HAPKIDO Project Report D5.2

Literature Review

***

(Quantum-Safe) Cryptographic Combiners and Hybrid Security

Serge Fehr[1,2], Yu-Hsuan Huang[1], and Alessandro Amadori[3]

[1]Centrum Wiskunde & Informatica (CWI), Amsterdam
[2]Mathematical Institute, Leiden University, Leiden
[3]Netherlands Organization for Applied Scientific Research (TNO), The Hague

November 13, 2023

This report is a result of the *HAPKIDO* project. It forms deliverable 5.1 and was produced within *Work Package 5: Cryptographic Tools*, in a collaboration between *CWI* and *TNO*. *HAPKIDO* stands for *"Hybrid Approach for quantum-safe Public Key Infrastructure Development for Organisations"*, and the project is a five-year initiative that aims to develop a roadmap for the transition to quantum-safe public key infrastructures. It is a collaboration between public and private parties and is financed by the *Dutch Research Council (NWO)*. For more information on the *HAPKIDO* project and for more project reports and deliverables, we refer to the project website `https://hapkido.tno.nl`.

Contents

1 Introduction

1.1 Background

The security of current public-key encryption and digital signature schemes relies on the (believed) hardness of factoring large integers and of computing discrete logarithms in certain large groups, like elliptic curves. These are two computational problems that would take millions or billions of years to solve on a conventional computer (using the best known algorithms). However, as was shown by Peter Shor in 1994 [Sho94], these problems are easy to solve by means of a quantum computer, by running what is now called *Shor's algorithm*. Thus, current public-key cryptography is vulnerable to potential future *quantum attacks*.

A *quantum computer* is a hypothetical computer that exploits the laws of quantum mechanics in order to offer computing capabilities that go beyond those of current conventional computers. These novel computing capabilities can then be used — in theory — to efficiently solve certain computational problems that were believed to be infeasible to solve. It is still an active and ongoing research area to understand which computational problems (beyond factoring and computing discrete logarithms) benefit from quantum computing capabilities.

A sufficiently large, universally programmable quantum computer, as needed for running Shor's algorithm on interesting problem instances, is currently still out of reach; nevertheless, we cannot afford to wait. Because of this, there is a huge international effort in migrating to new cryptographic schemes that are (believed to be) secure against quantum attacks; e.g., the *US National Institute of Standards and Technology (NIST)* is running processes for soliciting, evaluating and standardizing such quantum-secure schemes, sometimes also referred to as *post-quantum* cryptographic schemes.[1] These are schemes whose security is based on computational problems that are believed to be hard even when considering quantum computing capabilities. Examples of such computational problems arise for instance in the field called *geometry of numbers*, where mathematical objects referred to as *lattices* are studied.

While it is clear that in order to offer security against quantum attacks it is *necessary* to replace the computational problems, upon which the security of the schemes is supposed to rely, by ones that are (believed to be) hard to solve on a quantum computer (like lattice problems), this is *not sufficient* in general. The reason is that also the soundness of the design principle, which is meant to imply security of the scheme whenever the underlying computational problem is hard, may fail to hold when considering quantum attacks. Thus, also the security proof needs to be revisited.

Replacing the current cryptographic schemes by new ones bares various risks. Needless to say that the new schemes that will be standardized have — and are being — undergone tough scrutiny; nevertheless, they have been studied and analyzed significantly less than the schemes that are currently used. Therefore, while promising to offer security against quantum attacks, there is a risk that such a new scheme may actually be insecure (or less secure as believed) against classical attacks even. This could for instance be because the underlying computational problem turns out to be easy to solve after all (as in [CD23]), or easier than believed, or because there is flaw in the construction design and/or the security reduction that remained unnoticed (as in [BBD+23]), or because of implementation errors or insufficient side-channel protection, etc. Thus, in the worst case, instead of *improving* security we might end up *weakening* security.

One potential solution, very appealing certainly from the security perspective, is to make use of so-called (cryptographic) *combiners*. In full generality, combiners are techniques for turning two or more cryptographic schemes for a certain task, like encryption, into a new scheme for the same task, but which is then guaranteed to be as secure as the *most secure* of the component schemes. Thus, even if some but not all of the component schemes turn out to be insecure, the combined scheme remains secure. The combined scheme is sometimes also referred to as a *hybrid scheme*.

The benefit of using combiners in the context of the migration to new, quantum-secure public-key encryption and digital signature schemes is obvious: by using a combiner to combine a well-established, say, factoring-based scheme with a new(er) post-quantum secure scheme, whose security has not been

[1] See the official website `https://csrc.nist.gov/projects/post-quantum-cryptography`

under scrutiny for such a long time yet but is strongly believed to offer security against quantum attacks, we are armed against the upcoming of a quantum computer (assuming the post-quantum scheme to remain secure) *and* against a break of the post-quantum scheme by a classical computer (as long as we are still waiting for the quantum computer to come).

Even though the use of combiners sounds very appealing from a security perspective, it comes with practical issues. First of all, running the combined scheme essentially means running both component schemes, which makes the execution correspondingly slower, compare to just using one or the other scheme. Also, key-, signature- and ciphertext-sizes typically grow, which may result in compatibility issues with standards, for instance for PKI certificates. Furthermore, in particular in the context of the post-quantum cryptography migration, one may encounter situations where some parties have migrated to post-quantum security, possibly using combiners, while others have not (yet). Therefore, sometimes the predicate *hybrid* is also understood as offering the flexibility of falling back to non-quantum security in case certain instances have not migrated yet. Finally, even though naively constructing combiners may seem easy — sign twice for signing, and do a double encryption for encrypting — simple constructions tend to have pitfalls.

1.2 Contribution of this report — and what it does not provide

In this document, we report on a literature review, conducted by the authors of the document, on the topic of combiners and hybrid security. Due to their relevance in the context of PKIs, the main focus is on combiners for public-key encryption schemes (or KEMs) and digital signature schemes; combiners for other cryptographic tasks (like hash combiners etc.) are not covered.

