with current fabrication technology. For our experiments, we use the superconducting quantum computers on IBM Quantum Services. We optimise physical qubits and gate implementation using the t|ket⟩ compiler [4].

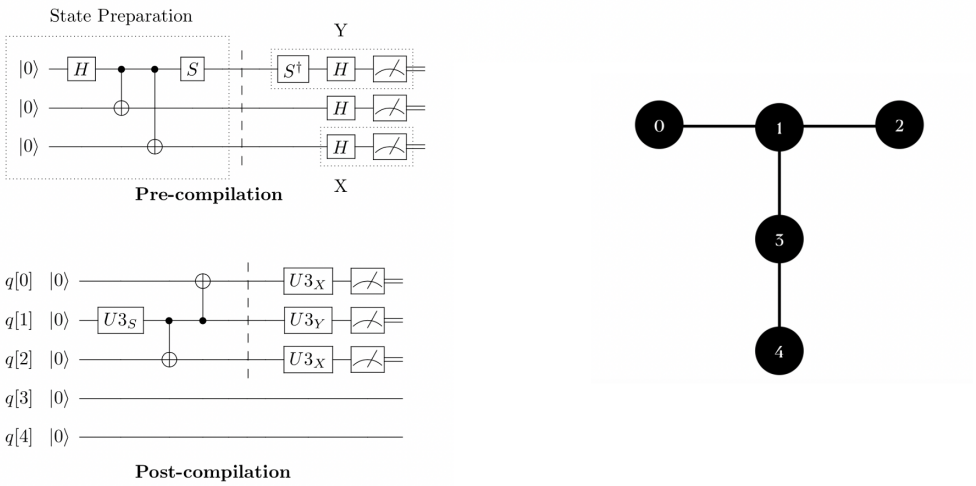

Figure 7.1: Left: One of the four circuits, representing the inputs $xyz = 011$, for the IBM quantum computers before (top) and after (bottom) compilation with t|ket⟩ [4]. State preparation (within the dashed box) is the same across all circuits, while measurements vary based on the inputs x, y, z. Right: Physical layout of qubits on IBM *ibmq_ourense* and *ibmq_valencia*.

Example circuits for *ibmq_ourense* and *ibmq_valencia* are shown in Figure 7.1 (left), before circuit optimisation (top) and after (bottom). The layouts of these devices are shown in Figure 7.1 (right). After qubit optimisation, the process of selecting qubits and qubit pairs with the lowest error rates, qubits 0, 1, and 2 (from

right of Figure 7.1) were chosen for circuit execution on both devices. At the time of execution (2020-2021), the IBM Quantum Services quantum computers operated at a fixed repetition rate of 2×10^3 circuits per second.

Ion-trap quantum computers. Ion-trap quantum computers use ions confined in electromagnetic traps as qubits, where quantum information is encoded in stable internal states of each ion (see, e.g., [194] or [203]). These qubits are attractive because of their high gate fidelity, low crosstalk and long qubit coherence times which are less prone to environmental noise. For our experiments, we used the ion-trap quantum computers from Quantinuum and AQT/UIBK. The circuit optimisation was performed manually for those from Quantinuum and for AQT/UIBK, the provider performed the optimisation. At the time of the experiment (2021), the Quantinuum devices executed roughly 13 circuits/second and the AQT/UIBK device 40 circuits/second.

7.1.3 Checking the no signalling condition

The no-signalling condition is, informally, that the input received by one party in the Bell test does not affect the output of any other party. Mathematically, for a probability distribution $\{\Pr(abc|xyz)\}_{x,y,z}^{a,b,c}$ associated to Alice (denoted A) Bob (denoted B) and Charlie (denoted C) performing a Bell test, the no-signalling condition from A to B says that

$$\sum_{a,c,z} \Pr(abc|0yz) = \sum_{a,c,z} \Pr(abc|1yz) \,, \tag{7.2}$$

for all b and y. For quantum computers, signalling effects may occur and we want a method to estimate their intensity. The no signalling condition can be modified to act as a signalling measure, defined as

$$s_{b,y}^{A\to B}(\{\Pr(abc|xyz)\}_{x,y,z}^{a,b,c}) := \left| \sum_{a,c,z} \Big(\Pr(abc|0yz) - \Pr(abx|1yz) \Big) \right| , \tag{7.3}$$

where $s_{b,y}^{A\to B} = 0$ for all b and y if there is no signalling from A to B. We call this measure the *signalling quantifier*. There exists a signalling quantifier for any pair

of parties and any of their corresponding input-output combinations. We denote the family of these by

$$s_{\gamma}^{\nu\to\nu'}(\{\Pr(abc|xyz)\}_{x,y,z}^{a,b,c}) \tag{7.4}$$

where $\gamma \in \{(a,x),(b,y),(c,z)\}$ represents any input-output tuples for combination of $a,b,c,x,y,z \in \mathbb{Z}_2$ and $\nu,\nu' \in \{A,B,C\}$ label the qubits, with $\nu \neq \nu'$. Using this measure, one can experimentally estimate the intensity of signalling effects in an experiment, for example, by computing the maximum signalling quantifier

$$\Lambda := \max_{\gamma,\nu\neq\nu'} s_{\gamma}^{\nu\to\nu'}(\{\Pr(abc|xyz)\}_{x,y,z}^{a,b,c}) \,. \tag{7.5}$$

7.1.4 Experimental results

Our experimental results for observed Mermin values, maximum amplifiable δ, and protocol efficiencies, are summarised in Table 7.1. All experiments were performed between 2020 and 2021. The number of experimental rounds for each device was: $n = 6\times10^4$ for AQT/UIBK, $n = 4\times10^4$ for Quantinuum H1, $n = 3\times10^4$ for Quantinuum H0, and $n = 10^7$ for each IBM device. Note that significantly fewer rounds were performed on the ion-trap devices compared to the superconducting devices, due to slower circuit speeds. The maximum amplifiable δ, the efficiency rate and enhanced efficiency rate are computed using $n = 10^8$, $\epsilon_{\text{sec}} \leq 2^{-32}$, $\Delta_f(n) = 0$, $\delta = 0.05$, MBQA and assuming any signalling effects are negligible. The efficiency $\mathcal{R}_{\text{eff}} = \frac{m_2}{n}$ represents the output length of the two-source extractor relative to the number of quantum device uses and the formula for m_2 is provided in Equation (6.40). The efficiency $\mathcal{R}^{\text{s}}_{\text{eff}} = \frac{m_S}{n}$ represents the output length of the seeded extractor relative to the number of quantum device uses and $m_S = (c-1)m_2$, where c is an integer related to δ computed in Section 6.2.8.

