Provably Secure Key Establishment Against Quantum Adversaries*

Aleksandrs Belovs[†1], Gilles Brassard[‡2], Peter Høyer[§3], Marc Kaplan[¶4], Sophie Laplante[‖5], and Louis Salvail[6]**

1 **University of Latvia, Riga, Latvia**
stiboh@gmail.com
2 **DIRO, Université de Montréal, Montréal, Canada and Canadian Institute for Advanced Research, Toronto, Canada**
brassard@iro.umontreal.ca
3 **Department of Computer Science, University of Calgary, Calgary, Canada**
hoyer@ucalgary.ca
4 **School of Informatics, University of Edinburgh, Edinburgh, Great Britain**
kapmarc@gmail.com
5 **IRIF, Université Paris Diderot, Paris, France**
laplante@irif.fr
6 **DIRO, Université de Montréal, Montréal, Canada**
salvail@iro.umontreal.ca

Abstract

At Crypto 2011, some of us had proposed a family of cryptographic protocols for key establishment capable of protecting quantum *and classical* legitimate parties unconditionally against a *quantum* eavesdropper in the query complexity model. Unfortunately, our security proofs were unsatisfactory from a cryptographically meaningful perspective because they were sound only in a worst-case scenario. Here, we extend our results and prove that for any $\varepsilon > 0$, there is a classical protocol that allows the legitimate parties to establish a common key after $O(N)$ expected queries to a random oracle, yet any quantum eavesdropper will have a vanishing probability of learning their key after $O(N^{1.5-\varepsilon})$ queries to the same oracle. The vanishing probability applies to a typical run of the protocol. If we allow the legitimate parties to use a quantum computer as well, their advantage over the quantum eavesdropper becomes arbitrarily close to the quadratic advantage that classical legitimate parties enjoyed over classical eavesdroppers in the seminal 1974 work of Ralph Merkle. Along the way, we develop new tools to give lower bounds on the number of quantum queries required to distinguish two probability distributions. This method in itself could have multiple applications in cryptography. We use it here to study average-case quantum query complexity, for which we develop a new composition theorem of independent interest.

* A full version of the paper is available at [6], https://arxiv.org/abs/1704.08182.

† The work of AB is supported in part by the ERC Advanced Grant MQC.

‡ The work of GB is supported in part by the Canadian Institute for Advanced Research (CIFAR), the Canada Research Chair program, Canada's Natural Sciences and Engineering Research Council (NSERC) and Québec's Institut transdisciplinaire d'information quantique.

§ The work of PH is supported in part by CIFAR and NSERC.

¶ The work of MK is supported in part by EPSRC grant number EP1N003829/1 Verification of Quantum Technology.

‖ The work of SL is supported in part by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 600700 (QALGO) and the French ANR Blanc grant RDAM ANR-12-BS02-005.

** The work of LS is supported in part by NSERC discovery grant and discovery accelerator supplements programs.

12th Conference on the Theory of Quantum Computation, Communication, and Cryptography (TQC 2017).
Editor: Mark M. Wilde; Article No. 3; pp. 3:1–3:17

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2012 ACM Subject Classification Theory of computation → Quantum complexity theory, Theory of computation → Cryptographic protocols, Security and privacy → Key management, Security and privacy → Mathematical foundations of cryptography

Keywords and phrases Merkle puzzles, Key establishment schemes, Quantum cryptography, Adversary method, Average-case analysis

Digital Object Identifier 10.4230/LIPIcs.TQC.2017.3

1 Introduction

Not taking classified work within secret services into consideration [28], Ralph Merkle is the first person to have asked – and solved – the question of secure communications over insecure channels [24]. In his seminal (rejected!) 1974 project for a Computer Security course at the University of California, Berkeley, he discovered that it is possible for two people who want to communicate securely to establish a secret key by communicating over an authenticated channel that provides no protection against eavesdropping. Merkle's solution to this conundrum offers *quadratic security* in the sense that if the legitimate parties – codenamed Alice and Bob – are willing to expend an effort in the order of N, for some security parameter N, they can establish a key that no eavesdropper – codenamed Eve – can discover with better than vanishing probability without expending an effort in the order of N^2.

This quadratic security may seem unattractive compared to the potential exponential security entailed by the subsequently discovered key establishment protocols of Diffie and Hellman [16] and Rivest, Shamir and Adleman [26], to name a few. However, the security of those currently ubiquitous cryptographic solutions will be compromised with the advent of full-scale quantum computers, as discovered by Peter Shor more than two decades ago [27]. And even if a quantum computer is never built, no one has been able to prove their security against classical attacks, nor that of quantum-resistant candidates based, for instance, on short vectors in lattices. Furthermore, Merkle had already understood in 1974 that quadratic security *could* be practical if the underlying one-way function (see below) can be computed very quickly: if it takes one nanosecond to compute the function and legitimate users are willing to spend one second each, a classical adversary who could only invert the function by exhaustive search would require fifteen expected *years* to break Merkle's original scheme.

The main interest of Merkle's solution is that it offers *provable* security, at least in the *query model* of computational complexity, a model closely related to the random oracle model. In this model, we assume the existence of a *black-box* function $f : D \to R$ from some domain D to some range R, so that the only way to learn something about this function is to query the value of $f(x)$ on inputs $x \in D$ that can be chosen arbitrarily. The *query complexity* of some problem given f is defined as the expected number of calls to f required to solve the problem, using the best possible algorithm. In our case of interest, we shall consider *random* black-box functions, meaning that for each $x \in D$, the value of $f(x)$ is chosen uniformly at random within R, independently of the value of $f(x')$ for any other $x' \in D$. Provided the size r of R is sufficiently large compared to the size d of D, such a random function is automatically one-to-one, except with vanishing probability. The main characteristic of these black-box random functions that is relevant to the proof of security of Merkle's scheme is that, given a randomly chosen point y in the image of f, the only (classical) approach to finding an x so that $f(x) = y$ is exhaustive search: we have to try x's one after another until a solution is found. Indeed, whenever we try some x' and find that $f(x') \neq y$, the *only* thing

we have learned is that this particular x' is not a solution. Provided the function is indeed one-to-one, we expect to have to query the function $d/2$ times on average in order to find the unique solution.

One may argue that black-box random functions do not exist in real life, but we can replace them in practice with one-way functions – provided *they* exist! – which is what Merkle meant by "one-way encryption" in his 1974 class assignment [24]. Thus, we can base the security of Merkle's scheme on the *generic* assumption that one-way functions exist, which is unlikely to be broken by a quantum computer, rather than the assumption that *specific* computational problems such as factorization or finding short vectors in lattices are difficult, at least the first one of which is known not to hold on a quantum computer. Can we do better than provable *quadratic* security in the query model? This question remained open for 35 years, and was finally settled in the negative by Boaz Barak and Mohammad Mahmoody-Ghidary [4], building on earlier work of Russell Impagliazzo and Steven Rudich [19]: any protocol by which the legitimate parties can obtain a shared key after $O(N)$ expected queries to a black-box random function can be broken with $O(N^2)$ expected queries to the same black box.

It was apparently noticed for the first time by one of us in 2005, and published a few years later [15], that Merkle's original 1974 scheme [24], as well as his better known subsequently published *puzzles* [25], are broken by Grover's algorithm [17] on a quantum computer. This attack assumes that the eavesdropper can query the function in quantum superposition, which is perhaps not reasonable if the function is provided as a *physical* classical black box, but is completely reasonable if it is given by the publicly-available *code* of a one-way function (as originally envisioned by Merkle). If the legitimate parties are also endowed with a quantum computer, the same paper [15] gave an obvious fix, by which the legitimate parties can establish a key after $O(N)$ quantum queries to the black-box function, but no quantum eavesdropper can discover it with better than vanishing probability without querying the function $O(N^{3/2})$ times. That paper made the explicit conjecture that this was best possible when quantum codemakers are facing quantum codebreakers in the game of provable security in the random black-box model. The issue of protecting classical codemakers against quantum codebreakers was not addressed in Ref. [15].

