Experimental Integration of Quantum Key Distribution and Post-Quantum Cryptography in a Hybrid Quantum-Safe Cryptosystem

Lydia Garms, Taofiq K. Paraïso, Neil Hanley, Ayesha Khalid, Ciara Rafferty, James Grant, James Newman, Andrew J. Shields, Carlos Cid, and Maire O'Neill*

Quantum key distribution (QKD) and post-quantum cryptography (PQC) are the two counter measures against cryptographic attacks via quantum computing. While QKD offers information theoretic security but limited authentication scalability, PQC facilitates scalable authentication in high density networks but is not information theoretic secure. Therefore, an ideal quantum-safe framework should efficiently leverage the complementarity of both techniques. However, despite growing efforts in integrating both, current realizations have focused on channel authentication, and a complete cryptosystem addressing both hybrid authentication and hybrid key exchange is yet to be demonstrated. Here, an authenticated hybrid key exchange protocol is introduced that incorporates PQC and QKD in a modular and information-theoretic secure architecture. The quantum-safe protocol is inherently resilient to catastrophic cryptographic failures and provides both forward and post-compromise security. As proof-of-concept implementation, the cryptosystem on a QKD hardware prototype is integrated, with the QKD processing, PQC key exchange and secret state masking via physical unclonable functions (PUFs) all running on a single field programmable gate array (FPGA). This work paves the way for the deployment of versatile and modular quantum-safe networks that exploit the complementarity of PQC and QKD.

1. Introduction

Authenticated key exchange allows two parties who have never met to agree on a shared secret, while verifying the identities of parties involved in the key exchange. This shared secret can subsequently be used for fast, efficient symmetric encryption. Provided all bits in the key are unpredictable and the key is as long as the message, the Vernam cipher or one-time pad (OTP) can be used to ensure information theoretic security of the cyphertext. However, due to the practical cost of the OTP for long messages, our modern encryption schemes are instead based on computational hardness assumptions. Assuming that secure schemes for symmetric encryption and authenticated key exchange are used, the resulting ciphertext will also be secure.[1] In the symmetric setting, parties can authenticate a message with a message authentication code (MAC) that makes use of a pre-shared key. Only the other party holding the same pre-shared key can verify the MAC. In the asymmetric setting, a party can use a secret key to sign a message with a digital signature scheme and anyone with access to their public key can verify the signature. Existing asymmetric cryptographic primitives such as Rivest-Shamir-Adleman (RSA) and elliptic curve cryptography (ECC) a are used as building blocks to perform authenticated key exchanges.

As we move closer to the realization of large-scale, fault-tolerant quantum computers,[2–8] current cryptographic

L. Garms, C. Cid
Information Security Group
Royal Holloway
University of London
Egham TW20 0EX, UK

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/qute.202300304

DOI: 10.1002/qute.202300304

T. K. Paraïso, J. Newman, A. J. Shields
Toshiba Europe Ltd.
Cambridge Research Laboratory
Cambridge CB4 0GZ, UK
E-mail: taofiq.paraiso@crl.toshiba.co.uk

N. Hanley, A. Khalid, C. Rafferty, J. Grant, M. O'Neill
Centre for Secure Information Technologies (CSIT)
Queen's University Belfast
Belfast BT3 9DT, UK

C. Cid
Simula UiB
Bergen 5006, Norway

C. Cid
Okinawa Institute of Science and Technology Graduate University
Okinawa 904-0495, Japan

standards are under threat. Quantum algorithms such as Shor's algorithm[9] and Grover's search algorithm[10] have the potential to completely break the widely used asymmetric RSA and ECC algorithms, while halving the security strength of symmetric-key algorithms such as Advanced Encryption Standard (AES). To date, only two different cryptographic approaches are deemed quantum-safe, i.e., immune to attacks by quantum computers. First, a physical approach, quantum key distribution (QKD), which produces information theoretic secure (ITS) symmetric encryption keys between two remote parties Alice and Bob, where the source of security is underpinned by the communication of quantum light signals.[11] QKD has reached a number of breakthroughs toward global intercontinental deployment, including high secret key generation rates,[12–15] long-distance quantum communications,[16–22] photonic integration,[23–27] and terrestrial and satellite-based networkdeployment.[28,29]Second, an algorithmic approach, post-quantum cryptography (PQC), which deploys newasymmetric protocols that can run on classical computers but are believed to be secure against known quantum computing attacks. In 2016, the National Institute of Standards and Technology (NIST) initiated a PQC competition to standardize quantum resistant algorithms (QRAs) for key exchange mechanisms (KEMs) and digital signature algorithms (DSAs).[30] The winning algorithms were announced in July 2022, with standardization to follow in 2024.[31]

Hybrid cryptography combines multiple cryptographic primitives to allow security to be based on a wide range of hard problems, helping to mitigate against unknown possible attacks. The security of a hybrid cryptosystem holds as long as one of the primitives is secure, thus reinforcing security over longer term. Historically, the term hybrid referred to a combination of symmetric and asymmetric algorithms.[32] In the context of PQC migration, the use of hybrid schemes combining classical and post-quantum public-key algorithms is already advocated, as they can guarantee inter-operability during the transition phase.[33–35] Further hybrid schemes combining the advantages of both QKD and PQC would greatly benefit our communication security. A natural first step in this direction is to enhance the scalability of QKD authentication using post-quantum (PQ) digital signatures.[36,37] Going beyond this step and completely merging QKD and PQC in a hybrid cryptosystem would further exploit the full potential of quantum-safe primitives and add considerable gain in terms of security: the encryption keys will remain secure even in the event of a catastrophic failure of one of the primitives caused by advances in quantum computing, or unforeseen attacks on less well established PQ algorithms.[38]

In this work, we present and demonstrate experimentally a hybrid quantum-safe cryptosystem capable of ITS authentication and key exchange. We incorporate QKD with a post-quantum DSA (PQ-DSA), a post-quantum KEM (PQ-KEM), their classical counterparts, an ITS MAC (IMAC) and a key derivation function (KDF). In addition, we provide an extra layer of hardware security by making use of a PUF for masking the secret state and tying it to the physical hardware device. The full protocol is implemented on a commercial QKD prototype where the PQ-KEM and PUF primitives are all executed by the same FPGA that executes the QKD functions. Our results constitute a new paradigm for hybrid quantum-safe cryptography, providing pathways toward information theoretic secure cryptosystems and their seamless deployment in existing quantum communication infrastructures for ultimate resilience against quantum computing threats.