The purpose of this report is two-fold: (1) we want to give a high-level overview on the questions and problems that are being considered and studied in the scientific literature on the topic of combiners and hybrid security, and (2) we want to give a brief summary of the results and achievements in the articles covered in this report. The latter in particular should facilitate the decision making, when one is searching for a particular study/result and one is unsure into which article to look in detail. Table 1 lists the articles that we found on the considered topic and that we covered in this report.

We stress that the purpose of this report is *not* to give advice on whether to deploy combiners or not, or what combiner to use and which schemes to combine. Various different factors weigh in and there is no universal right answer. For instance, the French *Agence nationale de la sécurité des système d'information (ANSSI)* and the German *Bundesachrichtendienst (BND)* in general recommend the use of combiners[2], certainly during the transition phase, while the US *National security agency (NSA)* is more reluctant.[3]

Also, we do not claim the list of papers in Table 1 to be complete. We did a thorough search but it is possible that we missed some articles.

2 Preliminaries

We assume the reader to be familiar with basic concepts from cryptography and with basic cryptographic primitives like *hash functions*, *pseudorandom functions* (PRFs), *public-key encryption* (PKE), *key-encapsulation mechanisms* (KEMs), *digital signature schemes*, etc., as well as standard security notions for such primitives, like *OW-CPA* and *IND-CCA* security for PKE and KEMs. We refer to [KL07] for a textbook that introduces and discusses these (and more).

In general, a (cryptographic) *combiner* is a means to turn multiple cryptographic schemes (for the same task) into a new scheme (typically for the same task again), so that the new scheme is secure if *at least one* of the component schemes is secure. Put differently: even if one (or more) of the component schemes turns out to be insecure, but not all of them, the combined scheme remains secure. We focus here on combiner for public-key encryption, actually KEMs, and for signature schemes.

[2] See `https://www.ssi.gouv.fr/en/publication/anssi-views-on-the-post-quantum-cryptography-transition/` and `https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Brochure/quantum-safe-cryptography.pdf`.

[3] See e.g. `https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI_CNSA_2.0_FAQ_.PDF`.

Reference	Authors	Year	Title	Page
[GHP18]	Giacon *et al.*	2018	KEM combiners	7
[CPS19]	Crockett *et al.*	2019	Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH	7
[BBF+19]	Bindel *et al.*	2019	Hybrid key encapsulation mechanisms and authenticated key exchange	8
[HDV21]	Huguenin-D. *et al.*	2021	FO-like combiners and hybrid post-quantum cryptography	8
[XGL+21]	Xu *et al.*	2021	Stateful KEM: towards optimal robust combiner for key encapsulation mechanism	9
[ADK+22]	Aviram *et al.*	2022	Practical (post-quantum) key combiners from one-wayness and applications to TLS	10
[DFH22]	Don *et al.*	2022	Adaptive versus static multi-oracle algorithms, and quantum security of a split-key PRF	10
[GM22]	Goncalves *et al.*	2022	Tightly secure PKE combiner in the quantum random oracle model	11
[SFG23]	Stebila *et al.*	2023	Hybrid key exchange in TLS 1.3 (draft IETF)	11
[TTB+23]	Tjhai *et al.*	2023	Multiple key exchanges in IKEv2 (draft IETF)	12
[SBM23]	Soroceanu *et al.*	2023	On multiple encryption for public-key cryptography	12
[BHMS17]	Bindel *et al.*	2017	Transitioning to a quantum-resistant public key infrastructure	12
[TLG+18]	Truskovsky *et al.*	2018	Multiple public-key algorithm X.509 certificates	13
[KPDG18]	Kampanakis *et al.*	2018	The viability of post-quantum X.509 certificates	14
[BBG+19]	Bindel *et al.*	2019	X.509-compliant hybrid certificates for the post-quantum transition	14
[Lyt21]	John Lytle	2021	Performance of hybrid signatures for public key infrastructure certificates	15
[FWZ+21]	Fan *et al.*	2021	Impact of post-quantum hybrid certificates on PKI, common libraries and protocols	15
[RCW+21]	Raavi *et al.*	2021	Performance characterization of post-quantum digital certificates	16
[GKP+23]	Ghinea *et al.*	2023	Hybrid post-quantum signatures in hardware security keys	16
[FvdHM+23]	Fischlin *et al.*	2023	Post-quantum security for the extended access control protocol	18
[GdNC+23]	Giron *et al.*	2023	Post-quantum hybrid KEMTLS performance in simulated and real network environments	18

Table 1: List of articles covered in this report.

2.1 KEM combiners

Following the above generic definition, a *KEM combiner* is a generic transformation that turns two or more KEMs (the *component* KEMs) into a new KEM (the *combined* KEM), which then has the property that the combined KEM is as secure as the most secure of its components. To emphasize that this property is indeed satisfied, one sometimes refers to the combiner as being *robust*. The aspired security is typically IND-CCA security, but sometimes some other notions are considered.

A natural construction design for a KEM combiner is to run the n component KEM schemes in parallel in order to obtain n key-ciphertext pairs $(k_1, c_1), \ldots, (k_n, c_n)$, and then derive the key k for the combined KEM by applying a suitable function f to the list of keys $(k_1, \ldots, k_n)$ and ciphertexts $(c_1, \ldots, c_n)$ obtained by running the component KEMs, i.e.,

$$k = f(k_1, \ldots, k_n, c_1, \ldots, c_n)\,.$$

The function f is then typically called *core function* or *key combiner*. Sometimes, the core function f only takes the keys $(k_1, \ldots, k_n)$ as input, and no ciphertexts.

Example 1. Common choices for the core function / KEM combiners are:

- A hash function modeled as a random oracle.
- A dual-PRF (see Def. 1 below).
- The nested dual-PRF N:

$$c, k_1, k_2 \mapsto \mathsf{PRF}_2\big(\mathsf{dualPRF}\big(\mathsf{PRF}_1(k_1), k_2\big), c\big)$$

  where PRF_1 and PRF_2 are pseudorandom functions, and $\mathsf{dualPRF}$ is a dual-PRF.
- XOR-then-MAC (XtM), where the combined KEM key k is obtained as the left half of $k_1 \oplus k_2$, and where additionally the ciphertext is augmented with an unconditional MAC tag, where the key for the latter is built from the right halves of k_1 and k_2.

Related to the definition of a KEM combiner is the notion of a dual-PRF, and more generally, a spilt-key PRF.

