IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 January 2024, accepted 29 January 2024, date of publication 8 February 2024, date of current version 16 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3364520

RESEARCH ARTICLE

Hybrid Keys in Practice: Combining Classical, Quantum and Post-Quantum Cryptography

SARA RICCI, PATRIK DOBIAS, LUKAS MALINA, JAN HAJNY, AND PETR JEDLICKA
Department of Telecommunications, Brno University of Technology, 61600 Brno, Czech Republic

Corresponding author: Lukas Malina (malina@vut.cz)

This work was supported by the Ministry of the Interior, Czech Republic, under Grant VJ01010008.

ABSTRACT Currently, with the threat of quantum computer attacks, the idea of combining several same-type primitives has reemerged. This is also the case for cryptographic keys where a hybrid quantum key exchange combination allows for preserving the security guarantees of pre-quantum schemes and achieving quantum resistance of post-quantum schemes. In this article, we present a concrete 3-key combiner system implemented on a Field Programmable Gate Arrays (FPGA) platform. Our system involves a pre-quantum Key EXchange scheme (KEX), a post-quantum key encapsulation mechanism, and a Quantum Key Distribution (QKD) algorithm. The proposed 3-key combiner is proven to be secure in the quantum standard model and it is INDistinguishable under a Chosen-Ciphertext Attack (IND-CCA). Our combiner can run in small FPGA platforms due to its relatively low resources usage. In particular, the key combiner without QKD is able to output up to 1 624 keys per second and the key combiner with QKD is able to output up to 9.2 keys per second.

INDEX TERMS Authentication, cryptography, key establishment, post-quantum cryptography, security, quantum key distribution (QKD), dual-PRF, key combiner.

I. INTRODUCTION

The idea of combining several same-type primitives, so that the resulting scheme is secure as long as one of the components remains secure, goes back to Even and Goldreich [1]. With the threat of quantum computer attacks, this concept has reemerged. Classical cryptographic methods, based on the hardness of mathematical assumptions such as Integer Factorization (IF) problem, Discrete Logarithm Problem (EC), and Elliptic Curve (EC)DLP, have long provided the foundation for securing communication and information. With the advent of quantum computers, the run of quantum-based algorithms could be used to break traditional public-key cryptography schemes. For instance, Shor's algorithm [2] allows attackers to solve DLP and IF problem and, therefore, breaks the most commonly used cryptographic protocols such as RSA, Diffie-Hellman scheme, and EC Digital Signature Algorithm (DSA), that are based on the aforementioned mathematical assumptions. Hybrid schemes permit mitigating the risk of quantum attacks and preserving common security guarantees by combining classically secure and quantum-resistant schemes. For instance, the National Cybersecurity Agency of France (ANSSI) [3] considers the role of hybridization in the cryptographic security crucial and mandatory for the next phases. An efficient way to achieve hybridization involves a scheme combiner where parallelization of the combined schemes can be provided. In this way, the slower Key EXchange (KEX) or Key Exchange Mechanism (KEM) scheme bounds the key generation speed. The combiners are designed to be fast and achieve an equal level of security to the involved schemes. Reinforcing this idea, the Federal Office for Information Security - Germany (BSI) report [4] suggests not using post-quantum cryptography in isolation, as it has not been equally well studied. At the same time, the report emphasizes the need to switch to quantum-safe schemes by combining post- and pre-quantum schemes. In fact, the usage of new algorithms can be a long and difficult process, where backward compatibility has to be maintained without introducing the risk of downgrade attacks [5]. Moreover, there is uncertainty about the hardness

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

of post-quantum assumption where new (even classical) attacks may show them to be vulnerable and, furthermore, the parameters choices not yet reliable [6]. Therefore, we are in a situation where there is a demand to protect assets from quantum computer threats, but not sufficient confidence in the security of post-quantum schemes. A hybrid approach facilitates the smooth transition to Post-Quantum (PQ) cryptography, retaining the time-tested trust on pre-quantum algorithms while incorporating the quantum resistance of PQ schemes [7].

Recently, several ways to combine either KEX protocols or KEMs into a secure hybrid system have been proposed [5], [7], [8], [9], [10]. A study from the European Network and Information Security Agency (ENISA) [11] suggests deploying post-quantum cryptography as an extra layer to pre-quantum cryptography. This is in accordance with Agence nationale de la sécurité des systèmes d'information (ANSSI) and the Federal Office for Information Security (BSI) specifications of hibridization. Specifically, ENISA proposes that the selected pre- and post-quantum scheme output keys are combined to generate the encryption key used, for example, as input in the AES-256-GCM scheme. It is important to notice that the Elliptic-Curve Diffie-Hellman (ECDH) is considered a candidate for the pre-quantum scheme whereas the Kyber scheme is one of the possible post-quantum KEMs. Furthermore, the first Internet Engineering Task Force (IETF) drafts have appeared that define and discuss hybrid approaches in various protocols, e.g., hybrid approach terminology,[1] hybrid key exchange methods in Transport Layer Security (TLS) 1.3[2] [12], and a combiner function for hybrid key encapsulation mechanisms.[3]

It is worth noting that Quantum Key Distribution (QKD) promises information-theoretic security [13], whereas classical and post-quantum schemes security is based on the intractability of selected computationally hard problems. This means that QKD can provide long-term security and that does not impose limits on the adversary's computational power [14]. Therefore, since they are based on different principles, QKD and post-quantum cryptography can be viewed as complementary methods that can be both deployed [4]. Accordingly, a hybrid system involving pre-quantum, post-quantum, and quantum schemes guarantees a smooth transaction to PQ cryptography, where QKD and post-quantum complement each other to strengthen the system. Nevertheless, it is essential to consider limitations of QKD, such as its limited range and associated costs..

A. CONTRIBUTION

Seeking to contribute to the knowledge gaps, we will focus on both a quantum-secure theoretically-proven combination of keys and its practical deployment. Specifically, we seek to answer the following Research Questions (RQ): RQ1) How securely and effectively can be combined 3 different key establishment methods to get a hybrid key? RQ2) How can a 3-key combiner be implemented in practice and how many hardware resources will be required at FPGA?

Specifically, we present a 3-key combiner system involving a pre-quantum KEX, a post-post quantum KEM, and a Quantum Key Distribution (QKD) algorithm. In particular, our work provides the following contributions:

1) **Extensive analysis of existing KEX and KEM combiners.** Our scheme requires combining 1 KEM and 2 KEXs. Therefore, we sought solutions in both domains and surveyed several works that provide theoretical and practical designs of either KEX or KEM combiners.
2) **Extension of the dual-PRF combiner to work with three keys as input.** One of the possible KEX candidates is dual-PRF [15] which is designed to have either 2 KEM output keys [5] or 2 KEX output keys [8] as input. Our system stems from the proposed dual-PRF, taking 2 KEX keys and 1 KEM key as input.
3) **Security proof of our system directly derived from the dual-PRF**, i.e., the proposed 3-key combiner is proven to be secure in the quantum standard model and it is indistinguishable under Chosen-Ciphertext Attack (IND-CCA).
4) **Concrete implementation of the proposed 3-key combiner in a Field Programmable Gate Arrays (FPGA) platform.** Most of the existing combiners remain theoretically described whereas we present a concrete deployment of our 3-key combiner. Our combiner can run in small FPGA platforms due to its relatively low resource usage (i.e., 4 532 LUT and 3 363 FF). In particular, the key combiner without QKD is able to output up to 1 624 keys per second and the key combiner with QKD is able to output up to 9.2 keys per second.

Furthermore, the dual-PRF is closely related to the key schedule used in TLS 1.3 [15] allowing our system to be smoothly integrated into TLS 1.3 and increasing its applicability. Our system is designed to be agile, i.e., easy and fast replacement of cryptographic components, and accessible, i.e., the quantum component can be easily not deployed. In fact, our combiner can be switched to either use or not the QKD algorithm. This allows the FPGA platform to be deployed with and without the quantum component and makes the solution more accessible from a market point of view. Note that Points 1, 2, and 3 help to answer to RQ1, while Point 4 and the above paragraph to RQ2.

The rest of this article is organized as follows. Section II extensively reviews the state-of-the-art for KEMs and KEXs combiners. Section III discusses some preliminaries. Section IV states the motivation of our selection and the design goal, presents the basic structure of the proposed key combiner, and lists the selected parameters and the implementation practical aspects. Section V provides the security analysis of the scheme. Section VI reports the experimental

[1] https://datatracker.ietf.org/doc/draft-driscoll-pqt-hybrid-terminology/
[2] https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/
[3] https://datatracker.ietf.org/doc/draft-ounsworth-cfrg-kem-combiners/

TABLE 1. Existing KEM combiners. "SM" states for standard model, "ROM" for random oracle model, "Q-SM" and "Q-ROM" for quantum-variant of SM and ROM, and "Impl." for implementation.

Combiner	Security Model	Model	Impl.
G1) Hashing [9]	IND-CCA	ROM	No
G2) XOR-then-PRF [9]	IND-CCA	ROM	No
B1) XOR-then-MAC [5]	Q-IND-CCA	Q-SM	No
G3) BLOCK-then-PRF [9]	IND-CCA	ROM	No
G4) PRF-then-XOR [9]	IND-CCA	SM	No
G5) opt PRF-then-XOR [9]	IND-CCA	SM	No
G6) Split-key PRF [9]	IND-CCA	SM	No
B2) dual-PRF [5]	Q-IND-CCA	Q-SM	No
B3) Nested dual-PRF [5]	Q-IND-CCA	Q-SM	No
ECDH & SIKE [7]	-	-	Yes
FO-wise primitives [10]	IND-CCA	Q-ROM	No

results. Section VII discusses potential use cases of the proposed hybrid system and sums up some open problems and potential extensions of our key combiner. The final section contains the conclusions.