At the Crypto 2011 conference [13], several of us disproved the conjecture of Ref. [15] with the introduction of a new quantum protocol that no quantum eavesdropper could break without querying the black-box functions $\Omega(N^{5/3})$ times.[1] We also offered the first protocol provably capable of protecting *classical* codemakers against *quantum* codebreakers, although $O(N^{13/12})$ queries in superposition sufficed for the quantum eavesdropper to obtain the not-so-secret key. Unfortunately, our security proofs were worked out in the traditional computational complexity *worst-case* scenario. In other words, it was only proved that any quantum eavesdropper limited to $o(N^{5/3})$ or $o(N^{13/12})$ queries, depending on whether the legitimate parties are quantum or classical, would be likely to fail *on at least one possible instance* of the protocol. This did not preclude that most instances of the protocol could result in insecure keys against an eavesdropper who would work no harder than the legitimate parties. Said otherwise, our Crypto 2011 result was of limited cryptographic significance.

In subsequent work [14], we claimed to have provided a proper average-case analysis of our protocols, rendering them cryptographically meaningful, so that any quantum eavesdropper has a vanishing probability of learning the key after only $o(N^{5/3})$ or $o(N^{7/6})$ queries[2], where

[1] The word "functions" is plural because the 2011 protocol required *two* black-box random functions.
[2] For classical legitimate parties, the $o(N^{13/12})$ of Ref. [13] had been improved to $o(N^{7/6})$ in Ref. [14].

the probabilities are taken not only over the execution of the eavesdropping algorithm but also over the instance of the protocol run by the legitimate parties. We also extended our results to two sequences of protocols based on the k-SUM problem (Definition 1 in Section 3), where $k \geq 2$ is an integer parameter, in which the legitimate parties query the black-box random functions $O(kN)$ times. It was claimed that any quantum eavesdropper had a vanishing probability of learning the key after $o(N^{\frac{1}{2}+\frac{k}{k+1}})$ or $o(N^{1+\frac{k}{k+1}})$ queries, against the classical or the quantum protocol parametrized by k, respectively. Again, this was claimed to hold not only in the cryptographically-challenged worst-case scenario, but also when the probabilities are taken over the protocols being run by the legitimate parties.

Unfortunately, all our average-case analyses in Ref. [14] were incorrect! The case $k = 2$ can be fixed rather easily, hence the insufficiency of $o(N^{5/3})$ queries for a quantum-against-quantum protocol and of $o(N^{7/6})$ queries for a classical-against-quantum protocol in a cryptographically significant setting can be derived from the incorrect arguments provided in Ref. [14]. However, we also claimed in Ref. [14] that the case $k > 2$ could be proved in ways "similar to" when $k = 2$. This was a mistake due to a fundamental difference in the k-SUM problem whether $k = 2$ or $k > 2$. Whereas the 2-SUM problem is easily seen to be random self-reducible, so that its hardness in worst case implies its hardness on average, this does not seem to be the case for the k-SUM problem when $k > 2$. In particular, the worst-case lower bound proved by Aleksandrs Belovs and Robert Špalek [8] on the difficulty of solving the k-SUM problem on a quantum computer does not extend in any obvious way to a lower bound on average. And without such an average lower bound, our results claimed in Ref. [14] go up in smoke for $k > 2$. Furthermore, for a technical reason explained later, even such an average lower bound would not suffice.

In this paper, we overcome all these problems and give a correct and cryptographically meaningful [3] security proof for all our protocols from Ref. [14]. Consequently, we prove that for any $\varepsilon > 0$ there is a classical protocol that allows the legitimate parties to establish a common key after $O(N)$ expected queries to black-box random functions, yet any quantum eavesdropper will have a vanishing probability of learning their key after $O(N^{1.5-\varepsilon})$ queries to the same oracle. The vanishing probability is over the randomness in the actual run of the protocol followed by that of the eavesdropper's algorithm. If we allow the legitimate parties to use quantum computers as well, their advantage over the quantum eavesdropper becomes arbitrarily close to the quadratic advantage that classical legitimate parties enjoyed over classical eavesdroppers in the seminal 1974 work of Ralph Merkle [24].

Our results require new tools in quantum query complexity, which are of independent interest. In particular, we introduce techniques to lower-bound the quantum query complexity of distinguishing between two probability distributions, which we use to extend the adversary lower bound method in order to handle average-case complexity, but they could have other uses in cryptography. This approach is necessary for the distributions of inputs considered here because the associated decision problems become trivial on average, which prevents us from applying the average-case method developed in Ref. [7]. Furthermore, we prove a composition theorem for this new lower bound method, extending that of Ref. [13], which was valid only to prove cryptographically irrelevant worst-case lower bounds [4]. Using these

[3] To be honest, it is not entirely cryptographically meaningful to restrict the analysis to the number of calls to the black-box functions, taking no account of the computing time that may be required outside those calls. However, if we also restrict the legitimate expected *time* to be in $O(N)$, then our quantum protocol with $k = 3$ remains valid and provably resists any $o(N^{7/4})$-time quantum eavesdropping attack, which was claimed in Ref [14], but with a fundamentally incorrect proof.

[4] Some parts of the proofs are omitted in the present version. They can be found in the extended version

two tools, we prove that any quantum eavesdropper who does not make a prohibitive number of calls to the black-box functions will fail to break a typical instance of the protocol, except with vanishing probability.

This work fits in the general framework of "Cryptography in a quantum world" [12], which addresses the question: "Is the fact that we live in a quantum world a blessing or a curse for codemakers?". It is a blessing if we allow quantum communication, thanks to Quantum Key Establishment (aka Quantum Key Distribution – QKD) [10], at least if the protocols can be implemented faithfully according to theory [29, 22]. On the other hand, it is a curse if we continue to use the current cryptographic infrastructure, which pretends to secure the Internet at the risk of falling prey to upcoming quantum computers. However, it is mostly a draw in the realm of provable query complexity in the black-box model considered in this paper since codemakers enjoy a quadratic (or arbitrarily close to being quadratic) advantage over codebreakers in both an all-classical or an all-quantum world, at least in terms of query complexity (but see footnote 4 again). Furthermore, the known proof that quadratic security is best possible in an all-classical world [4] does not extend to the all-quantum world, and hence the (unlikely) possibility remains that a more secure protocol could exist in our quantum world.

The rest of the paper is organized as follows. Section 2 lists all the techniques and related notations that are used throughout the paper. Section 3 recalls the classical and quantum protocols from Refs [13, 14]. In Section 4, we introduce a new method to prove lower bounds on the difficulty of distinguishing between two probability distributions, which we use to study average-case quantum query complexity. This method extends the extensively studied adversary method. We then apply this method to the k-SUM problem in Section 5, which is at the heart of our hardness result. Finally, in Section 6, we prove a composition theorem for the new adversary method introduced in Section 4. This allows us to conclude that typical runs of the protocols from Refs [13, 14] are indeed secure against quantum adversaries.

2 Preliminaries and Notation

At the heart of this work is a lower bound on the quantum query complexity of a generalisation of the k-SUM problem. Many techniques have been given to prove such lower bounds in the worst-case scenario, including the adversary method [2, 18, 21]. This method is based on the spectral norm of a matrix, Γ, indexed in the rows and columns by inputs to the problem. Roughly, each entry of the matrix $\Gamma[x, y] \in \mathbb{R}$ can be thought of as representing the hardness of distinguishing inputs x and y. It is known that for Boolean functions, the (negative) adversary bound is multiplicative under function composition [18]. For non-Boolean functions, a general composition theorem fails to hold, as counterexamples can be found. Nevertheless, it was shown in Ref. [13] that the adversary method *is* multiplicative under composition with (non-Boolean) unstructured search problems.

In this paper, we extend the quantum adversary method to average-case complexity, which is crucial for cryptographic applications, and we show that a similar composition property holds for this measure. As for the adversary bound, this method is based on the spectral norm of matrices, and involves probability distributions. Below, we summarize the notation related to functions, algebra and probabilities, used throughout the paper.