Definition 1 (dual-PRF)**.** A function $f(x, y)$ on two inputs is a *dual-PRF* if it is a PRF *both* as a function of y with key x *and* as a function of x with key y.

Definition 2 (split-key PRF)**.** A function $f(k_1, \ldots, k_n, x)$ is a split-key PRF if, for each $i \in [n]$, it is a PRF as a function of $(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_n, x)$ with key k_i.[4]

2.2 Signature combiners

In the same spirit, a *signature combiner* is a generic transformation that turns two or more digital signature schemes (the *component* schemes) into a new digital signature scheme (the *combined* scheme), with the property that the combined scheme is as secure as the most secure of the component schemes. Here, by default the aspired security is (strong) unforgeability under chosen message attack, i.e., (S)EU-CCA.

A very natural construction for a signature combiner is to simply concatenate the signatures:

Example 2. Given two signature schemes, a combined signature scheme can be obtained by signing the considered message m individually with the two schemes, resulting in signatures σ_1 and σ_2, and to let the pair $\sigma = (\sigma_1, \sigma_2)$ then be the signature for the combined scheme. This signature combiner is sometimes referred to as *concatenation*.

Unless signing is deterministic, we note however that this simple signature combiner does not preserve *strong* unforgeability under chosen message attack, since the attacker can ask twice to get the same message m signed, resulting in signatures $\sigma = (\sigma_1, \sigma_2)$ and $\sigma' = (\sigma'_1, \sigma'_2)$, and can then output, say, (σ_1, σ'_2) as a forged signature for m. Thus, this shows that one has to be very careful with trivial combiner constructions; whether a combiner "does its job" or not very much also depends on the considered security notion.

Example 3. Given two signature schemes, a combined signature scheme can be obtained by signing the considered message m with the first scheme to obtain signature σ_1, and then signing the pair (m, σ_2) with the second signature scheme to obtain σ_2, and to let the pair $\sigma = (\sigma_1, \sigma_2)$ then be the signature for the combined scheme. This is sometimes referred to as *(strong) nesting*.

3 Summaries

In this main part of the report, we provide concise summaries of the articles listed in Table 1 on the topic of cryptographic combiners and hybrid security. Section 3.1 covers the articles that focus on KEMs, Section 3.2 those that focus on digital signatures, and Section 3.3 those that consider both. In some cases, the summary is accompanied with a box that provides some more detailed information.

[4] The notion considered in, e.g., [GHP18] is slightly weaker: any x may be queried only once.

3.1 Hybrid KEMs

KEM combiners ([GHP18])

Federico Giacon, Felix Heuer, and Bertram Poettering
PKC 2018
`https://eprint.iacr.org/2018/024`

Summary. The paper proposes and analyzes several constructions of KEM combiners, showing CCA-security of the combined KEM if at least one of the component KEMs is CCA-secure. The KEM combiners are classified in terms of their core functions (see Sect.. 2), denoted W in the paper.

Concretely, the paper shows CCA-security if W is *any* split-key PRF as defined in Def. 2, as well as for the specific core function (which is not a split-key PRF)

$$W(k_1, \ldots, k_n, c_1, \ldots, c_n) = \bigoplus_i F_i(k_i, c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n)$$

where each F_i is a PRF. Additionally, the paper shows that

$$W(k_1, \ldots, k_n, c_1, \ldots, c_n) = \bigoplus_i F_i(k_i, c_1, \ldots, c_n)$$

is a split-key PRF (and thus a secure KEM combiner). On top, it shows that some hash-based constructions as split-key PRF are well (see the box below), in the ROM then.

> In more detail, it is shown that the following hash-based constructions are split-key PRFs (in the ROM)
>
> $$H(k_1, \ldots, k_n, c_1, \ldots, c_n)\,,\ H(k_1 \oplus \cdots \oplus k_n, c_1, \ldots, c_n) \text{ and } H\big(\pi(k_n, \ldots, \pi(k_2, \pi(k_1, 0)) \cdots), c_1, \ldots, c_n\big)\,,$$
>
> where $\pi(k, x)$ is a pseudorandom permutation of input x and key k. In the very last construction, H can be traded by a PRF when π is modeled as in *ideal cipher* instead. The first two constructions can be unified and generalized to
>
> $$W(k_1, \ldots, k_n, c_1, \ldots, c_n) = H\big(g(k_1, \ldots, k_n), c_1, \ldots, c_n\big)$$
>
> where the function g is such that for any i and any $k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_n$, and for a uniformly random K_i, the random variable $g(k_1, \ldots, k_{i-1}, K_i, k_{i+1}, \ldots, k_n)$ has high min-entropy, i.e., is hard to guess.

Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH ([CPS19])

Eric Crockett, Christian Paquin, and Douglas Stebila
IACR eprint (2019)
`https://eprint.iacr.org/2019/858`

Summary. The paper explores how two major Internet security protocols, the Transport Layer Security (TLS) and Secure Shell (SSH) protocols, can be adapted to use post-quantum cryptography.

First, the paper examines various design considerations for integrating post-quantum and hybrid key exchange and authentication into communications protocols generally, and in TLS and SSH specifically. These include issues such as how to negotiate the use of multiple algorithms for hybrid cryptography, how to combine multiple keys, and more.

Subsequently, the paper reports on several specific case studies, using KEMs and signature scheme families from to the NIST Round 2 submission:

- TLS 1.2: Post-quantum and hybrid key exchange, in OpenSSL 1.0.2s and Amazon s2n.
- TLS 1.3: Post-quantum and hybrid key exchange, and post-quantum and hybrid authentication, in OpenSSL 1.1.1c.

- SSH 2: Post-quantum and hybrid key exchange, and post-quantum and hybrid authentication, in OpenSSH 7.9.

In general, the results are pretty good; problems arose mainly from large message sizes: sizes that were bigger than the protocol specification allowed, or sizes that were within protocol specification tolerances but where the implementation in question had internal buffers or parameters set smaller than the maximum size permitted by the specification.

The conclusion of the paper is that there are several ways to extend TLS and SSH for both KEM and authentication. Each comes with their own pros and cons. What the best solution is, should depend on the application where the protocols are employed. Changes in the implementation of the protocols should also be tackled carefully since it might cause compatibility issues (like managing the sizes of the messages).