II. RELATED WORK

The possibility to combine more KEMs has been independently explored in the so-called KEM combiners. Table 1 provides an overview of the existing KEM combiners along with their main features. Note that only one article offers a concrete implementation of a proposed KEM combiner, nevertheless, no security proofs are given. Giacon et al. [9] proposed KEM combiners with the main purpose of developing generic methods that allow combining more KEMs in a way that security of any implies security of their combination. Their proposals focus on minimizing overhead with respect to the deployed KEMs and, therefore, achieving their easy adoption. The proposed solutions vary based on security requirements, i.e., whether they prioritize IND-CCA or INDistinguishability under Chosen-Plaintext Attacks (IND-CPA), and performance characteristics. In particular, their constructions combine hashing, Pseudo-Random Function (PRF), and XOR-ing of key and ciphertext pairs. A total of 6 combiners are introduced: G1) a simple hashing of the KEMs keys and ciphertexts; G2) an optimization of the previous one, namely XOR-then-PRF, involving the XOR-ing of the keys and the concatenation of the ciphertexts; G3) a BLOCK-then-PRF, where BLOCK stands for a secure block cipher, i.e., a chain of block cipher invocations is applied with inputs 0 and keys derived from the KEMs; G4) a PRF-then-XOR, where each KEM key and the concatenation of all ciphertexts pass through a PRF and the results are then XOR-ed; G5) a slight improvement of the previous one with the reduction of the ciphertext input; at last G6) a combiner based on a split-key pseudorandom function. Their combiners are proven to be secure either in the standard model or in the random oracle model, taking two IND-CCA KEM and outputing another IND-CCA KEM. In our case, we would like to combine one KEM and two KEX protocols. Note that the KEXs have not encapsulated keys and, therefore, the aforementioned combiners do not permit a straightforward combination of pre-shared keys, e.g., KEX schemes.

Furthermore, within the cryptography literature, there have been several articles [5], [7], [12] exploring the coupling of pre-quantum and post-quantum cryptography through KEM combiners. Bindel et al. [5] focused on hybrid KEM combiners and authenticated key exchange. They propose a new KEM combiner, introduce new security models for KEM combiners that account for quantum-capable adversaries and, through the new definitions, analyze the security of three KEM combiners. Specifically, the analyzed methods consist of B1) an XOR-then-MAC combiner derived from XOR-then-PRF proposed in [9]; B2) a dual-PRF which follows the TLS 1.3 construction proposed in [15]; and B3) a nested dual-PRF developed from [16]. Accordingly, these combiners are quantum-secure if at least one of the selected KEM is quantum resistant [5]. While previous works focused on the theoretical design of robust combiners, others considered the applicability and practicability these hybrid solutions. Stebila et al. [12] propose an evaluation of the applicability of the dual-PRF combiner proposed by [9]. They list the scope, goals, benefits, and drawbacks that a combiner should have in the post-quantum era. In particular, the dual-PRF was chosen due to its features: 1) backward compatibility, i.e., endpoints and middle-boxes need to remain compatible with clients and servers even if they are not aware of the hybrid combiner; 2) high performance, i.e. the use of hybrid key exchange should not be prohibitive on the performance terms; 3) low latency, i.e., the use of hybrid KEXs should not significantly increase the connection latency; 4) no extra round trips in the negotiation of the KEX; and 5) minimal duplicate information in the negotiation communication. Aviram et al. [8] focuses on the practical (implementation-wise) construction of the dual-PRF combiner [12]. In this work, a proven-secure method to combine KEXs is presented by deploying the dual-PRF. They also compare several key-combiners currently used in practice. Finally, Poettering and Rastikian [17] explore the use of KEMs beyond their typical application in constructing public key encryption and secure channels. Notably, they employ combiners suggested by [5] and [9], supporting the relevance of a dual-PRF combiner.

Furthermore, Azarderakhsh et al. [7] and Huguenin-Dumittan and Vaudenay [10] take another direction. In [7], the authors propose a quantum-secure combiner that couples ECDH and Supersingular Isogeny Key Encapsulation (SIKE) [18] protocols. It is important to notice that SIKE is an isogeny-based protocol and, therefore, it runs on elliptic curves as well as ECDH protocol making the merging more effective. They implemented their proposal on a FPGA platform. In [10], the authors propose a solution that does not require extra primitives such as special types of PRFs or MACs. They focus on bypassing the intermediate KEM constructions by involving much higher-level primitives, i.e. Fujisaki-Okamoto (FO) transform-like primitives, with respect to the previous proposals. However, their work is proven to be secure only in the Quantum-Random Oracle Model (Q-ROM).

On the other hand, several articles focused on providing post-quantum security proofs for existing KEX protocols, such as Signal [19] and TLS. In case of TLS, the literature includes draft standards [12], [20], theoretical articles [5], [21], [22] and industry experiments [23], [24]. From the industry perspective, Google and Cloudfare jointly experimented with the integration of two post-quantum KEXs, namely isogeny-based SIKE and lattice-based HRSS, in TLS [23]. However, all these works rely on the fact that the key-combiner needs to be modeled as a dual-PRF [8].

In a study conducted by Giron et al. [25], various hybrid KEX combiners are surveyed and classified on their efficiency and security. They suggest that the compared combiners have acceptable performance for important applications, making them fundamental for secure network communications. From a security perspective, they report that the PRF-based combiners are the only one proven secure in the standard model against a quantum adversary as we also mentioned. They also highlight that, conventionally, most studies use only two KEXs, emphasizing the need to explore the potential of hybrid designs involving more than two algorithms. Finally, they mention that there has been no exploration of "PQ-PQ combiners," where two or more post-quantum algorithms are combined in a new scheme. This area is left as an open research problem.

III. PRELIMINARIES

In this section, at first, we outline the used notation. Then, we recall the definition of ϵ-regular, PRF, dual-PRF, KEM combiner and briefly review the primitives on which our protocol is based. From now on, the symbol ":" means "such that", "$|x|$" is the bitlength of x and "$\|$" denotes the concatenation of two binary strings. A secure hash function is denoted as $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^\kappa$, where κ is a security parameter. We write $x \xleftarrow{\$} U$ when x is sampled uniformly at random from U.

A. PRELIMINARY DEFINITIONS

In this section, we review the definitions of ϵ-Regular function, PRF, dual-PRF, and computational extractor [8]. These definitions are necessary to understand our construction and prove its security.

Definition 1 (ϵ-Regular): Let U_ℓ be the uniform distribution over n bits. A function $\mathcal{H} : \{0,1\}^n \rightarrow \{0,1\}^m$ is ϵ-regular if

$$\frac{1}{2}\sum_x |\Pr[\mathcal{H}(U_n) = x] - \Pr[U_m = x]| \leq \epsilon$$

Definition 2 (Pseudorandom Function (PRF)): Let $\mathcal{F} = \{f_\lambda : K_\lambda \times X_\lambda \rightarrow Y_\lambda\}_{\lambda\in\mathbb{N}}$ be a function family ensemble and let $\mathcal{G} = \{G_\lambda\}_{\lambda\in\mathbb{N}} = \{g_\lambda : X_\lambda \rightarrow Y_\lambda\}_{\lambda\in\mathbb{N}}$ be the set of all functions mapping $X\lambda$ to Y_λ. We say that $\mathcal{F}$ is a (t,ϵ)-pseudorandom function family (PRF) if no t-size adversary can distinguish $f_\lambda(k,\cdot)$ from $g_\lambda(\cdot)$ for k chosen uniformly from the set K_λ and where g_λ is chosen uniformly from G_λ with probability better than ϵ. More precisely, for an adversary $A = \{A_\lambda\}_{\lambda\in\mathbb{N}}$, we define

$$Adv^{prf}_{f,A}(\lambda) = \Pr_{k\leftarrow K_\lambda}[A_\lambda^{f_\lambda(k,\cdot)}(1^\lambda) = 1] - \Pr_{g_\lambda\leftarrow G_\lambda}[A_\lambda^{g_\lambda(\cdot)}(1^\lambda) = 1].$$

The family $\mathcal{F}$ is a (t,ϵ)-PRF if for every size-s adversary A it holds that $Adv^{prf}_{f,A}(\lambda) \leq \epsilon(\lambda)$.

Definition 3 (Dual-PRF): Let $\mathcal{F} = \{f_\lambda : X^0_\lambda \times X^1_\lambda \rightarrow Y_\lambda\}_{\lambda\in\mathbb{N}}$ be an ensemble of functions. Let $\bar{\mathcal{F}} = \{\bar{f}_\lambda : X^0_\lambda \times X^1_\lambda \rightarrow Y_\lambda\}_{\lambda\in\mathbb{N}}$ be the ensemble of dual functions such that each $\bar{f} \in \bar{\mathcal{F}}$ is defined as $\bar{f}(y,x) = f(x,y)$ We say $\mathcal{F}$ is a (t,ϵ)-dual-PRF if both $\mathcal{F}$ and $\bar{\mathcal{F}}$ are (t,ϵ)-PRFs.

Definition 4 (Computational Extractor): We say that a function $F : K_\lambda \rightarrow Y_\lambda$ is a (t,ϵ)-computational extractor with respect to a (leakage) function $g : K_\lambda \rightarrow \{0,1\}^m$ if for any circuit A of size at most t it holds that

$$|\Pr_{k\leftarrow K_\lambda}[A(g(k), F(k))] - \Pr_{k\leftarrow K_\lambda, R\leftarrow Y_\lambda}[A(g(k), R] | \leq \epsilon$$

Note that a computational extractor takes as input a random k in K_λ and outputs an element y in Y_λ that is computationally indistinguishable from a random element in Y_λ. Moreover, the extractor is secure even in the presence of a leakage (one-way) function g, i.e., given $g(k)$, an adversary $\mathcal{A}$ cannot distinguish between the output of the extractor and a random element in Y_λ.

B. HASH-BASED MESSAGE AUTHENTICATION CODE

Hash-based Message Authentication Code (HMAC) [26] is a cryptographic hash-function-based PRF with

$$\mathtt{HMAC}(K,m) = \mathcal{H}(K' \oplus opad \parallel \mathcal{H}(K' \oplus ipad \parallel m)),$$

where m is some message, $opad$ is the byte-string $0 \times 5C_L$, and $ipad$ is the byte-string $0 \times 36_L$. Note that L represents the block size of $\mathcal{H}$ and $0 \times 5C_L$ means $0 \times 5C$ repeated L times. Moreover, $K' = \mathcal{H}(K)$ if $|K| > L$ and $K' = K$ otherwise. Proofs of HMAC security are given in [27] and [28].