On IBM's superconducting quantum computer *ibmq_toronto*, we observed a Mermin value of $M_{\text{obs}} = 3.62$. In the MBQA setting described above, this corresponds to a maximum amplifiable RNG quality of $\delta \leq 0.073$ and protocol efficiencies for $\delta = 0.05$ of $\mathcal{R}_{\text{eff}} = 26\%$ and $\mathcal{R}^{\text{s}}_{\text{eff}} = 181\%$. As expected, on ion-trap devices,

Results from quantum computer implementations					
Device	**Type**	M_{obs}	$\max \delta$	$\mathcal{R}_{\text{eff}}$	$\mathcal{R}^{\text{s}}_{\text{eff}}$
AQT/UIBK	Ion-trap	3.9	0.120	0.71	4.98
Quantinuum H1	Ion-trap	3.88	0.114	0.66	4.64
Quantinuum H0	Ion-trap	3.83	0.103	0.56	3.91
IBM *ibmq_toronto*	Superconducting	3.62	0.073	0.26	1.81
IBM *ibmq_ourense*	Superconducting	3.35	0.052	0.02	0.17
IBM *ibmq_valencia*	Superconducting	3.11	0.038	0	0

Table 7.1: Observed Mermin values, maximum amplifiable δ (using the Circulant two-source extractor), and protocol efficiencies using $\delta = 0.05$, where $\mathcal{R}_{\text{eff}}$ is the efficiency for randomness amplification and privatisation, whilst $\mathcal{R}^{\text{s}}_{\text{eff}}$ is the enhanced efficiency for randomness amplification only, for different quantum computers.

we observed higher Mermin values than on superconducting devices. Quantinuum's H1 system gave $M_{\text{obs}} = 3.88$, allowing for $\delta \leq 0.114$ and efficiencies of $\mathcal{R}_{\text{eff}} = 66\%$ and $\mathcal{R}^{\text{s}}_{\text{eff}} = 464\%$ (with $\delta = 0.05$). Similarly, AQT/UIBK's ion-trap device achieved $M_{\text{obs}} = 3.9$, corresponding to $\delta \leq 0.120$, and, for $\delta = 0.05$, $\mathcal{R}_{\text{eff}} = 71\%$ and $\mathcal{R}^{\text{s}}_{\text{eff}} = 498\%$ with the seeded extractor. These experimental results also serve as a metric to compare the circuit fidelity of the different devices, which may be of independent interest.

Figure 7.2 shows $\mathcal{R}_{\text{eff}}$ as a function of device uses n for an imperfect RNG with SV parameter $\delta = 0.05$. Results are given for the best-performing ion-trap device ($M_{\text{obs}} = 3.9$) and superconducting device ($M_{\text{obs}} = 3.62$). For simplicity, we again set the security parameter $\epsilon_{\text{sec}} \leq 2^{-32} \approx 10^{-10}$ and $\Delta_f(n) = 0$.

Computing the maximum signalling quantifier. Among the devices used to experimentally test our protocol, all exhibited very small values for the maximum signalling quantifier Λ. The largest values were observed on *ibmq_ourense* and *ibmq_valencia*, which is expected due to the extremely low crosstalk in ion-trap devices [204]. The results for these devices, computed over 10 experiments with $n = 10^7$, are summarized in Table 7.2. Given the small values of Λ, even on these

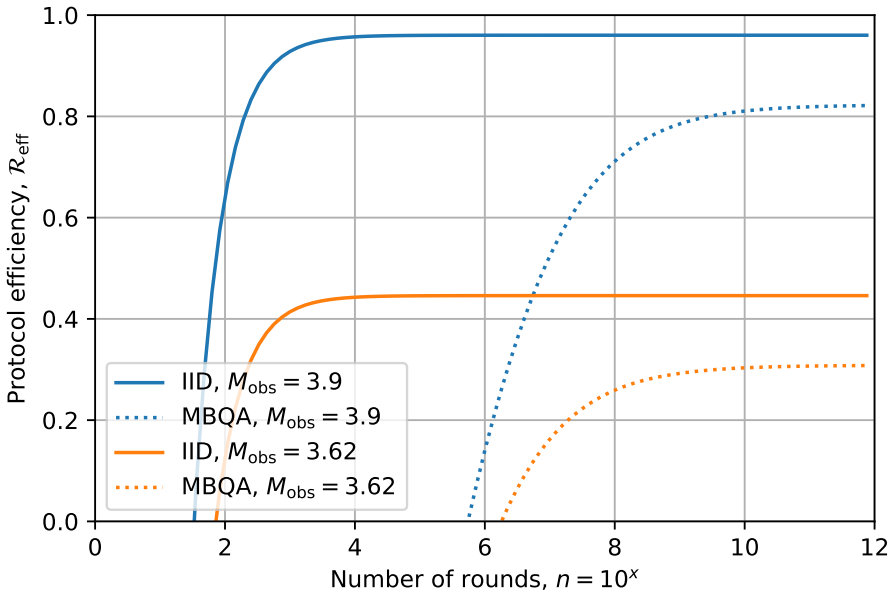

Figure 7.2: The protocol efficiency $\mathcal{R}_{\text{eff}}$ at the output of the Circulant extractor plotted against the number of uses of the quantum device, for $\delta = 0.05$, for the highest Mermin inequality violations from an ion-trap and superconducting device we observed.

devices, the impact of accounting for signalling – using, for instance, the method detailed in [17, Appendix F] – on the results in Table 7.1 is negligible and therefore ignored. Interestingly, *ibmq_ourense* displayed nearly double the signalling of *ibmq_valencia*, despite achieving a higher Mermin inequality violation (see Table 7.1).

ibmq_ourense	
Metric	Λ
Average	0.0046
Max	0.0063
Min	0.0025

ibmq_valencia	
Metric	Λ
Average	0.0026
Max	0.0042
Min	0.0014

Table 7.2: Maximum signalling quantifier, Λ, for the quantum computers *ibmq_ourense* and *ibmq_valencia* from 10 experiments with $n = 10^7$.

Next, we calculate the randomness generation speeds of our experiments using MBQA analysis with the parameters described above and $\delta = 0.05$. Assuming the imperfect RNG generates random numbers faster than the quantum computer exe-

cutes the required circuit for the protocol, the generation speed is determined by the number of circuits executed per second, multiplied by the protocol's efficiency.

Generation speeds on superconducting devices. IBM's *ibmq_toronto* achieved the highest Mermin value of all superconducting devices, $M_{\text{obs}} = 3.62$, resulting in an efficiency rate of $\mathcal{R}_{\text{eff}} = 26\%$ and an enhanced efficiency rate of $\mathcal{R}^{\text{s}}_{\text{eff}} = 181\%$. When the experiment was performed, all IBM devices had a fixed circuit repetition rate of $r = 2 \times 10^3$ circuits per second. Therefore, the resulting randomness generation speed for *ibmq_toronto* was $\mathcal{R}_{\text{eff}} \times r = 520$ bits per second for randomness amplification and privatisation, and $\mathcal{R}^{\text{s}}_{\text{eff}} \times r = 3620$ bits per second for randomness amplification alone.

We note that the generation speed would be significantly improved if the repetition rate restriction was removed. A single round of the protocol roughly amounts to performing two controlled-NOT gates, as the execution time is dominated by the two-qubit operations. On *ibmq_toronto*, this would take about 10^3 nanoseconds, implying a potential execution rate of 10^6 circuits per second, increasing the generation speed by a factor of approximately 10^3.