Hybrid key encapsulation mechanisms and authenticated key exchange ([BBF+19])

Nina Bindel, Jacqueline Brendel, Marc Fischlin, Brian Goncalves, and Douglas Stebila
PQCrypto 2019
`https://eprint.iacr.org/2018/903.pdf`

Summary. The paper introduces and studies similar fine-grained distinctions between classical and quantum attacks as in [BHMS17], but now in the context of KEMs. In more detail, it considers the notion of X^yZ-security with $X, Z \in \{\mathsf{Q}, \mathsf{C}\}$ and $y \in \{\mathsf{q}, \mathsf{c}\}$, where X being C or Q means that the attacker is classical or quantum *at the time it can make decryption (or decapsulation) queries*, y being c or q means that these *decryption queries* are classical or quantum, and Z being C or Q means that the attacker is classical or quantum *after having made all decryption queries*. The paper first shows various relations for these security notions for KEMs (see the box).

$$\mathsf{C^cQ}\text{-ind-cca} \geq \mathsf{Q}\text{-ind-cpa} \geq \mathsf{C}\text{-ind-cpa}$$
$$\mathsf{C}\text{-ind-cpa} \leq \mathsf{C^cC}\text{-ind-cca} \lneq \mathsf{C^cC}\text{-ind-cca} \lneq \mathsf{C^cQ}\text{-ind-cca} \lneq \mathsf{Q^cQ}\text{-ind-cca} \lneq \mathsf{Q^qQ}\text{-ind-cca}$$
$$\mathsf{C^cC}\text{-ind-cca} \not\leq \mathsf{Q}\text{-ind-cpa} \not\leq \mathsf{C^cC}\text{-ind-cca}\ ,$$

Then, the paper shows for several KEM combiners (namely for $\mathsf{XtM}, \mathsf{dualPRF}$ and N) that they preserve the notions of $\mathsf{C^cC}$, $\mathsf{C^cQ}$ and $\mathsf{Q^cQ}$-ind-cca security, meaning that if at least one of the component KEMs satisfies the considered notion then so does the resulting combined KEM.

Remark 1. *It remains unclear to us what should be the motivation to consider security notions weaker than* $\mathsf{Q^cQ}$-ind-cca *(like* $\mathsf{C^cQ}$-ind-cca*), when considering quantum attacks.*

The paper also considers similar fine-grained security notions for authenticated key exchange (AKE), and shows how to build hybrid AKE from hybrid KEMs, relying on the Krawczyk's SigMA-compiler, which uses signatures and MACs to obtain security against active adversaries. Intuitively, one would expect that the "weakest primitive" determines the overall security of the compiled protocol; however, it is shown that this intuition is not entirely correct for partially quantum adversaries.

FO-like combiners and hybrid post-quantum cryptography ([HDV21])

Loïs Huguenin-Dumittan and Serge Vaudenay
IACR eprint (2021)
`https://eprint.iacr.org/2021/1288`

Summary. The paper considers and analyzes two ways to combine two (or more) PKE schemes into a KEM in an FO-like manner. Concretely, the two combiners have its ciphertext $c = (c_1, c_2)$ and combined key K computed as follows, with randomly sampled $k_b \leftarrow \mathcal{M}_b$ from the respective message spaces,

$$c_b = \mathsf{enc}_b(\mathsf{pk}_b, k_b, G(k_b)) \qquad \text{and} \qquad K = H(k_1, k_2)\ ,$$

and

$$c_b = \mathsf{enc}_b(\mathsf{pk}_b, k_b; G(b, k_1, k_2)) \qquad \text{and} \qquad K = H(k_1 \oplus k_2)$$

respectively, where H and G are hash functions, and enc_b for $b \in [2]$ denotes the encryption of the corresponding component PKE scheme.

Using the terminology from the FO-transform, the former construction can be understood as applying the $\mathsf{U}^{\perp}_m$-transform to the *parallel composition* of two T-transforms (denoted as $\mathsf{T}_{\|}$ in the paper). The second construction (denoted $\mathsf{UT}_{\|}$) has the execution of $\mathsf{U}^{\perp}_m$ and the two executions of T intertwined in the above way.

The paper proves that the two KEMs are IND-CCA secure provided that one of the component PKE's is OW-CPA secure. These proofs are in the ROM. The paper also argues security in the QROM if the ciphertext is expanded with an additional *confirmation hash*

$$d = H'(k_1, k_2)\,,$$

where H' is yet another hash function (this is in line with the $\mathsf{QU}^{\perp}$ transform). The paper conjectures that this confirmation hash is not necessary for QROM security.

The paper claims that such a construction that combines weakly secure components into a strongly secure KEM is favourable over constructions that require the components to already have strong security, like is the case for combiners based on XtM, dual-PRF, and N (see Example 1).

Remark 2. *There may be arguments for the above claim. On the other hand, it is natural that one would combine schemes that have been standardized, or have some other official approval, and those scheme have strong security most of the time anyway.*

Stateful KEM: towards optimal robust combiner for key encapsulation mechanism ([XGL+21])
Jia Xu, Yiwen Gao, Hoon Wei Lim, Hongbing Wang, and Ee-Chien Chang
IACR erint (2021)
`https://eprint.iacr.org/2021/989`

Summary. This paper propose stateful KEM combiners that combines multiple stateless KEMs. The combiner is claimed to have amortized sublinear blow-up in terms of running time per encapsulation/decapsulation, as opposed to stateless combiners constructed in related works that essentially need to run at least every component KEMs once per encapsulation/decapsulation.