We consider *decision* or *search* problems denoted F, G or H. These problems are on abelian groups, which are denoted $\mathbb{G}$, or $\mathbb{G}_m$ when we want the order m of the group to

of this work [6].

appear explicitly. The group operation is denoted "+" and its inverse "−". For a decision problem F, the inputs in the language F are called *positive* and the inputs not in the language are *negative*. We compose our problems with an unstructured search problem to make them harder. To do so, we need to add to the alphabet an element that does not belong to $\mathbb{G}$. We denote this element "$\star$".

Fix two problems $\mathsf{F} : A^n \to B$ and $\mathsf{G} : C \to A$ for some $n \in \mathbb{N}$. Then, the composed problem $\mathsf{F} \circ \mathsf{G}^n : C^n \to B$ is defined by $\mathsf{F} \circ \mathsf{G}^n(x_1, \dots x_n) = \mathsf{F}(\mathsf{G}(x_1), \dots, \mathsf{G}(x_n))$ for $(x_1, \dots x_n) \in C^n$.

For any positive integer n we use $[n]$ to denote the set of n elements $\{0, 1, 2, \dots, n-1\}$. We only make use of basic concepts of quantum computing: states, unitary operations and measurements. These notions are used in Section 4, but even there, the calculations boil down to basic linear algebra. The entries of an $n \times m$ matrix Γ are denoted $\Gamma[x, y]$, where $x \in [n]$ and $y \in [m]$. For $X \subseteq [n]$ and $Y \subseteq [m]$, $\Gamma^{X,Y}$ is the restriction of Γ to the rows and columns in X and Y, respectively.

The direct sum of spaces, operators, matrices or vectors is denoted "$\bigoplus$". The inner product of two states (or vectors in an Hilbert space) ψ and ϕ is $\langle \psi, \phi \rangle$. For a matrix A, we use $\|A\|$ for its spectral norm, that is, its largest singular value, and $\|A\|_{\mathrm{F}}$ for the Frobenius norm, that is, the square root of the sum of the squares of the moduli of its elements. For two matrices A and B, we denote $A \circ B$ their entrywise (or Hadamard) product. We make use of the two following matrices: the $n \times n$ identity matrix I_n and the $n \times n$ all-one matrix J_n.

We use $\mathcal{P}$ and $\mathcal{Q}$ for probability distributions over inputs to the problems. The *support* of a distribution is the set of elements with non-zero probability. We sometimes identify distributions with vectors. More precisely, if p_x is the probability of x in $\mathcal{P}$, we can consider the vector $\mathcal{P}$ given by the entries $\mathcal{P}[x] = p_x$. We use "$X \sim \mathcal{P}$" to denote that the random variable X is sampled from $\mathcal{P}$. In this case, it is the variable whose probability is given by $\Pr[X = x] = p_x$. In the specific case of sampling an element x uniformly at random from a set D, we use $x \in_R D$. We also use the indicator function $1_{x \neq y}$ whose value is 1 if $x \neq y$ and 0 otherwise.

We sometimes consider sequences of probabilities, such as the accepting probability ν_n of an algorithm (for a decision problem) as a function of the input size n. For simplicity, we often omit the subscript n, in which case "ν" should be understood as a function of n. We call such a sequence ν *vanishing* if $\nu = o(1)$. If ν decreases faster than the inverse of any polynomial, we say that the event is *negligible*.

3 Provably Secure Key Establishment Protocols

With the exception of Merkle's more famous "puzzles" [25], all key establishment protocols based on black-box random functions (which Merkle called "one-way encryption") begin in a way that is essentially identical to Merkle's original 1974 idea [24], with possible inessential differences [5]. Given a black-box random function $f : D \to R$ from some domain D to some range R, Alice chooses random elements $x_i \in_R D$ and she obtains $y_i = f(x_i)$, which she sends to Bob over an authenticated channel on which Eve can freely eavesdrop. This defines the sets X of x_i's and Y of y_i's, of which X is private information kept by Alice whereas Y becomes known to all parties, including Eve. Upon receiving this information, Bob's first task is to find one or several preimage(s) under f of *any* of the points sent by Alice.

[5] In Merkle's original scheme, there is no asymmetry between Alice and Bob, as they both "guess at keywords" and share and compare their one-way encryptions until they discover that they have guessed at the same keyword. In all the protocols considered here, Alice goes first and Bob works from there.

The various schemes that were considered in Refs [24, 15, 13, 14] differ in how Bob proceeds to find the preimage(s), how many such preimages he needs to find, and how he informs Alice of which preimage(s) he has found. In Merkle's original scheme [24], he needs to find a single preimage. This is done by querying f on random points in its domain until some x is found such that $f(x) = y \in Y$. Afterwards, Bob sends y back to Alice, who can find efficiently the corresponding x because it is among her set X, which she had kept. This shared x becomes their secret key. The intuition behind the security of this scheme stems from the freedom in Bob's task to invert f on any element of Y, compared to how stringent Eve's is since she must invert it on the specific element that Bob had inverted by chance.

To be more precise, let N be a safety parameter, let the domain of f contain N^2 points and its range be of size N^5, which is large enough to ensure that f is one-to-one except with vanishing probability. If Alice chooses N random points in the domain of f and Bob tries random such points as well until he hits upon an x such that $f(x) \in Y$, it is easy to see that both Alice and Bob need query function f an expected number of N times. However, a classical Eve requires an expected $N^2/2$ queries, which gives a quadratic advantage to the legitimate parties.

Unfortunately, inverting one specific point in the image of f with the help of a quantum computer requires only $\frac{\pi}{4}\sqrt{N^2} = \frac{\pi}{4}N$ queries to f by way of Grover's algorithm [17], which is slightly *fewer* than the effort required by the legitimate parties. This is why Merkle's original scheme is totally broken against a quantum adversary, as first pointed out in Ref. [15]. In order to restore security, two main modifications to Merkle's original scheme have been considered, as we now proceed to describe.

3.1 Variations on Merkle's Idea

If we require Bob to find k distinct preimages among the N points sent by Alice, for some $k > 1$, rather than a single one, he will only have to work roughly k times as hard, provided $k \ll N$. The key shared by Alice and Bob could then be the concatenation of those preimages in the order in which the corresponding images were sent by Alice in the first step. But how can Bob tell Alice which preimages he was able to find in a way that will force Eve to make much more queries than her? A first solution was proposed in Ref. [13] for the case $k = 2$, but a much simpler one was given subsequently in Ref. [14] for arbitrary k. The idea is to introduce a second black-box random function t from the same domain to some sufficiently large group $\mathbb{G}$. If Bob finds preimages $x_{i_1}, x_{i_2}, \ldots, x_{i_k} \in X$, with $1 \le i_1 < i_2 < \cdots < i_k \le N$, and sends $w = t(x_{i_1}) + t(x_{i_2}) + \cdots t(x_{i_k})$ to Alice, she needs only call black-box function t on the N points she had kept in X in order to determine Bob's k preimages, provided the order of $\mathbb{G}$ was chosen sufficiently large to ensure the uniqueness of the solution, except with vanishing probability. Taking the order to be N^{4k+1} is sufficient to ensure this. Furthermore, she can do this efficiently, in terms of computing time, when $k = 2$. Hence, Alice needs to query each of functions f and t exactly N times, whereas Bob needs to query function f an expected $O(kN)$ times and function t exactly k times.