2. Results

2.1. Protocol Description

A hybrid cryptosystem combines cryptographic material from different protocols to ensure that the resulting protocol is secure provided at least one of these building blocks remains secure. Our protocol is based on the proposal by Dowling, Hansen and Patterson,[39] who provided the first security framework for a hybrid authenticated key exchange (HAKE) protocol combining QKD and PQC and proved the protocol secure in this model. While the original protocol did not include PQ-DSA and did not provide a complete ITS implementation, we propose a modular protocol, Muckle++, that incorporates both PQ-DSA and advanced cryptographic building blocks to support broader functionality, and which can easily be replaced to meet application requirements of efficiency and security.

The security of a cryptographic protocol is strongly linked to its authentication mechanism. QKD nodes are conventionally authenticated using pair-wise pre-shared keys. While this approach is suitable for a small number of nodes, it may not easily scale to high-density networks, which could even include links without quantum capable hardware, such as very long distance links or wireless links. We therefore propose the flexibility of authenticating with or without pre-shared keys and provide security proofs in both cases. This modular feature is illustrated in **Figure** 1a,b. In the absence of pre–shared keys, the protocol makes use of a classical and a PQ digital signature scheme. As authentication must now rely on long-term asymmetric secrets, information theoretical security is not achievable in this setting, though the protocol remains computationally quantum-safe. When we can assume that nodes in the network have established pre-shared keys, our protocol can authenticate with a MAC combined with pre-shared key material. In this optimal instantiation, our protocol is provably ITS for both authentication and key exchange and does not rely on computational hardness assumptions. In Notes S2 and S3 (Supporting Information), we provide detailed security proofs for both settings within an updated HAKE security framework.

In both cases, the HAKE architecture further guarantees noteworthy security features illustrated in Figure 1c:

1. The key exchange protocol remains secure, provided at least any one of three ingredients: QKD, classical key exchange and quantum resistant key exchange, remains secure.
2. The secret state allows for *post-compromise security*;[40] that is, security can be recovered in the event of session keys being leaked.
3. *Forward security* is guaranteed, ensuring that a security breach will not affect the security of previous keys.

Figure 2 gives a flow diagram of the protocol. The process is summarized as follows:

(i) *Session initialization*. Before the first communication session, the initiator Alice and responder Bob are set up with identifiers, a pre-shared key (or alternatively long-term

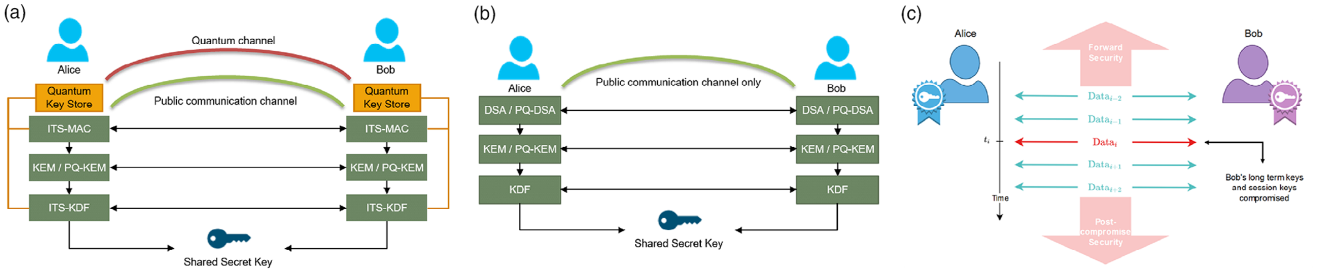

Figure 1. Protocol security features. a) Optimal instantiation: in the presence of QKD hardware, quantum key material is combined to an ITS MAC and to the KDF to achieve an ITS cryptosystem. b) In the absence of QKD hardware, e.g. wireless links, authentication is done using hybrid DSA and keys are constructed using a standard KDF. The protocol is quantum-safe but not ITS. c) Forward and post-compromise security. At time t_i the security of Bob's long term keys and session key is compromised by an attacker. Forward secrecy ensures that messages exchanged at earlier times remain protected. Post-compromise security ensures that future messages remain protected.

classical and post-quantum public keys and their own secret keys), the secret state *SecState* (initially set to 0) and a counter *ctr* (initially set to 0). In subsequent communication sessions the secret state and counter are retained from the previous session.

(ii) *Public Key Generation*. The initiator generates the public/private key pairs using both the classical KEM and the PQ-KEM. The quantum key material generated from a QKD protocol is interpreted as a shared array of random bits that both parties can access. A first message, m_0, is then formed to send the classical and PQ public keys ($lcpk_A$, $lqpk_A$) to the responder along with a header containing the type of KEM primitives used and an identifier of the party sending the message.

(iii) *Message authentication*. Authentication is needed before sending the message. If a pre-shared key is available, the message is authenticated with an IMAC tag (τ_0 / τ_1), where the secret key ($mkey_A$ / $mkey_B$) combines the pre-shared keys of both parties, and the secret state saved from the previous session. Otherwise, a MAC tag is generated conventionally and both the message and MAC tag are signed with both a classical and a PQ signatures (σ_0 / σ_1 and σ'_0 / σ'_1) using the long-term keys.