> More concretely, the stateful KEM combiners work as follows. A combined KEM have a public key $\mathsf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$ and a secret key $\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ concatenated from ones of its component KEMs. For every $1 \le i \le n$, let $\mathsf{Encap}_i, \mathsf{Decap}_i$ be the encapsulation, decapsulation of the ith component KEM respectively. At each of the ith session, the session key K_i is generated via
>
> $$S_i := H\left(\sum_{0 \le j < n} k_{i-n+1+j} \cdot r^{j+1} \bmod p\right),$$
>
> where p is a large enough prime number, $r \leftarrow \mathbb{F}_p{}^{\times}$ is a randomly sampled non-zero elements, $H(x \cdot k + y)$ is a PRF as a function of (x, y) and key k of suitable format, and k_j is generated via the component encapsulation $\mathsf{Encap}_{j \bmod n}$. The ciphertext C_i of that session is then $(c_{i-n+1}, \ldots, c_i)$ where every c_j is the ciphertext corresponding to k_j. Decapsulation is then performed in the obvious way. To reach a stronger CCA-flavored security, an additional MAC tag $\mathsf{MAC}(i, C_i)$ is included as part of the ciphertext, which is then verified in decapsulation with explict abort if it fails. The state of such a combined KEM essentially keep track of $\sum_{0 \le j < n} k_{i-n+1+j} \cdot r^{j+1} \bmod p$ for every i so that S_i can be evaluated efficiently enough per encapsulation/decapsulation.

In terms of security guarantee, assuming that one of the component KEM is IND-CPA secure, then for a combiner constructed in the paper, the combined stateful KEM satisfies a customized notion called *selective-session* IND-CPA, and another construction satisfies its IND-CCA counterpart.

Remark 3. *If one allows "stateful" KEMs as considered in this work then one can use an off-the-shelf pseudorandom generator to stretch a single key, obtained from any KEM combiner, arbitrarily. We do not see why this would not beat the construction suggested in this work.*

Practical (post-quantum) key combiners from one-wayness and applications to TLS ([ADK+22])

N. Aviram, B. Dowling, I. Komargodski, K. Paterson, E. Ronen, and E. Yogev
IACR eprint (2022)
`https://eprint.iacr.org/2022/065.pdf`

Summary. The paper has the following three main contributions.

1. The paper revisits HKDF, a particular hash-based key derivation function that is used in real-world protocols as TLS 1.3 and the Signal protocol, and shows that the security assumptions made on HKDF when analyzing these protocols do not match the existing literature on HKDF. In particular, the paper argues that in the analyses of these protocols, HKDF.Extract is assumed to be a dual-PRF (see Def. 1); however, there is no proof in the literature that would show that HMAC is indeed a dual-PRF under standard assumptions.

> For completeness, we briefly discuss HKDF, which is separated into the phases HKDF.Extract and HKDF.Expand. HKDF.Extract extracts a pseudorandom key from high-entropy (but not necessarily uniformly random) initial key material ikm and from uniformly random (but non-secret) salt. It is specified as $\mathsf{HKDF.Extract}(salt, ikm) := \mathsf{HMAC}(salt, ikm)$, where the salt is used as key in HMAC, and where
>
> $$\mathsf{HMAC}(K, m) := \mathsf{HASH}\big((K' \oplus opad) \| \mathsf{HASH}((K' \oplus ipad) \| m)\big),$$
>
> for fixed padding strings, and where $K' := \mathsf{HASH}(K)$ if K is larger than the block size of HASH, and $K' := K$ otherwise. On the other hand, roughly speaking, HKDF.Expand maps a pseudorandom key of a given size into a longer pseudorandom key with a size that can be specified.

2. The paper introduces the following construction

$$\begin{array}{ll} k_1 \leftarrow F(K_1), & u_1 \leftarrow g(K_1), \\ k_2 \leftarrow F(K_2), & u_2 \leftarrow g(K_2), \end{array} \qquad Y \leftarrow H\big(\mathsf{PRF}(k_1, 2\|u_2) \oplus \mathsf{PRF}(k_2, 1\|u_1)\big)$$

   and proves that it is a dual-PRF under certain standard assumptions on F, g, PRF and H. In more detail, g must be an injective oneway function, F a computational-extractor with respect to g, PRF a pseudorandom function, and H (almost-)regular.

3. The paper makes a concrete suggestion on how to instantiate (a salted version of) the above dual-PRF construction, arguing it to provide a practical and provably secure key combiner.

> Concretely, the *injective onewayness* is reduced to 2-*universality* and *same-input onewayness*, and it is argued that these assumptions are much weaker than collision resistance (and may thus hold e.g. for MD5). As for the extractor F, it is proposed to set $F(K, salt) := \mathsf{HMAC}(salt, K)$, using an optional salt value (which by default is all zero). This choice for F is well-established: if the compression function underlying the hash function is a good extractor then so is HMAC. Finally, it is suggested to set $\mathsf{PRF}(k, u) := \mathsf{HMAC}(k, u)$ and $H := \mathsf{HASH}$.

   Furthermore, the paper provides benchmarks (Table 1) for the proposed construction, and it is explained how to apply the proposed construction to the key schedule of TLS 1.3.

Adaptive versus static multi-oracle algorithms, and quantum security of a split-key PRF ([DFH22])

Jelle Don, Serge Fehr, and Yu-Hsuan Huang
TCC 2022
`https://eprint.iacr.org/2022/773`

Summary. On the combiner front, the paper follows up on the observation that the hash based constructions of split-key PRFs in [GHP18] (discussed on page 7) are proven in the classical ROM only, and thus are not proven secure against quantum attacks. Indeed, motivated by this observation, it offers a security proof in the QROM of the generic and particular efficient split-key PRF constructed

$$W(k_1, \ldots, k_n, x) = H(g(k_1, \ldots, k_n), x) \ ,$$

where g satisfies the statistical property as outlined in the box in the discussion of [GHP18]). As a direct consequence, KEM combiners obtained by such a split-key PRF offers security against quantum attacks (assuming that at least one of the underlying KEMs is secure against quantum attacks).

A crucial ingredient to the proof is a new, generic compiler that turns any oracle algorithm $\mathcal{A}^{\mathcal{O}_1,\ldots,\mathcal{O}_n}$, which has (possibly quantum) access to multiple oracles $\mathcal{O}_1, \ldots, \mathcal{O}_n$ and may decide *adaptively* for each query which oracle to query, into an oracle algorithm $\mathcal{B}[\mathcal{A}]^{\mathcal{O}_1,\ldots,\mathcal{O}_n}$ that has a predefined pattern on which oracle to query when, *and* that has a mild blow-up in the *individual* query complexities to the different oracles: if $\mathcal{A}$ makes q_i queries to $\mathcal{O}_i$ then $\mathcal{B}[\mathcal{A}]$ makes at most nq_i queries to $\mathcal{O}_i$.