Generation speeds on ion-trap devices. On Quantinuum's H1 ion-trap device, operating at 13 circuits per second, the experiment results in $\mathcal{R}_{\text{eff}} = 66\%$ and $\mathcal{R}^{\text{s}}_{\text{eff}} = 464\%$ for $\delta = 0.05$. This resulted in randomness generation speeds of approximately 8.6 and 60 bits per second, respectively. Similarly, on AQT/UIBK's ion-trap device, operating at 40 circuits per second, the experiment results in $\mathcal{R}_{\text{eff}} = 71\%$ and $\mathcal{R}^{\text{s}}_{\text{eff}} = 498\%$, yielding a randomness generation speed of approximately 28 and 200 bits per second, respectively.

7.2 Statistical testing of RNG amplification

Statistical testing evaluates numerous statistical properties of a set of random numbers to determine whether there is evidence to reject the possibility that they are uniformly distributed. A test is considered failed if sufficient evidence indicates that the set of numbers can be distinguished from uniformly distributed, with a certain confidence level.

Many certification bodies, including the NIST and the Bundesamt für Sicherheit in der Informationstechnik, require statistical testing as a key consistency check for RNGs used in cryptography. Due to their importance, several statistical test *suites* have been developed, which comprise of many individual algorithms designed to perform a specific statistical test. The most well-known test suites include NIST [205] and Dieharder [206], alongside others like ENT [207], PractRand [208], and TestU01 [209].

As a consistency check, we conduct statistical testing on the randomness generated by our randomness amplification protocol implemented on quantum computers, using various different imperfect RNGs as input. Specifically, we perform the NIST test suite [205] on 5 distinct output sets, each of 100Mb. Each 100Mb of data is divided into 100 substrings of 1Mb (as per the user guidance). Statistical analysis is then performed on each substring, and the cumulative results analysed to produce pass/fail results. We note that the NIST test suite is used primarily because it requires significantly less data for testing compared to other suites.

To generate the data for statistical testing, we implement the protocol using each of four different imperfect RNGs, spanning PRNGs, CRNGs, and QRNGs, on the *ibmq_ourense* quantum computer and set $\delta = 0.05$. Specifically the different imperfect RNGs are:

- **MMIX pseudo-RNG:** A 64-bit linear congruential generator (http://mmix.cs.hm.edu/index.html).
- **32-bit LFSR:** A 32-bit linear feedback shift register.
- **In-house classical RNG:** A CRNG built at Quantinuum, based on the chaotic avalanche effect in a reverse-biased diode, similar to, for example, the design in [210].
- **ID Quantique's Quantis QRNG:** A QRNG based on the quantum phenomenon of photons being reflected or transmitted upon striking a semi-transparent mirror [211].

We note that, since the quantum computer was accessed remotely, there is no guarantee – and it is highly likely untrue – that its output remains private. As such, the

generated output is unsuitable for cryptographic applications. The results of this statistical testing, both before and after processing each imperfect RNG with our protocol, are summarised in Table 7.3.

RNG	NIST pass rate	NIST pass rate (after amplification)
MMIX	6/15	15/15
32-bit LFSR	10/15	15/15
Reverse-biased diode	15/15	15/15
Quantis-USB QRNG	15/15	15/15

Table 7.3: Proportion of NIST statistical tests passed for each RNG, averaged over 5 tests on distinct samples.

Classical amplification. We also tested a classical amplification method using the two-source extractor from Dodis et al. [74], applied directly to two imperfect RNGs. This method generates near-perfect randomness under the assumptions that the δ SV condition holds for both imperfect RNGs and that they are independent. We paired the MMIX pseudo-RNG with the 32-bit LFSR, as both failed some initial NIST tests individually but succeeded when amplified using our quantum protocol. Assuming independence and $\delta = 0.05$, we used the classical-proof Dodis et al. extractor (described in Section 3.3.2). This classical amplification method did not improve performance, with only $6/15$ NIST tests passed on average across the 5 tests. In contrast, our quantum protocol enabled both RNGs to pass all $15/15$ NIST tests, highlighting the advantage of quantum resources in randomness amplification

A more extensive statistical analysis covering various imperfect RNGs, different amplification methods and an improved approach to modelling the quality of the imperfect RNGs is presented in [173].

7.3 Conclusion and discussion

In this chapter, we demonstrated that quantum computers, with minimal additional assumptions, can effectively implement our protocol for randomness amplification and privatisation introduced in Chapter 6. We began by discussing the suitability of quantum computers for performing Bell tests, focussing on the additional assumptions required. Notably, since quantum computers do not close the locality

loophole, signalling effects may be present. We then described and analysed the necessary additional assumptions to implement our protocol on such devices.

Next, we implemented our protocol on ion-trap and superconducting quantum computers from AQT/UIBK, Quantinuum, and IBM Quantum Services. By optimising circuit designs and protocol parameters, we achieved significant Bell inequality violations across all tested quantum architectures, observing the highest values reported in the literature: $M_{\text{obs}} = 3.9$ on ion-trap devices and $M_{\text{obs}} = 3.62$ on superconducting machines. These high Bell inequality violations enabled randomness amplification from various qualities of imperfect RNGs, achieving high efficiency rates for amplification and privatisation.

Additionally, as a validation, we performed our full protocol using several different imperfect RNGs. Notably, some RNGs initially failed statistical tests but passed all tests after being processed through our protocol with $\delta = 0.05$. This result shows that randomness amplification can be successfully achieved from a statistical perspective on current hardware in a semi-device-independent setting.

In the context of resource-efficient quantum cryptography, we presented techniques and analysis that enabled our protocol from Chapter 6 to be implemented on devices not specifically designed for loophole-free Bell inequality tests. This reduces the engineering challenges of implementation, allowing our protocol for randomness amplification (and privatisation) to generate secure random numbers on existing devices today. Several promising future directions can also be explored.

Open Problem 18. Since our experiments, quantum computers have improved significantly. Implementing our protocol on newer devices and architectures better suited to Bell inequality tests, such as photon-based quantum systems, could provide even stronger results.

Open Problem 19. It would be interesting to implement different approaches to account for the signalling effects and compare their performance, for example, using the approach of [198] or [200].

Open Problem 20. Quantum computers, while powerful and readily available, are not purpose-built for device-independent randomness amplification. It would be

intriguing to develop and explore randomness amplification (and privatisation) protocols better suited to such devices, particularly those leveraging quantum computational advantage. Several interesting directions for research in this direction include:

a. Recent advancements in compiled non-local games enable Bell tests on a single device under certain cryptographic hardness assumptions. Adapting such protocols, for example [212], to randomness amplification would be valuable.
b. Investigating how techniques from device-independent randomness amplification can be translated to this setting, such as the entropy accumulation theorem [153].
c. Developing randomness amplification protocols that explicitly exploit quantum computational advantage, for example, by adapting Aaronson and Hung's protocol [213] for randomness amplification.

Appendix A

Useful primes and primes with 2 as a primitive root

Integer	Closest element in $\mathbb{P}$	Closest element in $\mathbb{N}_A$
10^1	11	11
10^2	101	101
10^3	997	1019
10^4	10007	10037
10^5	100003	100003
10^6	1000003	1000003
10^7	9999991	9999973
10^8	100000007	99999989
10^9	1000000007	1000000021
10^{10}	10000000019	10000000019
10^{11}	100000000003	100000000003
10^{12}	999999999989	999999999989

Table A.1: Closest primes and primes with 2 as a primitive root to different powers of 10.