(iv) *Key exchange*. Upon reception of the first message, tag and signatures, the responder verifies the tag against the message and if successful proceeds to the key encapsulation steps. The encapsulated keys are sent in an second message m_1, authenticated with the same procedure as m_0. After verification of the second message and key decapsulation, the classical and PQ KEMs are completed and the link is authenticated for the generation of quantum key material (QKM) using a QKD protocol.

(v) *Key derivation*. The resulting session keys sk_A, sk_B are derived from the messages sent, the symmetric keys generated by the classical and post-quantum KEM, QKM, as well as the secret state to ensure post-compromise security. The QKM can be interpreted as a shared array of quantum random bits that both parties can access. These elements are successively input to a PRF.

(vi) *Secret state*. Finally, the output of the final PRF is split into the secret session keys of both users, as well as the secret state for the next session.

2.2. Selected Cryptographic Primitives

Table 1 summarizes the concrete instantiation of each cryptographic building block in our protocol. A detailed introduction to the different primitives is provided in Note S1 (Supporting Information). For the PQ-KEM, CRYSTALS Kyber,[41] on the NIST standardization track[31] was selected, while for the PQ digital signature, Falcon,[42] which is also on the standardization track, was selected. Both are lattice-based schemes, with Falcon the most compact of the selected NIST digital signature candidates. Lattice-based cryptographic schemes have been widely studied, and in general have computation speeds similar to, if not faster than, their classical equivalents. However, this is offset by the fact that their keys and ciphertexts can be significantly larger. The quantum key material is generated via a T12 protocol.[43] To instantiate the QPRF, a keyed-hash MAC (HMAC)-based KDF (HKDF)[44] was modified by xor-ing quantum key material to the PRF output.

For the IMAC, the Poly1305-AES MAC[45] construction was combined with QKM to ensure ITS assuming the security of the quantum key distribution holds. In Note S4 (Supporting Information) this scheme is detailed in full, alongside with proofs that the construction remains computationally secure if the security of the QKD is broken, and proof that ITS is achieved if the QKD remains secure.

2.3. System Architecture and Hardware Implementation

The cryptosystem was deployed on a high-speed commercial-grade QKD system prototype[13] consisting of two units, a quantum transmitter and a quantum receiver, configured to exchange quantum secure keys using the T12 protocol.[43] These are state-of-the-art systems able to generate QKD keys at rates of a few Mb/s, enabling the generation of a large store of pre-shared quantum key material between distant parties in a very short time. The overall architecture of the hybrid cryptosystem is shown in **Figure 3**. Each unit contains the discrete optics required to prepare or measure time-bin-encoded weak coherent pulses, the corresponding control electronics, an FPGA acting as the central QKD processor, and a server for the public channel communication.