How difficult is the cryptanalytic task for quantum Eve, who has seen the y's sent from Alice to Bob and the single w sent from Bob to Alice? We gave an explicit algorithm based on quantum walks [23] in Hamming graphs in Ref. [14], which allows her to discover the secret key after $O(N^{1/2+k/(k+1)})$ calls to the black-box functions. In the same paper, we claimed that a matching $\Omega(N^{1/2+k/(k+1)})$ lower bound holds for a typical instance of the protocol, which is formally stated in Theorem 8 below, but the proof proposed in Ref. [14] fails for $k > 2$ in a way that cannot be repaired. The main purpose of the present paper is to offer a correct proof of this theorem. It follows that for any fixed $\varepsilon > 0$, there is a *classical*

key establishment protocol (taking $k = \lfloor 1/\varepsilon \rfloor$) that allows the legitimate parties to establish a shared key after $O(N)$ expected queries to black-box random functions f and t, yet any *quantum* eavesdropper will have a vanishing probability of learning their key after $O(N^{1.5-\varepsilon})$ queries to the same oracle. If we take account of computational complexity in addition to query complexity, we must be content with $k = 2$, in which case the claim is much more modest, but still the quantum codebreaker must work more than linearly harder than the classical codemakers. Along the way, we need to develop in Section 4 new tools for the study of *average-case* quantum query complexity, which had essentially remained virgin territory despite its obvious importance, in particular but not only for cryptography.

The second modifications to Merkle's original scheme that has been considered [15, 13, 14] is to play a fair game in allowing the codemakers to use quantum computers as well. The first benefit is that we can enlarge the domain of f to contain N^3 points. If Alice proceeds exactly as before, Bob can use an extension of Grover's algorithm known as BBHT [11] in order to find random preimages of the N image points initially sent by Alice at the cost of $O(\sqrt{N^3/N}\,) = O(N)$ queries per preimage, provided $k \ll N$. This increase in the domain size of f, and correspondingly of t, makes it significantly harder for a quantum eavesdropper to solve the conundrum and discover the key shared by Alice and Bob. Indeed, we also prove Theorem 9, stated below, to the effect that no cryptanalytic attack can succeed on a typical instance of the protocol, except with vanishing probability, short of making $\Omega(N^{1+k/(k+1)})$ queries to the black-box functions. Again, this theorem was claimed in Ref. [14] but its proof was fundamentally flawed for $k > 2$. Taking k sufficiently large, this offers a quantum-against-quantum security that is arbitrarily close to the quadratic security that the original scheme of Merkle [24] offered in the classical-against-classical scenario. The second benefit to allowing the codemakers to use quantum computers is that now a quantum Alice can be efficient in terms of computation time, in addition to query complexity, even when $k = 3$. According to Theorem 9, we get an $\Omega(N^{7/4})$ security guarantee for a protocol that could become practical once sufficiently powerful quantum computers start to seriously threaten the security of the current Internet cryptographic infrastructure. This is the most secure *proven* solution ever discovered to the conundrum of post-quantum cryptography [12] when all parties have equal quantum computing capabilities, at least in the random oracle model, and its security is reasonably close to that of Merkle's provably optimal scheme in an all-classical world but otherwise in the same model.

3.2 The k-SUM Problem

The security of the protocols that we study is based on the k-SUM problem, which consists in searching for k elements among N in some abelian group $\mathbb{G}$ whose sum is a given value $w \in \mathbb{G}$.

▶ **Definition 1** (k-SUM problem)**.** Given an abelian group $\mathbb{G}$, a function $t : D \to \mathbb{G}$ for some domain D, a *target* $w \in \mathbb{G}$ and N distinct elements $x_1, x_2, \ldots, x_N \in D$, the problem is to find k indices $1 \leq i_1 < i_2 < \cdots < i_k \leq N$ such that $w = \sum_{j \in 1}^{k} t(x_{i_j})$, provided a solution exists. The *decision* version of k-SUM is to decide whether or not a solution exists.

It is crucial to understand that we are not interested in how much computation *time* would be required to find a solution, if one exists. Rather, we want to minimize the *number of calls* to function t that will be required. Naturally, a quantum algorithm is allowed to query t on superpositions of elements of D.

When $k = 1$, this is simply the *unstructured search problem*, which consists in finding i such that $t(x_i) = w$, provided it exists. When $k = 2$ and $\mathbb{G}$ is the group of bit strings of a given length under bitwise exclusive-or, k-SUM takes the name of 2-XOR. In turn, when

$w = 0$, 2-XOR becomes the search version of the Element Distinctness (ED) problem, which consists in finding a collision in a given function if it is not one-to-one.

▶ **Definition 2** (Element Distinctness (ED) problem)**.** Given a function $t : D \to R$, the *decision* element distinctness (ED) problem is to decide whether or not this function is one-to-one.

▶ **Definition 3** (Search version of ED)**.** Given a function $t : D \to R$, the *search* version of the element distinctness problem (SED) is to find a pair of distinct $x, x' \in D$ such that $t(x) = t(x')$, provided such a pair exists.

Quantum lower bounds have been proved on all these problems [1, 8, etc.], but only in the worst-case scenario, which is most frequently studied in the field of computational and query complexity. For some of these problems, such as ED, SED, 2-XOR and 2-SUM, a simple *classical* randomized reduction suffices for proving their difficulty on average from their difficulty in the worst case even in the quantum setting, at least if we add the promise that if there is a solution, then it is unique. However, this does not appear to be the case for k-SUM when $k > 2$. Our main mistake in Ref. [14] was to take such a reduction for granted for arbitrary k after having nearly proved it in the case $k = 2$. "Nearly" because the proof for $k = 2$ was flawed, albeit easy to repair. Not so for $k > 2$, however. In order to prove the security of the key establishment protocols described above in a cryptographically meaningful context, we need to prove the difficulty of k-SUM on average for arbitrary k, which requires new quantum lower bound techniques. In fact, we need to prove the difficulty on average of a *composed* version of k-SUM, defined below in Section 3.3, which does not follow by a classical reasoning from the average difficulty of plain k-SUM. Therefore, we also have to develop a new composition theorem that works on average as well.

The first quantum lower bound discovered among these problems was for the decision element distinctness problem. Aaronson and Shi [1] proved that this problem requires $\Omega(d^{2/3})$ queries to t in the worst case, where d is the cardinality of domain D. There was a technical condition in their original proof that required $r \geq d^2$, where r is the cardinality of range R, but that condition was subsequently lifted [3, 20]. Later, Belovs and Špalek [8] proved that solving k-SUM requires $\Omega(N^{k/(k+1)})$ queries to t in the worst case, provided $m \geq N^k$, where m is the order of group $\mathbb{G}$ and N is as in Definition 1.

Even though the technique used by Aaronson and Shi was adequate only to prove worst-case lower bounds, it is elementary to conclude by a classical reasoning that the hardness in worst-case of ED implies the same hardness on average for ED, SED and 2-XOR. But, as we said already, a completely new technique, which we develop in Section 4, is required to prove a matching hardness result for k-SUM on average, which is stated as Theorem 15 in Section 5.

However, even this is not sufficient to derive the security of the key establishment protocols described above in a cryptographically meaningful manner. Indeed, the eavesdropper is not faced with an instance of k-SUM, as specified in Definition 1. He learns the value of w when Bob transmits it to Alice, and he has access to black-box function t, but he does not know the x's, which are kept secret by Alice. Instead, he learns the image of those x's by function f, which we called the y's, when Alice sent them to Bob in the first step of the protocol. In fact, he has to solve the more difficult *Hidden* k-SUM problem, which we now proceed to describe.

3.3 Hidden and Composed k-SUM Problems

The hidden k-SUM problem, defined below, corresponds precisely to the task facing the eavesdropper.

▶ **Definition 4** (Hidden k-SUM problem)**.** Given two sets D and R, an abelian group $\mathbb{G}$, two functions $f : D \to R$ and $t : D \to \mathbb{G}$, N distinct elements $y_1, y_2, \ldots, y_N \in \mathsf{Im}(f)$, and a *target* $w \in \mathbb{G}$, the problem is to find k indices $1 \leq i_1 < i_2 < \cdots < i_k \leq N$ and a preimage x_{i_j} under f for each y_{i_j}, $1 \leq j \leq k$, meaning that $f(x_{i_j}) = y_{i_j}$, such that $w = \sum_{j=1}^{k} t(x_{i_j})$, provided a solution exists. The *decision* version of hidden k-SUM is to decide if a solution exists.