Power of 2	Closest element in $\mathbb{P}$	Closest element in $\mathbb{N}_A$
2^2	3	3
2^3	7	5
2^4	17	13
2^5	31	29
2^6	61	61
2^7	127	131
2^8	257	269
2^9	509	509
2^{10}	1021	1019
2^{11}	2053	2053
2^{12}	4093	4093
2^{13}	8191	8179
2^{14}	16381	16381
2^{15}	32771	32771
2^{16}	65537	65539
2^{17}	131071	131059
2^{18}	262147	262147
2^{19}	524287	524269
2^{20}	1048573	1048573
2^{21}	2097143	2097133
2^{22}	4194301	4194371
2^{23}	8388617	8388619
2^{24}	16777213	16777259
2^{25}	33554467	33554467
2^{26}	67108859	67108859
2^{27}	134217757	134217773
2^{28}	268435459	268435459
2^{29}	536870909	536870909
2^{30}	1073741827	1073741827
2^{31}	2147483647	2147483659
2^{32}	4294967291	4294967291
2^{33}	8589934583	8589934621
2^{34}	17179869209	17179869107
2^{35}	34359738337	34359738421
2^{36}	68719476731	68719476731
2^{37}	137438953481	137438953427
2^{38}	274877906951	274877906957
2^{39}	549755813881	549755813933
2^{40}	1099511627791	1099511627917

Table A.2: Closest primes and primes with 2 as a primitive root to different powers of 2.

Bibliography

[1] Ran Raz. Extractors with weak random seeds. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of computing*, pages 11–20, 2005.

[2] Cameron Foreman, Richie Yeung, Alec Edgington, and Florian J Curchod. Cryptomite: A versatile and user-friendly library of randomness extractors. *Quantum*, 9:1584, 2025.

[3] Cameron Foreman, Lewis Wooltorton, Kevin Milner, and Florian J Curchod. An efficient construction of raz's two-source randomness extractor with improved parameters. *arXiv preprint arXiv:2506.15547*, 2025.

[4] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. t|ket>: A retargetable compiler for NISQ devices. *Quantum Science and Technology*, 2020.

[5] Masahito Hayashi and Toyohiro Tsurumaru. More efficient privacy amplification with less random seeds via dual universal hash function. *IEEE Transactions on Information Theory*, 62(4):2213–2232, 2016.

[6] Wolfgang Mauerer, Christopher Portmann, and Volkher B Scholz. A modular framework for randomness extraction based on Trevisan's construction. *arXiv preprint arXiv:1212.0520*, 2012.

[7] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. IEEE, 1994.

[8] Stephen Wiesner. Conjugate coding. *ACM Sigact News*, 15(1):78–88, 1983.

[9] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical computer science*, 560:7–11, 2014.

[10] Artur Ekert. Quantum cryptography based on bell's theorem. *Physical Review Letters*, 67(6):661, 1991.

[11] Antonio Acín and Lluis Masanes. Certified randomness in quantum physics. *Nature*, 540(7632):213–219, 2016.

[12] Claude Crépeau. Quantum oblivious transfer. *Journal of Modern Optics*, 41(12):2445–2454, 1994.

[13] Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 643–652, 2002.

[14] Scott Aaronson. Quantum copy-protection and quantum money. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 229–242. IEEE, 2009.

[15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low t-gate complexity. In *Annual Cryptology Conference*, pages 609–629. Springer, 2015.

[16] Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. In *Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part III 18*, pages 92–122. Springer, 2020.

[17] Cameron Foreman, Sherilyn Wright, Alec Edgington, Mario Berta, and Florian J. Curchod. Practical randomness amplification and privatisation with implementations on quantum computers. *Quantum*, 7:969, 2023.

[18] Cameron Foreman and Lluis Masanes. Seedless extractors for device-independent quantum cryptography. *Quantum*, 9:1654, 2025.

[19] Christopher Portmann and Renato Renner. Security in quantum cryptography. *Reviews of Modern Physics*, 94(2):025008, 2022.

[20] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

[21] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 474–483. IEEE, 2014.

[22] Renato Renner. Security of quantum key distribution. *International Journal of Quantum Information*, 6(01):1–127, 2008.

[23] Carl W Helstrom. Quantum detection and estimation theory. *Journal of Statistical Physics*, 1:231–252, 1969.

[24] John Von Neumann. Various techniques used in connection with random digits. *John von Neumann, Collected Works*, 5:768–770, 1963.

[25] Ronen Shaltiel. An introduction to randomness extractors. In *International Colloquium on Automata, Languages, and Programming*, pages 21–41. Springer, 2011.

[26] Miklos Santha and Umesh Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of computer and system sciences*, 33(1):75–87, 1986.

[27] Yuval Peres. Iterating von neumann's procedure for extracting random bits. *The Annals of Statistics*, pages 590–597, 1992.

[28] Peter Elias. The efficient construction of an unbiased random sequence. *The Annals of Mathematical Statistics*, 43(3):865–870, 1972.

[29] Claude Gravel. A generalization of the Von Neumann extractor. *arXiv preprint arXiv:2101.02345*, 2021.

[30] Manuel Blum. Independent unbiased coin flips from a correlated biased source—a finite state markov chain. *Combinatorica*, 6:97–108, 1986.

[31] Benny Chor, Oded Goldreich, Johan Hasted, Joel Freidmann, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 396–407. IEEE, 1985.

[32] Yevgeniy Dodis. New imperfect random source with applications to coin-flipping. In *International Colloquium on Automata, Languages, and Programming*, pages 297–309. Springer, 2001.

[33] Ariel Gabizon, Ran Raz, and Ronen Shaltiel. Deterministic extractors for bit-fixing sources by obtaining an independent seed. *SIAM Journal on Computing*, 36(4):1072–1094, 2006.

[34] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2007.

[35] Gil Cohen and Igor Shinkar. Zero-fixing extractors for sub-logarithmic entropy. In *Automata, Languages, and Programming: 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I 42*, pages 343–354. Springer, 2015.

[36] Luca Trevisan and Salil Vadhan. Extracting randomness from samplable distributions. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 32–42. IEEE, 2000.

[37] Anindya De and Thomas Watson. Extractors and lower bounds for locally samplable sources. *ACM Transactions on Computation Theory (TOCT)*, 4(1):1–21, 2012.

[38] Jesse Kamp, Anup Rao, Salil Vadhan, and David Zuckerman. Deterministic extractors for small-space sources. *Journal of Computer and System Sciences*, 77(1):191–220, 2011.

[39] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. *Combinatorica*, 28(4):415–440, 2008.

[40] Jean Bourgain. On the construction of affine extractors. *GAFA Geometric And Functional Analysis*, 17(1):33–57, 2007.

[41] Anup Rao. Extractors for low-weight affine sources. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 95–101. IEEE, 2009.