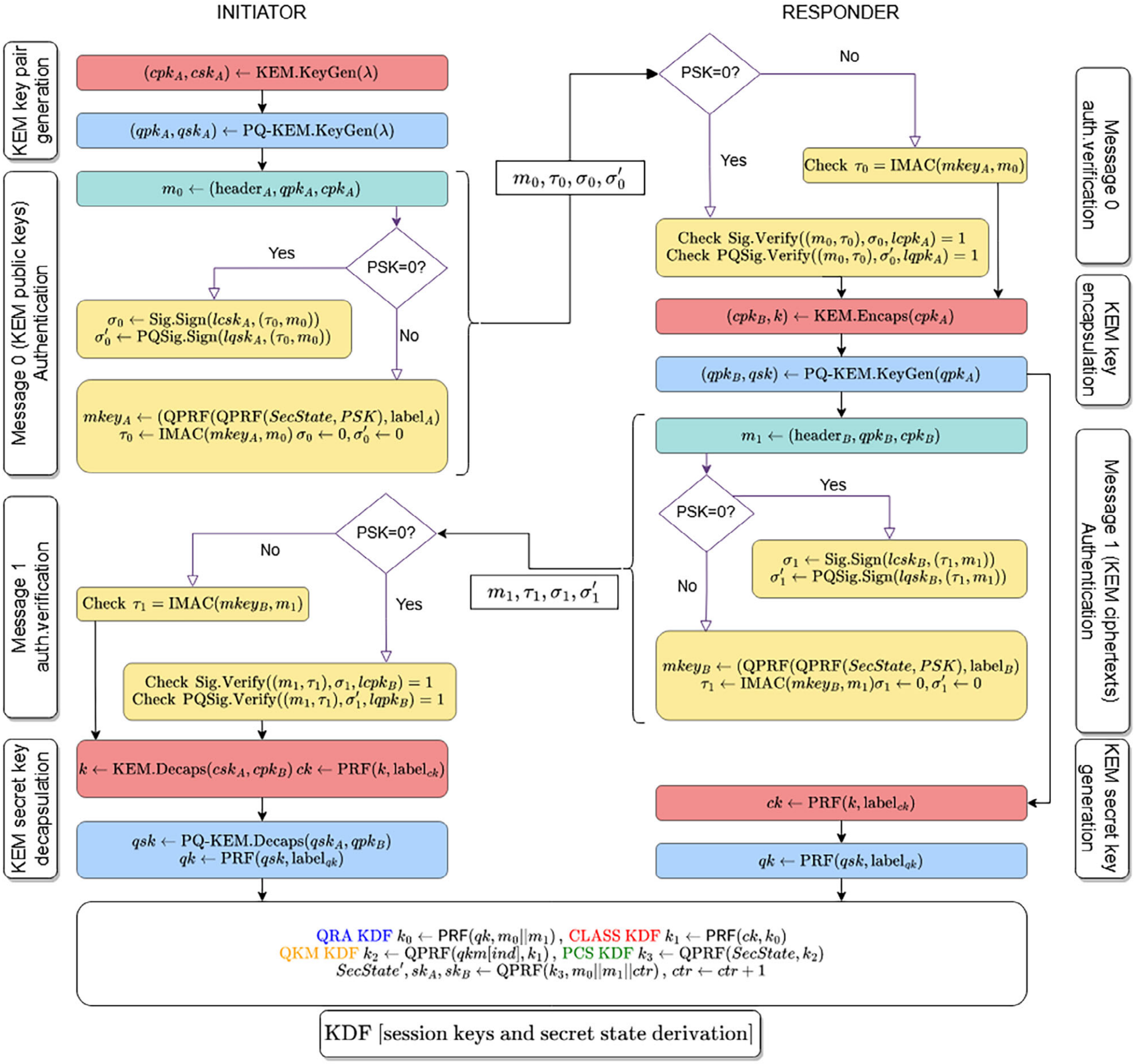
INITIATOR
RESPONDER
KEM key pair generation
$(cpk_A, csk_A) \leftarrow$ KEM.KeyGen(λ)
$(qpk_A, qsk_A) \leftarrow$ PQ-KEM.KeyGen(λ)
Message 0 (KEM public keys) Authentication
$m_0 \leftarrow (\text{header}_A, qpk_A, cpk_A)$
PSK=0?
Yes
No
$\sigma_0 \leftarrow$ Sig.Sign($lcsk_A, (\tau_0, m_0)$)
$\sigma'_0 \leftarrow$ PQSig.Sign($lqsk_A, (\tau_0, m_0)$)
$mkey_A \leftarrow$ (QPRF(QPRF($SecState, PSK$), label$_A$)
$\tau_0 \leftarrow$ IMAC($mkey_A, m_0$) $\sigma_0 \leftarrow 0, \sigma'_0 \leftarrow 0$
$m_0, \tau_0, \sigma_0, \sigma'_0$
PSK=0?
No
Yes
Check $\tau_0 =$ IMAC($mkey_A, m_0$)
Message 0 auth.verification
Check Sig.Verify($(m_0, \tau_0), \sigma_0, lcpk_A$) = 1
Check PQSig.Verify($(m_0, \tau_0), \sigma'_0, lqpk_A$) = 1
KEM key encapsulation
$(cpk_B, k) \leftarrow$ KEM.Encaps(cpk_A)
$(qpk_B, qsk) \leftarrow$ PQ-KEM.KeyGen(qpk_A)
$m_1 \leftarrow (\text{header}_B, qpk_B, cpk_B)$
PSK=0?
Yes
No
$\sigma_1 \leftarrow$ Sig.Sign($lcsk_B, (\tau_1, m_1)$)
$\sigma'_1 \leftarrow$ PQSig.Sign($lqsk_B, (\tau_1, m_1)$)
$mkey_B \leftarrow$ (QPRF(QPRF($SecState, PSK$), label$_B$)
$\tau_1 \leftarrow$ IMAC($mkey_B, m_1$)$\sigma_1 \leftarrow 0, \sigma'_1 \leftarrow 0$
Message 1 (KEM ciphertexts) Authentication
$m_1, \tau_1, \sigma_1, \sigma'_1$
Message 1 auth.verification
PSK=0?
No
Yes
Check $\tau_1 =$ IMAC($mkey_B, m_1$)
Check Sig.Verify($(m_1, \tau_1), \sigma_1, lcpk_B$) = 1
Check PQSig.Verify($(m_1, \tau_1), \sigma'_1, lqpk_B$) = 1
KEM secret key decapsulation
$k \leftarrow$ KEM.Decaps(csk_A, cpk_B) $ck \leftarrow$ PRF(k, label$_{ck}$)
$qsk \leftarrow$ PQ-KEM.Decaps(qsk_A, qpk_B)
$qk \leftarrow$ PRF(qsk, label$_{qk}$)
KEM secret key generation
$ck \leftarrow$ PRF(k, label$_{ck}$)
$qk \leftarrow$ PRF(qsk, label$_{qk}$)
QRA KDF $k_0 \leftarrow$ PRF($qk, m_0 \| m_1$) , CLASS KDF $k_1 \leftarrow$ PRF(ck, k_0)
QKM KDF $k_2 \leftarrow$ QPRF($qkm[ind], k_1$) , PCS KDF $k_3 \leftarrow$ QPRF($SecState, k_2$)
$SecState', sk_A, sk_B \leftarrow$ QPRF($k_3, m_0 \| m_1 \| ctr$) , $ctr \leftarrow ctr + 1$
KDF [session keys and secret state derivation]

Figure 2. Simplified flow diagram of the Muckle++ protocol. Hybrid classical/post-quantum asymmetric key exchange and authentication is used in conjunction with QKD to derive session keys and secret state. In the presence of pre-shared keys (PSKs), authentication is provided by an IMAC construction. Otherwise, hybrid classical and PQ digital signatures are used to sign the messages. The final key derivation step combines the different key materials by successively inputting the PQ key, the classical key, the QKD key, as well as the secret state for post-compromise security (PCS), in a series of PRFs. In the absence of QKD hardware, a standard PRF replaces the QPRF.

The FPGAs are used to control the optical hardware and for the sifting procedure, where the preparation and measurement bases are compared to only retain events where Alice and Bob selected matching bases. The resulting sifted keys are transferred from the FPGAs to the servers for the classical post-processing of the key. The first post-processing step is the error correction of the sifted keys, which is based on a low density parity check algorithm. From the error correction and sifting results, the quantum bit error rate and the photon statistics for the different intensity states (signal, decoy and vacuum) can be evaluated.[43] These parameters are essential to estimate and bound the single photon contribution to the key information and to perform the second post-processing step, i.e., privacy amplification.

To incorporate the HAKE on the QKD system, we implemented the main routine of the protocol as a c++ code executed on the server and featuring a hardware abstraction layer (HAL) providing access to the cryptographic components running in hardware built on a FPGA connected via Ethernet. We developed custom FPGA cores to execute the CRYSTALS Kyber PQ-KEM algorithm and the PUF functions for secret state masking alongside core QKD functions such as key sifting and photon statistics evaluation. The classical KEM, the DSA, the MAC and the

Table 1. Algorithms/protocols and parameters used to instantiate the cryptographic building blocks in the Muckle++ cryptosystem.