Tightly secure PKE combiner in the quantum random oracle model ([GM22])

Brian Goncalves, and Atefeh Mashatan
Cryptography 2022
`https://doi.org/10.3390/cryptography6020015`

Summary. This paper propose a public-key encryption (PKE) combiner QuAKe in a "semi-blackbox" setting. It combines a general PKE with one the follows the KEM-DEM construction design, i.e., that is obtained by agreeing on a key using a KEM, and then encrypting the message with a symmetric encryption scheme (which in this context is then called data encapsulation method). The paper argues that if either of the component PKE is IND-CCA, then so is the combined PKE.

Roughly speaking, the concrete description works in a cascading fashion, with additional (salted) de-randomization and re-encryption in order to test the validity of a ciphertext while decryption, and explicitly abort if the ciphertext is invalid.

> More in to the detail, let $(\mathcal{K}, \Pi_{sym})$ be a KEM-DEM scheme, with the encryption key being an encapsulation (resp. decapsulation) key ek (resp. dk) for the KEM $\mathcal{K}$, and let Π_{asym} be another public-key encryption scheme with encryption (resp. decryption) key being pk (resp. sk). The combiner QuAKe takes the concatenation of encryption (resp. decryption) key as its own key, and on input m to encrypt, it first samples a random salt $\delta \leftarrow \{0,1\}^\ell$, encapsulate a session key k with ciphertext C_{KEM} via $\mathcal{K}.\mathsf{Encap}(ek; H_1(\delta))$ but replacing the involved randomness to some hash output of a random oracle H_1. Then encrypt $m \| \delta$ via Π_{sym} under the key k, which produce a symmetric ciphertext C_{DEM}. Finally, deterministically encrypt C_{KEM} with the public key pk and produce a ciphertext via $C_{PKE} \leftarrow \Pi_{asym}.\mathsf{Enc}(pk, C_{DEM}, H_2(\delta))$ where H_2 is another random oracle. The overall ciphertext is then $C := (C_{KEM}, C_{PKE})$. The decryption of the combined PKE then works in the obvious way, but additionally check the validity of the ciphertext (C_{KEM}, C_{PKE}) via re-encryption (and abort if it is invalid).

The paper claims to be the first PKE combiner with IND-CCA security. The reduction is performed in the random oracle model, both classically and quantum. Furthermore, the paper features tight security reduction from the combined to the IND-CCA securities of both components, where the tightness holds against both quantum and classical adversaries.

Hybrid key exchange in TLS 1.3 ([SFG23])

Douglas Stebila, Scott Fluhrer, and Shay Gueron
IETF Online Document (2020)
`https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/`

Summary. The document specifies a couple of ways to implement hybrid key exchange in TLS 1.3 via the use of combiners: one is "X25519Kyber768Draft00" that combines "x25519" and "Kyber768"; the other is "SecP256r1Kyber768Draft00 " that combines "secp256r1" and "Kyber768". In both cases, the two component keys are combined by applying the key-derivation-function HKDF.Extract to the concatenation of the two keys (see [SFG23, Fig. 1]).

Multiple key exchanges in the internet key exchange protocol version 2 (IKEv2) ([TTB+23])
C. Tjhai, M. Tomlinson, G. Bartlett, S. Fluhrer, D. Van-Geest, O. Garcia-Morchon, and V. Smyslov
IETF Online Document (2023)
`https://datatracker.ietf.org/doc/draft-ietf-ipsecme-ikev2-multiple-ke/`

Summary. The document specifies a way to extend the IKEv2 protocol to run multiple key exchanges during the so-called security association (SA) set-up that computes a shared secret key via the use of a combiner. Furthermore, the component key exchanges are backward compatible, in the sense that, if either one of the peers in a component exchange do not agree/support to use the additional key, then it falls back to a shared secret previously specified. The document also takes into consideration (and handles) the issue that some post-quantum key exchanges may exceed the *maximum transmission unit* (MTU) size limit.

On multiple encryption for public-key cryptography ([SBM23])
Tudor Soroceanu, Nicolas Buchmann, and Marian Margraf
Cryptography 2023
`https://www.mdpi.com/2410-387X/7/4/49`

Summary. This is a survey paper on the combiners that result in PKEs, which the is referred to as *public-key multiple encryption schemes* (M-PKE) in the paper. Constructions proposed in related work are described and classified according to their design, e.g. whether ciphertexts are generated "sequentially" (e.g. by doing some sort of double encryption) or "in parallel" from its component schemes. The paper compares these proposals in [SBM23, Table 1] with respect to its design principle, security guarantee, proof model, ciphertext sizes, and whether there are additional primitives being used. In addition, it also briefly discuss is a paragraph the efficiency aspects, and provides recommendations on which combiner to use depending on four aspects: ciphertext sizes, run time, additional primitives, and quantum resistance.

3.2 Hybrid digital signature schemes

Transitioning to a quantum-resistant public key infrastructure ([BHMS17])
Nina Bindel, Udyani Herath, Matthew McKague, and Douglas Stebila
PQCrypto 2017
`https://eprint.iacr.org/2017/460`

Summary. This paper concerns hybrid signatures. The claimed contribution of this paper is three-fold:

1. The paper introduces and studies more fine-grained security notions for (ordinary and combined) digital signatures, depending on how quantum the adversary is. In more detail, it considers the notion of X^yZ-unforgeability with $X, Z \in \{\mathsf{Q}, \mathsf{C}\}$ and $y \in \{\mathsf{q}, \mathsf{c}\}$, where X being C or Q means that the attacker is classical or quantum *at the time it can interact with the signing oracle*, y being c or q means that these *signing queries* are classical or quantum, and Z being C or Q means that the attacker is classical or quantum *after the period during which it could interact with the signing oracle*. The paper confirms the expected implications, i.e., $\mathsf{Q}^\mathsf{q}\mathsf{Q}$-unforgeability implies $\mathsf{Q}^\mathsf{c}\mathsf{Q}$-unforgeability, which implies $\mathsf{C}^\mathsf{c}\mathsf{Q}$-unforgeability, which in turn implies $\mathsf{C}^\mathsf{c}\mathsf{C}$-unforgeability, and it proves that all these implications are strict.