[42] Matt DeVos and Ariel Gabizon. Simple affine extractors using dimension expansion. In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 50–57. IEEE, 2010.

[43] Amir Yehudayoff. Affine extractors over prime fields. *Combinatorica*, 31(2):245–256, 2011.

[44] Xin Li. A new approach to affine extractors and dispersers. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 137–147. IEEE, 2011.

[45] Jean Bourgain, Zeev Dvir, and Ethan Leeman. Affine extractors over large fields with exponential error. *computational complexity*, 25:921–931, 2016.

[46] Zeev Dvir, Ariel Gabizon, and Avi Wigderson. Extractors and rank extractors for polynomial sources. *Computational Complexity*, 18:1–58, 2009.

[47] Emanuele Viola. Extractors for turing-machine sources. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 663–671. Springer, 2012.

[48] Emanuele Viola. Extractors for circuit sources. *SIAM Journal on Computing*, 43(2):655–672, 2014.

[49] Salman Beigi, Omid Etesami, and Amin Gohari. Deterministic randomness extraction from generalized and distributed santha–vazirani sources. *SIAM Journal on Computing*, 46(1):1–36, 2017.

[50] Salman Beigi, Andrej Bogdanov, Omid Etesami, and Siyao Guo. Optimal deterministic extractors for generalized santha-vazirani sources. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[51] Ran Raz and Amir Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. *Journal of Computer and System Sciences*, 77(1):167–190, 2011.

[52] Eshan Chattopadhyay and Xin Li. Extractors for sumset sources. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 299–311, 2016.

[53] Eshan Chattopadhyay and David Zuckerman. New extractors for interleaved sources. In *31st Conference on Computational Complexity (CCC 2016)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2016.

[54] Eshan Chattopadhyay and Jyun-Jie Liao. Extractors for sum of two sources. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1584–1597, 2022.

[55] Omer Reingold, Salil Vadhan, and Avi Wigderson. A note on extracting randomness from santha-vazirani sources. *Unpublished manuscript*, 3, 2004.

[56] Divesh Aggarwal, Maciej Obremski, Joao Ribeiro, Luisa Siniscalchi, and Ivan Visconti. How to extract useful randomness from unreliable sources. In *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International*

Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part I 39, pages 343–372. Springer, 2020.

[57] Iddo Bentov, Ariel Gabizon, and David Zuckerman. Bitcoin beacon. *arXiv preprint arXiv:1605.04559*, 2016.

[58] Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, 2001.

[59] Xin Li. Two source extractors for asymptotically optimal entropy, and (many) more. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1271–1281. IEEE, 2023.

[60] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald De Wolf. Exponential separations for one-way quantum communication complexity, with applications to cryptography. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pages 516–525, 2007.

[61] Marshall Ball, Oded Goldreich, and Tal Malkin. Randomness extraction from somewhat dependent sources. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[62] Rotem Arnon-Friedman, Christopher Portmann, and Volkher B. Scholz. Quantum-Proof Multi-Source Randomness Extractors in the Markov Model. In *11th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2016)*, volume 61 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:34, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[63] Yevgeniy Dodis, Vinod Vaikuntanathan, and Daniel Wichs. Extracting randomness from extractor-dependent sources. In *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory*

and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part I 39, pages 313–342. Springer, 2020.

[64] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the forty-first annual ACM Symposium on Theory of Computing*, pages 601–610, 2009.

[65] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[66] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[67] Sergei Gorlatch. Programming with divide-and-conquer skeletons: A case study of fft. *The Journal of Supercomputing*, 12:85–97, 1998.

[68] Gilles Van Assche. *Quantum cryptography and secret-key distillation*. Cambridge University Press, 2006.

[69] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[70] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[71] Chi-Jen Lu, Omer Reingold, Salil Vadhan, and Avi Wigderson. Extractors: Optimal up to constant factors. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 602–611, 2003.

[72] Anindya De, Christopher Portmann, Thomas Vidick, and Renato Renner. Trevisan's extractor in the presence of quantum side information. *SIAM Journal on Computing*, 41(4):915–940, 2012.

[73] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *Journal of the ACM (JACM)*, 56(4):1–34, 2009.

[74] Yevgeniy Dodis, Ariel Elbaz, Roberto Oliveira, and Ran Raz. Improved randomness extraction from two independent sources. *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 334–344, 2004.

[75] Hugo Krawczyk. LFSR-based hashing and authentication. In *Annual International Cryptology Conference*, pages 129–139. Springer, 1994.

[76] Mario Berta and Fernando Brandao. Robust randomness generation on quantum computers. https://marioberta.info/wp-content/uploads/2021/07/randomness-theory.pdf, 2021.

[77] Marco Tomamichel, Christian Schaffner, Adam Smith, and Renato Renner. Leftover hashing against quantum side information. *IEEE Transactions on Information Theory*, 57(8):5524–5535, 2011.

[78] Irwin Kra and Santiago R Simanca. On circulant matrices. *Notices of the AMS*, 59(3):368–377, 2012.

[79] Tzvika Hartman and Ran Raz. On the distribution of the number of roots of polynomials and explicit weak designs. *Random Structures & Algorithms*, 23(3):235–263, 2003.

[80] Rodrigo Gallego, Lluis Masanes, Gonzalo De La Torre, Chirag Dhara, Leandro Aolita, and Antonio Acín. Full randomness from arbitrarily deterministic events. *Nature Communications*, 4(1):2654, 2013.

[81] Fernando G. S. L. Brandão, Ravishankar Ramanathan, Andrzej Grudka, Karol Horodecki, Michał Horodecki, Paweł Horodecki, Tomasz Szarek, and Hanna Wojewódka. Realistic noise-tolerant randomness amplification using finite number of devices. *Nature communications*, 7(1):11345, 2016.

[82] Max Kessler and Rotem Arnon-Friedman. Device-independent randomness amplification and privatization. *IEEE Journal on Selected Areas in Information Theory*, 1(2):568–584, 2020.

[83] Umesh Vazirani. Efficiency considerations in using semi-random sources. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 160–168, 1987.

[84] Yevgeniy Dodis and Roberto Oliveira. On extracting private randomness over a public channel. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 252–263. Springer, 2003.

[85] FRK Chung. Open problems of paul erdos in graph theory. *Journal of graph theory*, 25(1):3–36, 1997.

[86] Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating independence: New constructions of condensers, ramsey graphs, dispersers, and extractors. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 1–10, 2005.

[87] Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. *SIAM Journal on Computing*, 36(4):1095–1118, 2006.

[88] Jean Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 1(01):1–32, 2005.

[89] Anup Rao. Extractors for a constant number of polynomially small min-entropy independent sources. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 497–506, 2006.

[90] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2-source dispersers for n o (1) entropy, and ramsey graphs beating the frankl-wilson construction. *Annals of Mathematics*, pages 1483–1543, 2012.

[91] Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 688–697. IEEE, 2012.

[92] Xin Li. New independent source extractors with exponential improvement. In *Proceedings of the forty-fifth annual ACM Symposium on Theory of Computing*, pages 783–792, 2013.

[93] Xin Li. Extractors for a constant number of independent sources with polylogarithmic min-entropy. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 100–109. IEEE, 2013.