In order to prove lower bounds on the quantum cryptanalytic task of breaking typical instances of the protocols described in Section 3.1, we proceed in two steps. First we have to prove the hardness of the hidden k-SUM problem on average. Then, we have to exhibit a reduction that shows how to solve an average instance of the hidden k-SUM problem using an adversary who thinks he is breaking a typical instance of the key establishment protocol. To prove the hardness of the hidden k-SUM problem on average, it helps to consider a more structured version of it, which is given by the composition of k-SUM with a search problem called pSEARCH, defined below.

▶ **Definition 5** (pSEARCH problem)**.** Let A be some set and $\star$ a symbol not in A. Consider the set P of strings $(a_1, \ldots, a_\ell)$ in $(A \cup \{\star\})^\ell$ with the promise that exactly one value is not $\star$. The problem $\mathsf{pSEARCH}_\ell : P \to A$ consists in finding this non-$\star$ value by making queries that take i as input and return a_i, $1 \leq i \leq \ell$.

An equivalent formulation of the k-SUM problem would consist in a target w in abelian group $\mathbb{G}$ and a list $(t_1, t_2, \ldots, t_N)$ of elements of $\mathbb{G}$. The problem is to find k indices $1 \leq i_1 < i_2 < \cdots < i_k \leq N$ such that $w = t_{i_1} + t_{i_2} + \cdots + t_{i_k}$. We are charged for accessing each t_i given i. This is equivalent to Definition 1 simply by taking $t_i = t(x_i)$, but it is more convenient since it allows us to consider the composition of k-SUM with N instances of pSEARCH. Thus we define the *Composed* version of k-SUM as follows.

▶ **Definition 6** (Composed k-SUM problem)**.** Given a target w in abelian group $\mathbb{G}$ and N instances of the $\mathsf{pSEARCH}_\ell$ problem using $\mathbb{G}$ as set A, we want to solve the k-SUM problem with t_i being the only non-$\star$ element in the i^{th} instance of $\mathsf{pSEARCH}_\ell$. Said otherwise, this is the composition of k-SUM and $\mathsf{pSEARCH}_\ell$ denoted $k\text{-SUM} \circ \mathsf{pSEARCH}_\ell^N$.

The composed k-SUM problem (Definition 6) is similar to its hidden variant (Definition 4), except that it is more structured, hence easier. Specifically, the x_i's that serve to define $t_i = t(x_i)$ in the hidden version, $1 \leq i \leq N$, can be *a priori* any element of D, whereas they are put in N "buckets" of size ℓ in the composed version. If we choose the size of D to be the product of N and ℓ, any algorithm capable of solving the hidden version can serve directly to solve the composed version simply by taking no account of the additional information provided by the buckets. Moreover, a random instance of the composed version can be transformed into a random instance of the hidden version, essentially by mixing the buckets. It follows that any lower bound on the composed problem translates directly into the same lower bound on the hidden problem, *mutatis mutandis*.

In Sections 4 to 6, which are more technical, we give a lower bound on the composed problem in a series of steps. First, we give a new general method to prove lower bounds for the average-case quantum query complexity (Section 4). This method is closely related to the technique given in Ref. [9], albeit with essential differences. Second, building on techniques from Refs [8, 7], we show a lower bound on the average-case quantum query complexity of k-SUM (Section 5). Third, we show a composition theorem for average-case quantum query complexity, which allows us to conclude with Theorem 18 (Section 6).

When we apply this theorem with the parameters that correspond to the protocols described in Section 3.1, we should take $n = N$, which is the number of images sent by

Alice in the first step of any of these protocols and therefore also the number of buckets. Furthermore, we should take the product of ℓ, the size of the buckets, with n, the number of buckets, to correspond to the size of the domain D used in the protocols.

Putting it all together, Theorem 18 gives us the following lower bound on the difficulty to solve the hidden k-SUM problem if the domain D of functions f and t contains d elements.

▶ **Theorem 7.** *Any quantum algorithm that uses at most T queries to find a solution to the hidden k-SUM problem with success probability at least $\nu_N > 0$ on average over the uniform distribution on positive instances requires*

$$\frac{T}{\nu_N} = \Omega\left(\sqrt{d/N - 1}\, N^{k/(k+1)}\right)$$

provided $m = \omega\left(N^{k+\frac{2}{k+1}}\right)$, where m is the order of the underlying abelian group.

3.4 The Security of Key Establishment

We proved (correctly!) in Ref. [14] that any eavesdropper who succeeds in obtaining the key with non-vanishing success probability ν in any of the protocols described in Section 3.1, after making no more than T queries, on average over the runs of the protocol, can be used to solve the hidden k-SUM problem with the same parameters. Therefore, using the fact that $d = N^2$ for the classical protocols and $d = N^3$ for the quantum protocol, we can apply Theorem 7 to conclude that the protocols are secure according to the following theorems.

▶ **Theorem 8.** *Any quantum eavesdropping strategy that makes $o\left(N^{\frac{1}{2}+\frac{k}{k+1}}\right)$ queries to the black-box functions against a typical run of the classical protocol using parameter k will fail to recover the key, except with vanishing probability.*

▶ **Theorem 9.** *Any quantum eavesdropping strategy that makes $o\left(N^{1+\frac{k}{k+1}}\right)$ queries to the black-box functions against a typical run of the quantum protocol using parameter k will fail to recover the key, except with vanishing probability.*

Furthermore, we showed in Ref. [14] that these bounds are tight.

4 Average-Case Quantum Adversary Lower Bound Method

We generalize the adversary lower bound method to handle average-case complexity. A similar bound from Ref. [9] already gives a lower bound technique on average-case query complexity, but it cannot be applied directly here, as we explain below.

We use the following complexity measure, closely related to the adversary bound [2, 18]. We give a formulation tailored to the following problem. Given two distributions $\mathcal{P}$ and $\mathcal{Q}$, and an algorithm that attempts to distinguish between them, we consider the number of queries this algorithm must make in order to succeed. The algorithm is given one input, and accepts if it thinks the sample it is given comes from $\mathcal{P}$ and rejects otherwise. The measure of success is given by the probabilities $s_{\mathcal{P}}$ and $s_{\mathcal{Q}}$, which are the probability of accepting when the algorithm is given samples from $\mathcal{P}$ and $\mathcal{Q}$, respectively.

▶ **Definition 10.** Let $\mathcal{P}$ and $\mathcal{Q}$ be two probability distributions on $\mathcal{D}$, and p_x and q_y denote probabilities of x and y in $\mathcal{P}$ and $\mathcal{Q}$, respectively. Let $s_{\mathcal{P}}, s_{\mathcal{Q}}$ be real numbers in $[0, 1]$ (representing the acceptance probability on distributions $\mathcal{P}$ and $\mathcal{Q}$, respectively). For a given matrix Γ, define the adversary bound with respect to $\Gamma, \mathcal{P}, s_{\mathcal{P}}, \mathcal{Q}, s_{\mathcal{Q}}$ as

$$\overline{\mathsf{Adv}}(\Gamma; \mathcal{P}, s_{\mathcal{P}}; \mathcal{Q}, s_{\mathcal{Q}}) = \Omega\left(\min_{j\in[n]} \frac{\delta_{\mathcal{P}}^{*}\Gamma\delta_{\mathcal{Q}} - \tau(s_{\mathcal{P}}, s_{\mathcal{Q}})\|\Gamma\|}{\|\Gamma \circ \Delta_j\|}\right). \tag{1}$$