C. KEY ENCAPSULATION MECHANISM (KEM) COMBINER

In this section, we briefly review the structure of a KEM. A KEM can be split into three algorithms $K = (K.Gen, K.Enc, K.Dec)$, where

- $(pk, sk) \leftarrow \mathtt{K.Gen}(1^\kappa)$: on the input of the system security parameter κ, the algorithm generates a public key pk and a secret key sk.
- $(k, c) \leftarrow \mathtt{K.Enc}(pk)$: on the input of the public key pk, the algorithm produces a session key k and a ciphertext c.
- $(k, \perp) \leftarrow \mathtt{K.Dec}(sk, c)$: on the input of the secret key sk and the ciphertext c, the algorithm outputs either a session key k or a rejection value $\perp$.

A KEM combiner is a mechanism specifying how a set of existing KEMs can be joined to generate a new KEM, i.e. a new session key k. Note that a KEM combiner should be at least as secure as any of its input KEMs. Normally, a parallel combination is suggested for KEM, i.e., one component per input KEM [9]. This follows the IND-CCA model that is considered as a standard security notion for KEM protocols [29] and stronger than IND-CPA.

D. A PRACTICAL DUAL-PRF DESIGN

Aviram et al. [8] provides a provable-secure practical construction for a dual-PRF. Informally a dual-PRF(k, c) is a PRF where at least one input value between k and c needs to be random. Moreover, dual-PRF(k, c) is equal to dual-PRF(c, k). In order to have a dual-PRF, a combiner needs the involvement of a practical one-way function g and a computational extractor F.

Let exp be an expansion factor, and B_i fixed public values, where $|B_i|$ is equal to $\mathcal{H}$ block size. Let x be an input value, we define

$$\mathcal{H}_i(x) = \mathcal{H}(x||B_i). \tag{1}$$

Therefore, Algorithm 1 gives a way on how to compute g. Note that Algorithm 1, Line 1 is applied only if the size of K is bigger than $\mathcal{H}$ block size.

Algorithm 1 $g(K, exp)$

```
1: $k_1, k_2, \ldots, k_n \leftarrow \text{SplitToBlocks}(K)$
2: $u \leftarrow \text{Empty}$
3: for $j = 1$ to n do
4:     $u \leftarrow u \parallel \mathcal{H}_1(k_j) \parallel \cdots \parallel \mathcal{H}_{exp-1}(k_j)$
5: end for
6: return $u$
```

Moreover, a computational extractor $F(K)$ with respect to g can be designed as follows:

$$F(K, salt) = \texttt{HMAC}(salt, K),$$

where the input value K is used as a message and the $salt$ as a key. Note that the default value of $salt$ is set to be all zero bytes.

Algorithm 2 shows a construction of a dual-PRF (please see [8] for more details). Let K_i be the output key of KEM$_i$, exp the expansion factor for g, and $salt$ the salt needed for HMAC, then

Algorithm 2 Dual-PRF$(K_1, K_2, exp, salt)$

```
1: $k_1 \leftarrow F(K_1, salt), k_2 \leftarrow F(K_2, salt)$
2: $u_1 \leftarrow g(K_1, exp), u_2 \leftarrow g(K_2, exp)$
3: $K \leftarrow \mathcal{H}(\texttt{HMAC}(k_1, 1 \parallel u_2) \oplus \texttt{HMAC}(k_2, 2 \parallel u_1))$
4: return $K$
```

IV. PROPOSED ARCHITECTURE

In this section, we present the cryptographic architecture of our hybrid key establishment system. We also describe our design goals, selection of cryptographic primitives and related parameters.

A. MOTIVATION FOR THE SELECTION

It is worth mentioning that the main issue with Random Oracles (ROs) is that it is very difficult to build a truly "random" oracle. Canetti et al. [30] proved that there exist signature and encryption schemes that are secure in the ROM, but for which any implementation of the RO, i.e., hash functions, results in insecure schemes. Therefore, being a secure hash function does not imply that this function is a RO. This leads us to move our attention to combiners that are secure in the Standard Model (SM) rather than in the RO Model (ROM).

Moreover, our combiner has hybrid inputs not only in the sense that considers quantum-resistant and pre-quantum schemes but also that combines schemes with different structures, i.e., one KEM and two KEX protocols. Note that Aviram et al. [31] prove that concatenating secrets and hashing them is, in general, vulnerable to the (Authenticated Post Office Protocol) APOP attack [32]. Accordingly, the Concatenate-then-Hashing (i.e. G1) in Table 1) is a secure KEM combiner, but this does not imply that is also a secure KEX combiner.

Therefore, we need to consider hybrid combiners that are proven secure for KEX protocols in the SM model. In Table 1, only three KEM combiners are proven secure in the Quantum SM (Q-SM), and, among them, the dual-PRF combiner [12] presents a construction proven secure for KEXs as inputs [8]. Finally, the dual-PRF architecture directly stems from the TLS 1.3 KEX combiner, allowing it to be integrated into TLS 1.3 more smoothly and increasing its applicability.

B. DESIGN GOALS

During the creation of cryptographic architecture, we considered mainly these design goals:

- **Security**: our approach combines classically secure and quantum-resistant schemes retaining the time-tested trust on pre-quantum primitives and mitigating the risk of quantum attacks. Moreover, we deployed only cryptographic primitives that have no known vulnerabilities and considered secure for medium-term future.
- **Post-quantum security**: our combiner achieves IND-CCA and Quantum IND-CCA, i.e., IND-CCA against quantum adversaries.
- **Compliance**: we considered only primitives that comply with the recommendations of renowned authorities, such as (National Institute of Standards and Technology) NIST [33], ANSSI [3], BSI [4], and National Cyber Security Center (NCSC) [34].
- **High performance**: primitives and their composition were chosen considering low computational and space complexity so that implementation on constrained devices is possible and practical. In particular, we were aiming at small space for the implementation on FPGAs, i.e., a solution with less than 100k LUTs and 100 FFs for all cryptographic components at the platform.
- **Cryptographic flexibility (agility)**: cryptographic flexibility (in terms of BSI: agility) means easy and fast replacement of cryptographic components in case some weaknesses arise. In case any algorithm becomes

vulnerable, it must be easy to replace it without affecting the remaining components in the resulting system.

C. SELECTED PRIMITIVES

We list all necessary components of the hybrid key establishment system. Our system is composed of:

- **Classical Cryptography Key Source**: we selected the Elliptic-Curve Diffie Hellman (ECDH) protocol [35] as it represents today's standard for classical key establishment schemes. ECDH is recommended by all major authorities, including NIST, ANSSI, BSI, and NCSC.
- **Quantum Key Distribution (QKD) Key Source**: the ID Quantique CLAVIS 3 QKD system [36] has been selected as the source of keys generated and agreed by quantum devices. The selection is rather pragmatic (as this QKD system is present in our lab [37]) and motivated by the fact, that ID Quantique is one of very few companies currently delivering full-fledged commercial QKD systems to customers.
- **PQC Key Source**: CRYSTALS-Kyber [38] was selected as the post-quantum source of keys, as this algorithm is, at the time of writing this article, the only key-establishment mechanisms approved for the standardization by NIST [33] and recommended by security agencies, including the National Security Agency (NSA) [39] in the USA. We used the Kyber.AKE version that ensures mutual authentication of communicating parties. This version requires 3x Encaps and 3x Decaps operations. Further, both parties have to use long-term pre-shared public keys (side A knows B's public key, side B knows A's public key). The Kyber.AKE version is more complex and secure than a simple non-authenticated Kyber version.
- **Key Derivation Function**: the current standard for hash function, the SHA-3 based on the Keccak algorithm, was selected as the fundamental construction for the key-derivation function. SHA-3 is recommended by all major authorities, including NIST, ANSSI, BSI and NCSC.
- **Symmetric Block Cipher**: we selected the today's de-facto standard for fast encryption of high amounts of data, the Advanced Encryption Standard (AES) algorithm. The algorithm, in its 256 bits variant, is considered quantum safe by major authorities, e.g. NSA [39]. We used the algorithm in the Galois-Counter Mode (GCM) which provides both confidentiality and integrity of transferred data.
- **Key Management System**: the Key Management System (KMS) provides the logics for the derivation and updating of encryption and decryption keys. Its functionality is given by the requirements of relevant cryptographic components, particularly of the GCM mode of AES and its requirements on periodic key updates after a certain amount of data encrypted.

D. CRYPTOGRAPHIC ARCHITECTURE

In our system, we need to combine a post-quantum secure KEM, namely Kyber, with two KEXs, namely ECDH and QKD. Our solution follows the architecture presented by Aviram et al. [8], where their proposal practically combines two KEXs using a dual-PRF as proposed by Bindel et al. [5].

The proposed construction needs the involvement of a practical one-way function g which is sketched in Algorithm 3. Please see Section III for more details. In our case, $\mathcal{H}$ is SHA3-512 that needs input in bitrate instead of blocksize. Therefore, we slightly change the g construction to work with bitrates. Let K be a session key generated by either a KEM or a KEX scheme that we want to combine and pp some public values related to K generation.

Algorithm 3 $g(K, pp)$

```
$l \leftarrow$ bitrate of $\mathcal{H}$
$pp_1, pp_2, pp_3 \leftarrow$ SplitToBits$(pp)$ with $|pp_i| = l$
$u \leftarrow u \parallel \mathcal{H}(K \parallel pp_1) \parallel \mathcal{H}(K \parallel pp_2) \parallel \mathcal{H}(K \parallel pp_3)$
**return** $u$
```

Let K_1, K_2, and K_3 be the session keys generated by Kyber, ECDH, and QKD schemes, respectively. Moreover, let c_1 be the Kyber ciphertext, and p_2 and p_3 the public parameters of ECDH and QKD schemes, respectively. Our construction uses c_1, p_2 and p_3 as fixed public values for expansion function g. The *exp* value shown in Algorithm 1 is fixed to 3 as suggested by the authors [8]. Note that the bitrate of SHA3-512 is 576 bits and therefore, we need that $|pp| \geq 3 * 676 = 1728$ bits. This causes to SHA3-512 to be "expanding" from 512 bits to 1536. The proposed hybrid combiner, namely `3-key Combiner`, is depicted in Algorithm 4. Note that our `3-key Combiner` is a component of the key management system that produces the private values for AES.