Specifically for signature schemes obtained via a combiner, the paper introduces the notion of *non-separability*, which requires that from a signature under the combined scheme it should be hard to extract a signature under any of the component schemes (for the same message); it is claimed that this notion is interesting in the context of a transition.

Remark 4. *We could not convince ourselves of the usefulness or necessity of the considered non-separability security notion. In a similar vein and in line with Remark 1, it remains unclear what should be the motivation to consider security notions weaker than* $\mathsf{Q}^{\mathsf{c}}\mathsf{Q}$*-unforgeability.*

2. The paper analyses the trivial combiner, which simply signs the given message individually under the two component signature schemes, and some variants of nested signing, in the light of the above security definitions. Obviously, the trivial combiner not non-separable. See the box for more details. In all the considered combiners, the key generation procedures simply concatenate public and secret keys, respectively, but the creation (and verification) of the combined signature $\sigma(m)$, for the to-be-signed message m, is different.

For the trivial "concatenation" combiner
$$\sigma(m) = \big(\sigma_1(m), \sigma_2(m)\big)$$
it is shown that if either of the component schemes is $X^{y}Z$-unforgeable, then so is the combined scheme, for any (meaningful) choices of X, y and Z; while, trivially, this combiner is not non-separable.

For the weakly nested combiner
$$\sigma(m) = \Big(\sigma_1(m), \sigma_2\big(\sigma_1(m)\big)\Big)$$
it is shown that if the first component is$X^{y}Z$-unforgeable then so is the combined scheme, and if the second component is $X^{y}Z$-unforgeable then the combined scheme is $X^{\mathsf{c}}Z$-non-separable.

For the strongly nested combiner
$$\sigma(m) = \Big(\sigma_1(m), \sigma_2\big(m, \sigma_1(m)\big)\Big)$$
it is shown that if either of the component schemes is $X^{y}Z$-unforgeable, then so is the combined scheme, and if the second component is $X^{y}Z$-unforgeable then the combined scheme is $X^{\mathsf{c}}Z$-non-separable.

3. This paper also performs experimental evaluation on hybrid certificates in several standard protocols: X.509, TLS, and S/MIME, in terms of not only performance, but also the so-called backward compatibility.

Multiple public-key algorithm X.509 certificates ([TLG+18])

Alexander Truskovsky, Philip Lafrance, Daniel Van Geest, Scott Fluhrer, Panos Kampanakis, Mike Ounsworth, and Serge Mister

IETF Online Document (2018)

`http://www.ietf.org/internet-drafts/draft-truskovsky-lamps-pq-hybrid-x509-00.txt`

Summary. This is an online document that expired in 2018. This document concerns backward compatibility for hybrid X.509 certificates, certificate revocation lists (CRLs) and PKCS #10 certificate signing requests (CRSs). Concretely, it specifies a method to include an alternative signature and a public key, e.g. generated via a post-quantum scheme, into an X.509v3 certificate, in addition to the existing "conventional" one. Moreover, the alternative signatures and public keys are included in such a way that still allow out-dated legacy systems to successfully issue and verify it. In the document, the format of such an extended certificate, as well as how it should be processed, are specified in detail.

The described extension to such a certificate is typically marked by its issuer as *non-critical*. Namely, it will only be processed by updated systems that can recognize such extension, but ignored by the out-dated legacy systems. This is in contrast to a *critical* extension, which will incur rejection whenever it cannot be recognized by the certificate verifier. In case such extension is recognizable, the verifier in effect treats such a certificate as one with signatures combined via the concatenation combiner.

The viability of post-quantum X.509 certificates ([KPDG18])

Panos Kampanakis, Peter Panburana, Ellie Daw, and Daniel Van Geest
IACR eprint (2018)
`https://eprint.iacr.org/2018/063`

Summary. This paper concerns scenarios when hybrid X.059 certificates, i.e. certificates containing both pre-quantum (in particular RSA, ECDSA) and post-quantum signatures (here, HSS) and public keys, are deployed to protocols that use X.059 certificates: TLS, DTLS, QUIC, and IKEv2. The paper provides pen-and-paper evaluations for potential issues of such certificates, and also some experimental support. In experiments, the implemented post-quantum part of certificates are based on the hash-based HSS signature scheme, and a combiner as in [BHMS17] (discussed on page 12) is implemented. The provided evaluations depend on protocols where the hybrid certificates are applied to:

- For (D)TLS, the effect of post-quantum signatures are evaluated in various aspect. First, the so-called fragmentation mechanism may be affected in that larger certificates is going to yield more fragments and delays in order to transmit the same piece of information. In experiment, with OpenSSL, it is reported that 70% more packets are required in order to complete a handshake. It is also reported, for a certificate (chain) that does not exceed 16KB TLS record limit, the fragmentation mechanism behaves correctly. Second, in terms of overhead per connection, it is pointed out that for certain protocols that uses TLS under the hood, e.g. HTTP/2, there exists a multiplexing mechanism that amortizes the overhead per connection. Such mechanism is absent in HTTP/1. Third, it is pointed out that caching a previous certificate may reduce the overhead, but it doesn't generally apply because of security concerns. Last, it is also pointed out that the so-called certificate compression mechanism for a post-quantum signature gives negligible improvement, while additionally introduces concerns about the Denial of Service (DoS).
- For QUIC, the paper reported that there is no practical issue.
- For IKEv2, there is no authenticated message being exchanged after a "tunnel" is established. Therefore, during the lifetime of a tunnel, the amortized overhead per message is expected to be small. The paper further confirms that it works correctly with post-quantum signatures.

In conclusion, the paper examines the viability of using a post-quantum signature in X.059 hybrid certificates, and in terms of correctness no particular issue were raised (when without resource constraints).

X.509-compliant hybrid certificates for the post-quantum transition ([BBG+19])

Nina Bindel, Johannes Braun, Luca Gladiator, Tobias Stöckert, and Johannes Wirth
Journal of Open Source Software 4 (2019)
`https://www.theoj.org/joss-papers/joss.01606/10.21105.joss.01606.pdf`

Summary. This paper reports on a Java implementation for BouncyCastle that realizes the hybrid certificates suggested in [BHMS17] fully compliant to the X.509 standard. The implementation is available at `https://github.com/CROSSINGTUD/bc-hybrid-certificates` ("Hybrid Certificates - Java, Bouncy Castle integration", 2019).