Here, $\circ$ denotes entrywise (or Hadamard) product, and $\|A\|$ denotes the spectral norm of A (which is equal to its largest singular value). The vectors $\delta_{\mathcal{P}}[x] = \sqrt{p_x}$ and $\delta_{\mathcal{Q}}[y] = \sqrt{q_y}$ are unit vectors in $\mathbb{R}^{\mathcal{D}}$; for $j \in [n]$, the $|\mathcal{D}| \times |\mathcal{D}|$ matrix Δ_j is defined by $\Delta_j[x, y] = 1_{x_j \neq y_j}$; and

$$\tau(s_{\mathcal{P}}, s_{\mathcal{Q}}) = \sqrt{s_{\mathcal{P}} s_{\mathcal{Q}}} + \sqrt{(1 - s_{\mathcal{P}})(1 - s_{\mathcal{Q}})}. \tag{2}$$

▶ **Theorem 11.** *Assume $\mathcal{A}$ is a quantum algorithm that makes T queries to the input string $x = (x_1, \ldots, x_n) \in \mathcal{D}$, and then either accepts or rejects. Let $\mathcal{P}$ and $\mathcal{Q}$ be two probability distributions on $\mathcal{D}$. Let $s_{\mathcal{P}}$ and $s_{\mathcal{Q}}$ be acceptance probability of $\mathcal{A}$ when x is sampled from $\mathcal{P}$ and $\mathcal{Q}$, respectively. Then,*

$$T \geq \overline{\mathsf{Adv}}(\Gamma; \mathcal{P}, s_{\mathcal{P}}; \mathcal{Q}, s_{\mathcal{Q}}),$$

for any $|\mathcal{D}| \times |\mathcal{D}|$ matrix Γ.

If $\mathcal{P}$ and $\mathcal{Q}$ have partial supports, then we may use a matrix Γ whose rows are indexed by elements in the support of $\mathcal{P}$ and columns by elements of the support of $\mathcal{Q}$. In that case we can extend the matrix Γ by adding all-0 rows and columns. Notice that this does not alter the value of $\overline{\mathsf{Adv}}$.

First let us consider why we need two distributions $\mathcal{P}, \mathcal{Q}$ on the inputs (and why we cannot use existing techniques such as Theorem 33 from Ref. [9] for decision problems, where $\mathcal{P} = \mathcal{Q}$). The distribution we care about is the uniform distribution over the positive instances. Under this distribution, the decision problem is of course trivial. Using this distribution as both $\mathcal{P}$ and $\mathcal{Q}$ as in Ref. [9] would give a trivial bound.

Instead, Theorem 11 gives a lower bound on the query complexity of an algorithm that attempts to distinguish between two distributions $\mathcal{P}$ and $\mathcal{Q}$. Taking $\mathcal{P}$ as the uniform distribution over positive instances, and $\mathcal{Q}$ as the uniform distribution over all instances implies a lower bound for the *search* problem of finding k elements that sum to w with the promise that the instance is positive, by the following argument. Assume an algorithm solves the search problem with T queries with non-vanishing probability. Then we can transform this algorithm into a distinguishing algorithm with one-sided error: if the algorithm outputs a candidate solution $a_1, \ldots, a_k$, make k additional queries and check that they sum to w. If they do, accept, else reject. Then the acceptance probability on negative instances is 0. Since most instances are negative, the acceptance probability on the uniform distribution is close to 0. We are interested in the acceptance probability on the positive instances, as a function of the number of queries T.

We now proceed to the proof of Theorem 11. Our proof is closely related the proof of the worst-case negative-weighted adversary bound from Ref. [18]. We follow a slightly simplified version of the proof from Ref. [5]. As usual, we introduce a progress function, show that initially, the progress function is large (Claim 12), at the end, it is small (Claim 13), and that at each step, the decrease is bounded (Claim 14).

Proof of Theorem 11. Recall that a quantum query algorithm is given by the following sequence of operations

$$U_0 \to O_x \to U_1 \to O_x \to U_2 \to \cdots \to U_{T-1} \to O_x \to U_T,$$

where O_x denotes the input oracle, and the U_is are arbitrary unitary transformations. The operator O_x is defined by $O_x|a\rangle|i\rangle = |a + x_i\rangle|i\rangle$ which can be decomposed as

$$O_x = \bigoplus_{j=0}^{n} O_{x_j}, \tag{3}$$

where for $b \in \mathbb{G}_m$, $O_b\colon |a\rangle|i\rangle \mapsto |a+b\rangle|i\rangle$. The addition in the first register is the group operation of $\mathbb{G}_m$.

For an integer t between 0 and T, and $x \in \mathcal{D}$, let

$$\psi_x^{(t)} = U_t O_x U_{t-1} O_x \cdots U_1 O_x U_0 |0\rangle. \tag{4}$$

be the state of the algorithm on the input x after t queries. We define the quantity called the *progress function* as follows

$$W^{(t)} = \sum_{x,y\in\mathcal{D}} \sqrt{p_x q_y}\, \Gamma[x,y] \langle \psi_x^{(t)}, \psi_y^{(t)} \rangle. \tag{5}$$

The proof is split into three parts: proving that $W^{(0)}$ is large, and that both $W^{(T)}$ and $W^{(t)} - W^{(t+1)}$ are small. The proofs of the claims appear in the extended version of the paper [6].

▶ **Claim 12.** $W^{(0)} = \delta_{\mathcal{P}}^* \Gamma \delta_{\mathcal{Q}}$.

▶ **Claim 13.** $W^{(T)} \le \left(\sqrt{s_{\mathcal{P}} s_{\mathcal{Q}}} + \sqrt{(1-s_{\mathcal{P}})(1-s_{\mathcal{Q}})} \right) \|\Gamma\|$.

▶ **Claim 14.** $|W^{(t)} - W^{(t+1)}| \le 2 \max_{j\in[n]} \|\Gamma \circ \Delta_j\|$. ◀

5 Average-Case Complexity of k-SUM

Recall the k-SUM problem on n elements in an abelian group $\mathbb{G}_m$ where m is the order of the group. Let w be a fixed element of $\mathbb{G}_m$. An input $x = (x_1, \ldots, x_n)$ is called *positive* if there exists a k-subset $V = \{t_1, \ldots, t_k\} \subseteq [n]$ such that $x_{t_1} + \cdots + x_{t_k} = w$ in $\mathbb{G}_m$. Otherwise, the input is called *negative*.

Consider the following probability distribution $\mathcal{P}$ on positive inputs:

- Select a k-subset U of $[n]$ uniformly at random;
- assign to U a uniformly random string in $\mathbb{G}_m^{|U|}$ whose sum is w;
- choose the remaining elements uniformly at random.

▶ **Theorem 15.** *Assume $\mathcal{S}$ is a quantum algorithm for the search problem k-SUM that makes T queries and succeeds with probability $\nu > 0$ over inputs sampled from the distribution $\mathcal{P}$.*

Then,

$$\frac{T}{\nu} = \Omega\left(n^{k/(k+1)}\right),$$

provided that $\nu = \omega\left(n^{-1/(k+1)}\right)$ and $m = \Omega\left(n^{k+\frac{2}{k+1}}\right)$ is again the order of the underlying abelian group.

This theorem uses the following claim, whose proof appears in the extended version of the paper [6].

▶ **Claim 16.** *Let the distribution $\mathcal{P}$ be as above, and $\mathcal{Q}$ be the uniform distribution on all the inputs. There exists a matrix Γ satisfying the following constraints:*

$$\delta_{\mathcal{P}}^* \Gamma \delta_{\mathcal{Q}} = n^{k/(k+1)}, \qquad \|\Gamma\| \le \left(1 + O\left(n^{-1/(k+1)}\right)\right) n^{k/(k+1)}, \quad \text{and} \qquad \|\Gamma \circ \Delta_j\| = O(1)$$

in the notation of Theorem 11.

Proof of Theorem 15. Let $\mathcal{S}$ be the algorithm of Theorem 15. We apply Theorem 11 to the algorithm $\mathcal{A}$ defined as follows, using the constraints from Claim 16 to evaluate $\overline{\mathsf{Adv}}$. First, $\mathcal{A}$ executes $\mathcal{S}$ on its input. Let $\{t_1, \ldots, t_k\}$ be the output of $\mathcal{S}$. The algorithm $\mathcal{A}$ then queries the elements $x_{t_1}, \ldots, x_{t_k}$. It accepts if $x_{t_1} + \cdots + x_{t_k} = w$, and rejects otherwise.