In our case, we need to combine three keys. Therefore, we modified the input of `HMAC` (Algorithm 4, Line 5, highlighted in red) to involve results computed from each key. For instance, in order to retrieve $u_2 \parallel u_3 \parallel 1$ in `HMAC`$(k_1, u_2 \parallel u_3 \parallel 1)$, an attacker would need to know both K_2 and K_3 and, accordingly, break both ECDH and QKD. This technique was already proposed by Bindel et al. [5]. Furthermore, for performance purposes, we changed the order of the concatenated elements in `HMAC` (Algorithm 4, Lines 5 and 7). For instance, $1 \parallel u_2 \parallel u_3$ becomes $u_2 \parallel u_3 \parallel 1$. We refer to Section VI for more details.

Note that `HMAC`$(salt, K)$ is a computational extractor with respect to g, where default value *salt* is set to be all zero bytes. In fact, HMAC is proven to be a good extractor if the compression function underlying the hash function is a PRF and the dual of the compression function is a good extractor [8], [40].

Our `3-key Combiner` can be easily switched to either use or not use the QKD algorithm. This allows the FPGA platform to be used with and without the quantum component

Algorithm 4 `3-Key Combiner`$(K_1, K_2, K_3, c_1, p_2, p_3, salt)$

```
$k_1 \leftarrow \mathtt{HMAC}(salt, K_1)$, $u_1 \leftarrow g(K_1, c_1)$
$k_2 \leftarrow \mathtt{HMAC}(salt, K_2)$, $u_2 \leftarrow g(K_2, p_2)$
if QKD then
    $k_3 \leftarrow \mathtt{HMAC}(salt, K_3)$, $u_3 \leftarrow g(K_3, p_3)$
    $K \leftarrow \mathcal{H}(\mathtt{HMAC}(k_1, u_2 \| u_3 \| 1) \oplus \mathtt{HMAC}(k_2, u_1 \| u_3 \| 2) \oplus \mathtt{HMAC}(k_3, u_1 \| u_2 \| 3))$
else
    $K \leftarrow \mathcal{H}(\mathtt{HMAC}(k_1, u_2 \| 1) \oplus \mathtt{HMAC}(k_2, u_1 \| 2))$
end if
return $K$
```

and makes the solution more accessible from a market point of view.

In Figure 1, we illustrate the high-level cryptographic architecture of our quantum-safe encryption system designed for high-speed network interfaces. Our system architecture is deployed on both sides, labelled as Side A and Side B, utilizing programmable FPGA network cards equipped with four 100 GbE interfaces, collectively referred as the trust zone. Each FGPA card allows a pre- and a post-quantum encryption capabilities, employing high-speed encryption, ECDH and Authenticated Key Exchange (AKE)-Kyber exchange methods, and a dedicated key combiner. Note that Algorithm 4 describes the keys combination which has XOR-ing, hashing and HMAC as main primitives. The trust zone interfaces serve distinct purposes: one interface connects the internal network, managing upstream and downstream data for two logical sessions, while another interfaces with the external network (Side B in a peer-to-peer configuration). The third interface establishes a connection to the external source responsible for quantum-generated keys (a QKD-KEM system) that uses its own optical connection lines between Side A and Side B.

All key-generation blocks (ECDH, Kyber, and QKD) independently generate 256-bit keys. These keys are then fed into the `3-key combiner`. The 3-key combiner combines the keys and outputs the 256 bits hybrid encryption and decryption keys. These hybrid keys, along with the randomly generated 32-bit salt, serves as inputs for the AES encrypted and decrypted blocks. These blocks, emplying AES in the GCM mode, are used to encrypt (resp. decrypt) traffic on the LAN (resp. WAN) interfaces. Note that the 32-bit salt values are also generated by the 3-key combiner.

E. SELECTION OF PARAMETERS, IMPLEMENTATION ASPECTS

Our architecture is not only theoretical, it was also used in a concrete implementation of our quantum-safe encryptor on the FPGA platform. For the practical implementation, the concrete parameters for cryptographic components and key management approach had to be selected. To balance the security and speed of our implementation, we selected NIST Security Strength Category 3 [41] as the baseline. The selected Level 3 is suggested for standard security applications according to [42].

TABLE 2. Values used in the `3-Key Combiner`. "Param" states for parameter.

Param	Component	value	bitsize
K_1	Kyber-768	generated key	256
c_1	Kyber-768	ciphertext	8 074
K_2	ECDH	generated key	256
p_2	ECDH	$Q_a \parallel Q_b \parallel Q$	3 426
K_3	QKD	generated key	256
p_3	QKD	keyID ‖ SHAKE("Logtail qkd")	1728

The selected parameters are:

- Classical Cryptography: we selected ECDH-512 (precisely the sect571k1 curve), as it exceeds Level 3 according to [43].
- Post-Quantum Cryptography: we selected KYBER-768, as it directly complies with NIST Security Level 3.
- Quantum Cryptography: we used the default QKD protocol for key distribution: the Coherent One Way (COW) protocol for establishing 256-bit keys.
- Block Cipher: we selected the highest security parameter of AES, i.e. 256-bit keys.

For the `3-key Combiner` system, we selected SHA3-512 as the hash function due to its security strength [21] and bitrate. We have to consider that SHA3-512 will be used as a component of g and `HMAC`. With this choice, function g needs that pp has a bitlength of (at least) $3*576 = 1728$ bits. We refer to Section IV-D for more details.

Table 2 lists the input values information used in our `3-key Combiner`, where c_1, p_2, and p_3 are the public values needed in g. Therefore, the first 1728 bits of c_1, p_2, and p_3 are Algorithm 3 inputs. In ECDH, Q_a, Q_b and Q present the communication points that SIDE A and SIDE B exchange, and the generator of the curve, respectively. Their size is 1142 bits each. Note that keyID is the value of size 256 bits transmitted with the generated key. In case of QKD, three possible solutions as input values were found:

- **Option 1: keyID ‖ SHAKE(certificate)**, where 256 bits are from keyID and the remaining 1472 bits are generated by applying SHAKE-512 to the QKD certificate that each device owns. However, certificates are static and less universal.
- **Option 2: keyID ‖ SHAKE(session settings info)**, where the input of SHAKE changes to the information about session settings. Note that these data are unique (i.e., dynamic), and with proper length between 1500 and 1800 bits per key. However, they depend on the length of the generated key.
- **Option 3: keyID ‖ SHAKE(service channel messages/"Logtail qkd")**. This option has larger spaces than session settings in Option 2, and the data should be also dynamic since they contain encoded information unique for session settings.

For our setting, Option 3 presents more compliant characteristics, i.e., dynamicity and right bit length.

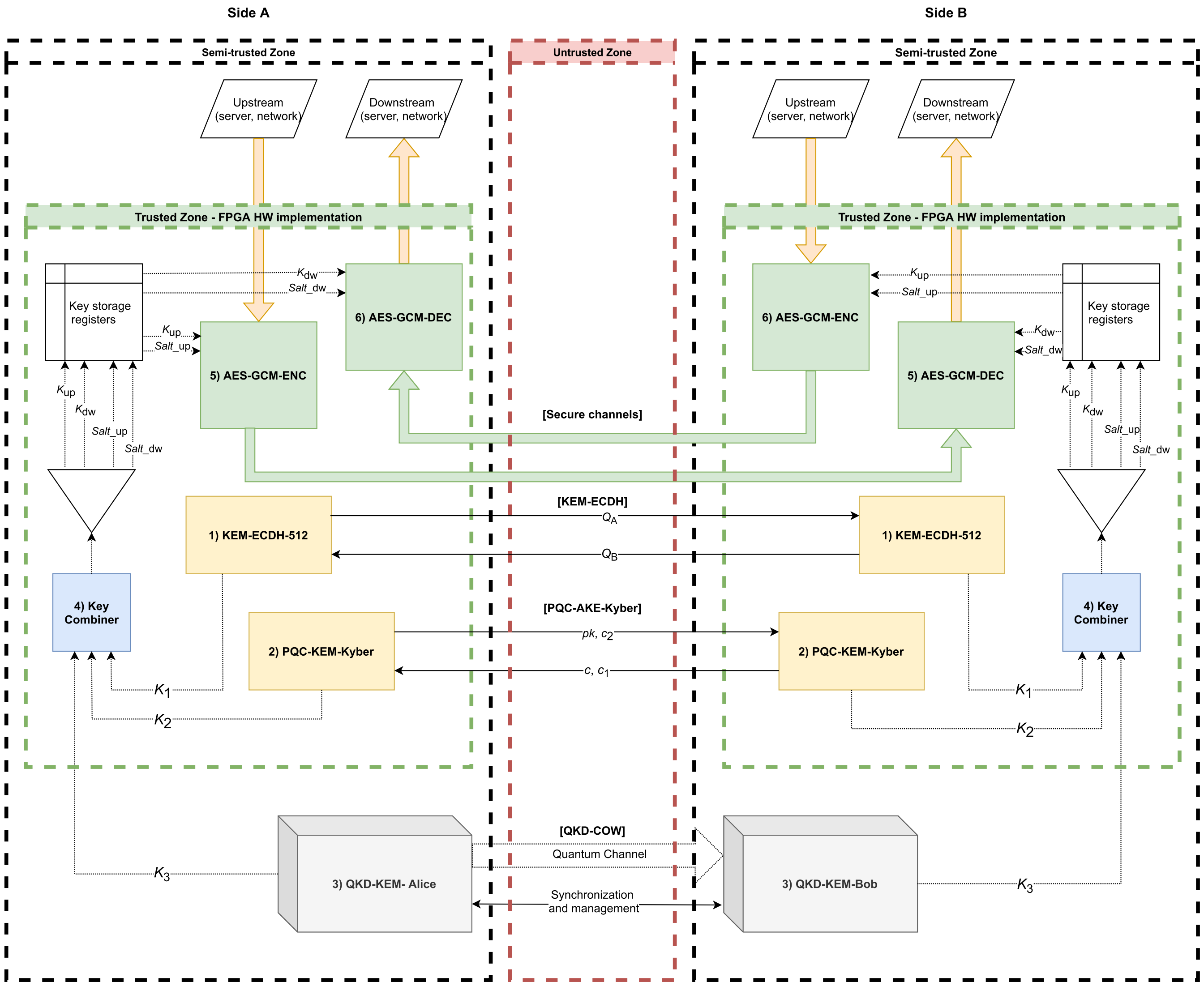


FIGURE 1. Basic system architecture.