[94] Gil Cohen. Two-source dispersers for polylogarithmic entropy and improved ramsey graphs. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 278–284, 2016.

[95] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 670–683, 2016.

[96] Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 168–177. IEEE, 2016.

[97] Gil Cohen and Leonard J Schulman. Extractors for near logarithmic min-entropy. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 178–187. IEEE, 2016.

[98] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors, and almost optimal privacy amplification protocols. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 158–167. IEEE, 2016.

[99] Gil Cohen. Making the most of advice: New correlation breakers and their

applications. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 188–196. IEEE, 2016.

[100] Gil Cohen. Towards optimal two-source extractors and ramsey graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1157–1170, 2017.

[101] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1144–1156, 2017.

[102] Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. *arXiv preprint arXiv:1804.04005*, 2018.

[103] Mark Lewko. An explicit two-source extractor with min-entropy rate near 4/9. *Mathematika*, 65(4):950–957, 2019.

[104] Roy Kasher and Julia Kempe. Two-source extractors secure against quantum adversaries. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 656–669. Springer, 2010.

[105] Kai-Min Chung, Xin Li, and Xiaodi Wu. Multi-source randomness extractors against quantum side information, and their applications. *arXiv preprint arXiv:1411.2315*, 2014.

[106] Patrick Hayden, Richard Jozsa, Denes Petz, and Andreas Winter. Structure of states which satisfy strong subadditivity of quantum entropy with equality. *Communications in mathematical physics*, 246:359–374, 2004.

[107] Umesh Vazirani. Efficiency considerations in using semi-random sources. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 160–168, New York, NY, USA, 1987. Association for Computing Machinery.

[108] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993.

[109] N. Alon, O. Goldreich, J. Hastad, and R. Peralta. Simple construction of almost k-wise independent random variables. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 544–553 vol.2, 1990.

[110] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7(4):567–583, 1986.

[111] Raghu Meka, Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Fast pseudorandomness for independence and load balancing. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming*, pages 859–870, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[112] Daniel Lemire, Owen Kaser, and Nathan Kurz. Faster remainder by direct computation: Applications to compilers and software libraries. *Software: Practice and Experience*, 49(6):953–970, 2019.

[113] Tolga Acar and Dan Shumow. Modular reduction without pre-computation for special moduli. *Microsoft Research, Redmond, WA, USA*, 2, 2010.

[114] Tanvirul Islam. Toeplitz extractor. https://github.com/tanvirulz/toeplitz_extractor, 2018.

[115] Mayank Kharbanda. Randomness extractors. https://github.com/MayankKharbanda/randomness_extractors, 2020.

[116] BYUCamachoLab. ottoeplitz. https://github.com/BYUCamachoLab/ottoeplitz, 2022.

[117] Rok Zitko. Toeplitz. https://github.com/rokzitko/toeplitz, 2022.

[118] Wolfgang Mauerer. libtrevisan. https://github.com/wolfgangmauerer/libtrevisan, 2014.

[119] Michele Mancusi. libtrevisan. https://github.com/michelemancusi/libtrevisan, 2019.

[120] Mario Berta and Fernando Brandao. Randomness generation. https://github.com/aws/amazon-braket-examples/blob/main/examples/advanced_circuits_algorithms/Randomness/Randomness_Generation.ipynb, 2021.

[121] Valerio Scarani, Helle Bechmann-Pasquinucci, Nicolas J Cerf, Miloslav Dušek, Norbert Lütkenhaus, and Momtchil Peev. The security of practical quantum key distribution. *Reviews of modern physics*, 81(3):1301–1350, 2009.

[122] Roger Colbeck and Adrian Kent. Private randomness expansion with untrusted devices. *Journal of Physics A: Mathematical and Theoretical*, 44(9):095305, 2011.

[123] Peter J Brown, Sammy Ragy, and Roger Colbeck. A framework for quantum-secure device-independent randomness expansion. *IEEE Transactions on Information Theory*, 66(5):2964–2987, 2019.

[124] Nitin Jain, Hou-Man Chin, Hossein Mani, Cosmo Lupo, Dino Solar Nikolic, Arne Kordts, Stefano Pirandola, Thomas Brochmann Pedersen, Matthias Kolb, Bernhard Ömer, Christoph Pacher, Tobias Gehring, and Ulrik L. Andersen. Practical continuous-variable quantum key distribution with composable security. *Nature Communications*, 13(1):1–8, 2022.

[125] Marco Avesani, Hamid Tebyanian, Paolo Villoresi, and Giuseppe Vallone. Semi-device-independent heterodyne-based quantum random-number generator. *Physical Review Applied*, 15(3):034034, 2021.

[126] Christian Gabriel, Christoffer Wittmann, Denis Sych, Ruifang Dong, Wolfgang Mauerer, Ulrik L Andersen, Christoph Marquardt, and Gerd Leuchs. A generator for unique quantum random numbers based on vacuum states. *Nature Photonics*, 4(10):711–715, 2010.

[127] Johannes Thewes, Carolin Lüders, and Marc Aßmann. Eavesdropping attack on a trusted continuous-variable quantum random-number generator. *Physical Review A*, 100(5):052318, 2019.

[128] John F Clauser, Michael A Horne, Abner Shimony, and Richard A Holt. Proposed experiment to test local hidden-variable theories. *Physical review letters*, 23(15):880, 1969.

[129] Yukun Wang, Xingyao Wu, and Valerio Scarani. All the self-testings of the singlet for two binary measurements. *New Journal of Physics*, 18(2):025021, 2016.

[130] Jesse Kamp, Anup Rao, Salil Vadhan, and David Zuckerman. Deterministic extractors for small-space sources. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 691–700, 2006.

[131] Harry Buhrman, Richard Cleve, Serge Massar, and Ronald De Wolf. Nonlocality and communication complexity. *Reviews of modern physics*, 82(1):665, 2010.

[132] Lluis Masanes. Universally composable privacy amplification from causality constraints. *Physical Review Letters*, 102(14):140501, 2009.

[133] Lewis Wooltorton, Peter Brown, and Roger Colbeck. Device-independent quantum key distribution with arbitrarily small nonlocality. *Physical Review Letters*, 132(21):210802, 2024.

[134] Máté Farkas. Unbounded device-independent quantum key rates from arbitrarily small nonlocality. *Physical Review Letters*, 132(21):210803, 2024.

[135] Ravishankar Ramanathan, Fernando G. S. L. Brandão, Karol Horodecki, Michał Horodecki, Paweł Horodecki, and Hanna Wojewódka. Randomness amplification under minimal fundamental assumptions on the devices. *Physical review letters*, 117(23):230501, 2016.

[136] Ravishankar Ramanathan, Michał Horodecki, Hammad Anwer, Stefano Pironio, Karol Horodecki, Marcus Grünfeld, Sadiq Muhammad, Mohamed Bourennane, and Paweł Horodecki. Practical no-signalling proof randomness amplification using hardy paradoxes and its experimental implementation. *arXiv preprint arXiv:1810.11648*, 2018.