Building Blocks (see Note S1, Supporting Information)	Algorithm/Protocol	Parameters
KEM	Ephemeral elliptic curve Diffie-Hellman key exchange[47]	Curve25519[46]
PQ-KEM	CRYSTALS Kyber[41]	$k = 3$
PRF	HKDF[44]	256-bit keys
QPRF	HKDF xored with quantum key material	256-bit keys
IMAC (see Note S4, Supporting Information)	Poly1305-AES[45] combined with quantum key material	256-bit AES keys
DSA	Elliptic curve digital signature algorithm (ECDSA)[48]	Elliptic curve NIST P-256 and SHA-256 hash function
PQ-DSA	Falcon[42]	$(n = 512, q = 12289)$
QKD	T12[43]	1-Mbit keys, $\epsilon < 10^{-10}$

QPRF are performed on the server. A common daemon enables the communication between the main software and the different hardware cores.

2.4. Platform Authentication

FPGA-based PUFs are used for authentication of the physical FPGA devices, providing an additional layer of security. Prior to storing the Muckle++ secret state in memory after a key exchange, it is first masked by the PUF response. This PUF response is based on the unique manufacturing variations of the physical hardware device itself, and can be viewed as an intrinsic digital fingerprint of the device. Therefore, in order to subsequently unmask the secret state to generate the next session key, the code must be run on physically the same hardware device in order to generate the required PUF response. This ensures that the underlying FPGA, where the QKD processing occurs, has not been modified, providing additional protection for the generated keys and binding the exact FPGA device to the server. A symmetric encryption scheme is used to mask the state, with the specific instantiation being AES[49] in Galois/counter mode (GCM) with a 256-bit key derived from the PUF response. The PUF is a modified version of that proposed in Ref. [50], with a majority vote of 3 and a $[23, 12, 7]_2$ binary Golay code used for error correction. Both the majority vote and the algorithm are performed in hardware and abstracted away from the Muckle++ software.

2.5. Performance and Hybrid Key Generation

Following Figure 3, the software implementing the Muckle++ protocol was run on a pair of commercial servers connected via Ethernet to FPGAs. The FPGAs contained intellectual property (IP) cores for QKD processing, Kyber encapsulation and decapsulation and PUF operations. The resource requirements for the

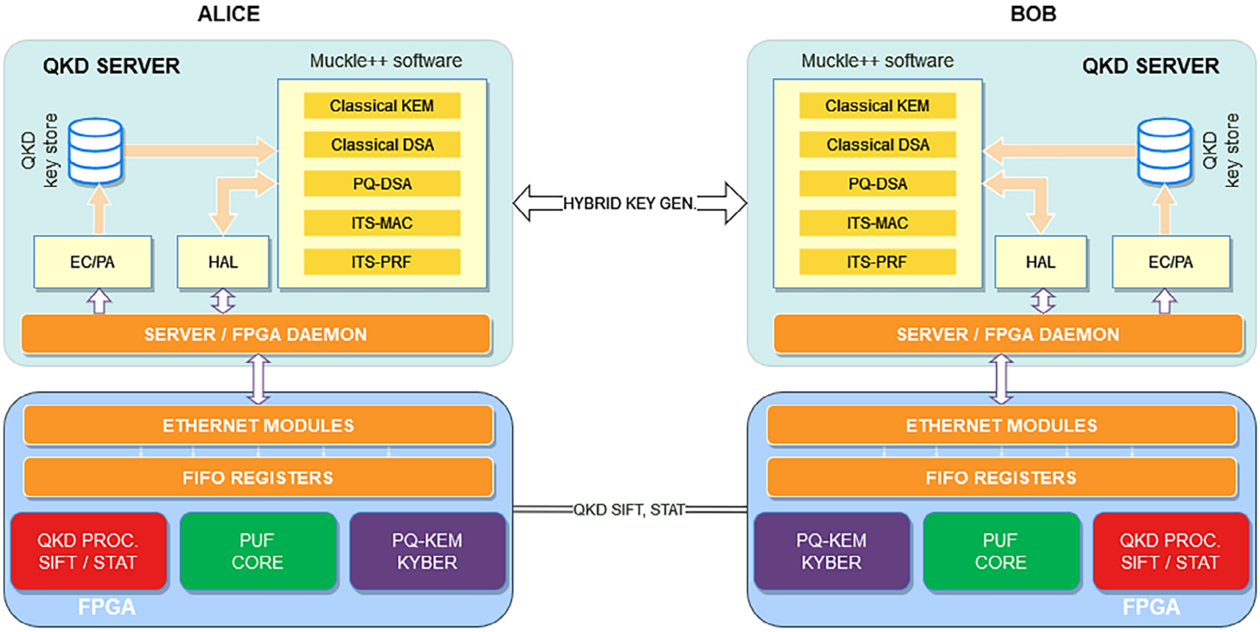

Figure 3. System architecture. The cryptosystem is deployed on a commercial QKD system prototype. The main software runs on the QKD servers where the classical primitives, the PQ-DSA, the MACs, and PRFs are executed alongwith the QKD error correction and privacy amplification. A hardware implementation of the PQ-KEM and the PUF was designed to execute them on the same FPGA that performs the core QKD functions.

Table 2. CRYSTALS-Kyber and PUF FPGA resource requirements.

Resource	Kyber	PUF
LUT	17339	4347
LUTRAM	2962	0
FF	7060	5404
BRAM	8.5	0
DSP	43	0

Table 3. CRYSTALS-Kyber Performance at 35 MHz.

	Latency		Operations
	(cycles)	[μs]	per second
Load Secret Key	609	17.4	57471
Encapsulation	19458	555.94	1798
Decapsulation	27746	792.74	1261

Kyber and PUF IP cores are given in **Table 2**, with the speed of the hardware Kyber implementation given in **Table 3**. The required time to readout the PUF response is minimal, with < 200 clock cycles required including error correction.