The 3-key Combiner system must be able to update encryption and decryption keys before their lifetime expires. The maximum key lifetime is given by the maximum amount of data that are encrypted using the same key and salt. For the GCM mode, 2^{32} messages can be encrypted using the same key and salt [44]. Considering using IP packets of length 1500 bytes as messages and the line speed of 100 Gbps, the expected key lifetime is around 500 seconds,[4] a bit above 8 minutes. To have some security margin, we selected to update the encryption and decryption keys every minute. Furthermore, our system is easily adaptable either to the involvement of a quantum component or not. Therefore, depending on the user demand, we have that the 3-key Combiner uses:

1) Two-methods: Kyber + ECDH.
2) Three-methods: Kyber + ECDH + QKD.

V. SECURITY ANALYSIS

Bellare and Lysyanskaya [45] proved the IND-CCA security of dual-PRFs in the SM. Moreover, Aviram et al. [8] present a concrete construction of a dual-PRF. The IND-CCA security of our 3-key Combiner follows the same proof technique of [8]. In other words, we prove that our 3-key Combiner outputs a key K that is indistinguishable from random. We omit the public value inputs c_1, p_2, p_3 in g for brevity. Note that c_1, p_2, p_3 are public values and, therefore, an adversary $\mathcal{A}$ knows them.

Lemma 1: Assume that g is an injective (t, ϵ)-one way function, that F is a (t, ϵ)-computational-extractor with respect to g, that HMAC *is a standard (t, ϵ)-PRF, and that $\mathcal{H}$ is ϵ-regular. Then, the following construction is a $(t, 3\epsilon)$-dual-PRF. On input K_1, K_2 compute the following*

[4] $8 * 1500 * 2^{32}/10^1 1 = 515, 39$ s.

1) $k_1 \leftarrow F(K_1)$, $u_1 \leftarrow g(K_1)$
2) $k_2 \leftarrow F(K_2)$, $u_2 \leftarrow g(K_2)$
3) *Output:* $K \leftarrow \mathcal{H}(\texttt{HMAC}(k_1, u_2 \parallel 1) \oplus \texttt{HMAC}(k_2, u_1 \parallel 2))$

Proof: This construction is equivalent to the one proposed by Aviram et al. Therefore, the proof follows straightforward from Theorem 1, [8]. □

Lemma 2: Assume that g is an injective (t, ϵ)-one way function, that F is a (t, ϵ)-computational-extractor with respect to g, that HMAC *is a standard (t, ϵ)-PRF, and that $\mathcal{H}$ is ϵ-regular. Then, the following construction is a $(t, 3\epsilon)$-dual-PRF. On input K_1, K_2, K_3 compute the following*

1) $k_1 \leftarrow F(K_1)$, $u_1 \leftarrow g(K_1)$
2) $k_2 \leftarrow F(K_2)$, $u_2 \leftarrow g(K_2)$
3) $k_3 \leftarrow F(K_3)$, $u_3 \leftarrow g(K_3)$,
4) *Output:* $K \leftarrow \mathcal{H}(\texttt{HMAC}(k_1, u_2 \parallel u_3 \parallel 1) \oplus \texttt{HMAC}(k_2, u_1 \parallel u_3 \parallel 2) \oplus \texttt{HMAC}(k_3, u_1 \parallel u_2 \parallel 3))$.

Proof: We need to show that if K_1 is uniform at random, and K_2 and K_3 are malicious, then an adversary $\mathcal{A}$ is not able to distinguish the output from a uniform value. This needs to be valid also for the cases 1) K_2 uniform, and K_2 and K_3 malicious, and 2) K_3 uniform, and K_1 and K_3 malicious which follow the same proof structure.

We assume that K_1 is chosen uniformly at random and $\mathcal{A}$ performs queries while choosing values for K_2 and K_3. Let ϵ_0 and ϵ_i be the advantage $\mathcal{A}$ in the original construction, namely Construction 0, and in Construction i, respectively.

Let $\mathcal{A}$ be an adversary that runs in time t. For each construction the difference from the previous construction is marked in red. We bound the differences between ϵ_i and $\epsilon_i + 1$ for $i = 0, 1, 2, 3$.

Construction 1:

1) $k_1 \xleftarrow{\$} U$, $u_1 \leftarrow g(K_1)$,
2) $k_2 \leftarrow F(K_2)$, $u_2 \leftarrow g(K_2)$
3) $k_3 \leftarrow F(K_3)$, $u_3 \leftarrow g(K_3)$,
4) Output: $K \leftarrow \mathcal{H}(\texttt{HMAC}(k_1, u_2 \parallel u_3 \parallel 1) \oplus \texttt{HMAC}(k_2, u_1 \parallel u_3 \parallel 2) \oplus \texttt{HMAC}(k_3, u_1 \parallel u_2 \parallel 3))$.

Claim: $\mid \epsilon_0 - \epsilon_1 \mid \leq \epsilon$

This follows directly from the definition of (t, ϵ)-computational extractor with respect to the function g. For g, $\mathcal{A}$ gets as input $u_1 \leftarrow g(K_1)$ and k_1, where k_1 is either random, i.e., $k_1 \xleftarrow{\$} U$, or $k_1 \leftarrow F(K_1)$. Therefore, $\mathcal{A}$ can simulate the rest:

1) sample K_2 and K_3,
2) compute $k_2 \leftarrow F(K_2)$, $u_2 \leftarrow g(K_2)$
3) compute $k_3 \leftarrow F(K_3)$, $u_3 \leftarrow g(K_3)$,
4) compute $K \leftarrow \mathcal{H}(\texttt{HMAC}(k_1, u_2 \parallel u_3 \parallel 1) \oplus \texttt{HMAC}(k_2, u_1 \parallel u_3 \parallel 2) \oplus \texttt{HMAC}(k_3, u_1 \parallel u_2 \parallel 3))$.

Accordingly, the probability of distinguishing between Construction 0 and Construction 1 is equivalent to the probability of distinguishing in the PRF game (Definition 4), i.e., ϵ.

Construction 2:

1) $k_1 \xleftarrow{\$} U$, $u_1 \leftarrow g(K_1)$,
2) $k_2 \leftarrow F(K_2)$, $u_2 \leftarrow g(K_2)$
3) $k_3 \leftarrow F(K_3)$, $u_3 \leftarrow g(K_3)$,
4) Output: $K \leftarrow \mathcal{H}(U \oplus \texttt{HMAC}(k_2, u_1 \parallel u_3 \parallel 2) \oplus \texttt{HMAC}(k_3, u_1 \parallel u_2 \parallel 3))$.

Claim: $\mid \epsilon_1 - \epsilon_2 \mid \leq \epsilon$

We can assume without loss of generality that $\mathcal{A}$ performs unique queries $q_1, \ldots, q_t$. Since g is injective and $q_i \neq q_j$ for all $i, j = 1, \ldots, t$ and $i \neq j$, then $y_1 = g(q_1), \ldots, y_t = g(q_t)$ are distinct.

The claim follows from the fact that HMAC is a PRF. Note that k_1 can be input only for HMAC. Therefore, since HMAC is queried only with distinct inputs, the probability of distinguishing between Construction 1 and Construction 2 is equivalent to the probability of distinguishing in the PRF game (Definition 4).

Construction 3:

1) $k_1 \xleftarrow{\$} U$, $u_1 \leftarrow g(K_1)$,
2) $k_2 \leftarrow F(K_2)$, $u_2 \leftarrow g(K_2)$
3) $k_3 \leftarrow F(K_3)$, $u_3 \leftarrow g(K_3)$,
4) Output: $K \leftarrow U$.

Claim: $\mid \epsilon_2 - \epsilon_3 \mid = 0$

Since U is uniform at random for each query of $\mathcal{A}$, we have that U is distributed exactly as $U \oplus \texttt{HMAC}(k_2, 2 \parallel u_1 \parallel u_3) \oplus \texttt{HMAC}(k_3, 3 \parallel u_1 \parallel u_2)$.

Construction 4:

1) $k_1 \xleftarrow{\$} U$, $u_1 \leftarrow g(K_1)$,
2) $k_2 \leftarrow F(K_2)$, $u_2 \leftarrow g(K_2)$
3) $k_3 \leftarrow F(K_3)$, $u_3 \leftarrow g(K_3)$,
4) Output: $K \leftarrow \mathcal{H}(U)$.

Claim: $\mid \epsilon_3 - \epsilon_4 \mid \leq \epsilon$

This follows directly from the fact that $\mathcal{H}$ is ϵ-regular (Definition 1).

Therefore, the distance between Construction O and Construction 4 is

$$\mid \epsilon_0 - \epsilon_4 \mid = \mid (\epsilon_0 - \epsilon_1) + (\epsilon_1 - \epsilon_2) + (\epsilon_2 - \epsilon_3) + (\epsilon_3 - \epsilon_4) \mid \leq 3\epsilon.$$

Note that Construction 4 is a random oracle and, therefore, ϵ_4 is equal to 0. Accordingly, $\epsilon_0 \leq 3\epsilon$. □

Theorem 1: Algorithm 4 is a $(t, 3\epsilon)$-dual-PRF.