The query complexity of $\mathcal{A}$ is $T+k = T+O(1)$. The acceptance probability on distribution $\mathcal{P}$ is $s_{\mathcal{P}} = \nu$. Also, since $\mathcal{A}$ always rejects a negative input,

$$s_{\mathcal{Q}} \leq \Pr_{x\sim\mathcal{Q}}\big[\text{the input } x \text{ is positive}\big] \leq \frac{1}{m}\binom{n}{k},$$

the last inequality following from the union bound. Thus, we have the following estimate on $\tau(s_{\mathcal{P}}, s_{\mathcal{Q}})$:

$$\tau(s_{\mathcal{P}}, s_{\mathcal{Q}}) = \sqrt{s_{\mathcal{P}} s_{\mathcal{Q}}} + \sqrt{(1-s_{\mathcal{P}})(1-s_{\mathcal{Q}})} \leq \sqrt{\frac{1}{m}\binom{n}{k}} + 1 - \frac{\nu}{2},$$

and using the conditions on m and ν, we obtain:

$$\begin{aligned}\frac{\delta^*_{\mathcal{P}}\Gamma\delta_{\mathcal{Q}} - \tau(s_{\mathcal{P}}, s_{\mathcal{Q}})\|\Gamma\|}{\|\Gamma \circ \Delta_j\|} &= \frac{n^{k/(k+1)} - \big(1-\Omega(\nu)\big)\Big(1+O\big(n^{-1/(k+1)}\big)\Big)n^{k/(k+1)}}{O(1)} \\ &= \Omega\Big(\nu n^{k/(k+1)}\Big).\end{aligned}$$

◀

6 Composition Theorem for the Average-Case Adversary Bound

We now prove the last remaining theorem needed to obtain the lower bound on the average case complexity of k-$\mathsf{SUM}\circ\mathsf{pSEARCH}^n_\ell$ (see Section 3.3). Recall that in this version, each input variable $x_i \in \mathbb{G}_m$ is embedded into a "bucket", that is, a sequence $(x_{i1}, \ldots, x_{i\ell}) \in (\mathbb{G}_m \cup \{\star\})^\ell$ in which exactly one element is non-$\star$. To apply our average-case adversary lower bound method, we need to define the probability distributions and the matrix that appears in Eq. 1 for the composed problem. Intuitively, this is done by tensoring the matrix of the two problems that are composed, as well as the vectors that represent the probability distributions. However, defining the matrix correctly to get a lower bound for the composed problem requires a careful analysis.

We use the distributions $\mathcal{P}_{\mathsf{F}}$ and $\mathcal{Q}_{\mathsf{F}}$ to pick inputs to the outer function F, and the uniform distribution to place each element of the input independently in its bucket. Formally, we write $\mathcal{P} = \mathcal{P}_{\mathsf{F}} \otimes U_\ell^{\otimes n}$, where U_ℓ is the uniform distribution over $[\ell]$ and the distributions are viewed as real-valued vectors indexed by elements of their supports. The definition of $\mathcal{Q}$ is similar, starting from $\mathcal{Q}_{\mathsf{F}}$.

▶ **Lemma 17.** *Let $\mathsf{F} : A^n \to B$, $\mathsf{pSEARCH}_\ell : P \to A$ where $P \subseteq (A \cup \{\star\})^\ell$ is the set of all possible buckets, $\mathsf{H} = \mathsf{F} \circ \mathsf{pSEARCH}^n_\ell$, and $\mathcal{P}_{\mathsf{F}}$, $\mathcal{Q}_{\mathsf{F}}$, $\mathcal{P}$ and $\mathcal{Q}$ defined as above. Then for any real numbers $s_{\mathcal{P}}$, $s_{\mathcal{Q}} \in [0,1]$ and matrix Γ_{F}, there exists a matrix Γ_{H} such that*

$$\overline{\mathsf{Adv}}(\Gamma_{\mathsf{H}}; \mathcal{P}, s_{\mathcal{P}}; \mathcal{Q}, s_{\mathcal{Q}}) \geq \overline{\mathsf{Adv}}(\Gamma_{\mathsf{F}}; \mathcal{P}_{\mathsf{F}}, s_{\mathcal{P}}; \mathcal{Q}_{\mathsf{F}}, s_{\mathcal{Q}})\sqrt{\ell - 1}\,.$$

▶ **Theorem 18.** *Any algorithm that finds a solution to the search version of k-$\mathsf{SUM}\circ\mathsf{pSEARCH}^n_\ell$ within T queries with probability $\nu > 0$ on average over the uniform distribution on positive instances requires*

$$\frac{T}{\nu} = \Omega\Big(\sqrt{\ell-1}\, n^{k/(k+1)}\Big)$$

provided $m = \omega\Big(n^{k+\frac{2}{k+1}}\Big)$.

The rest of this section is devoted to the proof of Theorem 18. It follows closely the proof of the composition theorem in Ref. [13], and in particular the adversary matrix for H we use here has the same structure as the matrices considered in that paper. This allows us to re-use some of the calculations from that paper (see Claims 20 and 21).

We use the following notation. Let $\mathsf{X}, \mathsf{Y} \in A^n$ denote inputs to F. Its components are $\mathsf{X}_i \in A$. The value $\Gamma_{\mathsf{F}}[\mathsf{X}, \mathsf{Y}]$ is a scalar. Notice that for the k-SUM problem, the rows of the matrix defined in the previous section are only defined for positive inputs. In order to reuse the norm calculations from the composition theorem in Ref. [13], we need to extend it to all possible inputs. We do so by extending the matrix for k-SUM with rows of zeros. This transformation does not change the norm of the matrix. Similarly, the vector $s_{\mathcal{P}_{\mathsf{F}}}$ can be extended with zeros to be defined for any input.

Proof of Lemma 17. The adversary matrix for the composed problem H is denoted Γ_{H}. We consider blocks of Γ_{H} indexed by values X, Y, which we denote $\Gamma_{\mathsf{H}}^{\mathsf{X},\mathsf{Y}}$. (These $\ell^n \times \ell^n$ blocks are a submatrix corresponding to all the inputs for which the input to F is X, in the rows, and Y, in the columns.) As in Ref. [13], we define Γ_{H} by blocks as follows:

$$\Gamma_{\mathsf{H}}^{\mathsf{X},Y} = \Gamma_F[\mathsf{X}, \mathsf{Y}] \cdot \bigotimes_{i \in [n]} \overline{\Gamma}^{\mathsf{X}_i, \mathsf{Y}_i},$$

where for $a, b \in A$,

$$\overline{\Gamma}^{a,b} = \begin{cases} \|\mathsf{J}_\ell - \mathrm{I}_\ell\| \cdot \mathrm{I}_\ell & \text{if } a = b \\ \mathsf{J}_\ell - \mathrm{I}_\ell & \text{otherwise.} \end{cases}$$

An optimal adversary matrix for pSEARCH can be obtained by taking $\mathsf{J}_\ell - \mathrm{I}_\ell$ for all blocks except the diagonal ones that are all zeroes. But if we were using it, a block $\Gamma_{\mathsf{H}}^{X,Y}$ would be zero whenever there is an i such that $X_i = Y_i$. Using the matrix $\overline{\Gamma}$, with modified diagonal blocks, overcomes this issue.

From the distributions $\mathcal{P}_F$ and $\mathcal{Q}_F$, we define the vector $\delta_{\mathcal{P}_F} = \sqrt{\mathcal{P}_F}$, that is, $\delta_{\mathcal{P}_{\mathsf{F}}}[\mathsf{X}] = \sqrt{\Pr_{\mathsf{X} \sim \mathcal{P}_F}[\mathsf{X}]}$ (similarly for $\delta_{\mathcal{Q}_{\mathsf{F}}}$). Again, we can split $\delta_{\mathcal{P}_{\mathsf{F}}}$ into blocks $\delta_{\mathcal{P}_{\mathsf{F}}}^{\mathsf{X}}$.