[137] Ravishankar Ramanathan, Michał Banacki, and Paweł Horodecki. No-signaling-proof randomness extraction from public weak sources. *arXiv preprint arXiv:2108.08819*, 2021.

[138] Lluis Masanes, Stefano Pironio, and Antonio Acín. Secure device-independent quantum key distribution with causally independent measurement devices. *Nature Communications*, 2(1):1–7, 2011.

[139] I Gelfand and M Neumark. On the imbedding of normed rings into the ring of operators in hilbert space. *Contemporary Mathematics*, 167:3–3, 1994.

[140] Devashish Tupkary, Ernest Y-Z Tan, and Norbert Lütkenhaus. Security proof for variable-length quantum key distribution. *Physical Review Research*, 6(2):023002, 2024.

[141] Rotem Arnon-Friedman, Frédéric Dupuis, Omar Fawzi, Renato Renner, and Thomas Vidick. Practical device-independent quantum cryptography via entropy accumulation. *Nature Communications*, 9(1):459, 2018.

[142] Florence J. MacWilliams and Neil J. A. Sloane. The theory of error-correcting codes. *Elsevier Science Publishers BV*, 2:39–47, 1977.

[143] Tadao Kasami, Toru Fujiwara, and Shu Lin. An approximation to the weight

distribution of binary linear codes. *IEEE transactions on information theory*, 31(6):769–780, 1985.

[144] Yoshihisa Desaki, Toru Fujiwara, and Tadao Kasami. The weight distributions of extended binary primitive bch codes of length 128. *IEEE Transactions on Information Theory*, 43(4):1364–1371, 1997.

[145] Anne Canteaut and Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense bch codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.

[146] Amit Berman, Yaron Shany, and Itzhak Tamo. Efficient algorithms for constructing minimum-weight codewords in some extended binary bch codes. *IEEE Transactions on Information Theory*, 2024.

[147] Stefano Pironio, Antonio Acín, Serge Massar, A Boyer de La Giroday, Dzmitry N Matsukevich, Peter Maunz, Steven Olmschenk, David Hayes, Le Luo, T Andrew Manning, et al. Random numbers certified by bell's theorem. *Nature*, 464(7291):1021–1024, 2010.

[148] Richard Arratia and Louis Gordon. Tutorial on large deviations for the binomial distribution. *Bulletin of mathematical biology*, 51(1):125–131, 1989.

[149] Rutvij Bhavsar, Sammy Ragy, and Roger Colbeck. Improved device-independent randomness expansion rates using two sided randomness. *New Journal of Physics*, 25(9):093035, 2023.

[150] Le Phuc Thinh, Lana Sheridan, and Valerio Scarani. Bell tests with min-entropy sources. *Physical Review A—Atomic, Molecular, and Optical Physics*, 87(6):062121, 2013.

[151] Miguel Navascués, Stefano Pironio, and Antonio Acín. A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations. *New Journal of Physics*, 10(7):073013, 2008.

[152] Lluis Masanes, Renato Renner, Matthias Christandl, Andreas Winter, and Jonathan Barrett. Full security of quantum key distribution from no-signaling constraints. *IEEE Transactions on Information Theory*, 60(8):4973–4986, 2014.

[153] Frederic Dupuis, Omar Fawzi, and Renato Renner. Entropy accumulation. *Communications in Mathematical Physics*, 379(3):867–913, 2020.

[154] Mario Stipčević and Çetin Kaya Koç. True random number generators. In *Open Problems in Mathematics and Computational Science*, pages 275–315. Springer, 2014.

[155] Mario Stipčević. Quantum random number generators and their applications in cryptography. In *Advanced Photon Counting Techniques VI*, volume 8375, page 837504. International Society for Optics and Photonics, 2012.

[156] Miguel Herrero-Collantes and Juan Carlos Garcia-Escartin. Quantum random number generators. *Reviews of Modern Physics*, 89(1):015004, 2017.

[157] Roger Colbeck and Renato Renner. Free randomness can be amplified. *Nature Physics*, 8(6):450–453, 2012.

[158] Hanna Wojewódka, Fernando GSL Brandão, Andrzej Grudka, Karol Horodecki, Michał Horodecki, Paweł Horodecki, Marcin Pawłowski, Ravishankar Ramanathan, and Maciej Stankiewicz. Amplifying the randomness of weak sources correlated with devices. *IEEE Transactions on Information Theory*, 63(11):7592–7611, 2017.

[159] Kai-Min Chung, Yaoyun Shi, and Xiaodi Wu. General randomness amplification with non-signaling security. *Available: https://ix.cs.uoregon.edu/xiaodiwu/papers/csw16.pdf*, 2016.

[160] Kai-Min Chung, Yaoyun Shi, and Xiaodi Wu. Physical randomness extractors: generating random numbers with minimal assumptions. *arXiv preprint arXiv:1402.4797*, 2014.

[161] Ravishankar Ramanathan, Fernando G. S. L. Brandão, Karol Horodecki, Michał Horodecki, Paweł Horodecki, and Hanna Wojewódka. Randomness amplification under minimal fundamental assumptions on the devices. *Phys. Rev. Lett.*, 117:230501, Nov 2016.

[162] Ravishankar Ramanathan, Michał Horodecki, Stefano Pironio, Karol Horodecki, and Paweł Horodecki. Generic randomness amplification schemes using hardy paradoxes. *arXiv preprint arXiv:1810.11648*, 2018.

[163] Stephen Checkoway, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz, Hovav Shacham, and Matthew Fredrikson. On the practical exploitability of dual {EC} in {TLS} implementations. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 319–335, 2014.

[164] Daniel J. Bernstein, Tanja Lange, and Ruben Niederhagen. Dual ec: A standardized back door. In *The New Codebreakers*, pages 256–281. Springer, 2016.

[165] Stephen Checkoway, Jacob Maskiewicz, Christina Garman, Joshua Fried, Shaanan Cohney, Matthew Green, Nadia Heninger, Ralf-Philipp Weinmann, Eric Rescorla, and Hovav Shacham. A systematic analysis of the juniper dual ec incident. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 468–479, 2016.

[166] Yevgeniy Dodis, David Pointcheval, Sylvain Ruhault, Damien Vergniaud, and Daniel Wichs. Security analysis of pseudo-random number generators with input: /dev/random is not robust. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 647–658, 2013.

[167] Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. Mining your ps and qs: Detection of widespread weak keys in network

devices. In *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*, pages 205–220, 2012.

[168] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Cryptanalytic attacks on pseudorandom number generators. In *International workshop on fast software encryption*, pages 168–188. Springer, 1998.

[169] YongBin Zhou and DengGuo Feng. Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. *IACR Cryptol. ePrint Arch.*, 2005:388, 2005.

[170] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults. In *International conference on the theory and applications of cryptographic techniques*, pages 37–51. Springer, 1997.

[171] Denis Rosset, Raphael Ferretti-Schöbitz, Jean-Daniel Bancal, Nicolas Gisin, and Yeong-Cherng Liang. Imperfect measurement settings: Implications for quantum state tomography and entanglement witnesses. *Physical Review A*, 86(6):062325, 2012.