Proof: We demonstrate this theorem using Lemmas 1 and 2 that prove that both the combiners in our construction are dual-PRF with the right selection of primitives. To do so, we need to have that:

- g is an injective (t, ϵ)-one way function. This is proven in Section 5.1, [8].
- $\mathcal{H}$ is ϵ-regular. In our case, we consider SHA3-512 which is safe to consider ϵ-regular since if it is applied to uniformly random inputs, then it outputs (very close to) hash uniformly distributed hash digests.
- $\texttt{HMAC}(K, salt)$ is a (t, ϵ)-computational-extractor with respect to $g(K)$. This is proven in [40].
- HMAC is a standard (t, ϵ)-PRF. This is proven in [27].

□

Note that both HMAC and g involve SHA3-512 in out system. This also allows reducing the implemented primitives.

In Bindel et al. [5], the dual-PRF is proven to be a robust KEM combiner, i.e., the resulting KEM has the security of the strongest of the two input KEMs. Moreover, they show that their construction is IND-CCA secure in the post-quantum setting if `HMAC` is a post-quantum secure dual PRF, $\mathcal{H}$ is a post-quantum secure PRF, and at least one of the two KEMs is post-quantum IND-CCA secure. Note that this property holds for our system since Kyber is post-quantum IND-CCA secure.

VI. IMPLEMENTATION RESULTS

In this section, we provide the implementation details and benchmarking of our key combiner.

A. IMPLEMENTATION DETAILS

The `3-key combiner` component was designed with the aim of low resource utilization and ease of incorporation with existing components for ECDH and Kyber. The architecture of the `3-Key combiner` is shown in Figure 2. This component accepts three different keys, each with a size of 256 bits. For the ciphertext and public parameters, 512-bit transactions are used. To optimize resource usage, the key combiner works with 64-bit transactions internally.

Initially, the 256-bit and 512-bit transactions are split into 64-bit transactions. Since the values u_1, u_2, and u_3 are used multiple times, they are computed and stored using FIFO. Subsequently, k_1 is computed, and the HMAC $(k_1, u_2 \parallel u_3 \parallel 1)$ is calculated. This approach eliminates the need to store k_1, as it can be directly fed from the HMAC output back to the input. The same process is repeated for k_2 and k_3. The resulting values are XORed together and passed through the SHA3-512 hash function. The output of this hash function is used as the final key.

To reduce the resource utilization further, we use only one Keccak component, which is shared between the HMAC and the practical one-way function g, described in Algorithm 3 to serve as postfixes. This modification allows SHA3-512 to absorb a 64-bit transaction of u_1, u_2, and u_3. In contrast, if we were to use these prefixes as intended, we would need to read 56 bits of u_i while reserving 8 bits for the next transaction, which would lead to increased resource usage.

To accommodate variants with and without QKD support, the component was implemented with a generic parameter. This allows for easy switching between those variants based on needs.

B. IMPLEMENTATION RESULTS

The implementation results of the key combiner and its components are presented in Table 5. The implementation was performed using Vivado, targeting the FPGA Virtex Ultrascale+ (xcvu9p-flgb2104-2-i). Table 3 shows available hardware resources and specifications of the Virtex UltraScale+ FPGA platform. Notably, Algorithm 4 results are presented as Key Combiner Component in Table 5. Moreover, Algorithm 3 is implemented as one primitive of Algorithm 4. Algorithm 3 functions as a hash component, and its results should correspond to those of SHA3-512, with potentially minor overhead for the final concatenation.

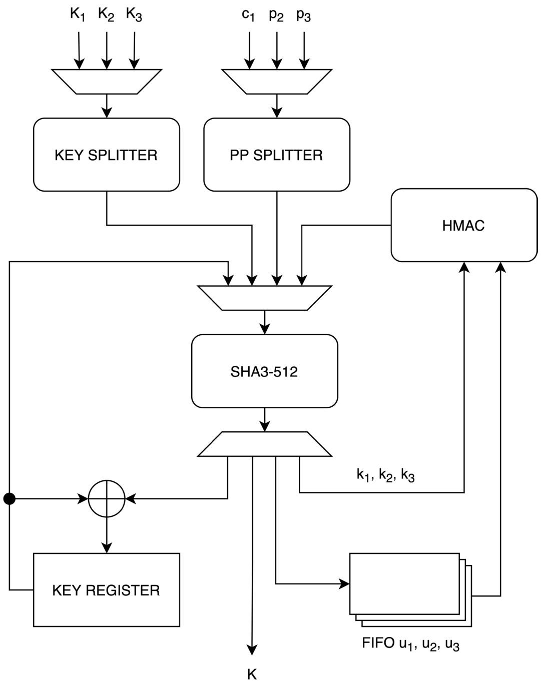


FIGURE 2. Key combiner component scheme.

TABLE 3. Hardware specification of Virtex UltraScale+. RAM and UltraRAM are given in MB.

LB	LUTs	FFs	RAM	UltraRAM
2586150	1182240	2364480	2160	960

The key combiner without QKD utilized 4471 Look-Up Tables (LUTs) and 3287 Flip-Flops (FFs), while the variant with QKD used 4532 LUTs and 3363 FFs. The increase in resource utilization with QKD is negligible. Moreover, when we use the combiner in our complete system, Vivado optimizes this difference out, leading to identical results for both variations.

Our combiner requires relatively low resources as shown in Table 5. This makes it feasible to utilize it on more constrained FPGAs, i.e., in small and medium platforms such as Artix-7. However, since we use the combiner with ECDH and Kyber schemes which require together 46464 LUTs, 59219 FFs, and 182 DSPs, a more powerful FPGA has to be deployed. We refer to [46] for a description of small, medium and large FPGA platforms.

After the implementation step, the maximum frequency reported for our key combiner without QKD was 399 MHz, whereas, for the key combiner with QKD, it was 388 MHz. The key combination process takes 1155 cycles for the variant without QKD and 2007 cycles for the variant with QKD support. Based on these values, the key combiner without QKD is able to output up to 345,454 keys per second and the key combiner with QKD is able to output up to 193,323 keys per second. In practice, these maximum key speed rates will be further reduced by the efficiency of all 3 KEX subsystems

TABLE 4. Measured average key rates with various QKD parameters.

QKD COW protocol	Route distance	Optical route loss (without attenuator)	Average Key rate
3-state COW	ca 7 km	2.1 dB	2684 bps
4-state COW	ca 7 km	2.1 dB	2365 bps
4-state COW	ca 2.2 km	6.6 dB	1230 bps

that generate the keys K_1, K_2, and K_3 as inputs for the key combiner.

All of these keys are computed in parallel. For K_1, we use Kyber with both parties authenticated key exchange. On the client side, it is necessary to do key generation, key encapsulation, and 2x key decapsulation, resulting in total 21,196 clock cycles. On the server side, 2x key encapsulation and key decapsulation are needed, resulting in total 15,920 clock cycles. For K_2, we use ECDH which requires 221,619 clock cycles. Since both run in parallel, the overall delay is determined by the slower scheme, which in this case is ECDH. For all these components combined, the reported frequency was 362 MHz. Based on this, the total delay for the key combination without QKD amounts to 222,774 cycles, which means generating 1,624 keys per second.

Table 4 shows experimentally obtained key rate results at our QKD CLAVIS 3 system deployed in various optical routes and settings. The results show how the type of the COW protocol, distance, and optical route loss (without added attenuators) can influence key rates. The longer distance and higher loss of the optical routes cause lower key rates. However, the manufacturer recommends deploying QKD systems on routes with a certain level of attenuation from 10 dB to 14 dB to prevent the malfunction of the QKD equipment. Further, the 4-state COW protocol produces slightly lower key rates than the 3-state COW protocol but 4-state COW promises higher robustness and security. Nevertheless, the efficiency and security of the COW variants are still open to research [47]. Finally, more parameters can affect quantum-bit error rate (QBER) and key rates, e.g., the types of connectors, their cleanness, visibility, line manipulation, and Raman noise, see more results in [48], [49], [50], and [51].

For the 4-state COW protocol in QKD, the average speed rate is 2365 bps at lines with less line loss (ca 2 dB). As we use 256 b keys, we can generate approximately 9.2 keys per second. Note that this is significantly slower than both ECDH key exchange and the Kyber scheme. Since these protocols run in parallel, QKD gives the maximum speed for the key combiner input values generation. Nevertheless, this total speed rate is still enough for 100 Gbps lines, see our calculations in Subsection IV-E. In order to temporally increase the total key speed rate with QKD, the system can preestablish a set of K_3 keys and stores them securely in FPGA during an initial stage.

We benchmarked the whole system (i.e., ECDH, Kyber, QKD, `3-key Combiner` and AES scheme) using the architecture presented in Figure 1 and we achieved the speed of 53.57 Gbps reported by iPerf3. It is worth noting that this speed did not utilize the system to its maximum capacity, as its theoretical maximum is approximately 100 Gbps. The discrepancy was attributed to the limitations of the testing server's speed.

TABLE 5. Resource utilization and power consumption of `3-key Combiner` and its components.

Component	LUT	FF	Power [W]
ECDH	26 625	24 090	2.505 / 2.022
Kyber	19 839	35 129	2.496 / 1.741
SHA3-512	2 901	1 702	2.514 / 0.643
HMAC	403	659	2.474 / 0.294
Key Combiner	4 471	3 287	2.480 / 0.746
Key Combiner with QKD	4 532	3 363	2.480 / 0.756
Total	67 464	79 777	2.557 / 5.468

The component's correctness was extensively tested using thousands of simulation runs using outputs of Python script implementing the key combiner based on hashlib[5] and hmac[6] libraries. Additionally, it was utilized for key establishment in the encryption project, further validating its correctness.

VII. DISCUSSION

In this section, we present a comparison of our system with existing combiners and we discuss potential use cases of the proposed hybrid system with its benefits. Secondly, we sum up some open problems and potential future extensions of the hybrid system.