To generate the QKD keys, the privacy amplification (PA) algorithm is executed on 96 blocks of error corrected (EC) keys (≈ 100 Mb of key data) to avoid important finite size effects. With a PA data throughput > 100Mb/s, the time needed to generate a secure QKD key is limited by the time needed to accumulate 96 blocks of EC data, and is directly related to the rate-distance limit. 1 Mb-long secure QKD keys are reliably saved in the final key store. From there a set of keys are transferred to a dedicated key store for the hybrid cryptosystem demonstration.

Running the entire system to continuously generate fresh key material led to stable operation, with one hybrid-quantum-safe key per second and autonomous run-times > 18 h, as shown in **Figure 4**.

3. Conclusion and Future Work

There is a clear and urgent need to strengthen all digital communications through the addition of quantum safe cryptographic solutions, preventing a catastrophic loss of privacy in a future world where large quantum computers exist. Our work is an effort to ease the transition of the conventional security solutions used today to quantum safe alternates. We present Muckle++, a novel, provably quantum secure, authenticated key exchange protocol deriving its security from combining QKD, PQC and currently used public-key cryptography. This hybrid approach is appealing as it helps maintain inter-operability during migration, ensures forward security in the case of accidental key leakage, and caters for different adversarial strengths (both quantum and classical). Muckle++ augments an earlier presented HAKE protocol, by excluding the need to pre-share keys and instead rely on quantum-resistant digital signature schemes for authentication.

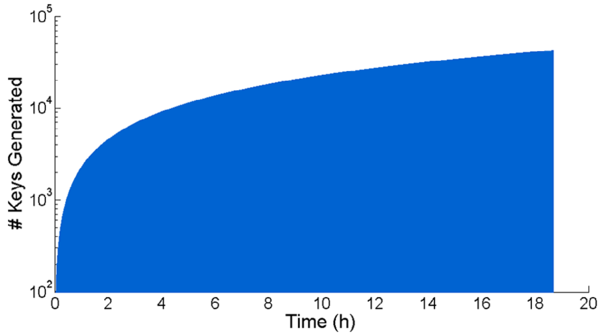

Figure 4. Hybrid key generation (log scale). Keys were generated steadily over more than 18 h.

The feasibility of Muckle++ was successfully tested through the seamless integration of a server running the protocol software, a commercially available QKD system, and hardware acceleration of the PQ-KEM (CRYSTALS-Kyber) on an FPGA. An additional layer of security was provided through the use of a PUF, to ensure that the secret state cannot be accessed unless the correct physical FPGA device is present. The prototype was successfully tested for a steady performance of hybrid key generation for several days without failures.

Our work aims to pave the way for future endeavors exploiting quantum and post-quantum technologies into a wider and viable quantum-safe security solutions for real-world operation. We are eager to extend the work in several directions. First regarding the implementation security, it would be important to undertake a vulnerability analysis of the physical security of integrated PQC-QKD designs including side channel analysis attacks and/or fault attacks. Extending the key generation throughput could be achieved by further leveraging a full hardware implementation of the entire protocol, taking advantage of FPGA-based QKD post-processing. Finally, investigating a greater range of potential use-case scenarios and applications would provide useful contributions to on-going standardization activities.

4. Experimental Section

Kyber Hardware Implementation: In order to accelerate the generation of shared keys, unused FPGA logic was utilized to implement the encapsulation and decapsulation functionality of the Kyber PQ-KEM algorithm. While this hardware Kyber core ran at 35 MHz, it could run at clock speeds up to 80 MHz. Build flags were added to the Muckle++ software to allow execution of Kyber instructions in hardware or software. Implementing Kyber required a SHA-3 ip core capable of supporting both SHA3 hashing and SHAKE functionality. This generic core could be reused to reduce resource requirements where additional building blocks were implemented in hardware.

Runtime: While the full system operated with continuous runtime ≃ 20 h, disabling the PUF let the system run for several days before it was shut down without error. Closer inspection revealed that further optimization of the PUF response reconstruction would prevent error in unmasking of the Muckle++ secret state and lead to continuous runtimes. In particular, the PUF error correction capability was designed for an error probability that was too high given continuous operation. Decreasing the error probability of the PUF to ensure correct operation for longer time periods could be achieved through the use of a different error correction scheme, or through the use of concatenated codes. This would largely involve trading off FPGA resources (slices/registers) and execution time for increased robustness. Optimizing the PUF and increasing the hybrid key generation throughput will be in scope for future works.

NIST Competition: During the preparation of this manuscript, NIST announced their endorsement of CRYSTALS-Kyber and Falcon, which were chosen in this work to instantiate the PQ-KEM and PQ-DSA, respectfully, among future standard quantum resistant algorithms. A standardization process on their implementation has now started and will be completed within the next 2 years.

Related Works: After completion of this manuscript, we became aware of a related work ref. [51] that also aims at extending the work by Dowling et al. [39] although without the emphasis on information theoretic security as it is devised in the current work.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

This work was supported by the Innovate UK funded project Agile Quantum Safe Communications (AQuaSec) and the EPSRC Quantum Communications Hub (EP/T001011/1). Authors L.G. and C.C. carried out the research described in this paper at Royal Holloway, University of London.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

L.G. and C.C. developed the software and derived the security analysis. N.H., A.K., C.R., and J.G. developed the FPGA hardware acceleration firmware including the puf engine. J.G. and J.N. developed the hardware interface layer between the software and the QKD FPGA with input from N.H. and T.P.. T.P. deployed the project on the QKD system with input from L.G., N.H., J.G., and J.N. and ran the experiment. L.G., T.P., N.H., and A.K. wrote the manuscript with comments from the other authors. A.J.S., C.C., and M.O.N. supervised the project.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

hybrid authenticated cryptosystem, information theoretic security, post-quantum cryptography, quantum cryptography, quantum key distribution

Received: September 13, 2023
Revised: November 21, 2023
Published online: February 18, 2024

[1] R. Cramer, V. Shoup, *SIAM J. Comput.* **2003**, *33*, 167.