With these definitions in hand, we can compute the terms that appear in Eq. 1 of Definition 10. This is done in Claims 19, 20, and 21. When referring to Ref. [13], we use $S_i = \mathsf{J}_\ell - \mathrm{I}\ell$ for all i $(1 \le i \le n)$.

▶ **Claim 19.** $\delta_{\mathcal{P}}^\dagger \Gamma_{\mathsf{H}} \delta_{\mathcal{Q}} = \delta_{\mathcal{P}_{\mathsf{F}}}^\dagger \Gamma_F \delta_{\mathcal{Q}_F} \cdot \|\mathsf{J}_\ell - \mathrm{I}_\ell\|^n$.

▶ **Claim 20.** *[13, claim on last line of page 409]* $\|\Gamma_{\mathsf{H}}\| = \|\Gamma_{\mathsf{F}}\| \cdot \|\mathsf{J}_\ell - \mathrm{I}_\ell\|^n$.

▶ **Claim 21.** *[13, claim near the end of page 410] For a query i that corresponds to index q in the bucket p,* $\|\Gamma_{\mathsf{H}} \circ \Delta_i\| = \|\Gamma_{\mathsf{F}} \circ \Delta_p\| \cdot \|\mathsf{J}_\ell - \mathrm{I}_\ell\|^{n-1} \cdot \|(\mathsf{J}_\ell - \mathrm{I}_\ell) \circ \Delta_q\|$.

Claims 20 and 21 were proven in the arXiv extended version of Ref. [13]. Although the claims in the original Crypto version of Ref. [13] consider specifically the Element Distinctness problem, the paper mentions that an explicit description of the adversary matrix is not needed (such a description was indeed unknown when this proof was given). For this reason, these two claims apply to any outer function F, and in particular to k-SUM. Note that the arXiv extended version of Ref. [13] contains the proofs for arbitrary outer functions. The proof of Claim 19 appears in the extended version of the paper [6].

Using the fact that $\|\mathsf{J}_\ell - \mathrm{I}_\ell\| = \ell - 1$ and $\|(\mathsf{J}_\ell - \mathrm{I}_\ell) \circ \Delta_q\| = \sqrt{\ell - 1}$ for any q, we immediately get Lemma 17 by substituting the values obtained in Claims 19, 20 and 21 into Definition 10. ◀

Proof of Theorem 18. Using the values computed in Section 5 we get

$$\begin{aligned} T &= \Omega\left(\frac{\delta^{\dagger}_{\mathcal{P}_\mathsf{F}}\Gamma_\mathsf{F}\delta_{\mathcal{P}_\mathsf{F}} - \tau(s_\mathcal{P}, s_\mathcal{Q})\|\Gamma_\mathsf{F}\|}{\|\Gamma_\mathsf{F} \circ \Delta_i\|}\sqrt{\ell - 1}\right) \\ &= \Omega\left(n^{k/(k+1)}\sqrt{\ell-1}\left(\frac{\nu}{2} - \sqrt{\frac{1}{m}\binom{n}{k}}\right)\right) \end{aligned}$$

Suppose that ν is non-vanishing. Since m is chosen large enough to make $\frac{1}{m}\binom{n}{k}$ arbitrarily small, we get

$$\frac{T}{\nu} = \Omega\Big(\sqrt{l-1}n^{k/(k+1)}\Big).$$ ◀

Acknowledgements. We are grateful to Kassem Kalach, with whom this work has initiated many years ago. Part of this work was performed when GB visited AB, then at *QuSoft* in Amsterdam.

References

1 S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM* **51**(4):595–605, 2004.

2 A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences* **64**:750–767, 2002.

3 A. Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing* **1**(1):37–46, 2005.

4 B. Barak and M. Mahmoody-Ghidary. Merkle puzzles are optimal – An $O(n^2)$–query attack on any key exchange from a random oracle. In *Advances in Cryptology – Proceedings of Crypto 2009*, pages 374–390, 2009.

5 A. Belovs. *Applications of the Adversary Method in Quantum Query Algorithms*. PhD thesis, University of Latvia, 2014.

6 A. Belovs, G. Brassard, P. Høyer, M. Kaplan, S. Laplante and L. Salvail. Provably secure key establishment against quantum adversaries. Extended version available at `http://arxiv.org/abs/1704.08182`.

7 A. Belovs and A. Rosmanis. On the power of non-adaptive learning graphs. *Computational Complexity* **23**(2):323–354, 2014.

8 A. Belovs and R. Špalek. Adversary lower bound for the k-sum problem. In *Proceedings of 4th ACM Innovations in Theoretical Computer Science*, pages 323–328, 2013.

9 A. Belovs. Variations on quantum adversary. `http://arxiv.org/abs/1504.06943`, April 2015.

10 C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of International Conference on Computers, Systems & Signal Processing, Bangalore*, pages 175–179, 1984. Republished in 30th Anniversary Commemorative Issue of *Theoretical Computer Science* **560**(Part 1):7–11, 2014.

11 M. Boyer, G. Brassard, P. Høyer and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik* **46**:493–505, 1998.

12 G. Brassard. Cryptography in a quantum world. In *Proceedings of SOFSEM 2016: Theory and Practice of Computer Science*, pages 3–16, 2016.

13 G. Brassard, P. Høyer, K. Kalach, M. Kaplan, S. Laplante and L. Salvail. Merkle puzzles in a quantum world. In *Advances in Cryptology – Proceedings of Crypto 2011*, pages 391–410, 2011. Extended version available at `http://arxiv.org/abs/1108.2316v1`.

14 G. Brassard, P. Høyer, K. Kalach, M. Kaplan, S. Laplante and L. Salvail. Key establishment à la Merkle in a quantum world. http://arxiv.org/abs/1108.2316v2, February 2015.

15 G. Brassard and L. Salvail. Quantum Merkle puzzles. *Proceedings of Second International Conference on Quantum, Nano, and Micro Technologies*, pages 76–79, 2008.

16 W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory* **22**(6):644–654, 1976.

17 L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters* **79**(2):325–328, 1997.

18 P. Høyer, T. Lee and R. Špalek. Negative weights make adversaries stronger. In *Proceedings of 39th Annual ACM Symposium on Theory of Computing*, pages 526–535, 2007. doi:10.1145/1250790.1250867.

19 R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of 21st Annual ACM Symposium on Theory of Computing*, pages 44–61, 1989.

20 S. Kutin. Quantum lower bound for the collision problem with small range. *Theory of Computing* **1**(1):29–36, 2005.

21 T. Lee, R. Mittal, B. W. Reichardt, R. Špalek and M. Szegedy. Quantum query complexity of state conversion. In *Proceedings of 52nd Annual IEEE Symposium on Foundations of Computer Science*, pages 344–353, 2011.

22 L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar and V. Makarov. Hacking commercial quantum cryptography systems by tailored bright illumination. *Nature Photonics* **4**(10):686–689, 2010.

23 F. Magniez, A. Nayak, J. Roland and M. Santha. Search via quantum walk. *SIAM Journal on Computing* **41**(1):142–164, 2011.

24 R. Merkle. Publishing a new idea. http://www.merkle.com/1974/.

25 R. Merkle. Secure communications over insecure channels. *Communications of the ACM* **21**(4):294–299, 1978.

26 R. L. Rivest, A. Shamir and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* **21**(2):120–126, 1978.

27 P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing* **26**:1484–1509, 1997.

28 P. Wayner. British document outlines early encryption discovery. http://www.nytimes.com/library/cyber/week/122497encrypt.html, New York Times Technology Cybertimes column, 24 December 1997.

29 Y. Zhao, C.-H. F. Fung, B. Qi, C. Chen and H.-K. Lo. Quantum hacking: Experimental demonstration of time-shift attack against practical quantum-key-distribution systems. *Physical Review A* **78**(4):042333, 2008.