[172] Darren Hurley-Smith and Julio Hernandez-Castro. Quantum leap and crash: Searching and finding bias in quantum random number generators. *ACM Transactions on Privacy and Security (TOPS)*, 23(3):1–25, 2020.

[173] Cameron Foreman, Richie Yeung, and Florian J Curchod. Statistical testing of random number generators and their improvement using randomness extraction. *arXiv preprint arXiv:2403.18716*, 2024.

[174] Burak Acar and Salih Ergün. A robust digital random number generator based on transient effect of ring oscillator. In *2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*, pages 1–4. IEEE, 2020.

[175] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im) possibility of cryptography with imperfect randomness. In *45th*

Annual IEEE Symposium on Foundations of Computer Science, pages 196–205. IEEE, 2004.

[176] Carl Bosley and Yevgeniy Dodis. Does privacy require true randomness? In *Theory of Cryptography Conference*, pages 1–20. Springer, 2007.

[177] James L. McInnes and Benny Pinkas. On the impossibility of private key cryptography with weakly random keys. In *Conference on the Theory and Application of Cryptography*, pages 421–435. Springer, 1990.

[178] Yevgeniy Dodis and Joel Spencer. On the (non) universality of the one-time pad. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 376–385. IEEE, 2002.

[179] Yevgeniy Dodis, Adriana López-Alt, Ilya Mironov, and Salil Vadhan. Differential privacy with imperfect randomness. In *Annual Cryptology Conference*, pages 497–516. Springer, 2012.

[180] Ravishankar Ramanathan. Finite device-independent extraction of a block min-entropy source against quantum adversaries. *arXiv preprint arXiv:2304.09643*, 2023.

[181] Esther Hänggi, Renato Renner, and Stefan Wolf. The impossibility of non-signaling privacy amplification. *Theoretical Computer Science*, 486:27–42, 2013.

[182] N David Mermin. Extreme quantum entanglement in a superposition of macroscopically distinct states. *Physical Review Letters*, 65(15):1838, 1990.

[183] Jonathan Barrett, Roger Colbeck, and Adrian Kent. Memory attacks on device-independent quantum cryptography. *Physical review letters*, 110(1):010503, 2013.

[184] Nicolas Brunner, Daniel Cavalcanti, Stefano Pironio, Valerio Scarani, and Stephanie Wehner. Bell nonlocality. *Reviews of Modern Physics*, 86(2):419, 2014.

[185] Adán Cabello, Álvaro Feito, and Antia Lamas-Linares. Bell's inequalities with realistic noise for polarization-entangled photons. *Physical Review A*, 72(5):052112, 2005.

[186] Gilles Pütz and Nicolas Gisin. Measurement dependent locality. *New Journal of Physics*, 18(5):055006, 2016.

[187] Erik Woodhead, Boris Bourdoncle, and Antonio Acín. Randomness versus nonlocality in the mermin-bell experiment with three parties. *Quantum*, 2:82, 2018.

[188] Rotem Arnon-Friedman, Renato Renner, and Thomas Vidick. Simple and tight device-independent security proofs. *SIAM Journal on Computing*, 48(1):181–225, 2019.

[189] Rotem Arnon-Friedman. Reductions to iid in device-independent quantum information processing. *arXiv preprint arXiv:1812.10922*, 2018.

[190] Daniela Frauchiger, Renato Renner, and Matthias Troyer. True randomness from realistic quantum devices. *arXiv preprint arXiv:1311.4547*, 2013.

[191] Tony Metger, Omar Fawzi, David Sutter, and Renato Renner. Generalised entropy accumulation. *Communications in Mathematical Physics*, 405(11):261, 2024.

[192] Yanbao Zhang, Honghao Fu, and Emanuel Knill. Efficient randomness certification by quantum probability estimation. *Physical review research*, 2(1):013016, 2020.

[193] John Clarke and Frank K Wilhelm. Superconducting quantum bits. *Nature*, 453(7198):1031–1042, 2008.

[194] Colin D Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2):021314, 2019.

[195] Markus Ansmann, H Wang, Radoslaw C Bialczak, Max Hofheinz, Erik Lucero, Matthew Neeley, Aaron D O'Connell, Daniel Sank, Martin Weides, James Wenner, et al. Violation of bell's inequality in josephson phase qubits. *Nature*, 461(7263):504–506, 2009.

[196] Juan M Pino, Jennifer M Dreiling, Caroline Figgatt, John P Gaebler, Steven A Moses, MS Allman, CH Baldwin, M Foss-Feig, D Hayes, K Mayer, et al. Demonstration of the trapped-ion quantum ccd computer architecture. *Nature*, 592(7853):209–213, 2021.

[197] Yu-Ao Chen, Tao Yang, An-Ning Zhang, Zhi Zhao, Adán Cabello, and Jian-Wei Pan. Experimental violation of bell's inequality beyond tsirelson's bound. *Physical review letters*, 97(17):170408, 2006.

[198] Dave Bacon and Benjamin F Toner. Bell inequalities with auxiliary communication. *Physical review letters*, 90(15):157904, 2003.

[199] Jonatan Bohr Brask and Rafael Chaves. Bell scenarios with communication. *Journal of Physics A: Mathematical and Theoretical*, 50(9):094001, 2017.

[200] Jonathan Silman, Stefano Pironio, and Serge Massar. Device-independent randomness generation in the presence of weak cross-talk. *Physical review letters*, 110(10):100504, 2013.

[201] Daniel M Greenberger, Michael A Horne, and Anton Zeilinger. Going beyond bell's theorem. In *Bell's theorem, quantum theory and conceptions of the universe*, pages 69–72. Springer, 1989.

[202] Michel H Devoret and Robert J Schoelkopf. Superconducting circuits for quantum information: an outlook. *Science*, 339(6124):1169–1174, 2013.

[203] Christopher Monroe and Jungsang Kim. Scaling the ion trap quantum processor. *Science*, 339(6124):1164–1169, 2013.

[204] Pedro Parrado-Rodríguez, Ciarán Ryan-Anderson, Alejandro Bermudez, and Markus Müller. Crosstalk suppression for fault-tolerant quantum error correction with trapped ions. *Quantum*, 5:487, 2021.

[205] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, Booz-allen and hamilton inc mclean va, 2001.

[206] Robert G Brown, Dirk Eddelbuettel, and David Bauer. Dieharder. *Duke University Physics Department Durham, NC*, pages 27708–0305, 2018.

[207] J Walker. *A Pseudorandom Number Sequence Test Program*.

[208] Chris Doty-Humphrey. practrand. webpage, 2022.

[209] Pierre L'ecuyer and Richard Simard. TestU01: AC library for empirical testing of random number generators. *ACM Transactions on Mathematical Software (TOMS)*, 33(4):1–40, 2007.

[210] Gabriel Guerrer. Rava: An open hardware true random number generator based on avalanche noise. *IEEE Access*, 11:119568–119583, 2023.

[211] ID Quantique. Quantis: Quantum random number generator, 2004.

[212] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. Quantum advantage from any non-local game. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1617–1628, 2023.

[213] Scott Aaronson and Shih-Han Hung. Certified randomness from quantum supremacy. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 933–944, 2023.