[2] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, et al., *Nature* **2019**, *574*, 505.

[3] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Huo, L. Li, N. Li, S. Li, Y. Li, F. Liang, C. Lin, J. Lin, H. Qian, D. Qiao, H. Rong, H. Su, L. Sun, et al., *Phys. Rev. Lett.* **2021**, *127*, 180501.

[4] J. M. Arrazola, V. Bergholm, K. Brádler, T. R. Bromley, M. J. Collins, I. Dhand, A. Fumagalli, T. Gerrits, A. Goussev, L. G. Helt, J. Hundal, T. Isacsson, R. B. Israel, J. Izaac, S. Jahangiri, R. Janik, N. Killoran, S. P. Kumar, J. Lavoie, A. E. Lita, D. H. Mahler, M. Menotti, B. Morrison, S. W. Nam, L. Neuhaus, H. Y. Qi, N. Quesada, A. Repingon, K. K. Sabapathy, M. Schuld, et al., *Nature* **2021**, *591*, 54.

[5] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, A. E. Lita, T. Gerrits, S. W. Nam, V. D. Vaidya, M. Menotti, I. Dhand, Z. Vernon, N. Quesada, J. Lavoie, *Nature* **2022**, *606*, 75.

[6] X. Xue, M. Russ, N. Samkharadze, B. Undseth, A. Sammak, G. Scappucci, L. M. K. Vandersypen, *Nature* **2022**, *601*, 343.

[7] A. Noiri, K. Takeda, T. Nakajima, T. Kobayashi, A. Sammak, G. Scappucci, S. Tarucha, *Nature* **2022**, *601*, 338.

[8] M. T. Mądzik, S. Asaad, A. Youssry, B. Joecker, K. M. Rudinger, E. Nielsen, K. C. Young, T. J. Proctor, A. D. Baczewski, A. Laucht, V. Schmitt, F. E. Hudson, K. M. Itoh, A. M. Jakob, B. C. Johnson, D. N. Jamieson, A. S. Dzurak, C. Ferrie, R. Blume-Kohout, A. Morello, *Nature* **2022**, *601*, 348.

[9] P. W. Shor, in *Proc. 35th Annual Symposium on Foundations of Computer Science*, IEEE, Piscataway, NJ **1994**, pp. 124–134.

[10] L. K. Grover, in Proc. 28th Annual ACM Symposium on Theory of Computing, STOC '96. ACM, New York, NY, USA **1996**, pp. 212–219.

[11] N. Gisin, G. Ribordy, W. Tittel, H. Zbinden, *Rev. Mod. Phys.* **2002**, *74*, 145.

[12] N. T. Islam, C. C. W. Lim, C. Cahall, J. Kim, D. J. Gauthier, *Sci. Adv.* **2017**, *3*, e1701491.

[13] Z. Yuan, A. Murakami, M. Kujiraoka, M. Lucamarini, Y. Tanizawa, H. Sato, A. J. Shields, A. Plews, R. Takahashi, K. Doi, W. Tam, A. W. Sharpe, A. R. Dixon, E. Lavelle, J. F. Dynes, *J. Lightwave Technol.* **2018**, *36*, 3427.

[14] F. Grünenfelder, A. Boaron, G. V. Resta, M. Perrenoud, D. Rusca, C. Barreiro, R. Houlmann, R. Sax, L. Stasi, S. El-Khoury, E. Hänggi, N. Bosshard, F. Bussières, H. Zbinden, *Nat. Photonics* **2023**, *17*, 422.

[15] W. Li, L. Zhang, H. Tan, Y. Lu, S.-K. Liao, J. Huang, H. Li, Z. Wang, H.-K. Mao, B. Yan, Q. Li, Y. Liu, Q. Zhang, C.-Z. Peng, L. You, F. Xu, J.-W. Pan, *Nat. Photonics* **2023**, *17*, 416.

[16] M. Lucamarini, Z. Yuan, J. F. Dynes, A. J. Shields, *Nature* **2018**, *557*, 400.

[17] M. Pittaluga, M. Minder, M. Lucamarini, M. Sanzaro, R. I. Woodward, M.-J. Li, Z. Yuan, A. J. Shields, *Nat. Photonics* **2021**, *15*, 530.

[18] Y. Liu, W.-J. Zhang, C. Jiang, J.-P. Chen, C. Zhang, W.-X. Pan, D. Ma, H. Dong, J.-M. Xiong, C.-J. Zhang, H. Li, R.-C. Wang, J. Wu, T.-Y. Chen, L. You, X.-B. Wang, Q. Zhang, J.-W. Pan, *Phys. Rev. Lett.* **2023**, *130*, 210801.

[19] Y.-M. Xie, Y.-S. Lu, C.-X. Weng, X.-Y. Cao, Z.-Y. Jia, Y. Bao, Y. Wang, Y. Fu, H.-L. Yin, Z.-B. Chen, *PRX Quantum* **2022**, *3*, 020315.

[20] P. Zeng, H. Zhou, W. Wu, X. Ma, *Nat. Commun.* **2022**, *13*, 3903.

[21] L. Zhou, J. Lin, Y.-M. Xie, Y.-S. Lu, Y. Jing, H.-L. Yin, Z. Yuan, *Phys. Rev. Lett.* **2023**, *130*, 250801.

[22] H.-T. Zhu, Y. Huang, H. Liu, P. Zeng, M. Zou, Y. Dai, S. Tang, H. Li, L. You, Z. Wang, Y.-A. Chen, X. Ma, T.-Y. Chen, J.-W. Pan, *Phys. Rev. Lett.* **2023**, *130*, 030801.

[23] P. Sibson, C. Erven, M. Godfrey, S. Miki, T. Yamashita, M. Fujiwara, M. Sasaki, H. Terai, M. Tanner, C. Natarajan, R. Hadfield, J. O'Brien, M. Thompson, *Nat. Commun.* **2017**, *8*, 13984.