A. COMPARISON WITH STATE-OF-THE-ART SOLUTIONS

In the state-of-the-art proposals,it is crucial to differentiate between Key Exchange Mechanism (KEM) combiners, Key Exchange (KEX) combiners - whether implemented without security proof, theoretically proven secure in the Standard Model (SM), Random Oracle Model (ROM), Quantum Standard Model (Q-SM), and Quantum Random Oracle Model (Q-ROM) - or those that lack practical implementations. Notably, among these, there exists only one KEX-KEM implementation based on synergy (lacking security proofs) [7] and one KEX combiner proven secure in the SM model with a practical implementation [8]. To our knowledge, our scheme is the only one combining KEXs and KEMs scheme and having both the theoretically-proven security in the Standard Model (MS) in the post-quantum setting and being practically implemented. This versatility extends to its application for combining either only KEMs, only KEXs, or both. Unlike [7], our proposal is not bound to a specific post-quantum class, being adaptable to any KEX or KEM with chosen security characteristics.

Its nature to be a 3-key combiner, integrating pre-quantum, post-quantum, and quantum schemes, ensures long-term security without limitations on the adversary's computational power, and a smooth transaction to PQ cryptography.

[5] https://pypi.org/project/hashlib/

[6] https://pypi.org/project/hmac/

Notably, the existing combiner considers only two dimension, i.e., pre-quantum and post-quantum. Moreover, our system provides the flexibility to utilize or bypass QKD algorithm, making it adaptable to various deployment scenarios and enhancing market accessibility.

From an efficiency perspective, our proposal aligns with the existing combiners designed to minimize overhead by employing mechanisms such as Pseudo-Random Function (PRF), dual-PRF, and XOR-ing. Notably, most existing schemes most existing schemes lack real-world implementations, leaving the choice of primitives to the reader and hindering direct comparisons. Conversely, Azarderakhsh et al. [7], 2-key (opposed to our 3-key) combiner, merging ECDH and SIKE, offers a distinct design not reliant on hash counts and XOR-ing mechanisms. This distinct approach introduces complexities that compromise a straightforward comparison with our scheme.

B. USE CASES OF HYBRID KEY ESTABLISHMENT SYSTEM

The deployment of a complex hybrid key establishment system integrating three different methods based on classic cryptography, PQC, and QKD can be suitable for peer high-speed connections requiring high-security assurances such as between governmental bodies, security institutions, critical infrastructure nodes, data centers, telecommunications operators, cloud service providers, or in the financial sector. The main benefit of the proposed hybrid system is its high-security resistance against three types of attacks. Attackers have to break three different methods (i.e., pre- and post-quantum cryptography, and quantum cryptography). Specifically, until one of the key generation methods remains unbroken, the `3-key Combiner` is IND-CCA secure. It is important to note that combinations of several proven secure cryptographic components are not necessarily secure [52]. However, since our `3-key Combiner` is a dual PRF, its security is theoretically proven once the combined schemes are IND-CCA secure as well [45]. We refer to Section V for more details. The need to deploy three types of attacks to break the system gives obvious practical benefits such as minimizing transition risks of new PQC methods and less tested QKD systems, and protection against various attacks such as the Store Now Decrypt Later (SNDL) attacks that could leverage quantum computers in the future and be used on current data. The system is also suitable for use cases having long lifespan requirements.

C. FUTURE EXTENSIONS AND OPEN PROBLEMS

In potential future extensions, the system could be redesigned also for multiple non-peer-to-peer connections in QKD networks working with more QKD nodes that could connect to each other. Extending the system for larger QKD networks may pose challenges in terms of key management and storage. Accordingly, scaling our system for larger QKD networks may pose challenges in terms of key management and storage. A possible future work could involve re-designingthe system for non-peer-to-peer connections within QKD networks. This comprehensive redesign would address the setup phase, pre-shared value management, and key storage for multiple nodes.

Another extension could aim at the design of FPGA implementations that can be optimized to be more compact for the FPGA-based network cards having fewer hardware resources. This optimization would aim to enhance the efficiency of the system on FPGA-based platforms.

The slow distillation of the key from the QKD method is perhaps the main problem that will affect the high-speed connection efficiency. Future improvement tasks could aim at increasing the rates for establishing keys by QKD by studying the appropriate connection conditions (distances, optical connection quantity, etc.). In this case, the primary responsibility lies with QKD producers.

While QKD inherently provides unconditional security, threats such as quantum hacking and eavesdropping techniques targeting quantum communication channels should be carefully considered. Moreover, relay architectures, extending the reach of QKD-secured networks, introduce trust concerns. Mitigating vulnerabilities associated with trusted nodes involves incorporating verifiable and authenticated quantum devices. Ensuring the integrity of relay nodes is necessary for the overall security of the network.

While the proposed system offers versatility in combining classic cryptography, post-quantum cryptography, and QKD, further research can explore its adaptability to diverse cryptographic needs. This involves understanding the system's performance under different use cases and cryptographic requirements. Finally, in the rapidly evolving field of quantum computing, ensuring the long-term viability of our system is crucial. Factors such as technological advancements, emerging quantum algorithms, and evolving security standards need continuous attention. These can be solve with a continuous up-to-date analysis of current attack and security standard development, to keep the system up-to-date.

VIII. CONCLUSION

In this article, we present a concrete 3-key combiner system implemented on an FPGA platform. Our system involves a pre-quantum KEX, a post-quantum KEM, and a QKD algorithm that allows for a smooth transition to PQ cryptography. In fact, our combiner retains the time-tested trust in pre-quantum algorithms and the quantum-resistant of PQ schemes. Moreover, our architecture is an extension of the dual-PRF combiner designed to work with three keys as input. The proposed 3-key combiner has been proven to be secure in the quantum standard model and indistinguishable under a chosen-ciphertext attack.

Our system is, for instance, suitable for peer high-speed connections requiring high-security assurances such as between governmental bodies, security institutions, critical infrastructure nodes, data centers, telecommunications operators, cloud service providers, or in the financial sector. The main benefit of the proposed hybrid system is its high-security resistance against three types of attacks. Attackers have to break three different methods (i.e., pre- and post-quantum cryptography, and quantum cryptography).

It is important to note that our system is relatively independent of the KEM and KEXs involved. In fact, if one wants to achieve IND-CCA, then any KEM or KEX scheme proven to be IND-CCA is suitable. Moreover, if one of the combined schemes is also quantum-resistant, the key combiner presents the same property as our proposal. Finally, system can be easily switched to either use the QKD algorithm or not. This allows the FPGA platform to be deployed with and without the quantum component, making the solution more accessible from a market point of view.

Our combiner also runs in small FPGA platforms since it requires low resources usage (i.e., 4 532 LUTs and 3 363 FFs). In particular, the key combiner without QKD is able to output up to 1,624 keys per second and the key combiner with QKD is able to output up to 9.2 keys per second.

REFERENCES

[1] S. Even and O. Goldreich, "On the power of cascade ciphers," *ACM Trans. Comput. Syst.*, vol. 3, no. 2, pp. 108–116, May 1985.

[2] D. J. Bernstein, "Introduction to post-quantum cryptography," in *Proc. Post-Quantum Cryptogr.* Berlin, Germany: Springer, 2009, pp. 1–14.

[3] ANSSI. (2022). *Anssi Views on the Post-Quantum Cryptography Transition*. [Online]. Available: https://www.ssi.gouv.fr/uploads/2022/01/anssi-technical_position_papers-post_quantum_cryptography_transition.pdf

[4] BSI. (2022). *Quantum-Safe Cryptography—Fundamentals, Current Developments and Recommendations*. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Brochure/quantum-safe-cryptography.html

[5] N. Bindel, J. Brendel, M. Fischlin, B. Goncalves, and D. Stebila, "Hybrid key encapsulation mechanisms and authenticated key exchange," in *Proc. Post-Quantum Cryptogr., 10th Int. Conf., PQCrypto*. Chongqing, China: Springer, 2019, pp. 206–226.

[6] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, and T. Wunderer, "Estimate all the LWE, NTRU schemes!" in *Proc. Secur. Cryptogr. Netw., 11th Int. Conf. (SCN)*. Amalfi, Italy: Springer, Sep. 2018, pp. 351–367.

[7] R. Azarderakhsh, R. Elkhatib, B. Koziel, and B. Langenberg, "Hardware deployment of hybrid PQC: Sike+ ECDH," in *Proc. Secur. Privacy Commun. Netw., 17th EAI Int. Conf. (SecureComm)*. Springer, Sep. 2021, pp. 475–491.

[8] N. Aviram, B. Dowling, I. Komargodski, K. G. Paterson, E. Ronen, and E. Yogev, "Practical (post-quantum) key combiners from one-wayness and applications to TLS," *Cryptol. ePrint Arch.*, pp. 1–24, Feb. 2022.

[9] F. Giacon, F. Heuer, and B. Poettering, "Kem combiners," in *Proc. Public-Key Cryptogr. PKC 21st IACR Int. Conf. Pract. Theory Public-Key Cryptogr.* Rio de Janeiro, Brazil: Springer, Mar. 2018, pp. 190–218.

[10] L. Huguenin-Dumittan and S. Vaudenay, "Fo-like combiners and hybrid post-quantum cryptography," in *Proc. Cryptol. Netw. Secur., 20th Int. Conf. (CANS)*. Vienna, Austria: Springer, Dec. 2021, pp. 225–244.

[11] ENISA. (2022). *Post-Quantum Cryptography—Integration Study*. [Online]. Available: https://www.enisa.europa.eu/publications/post-quantum-cryptography-integration-study

[12] D. Steblia, S. Fluhrer, and S. Gueron, "Hybrid key exchange in TLS 1.3," Internet Eng. Task Force (IETF), Internet-Draft Draft-Ietf-Tls-Hybrid-Design-01, Wilmington, DE, USA, Tech. Rep. draft-ietf-tls-hybrid-design-09, 2020.

[13] R. Renner, N. Gisin, and B. Kraus, "Information-theoretic security proof for quantum-key-distribution protocols," *Phys. Rev. A, Gen. Phys.*, vol. 72, no. 1, Jul. 2005, Art. no. 012332.

[14] Y. Liang, H. V. Poor, and S. Shamai Shitz, "Information theoretic security," *Found. Trends Commun. Inf. Theory*, vol. 5, nos. 4–5, pp. 355–580, 2009.

[15] E. Rescorla, "The transport layer security (TLS) protocol version 1.3," Internet Eng. Task Force (IETF), Wilmington, DE, USA, Tech. Rep. RFC 8446, 2018.

[16] J. M. Schanck and D. Stebila, "A transport layer security (TLS) extension for establishing an additional shared secret," *IETF Draft*, pp. 1–19, Apr. 2017.

[17] B. Poettering and S. Rastikian, "A study of KEM generalizations," in *Proc. Int. Conf. Res. Secur. Standardisation*, Springer, 2023, pp. 53–77.

[18] D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. De Feo, B. Hess, A. Jalali, B. Koziel, B. LaMacchia, and P. Longa, "Supersingular isogeny key encapsulation," Post-Quantum Standardization Project, NIST, Gaithersburg, MD, USA, Tech. Rep., 2017.

[19] S. Stadler, V. Sakaguti, H. Kaur, and A. L. Fehlhaber, "Hybrid signal protocol for post-quantum email encryption," *Cryptol. ePrint Arch.*, pp. 1–20, Jun. 2021.

[20] M. Campagna and A. Petcher, "Security of hybrid key encapsulation," *Cryptol. ePrint Arch.*, pp. 1–15, Nov. 2020.

[21] B. Dowling, T. B. Hansen, and K. G. Paterson, "Many a Mickle makes a muckle: A framework for provably quantum-secure hybrid key exchange," in *Proc. Post-Quantum Cryptogr., 11th Int. Conf. (PQCrypto)*. Paris, France: Springer, Apr. 2020, pp. 483–502.

[22] P. Schwabe, D. Stebila, and T. Wiggers, "Post-quantum TLS without handshake signatures," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 1461–1480.

[23] K. Kwiatkowski and L. Valenta, "The TLS post-quantum experiment," *Post Cloudflare Blog*, 2019.

[24] A. Langley, "Real-world measurements of structured-lattices and supersingular isogenies in TLS," Tech. Rep., 2019. [Online]. Available: https://www.imperialviolet.org/2019/10/30/pqsivssl.html

[25] A. A. Giron, R. Custódio, and F. Rodríguez-Henríquez, "Post-quantum hybrid key exchange: A systematic mapping study," *J. Cryptograph. Eng.*, vol. 13, no. 1, pp. 71–88, Apr. 2023.

[26] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Proc. 16th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA: Springer, Aug. 1996, pp. 1–15.

[27] M. Bellare, "New proofs for NMAC and HMAC: Security without collision-resistance," in *Proc. CRYPTO*, vol. 4117. Berlin, Germany: Springer, 2006, pp. 602–619.

[28] P. Gaži, K. Pietrzak, and M. Rybár, "The exact PRF-security of NMAC and HMAC," in *Proc. 34th Annu. Cryptol. Conf.*, Santa Barbara, CA, USA. Berlin, Germany: Springer, Aug. 2014, pp. 113–130.

[29] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *SIAM J. Comput.*, vol. 33, no. 1, pp. 167–226, Jan. 2003.

[30] R. Canetti, O. Goldreich, and S. Halevi, "The random Oracle methodology, revisited," *J. ACM*, vol. 51, no. 4, pp. 557–594, 2004.

[31] N. Aviram, B. Dowling, I. Komargodski, K. Paterson, E. Ronen, and E. Yogev. (2021). *Concatenating Secrets May Be Dangerous*. [Online]. Available: https://github.com/nimia/kdf_public

[32] Y. Sasaki, G. Yamamoto, and K. Aoki, "Practical password recovery on an MD5 challenge and response," *Cryptol. ePrint Arch.*, pp. 1–11, Mar. 2007.

[33] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, and R. Peralta, "Status report on the third round of the nist post-quantum cryptography standardization process," U.S. Dept. Commerce, NIST, Gaithersburg, MD, USA, Tech. Rep. 8413, 2022.

[34] NCSC. (2019). *Factsheet Post-quantum Cryptography*. [Online]. Available: https://english.ncsc.nl/publications/factsheets/2019/juni/01/factsheet-post-quantum-cryptography

[35] V. S. Miller, *Use of Elliptic Curves in Cryptography*. Berlin, Germany: Springer, 1986.

[36] IDQ. *Idq Quantum-Safe Security Products*. Accessed: Feb. 10, 2024. [Online]. Available: https://www.idquantique.com/

[37] O. Klicnik, A. Tomasov, P. Munster, T. Horvath, and J. Hajny, "Long-term parameters monitoring of the IDQ clavis 3 QKD system," in *Proc. Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2022, pp. 1–4.

[38] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, "CRYSTALS-kyber: A CCA-secure module-lattice-based KEM," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Apr. 2018, pp. 353–367.

[39] NSA. (2022). *Announcing the Commercial National Security Algorithm Suite 2.0*. [Online]. Available: https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF

[40] H. Krawczyk, "Cryptographic extraction and key derivation: The HKDF scheme," in *Proc. CRYPTO*, vol. 6223. Berlin, Germany: Springer, 2010, pp. 631–648.

[41] NIST. (2016). *Submission Requirements and Evaluation Criteria for the Post-quantum Cryptography Standardization Process*. [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

[42] NIST. (2018). *Let's Get Ready To Rumble—The Nist Pqc*. [Online]. Available: https://csrc.nist.gov/CSRC/media/Presentations/Let-s-Get-Ready-to-Rumble-The-NIST-PQC-Competiti/images-media/PQCrypto-April2018_Moody.pdf

[43] E. Barker, "Nist special publication 800–57 part 1, revision 5," NIST, Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-57 Part 1 Rev. 5, 2020.

[44] M. Dworkin, "Nist special publication 800–38D," NIST, Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-38D, 2007.

[45] M. Bellare and A. Lysyanskaya, "Symmetric and dual prfs from standard assumptions: A generic validation of an hmac assumption," *Cryptol. ePrint Arch.*, pp. 1–17, Dec. 2015.

[46] L. Malina, S. Ricci, P. Dobias, P. Jedlicka, J. Hajny, and K.-K. Choo, "On the efficiency and security of quantum-resistant key establishment mechanisms on FPGA platforms," in *Proc. 19th Int. Conf. Secur. Cryptogr.*, 2022, pp. 605–613.

[47] M.-Y. Li, X.-Y. Cao, Y.-M. Xie, H.-L. Yin, and Z.-B. Chen, "Finite-key analysis for coherent one-way quantum key distribution," *Phys. Rev. Res.*, vol. 6, no. 1, Jan. 2024, Art. no. 013022.

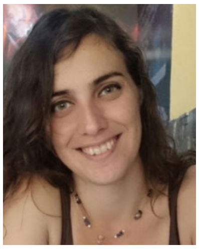
[48] O. Klicnik, K. Turcanova, P. Munster, A. Tomasov, T. Horvath, and J. Hajny, "Deploying quantum key distribution into the existing university data infrastructure," in *Proc. IEEE AFRICON*, 2023, pp. 1–3.

[49] J. Vojtech, R. Vohnout, O. Havliš, P. Pospíšil, M. Šlapák, R. Velc, L. Altmannová, T. Horváth, J. Kundrát, and M. Hažlinskỳ, "First cross-border trial of quantum key distribution sharing fiber line with data and accurate time transmissions," in *Proc. SPIE*, vol. 12238, 2022, pp. 101–107.

[50] O. Klicnik, P. Munster, and T. Horvath, "Multiplexing quantum and classical channels of a quantum key distribution (QKD) system by using the attenuation method," *Photonics*, vol. 10, no. 11, p. 1265, Nov. 2023.

[51] O. Klíčník, P. Münster, and T. Horváth, "Impact of optical connectors on QKD transmission," *Fiber Integr. Opt.*, vol. 42, no. 6, pp. 219–231, Nov. 2023.

[52] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proc. 42nd IEEE Symp. Found. Comput. Sci.*, 2001, pp. 136–145.



SARA RICCI received the M.Sc. degree in mathematics from the University of Pisa, Italy, in 2015, and the Ph.D. degree in computer engineering and mathematics security from Universitat Rovira i Virgili, Spain, in 2018. She is currently a Postdoctoral Researcher with the Department of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology, Czech Republic. Her research interests include theoretical cryptography, in particular lattice-based and elliptic curve cryptography and data privacy and security. She is also focused on the design of new privacy-preserving cryptographic protocols and their security analyses. She has been involved as the Task Leader of the SPARTA H2020 Project. She is also involved as a Co-Investigator and the WP Leader of the ERASMUS+ REWIRE Project.

PATRIK DOBIAS received the master's degree in information security from the Brno University of Technology, Czech Republic, where he is currently pursuing the Ph.D. degree. He is also focused on hardware implementation and optimization of cryptographic schemes.

LUKAS MALINA received the M.Sc. degree (Hons.) and the master's and Ph.D. degrees from the Brno University of Technology (BUT), Czech Republic, in 2010 and 2014, respectively. He is currently an Associate Professor with the Department of Telecommunications, BUT. He has published more than 90 papers in international journals and conferences. He has provided several invited research and teaching lectures abroad, e.g., University of Tampere, Finland, in 2013; URV Tarragona, Spain, in 2015; and KU Leuven, Belgium, in 2017. His research interests include applied cryptography, privacy-preserving protocols, and authentication systems. He has been involved as the Task Leader of the SPARTA H2020 Project (task: Privacy-by-Design) and a Senior Researcher in several Czech scientific projects focused on cybersecurity. He obtained the Dean's Prize for the master's degree.

JAN HAJNY received the Ph.D. degree from the Brno University of Technology (BUT), Czech Republic, in 2012. He is currently an Associate Professor with the Faculty of Electrical Engineering and Communication, BUT. He also deals with the research into cryptographic protocols for the privacy and digital identity protection. He is also the Co-Founder and lead of the Cryptology Research Group (http://crypto.utko.feec.vutbr.cz) and is responsible for managing the information security study program with the university. He is the author of more than 80 scientific publications and cooperates with renowned laboratories abroad.

PETR JEDLICKA received the master's degree in communication technologies from the Department of Radioelectronics, Brno University of Technology, Czech Republic, where he is currently pursuing the Ph.D. degree with the Department of Telecommunications. He also deals with hardware-accelerated post-quantum cryptography.

• • •