[24] D. Bunandar, A. Lentine, C. Lee, H. Cai, C. M. Long, N. Boynton, N. Martinez, C. DeRose, C. Chen, M. Grein, D. Trotter, A. Starbuck, A. Pomerene, S. Hamilton, F. N. C. Wong, R. Camacho, P. Davids, J. Urayama, D. Englund, *Phys. Rev. X* **2018**, *8*, 021009.
[25] T. K. Paraïso, T. Roger, D. G. Marangon, I. de Marco, M. Sanzaro, R. I. Woodward, J. F. Dynes, Z. Yuan, A. J. Shields, *Nat. Photonics* **2021**, *15*, 850.
[26] R. Sax, A. Boaron, G. Boso, S. Atzeni, A. Crespi, F. Grünenfelder, D. Rusca, A. Al-Saadi, D. Bronzi, S. Kupijai, H. Rhee, R. Osellame, H. Zbinden, *Photonics Res.* **2023**, *11*, 1007.
[27] K. Wei, X. Hu, Y. Du, X. Hua, Z. Zhao, Y. Chen, C. Huang, X. Xiao, *Photonics Res.* **2023**, *11*, 1364.
[28] J. F. Dynes, A. Wonfor, W. Tam, A. W. Sharpe, R. Takahashi, M. Lucamarini, A. Plews, Z. Yuan, A. R. Dixon, J. Cho, Y. Tanizawa, J.-P. Elbers, H. Greißer, I. White, R. Penty, A. J. Shields, *npj Quantum Inf.* **2019**, *5*, 101.
[29] Y.-A. Chen, Q. Zhang, T.-Y. Chen, W.-Q. Cai, S.-K. Liao, J. Zhang, K. Chen, J. Yin, J.-G. Ren, Z. Chen, S.-L. Han, Q. Yu, K. Liang, F. Zhou, X. Yuan, M.-S. Zhao, T.-Y. Wang, X. Jiang, L. Zhang, W.-Y. Liu, Y. Li, Q. Shen, Y. Cao, C.-Y. Lu, R. Shu, J.-Y. Wang, L. Li, N.-L. Liu, F. Xu, X.-B. Wang, et al., *Nature* **2021**, *589*, 214.
[30] D. Moody, R. Perlner, in *Lecture Notes in Computer Science*, **2016**.
[31] National Institute of Standards and Technology, Post-Quantum Cryptography: Selected Algorithms 2022, https://csrc.nist.gov/projects/post-quantum-cryptography/selected-algorithms-2022, (accessed: 2022).
[32] R. Barnes, K. Bhargavan, B. Lipp, C. Wood, RFC, **2020**, 9180. https://www.rfc-editor.org/rfc/rfc9180.html.
[33] E. Crockett, C. Paquin, D. Stebila, Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH, Cryptology ePrint Archive, Report, **2019**, 858.
[34] D. Stebila, Internet Engineering Task Force (IETF) draft, **2020**.
[35] J. Xu, Y. Gao, H. Lim, Cryptology ePrint Archive, Report, **2020**, 763.
[36] L.-J. Wang, K.-Y. Zhang, J.-Y. Wang, J. Cheng, Y.-H. Yang, S.-B. Tang, D. Yan, Y.-L. Tang, Z. Liu, Y. Yu, Q. Zhang, J.-W. Pan, *npj Quantum Inf.* **2021**, *7*, 1.
[37] Y.-H. Yang, P.-Y. Li, S.-Z. Ma, X.-C. Qian, K.-Y. Zhang, L.-J. Wang, W.-L. Zhang, F. Zhou, S.-B. Tang, J.-Y. Wang, Y. Yu, Q. Zhang, J.-W. Pan, *Opt. Express* **2021**, *29*, 25859.
[38] W. Beullens, Cryptology ePrint Archive, Report, **2022**, 214.
[39] B. Dowling, T. B. Hansen, K. G. Paterson, in *International Conference on Post-Quantum Cryptography*, Springer, Berlin, Heidelberg **2020**, pp. 483–502.
[40] K. Cohn-Gordon, C. Cremers, L. Garratt, in *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, IEEE, Piscataway, NJ **2016**, pp. 164–178.
[41] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, D. Stehlé, in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, Piscataway NJ **2018**, pp. 353–367.
[42] T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, Z. Zhang, Technical report, National Institute of Standards and Technology, **2018**, https://csrc.nist.gov/presentations/2018/falcon.
[43] M. Lucamarini, K. Patel, J. F. Dynes, B. Fröhlich, A. W. Sharpe, A. R. Dixon, Z. Yuan, R. Penty, A. J. Shields, *Opt. Express* **2013**, *21*, 24550.
[44] H. Krawczyk, Cryptology ePrint Archive, **2010**, 264, https://eprint.iacr.org/2010/264.
[45] D. J. Bernstein, Lecture Notes in Computer Science, **2005**, 3557, https://link.springer.com/chapter/10.1007/11502760_3
[46] D. J. Bernstein, Lecture Notes in Computer Science, **2006**, 3958, https://link.springer.com/chapter/10.1007/11745853_14.
[47] V. S. Miller, Lecture Notes in Computer Science, **1985**, 218, https://link.springer.com/chapter/10.1007/3-540-39799-X_31.
[48] D. Johnson, A. Menezes, S. Vanstone, *IJIS* **2001**, *1*, 36.
[49] National Institute of Standards and Technology, Federal Information Processing Standards Publications **2001**, 197.
[50] C. Gu, M. O'Neill, in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, IEEE, Piscataway NJ **2015**, pp. 934–937.
[51] S. Bruckner, S. Ramacher, C. Striecks, Muckle+: End-to-end hybrid authenticated key exchanges, Cryptology ePrint Archive, Paper 2023/653, **2023**, https://eprint.iacr.org/2023/653.