In this implementation, the standard signature and public-key fields of the X.509 certificate are used for one of the signature schemes. For the post-quantum transition, the standard fields are used for the classical scheme. This allows compatibility with clients that do not support hybrid signatures. The second signature scheme, using qTESLA as an example, is integrated using two non-critical X.509 extensions. One of the extensions contains the public key associated with the second scheme, while the other contains the second signature on the certified data. To fully support legacy entities in a controlled manner, the extension containing the second public key may optionally be left out. This explicitly states that the certified entity does not support post-quantum schemes yet, while the certificate contents themselves are still protected in a hybrid fashion.

Performance of hybrid signatures for public key infrastructure certificates ([Lyt21])

John Lytle

Master's Thesis, 2021.

`https://apps.dtic.mil/sti/trecms/pdf/AD1204814.pdf`

Summary. This master's thesis provides a nice overview of relevant conventional and post-quantum signature schemes, and approaches of combining them to obtain hybrid security. It considers concatenation and nesting, but has a particular focus on what is called *true hybrid schemes* in the thesis. If we understand correctly, the latter refers to hybrid schemes that are *not* obtained by combining two schemes *in a black-box way*. For example, "concatenating" two interactive proofs of knowledge and then applying the Fiat-Shamir transformation is considered a true hybrid scheme, in contrast to first applying the Fiat-Shamir transformation individually and then combining the two resulting signature schemes via concatenation.

In order to compare and contrast performance, the true hybrid digital signature schemes considered in the thesis are implemented within a common cryptographic framework, and their performance is evaluated against traditional hybrid techniques. The results show that specific true hybrid signature schemes introduce negligible overhead when compared to concatenated hybrid schemes using the same component algorithms. The results also show that certain true hybrid combinations add additional computational overhead; in these examples, the efficiency decrease is directly influenced by how the true hybrid scheme interacts with the component algorithms.

Remark 5. *The relevance of* true *hybrid schemes remains unclear to us, and hence why the particular focus of this thesis. We have not come across this notion of* true hybrid schemes *in any other work.*

The thesis also explores how hybrid digital signatures could be integrated into existing X.509 certificates and examines their performance by integrating both into the TLS 1.3 protocol. The thesis confirms that the larger size of hybrid digital certificates and the increase in computational processing required to run two digital signature algorithms within a hybrid scheme have a significant impact on the total handshake time. Additionally, it is observed that integrating hybrid signatures into existing protocols within crytographic libraries is a non-trivial task: implementation requires an in-depth knowledge of the existing protocol standards, the cryptographic library internals, and the security features of the programming language it is written in.

Impact of post-quantum hybrid certificates on PKI, common libraries and protocols ([FWZ+21])

Jinnan Fan, Fabian Willems, Jafar Zahed, John Gray, Serge Mister, Mike Ounsworth, and Carlisle Adams

Int. J. Security and Networks (2021)

`https://www.inderscienceonline.com/doi/pdf/10.1504/IJSN.2021.117887`

Summary. This paper asses the impact of particular post-quantum (PQ) cryptography on public key infrastructures, with a special focus on the most significant change from traditional cryptography: large public keys and digital signatures. To do so, the authors employs the template for hybrid certificates as defined in the IETF Internet Draft for hybrid certificates [TLG+18] (discussed on page 13), and they then consider modified CA that is capable of issuing hybrid certificates, which contain both an RSA and a post-quantum public key and signature (using SPHINCS$^+$ for the latter). The impact of using these certificates is then tested on various existing protocols, including TLS, OCSP, CMP, and EST, with open-source libraries OpenSSL and CFSSL, and with a commercially available cryptographic toolkit.

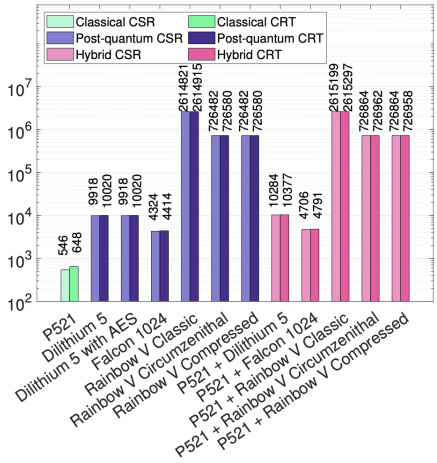
A particular goal is to evaluate the backwards compatibility of these protocols when using post-quantum hybrid certificate of varying sizes. Based on the test results, the paper highlights the protocols that will not require changes, and it determines the maximum certificate size that will not have backwards compatibility issues with the software we tested.

The paper finds that most of the protocols and libraries that were tested are well equipped to work with hybrid certificates, and some of the failures could be overcome with minor patches and updates to the existing software.

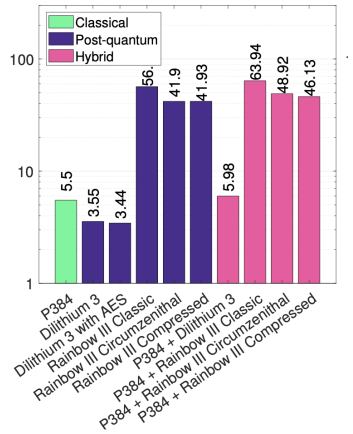
Performance characterization of post-quantum digital certificates ([RCW+21])

Manohar Raavi, Pranav Chandramouli, Simeon Wuthier, Xiaobo Zhou, and Sang-Yoon Chang
ICCCN 2021
`https://par.nsf.gov/servlets/purl/10324007`

Summary. The paper analyses extensively the effect on the performance when making the X.509 PKI post-quantum secure, with a focus on the computation time and the memory overhead for *key-pair generation*, *certificate signing request generation*, *certificate generation* and *certificate verification*. For this purpose, the paper compares the performance of the post-quantum algorithms (using the NIST finalists for this purpose), the classical algorithms (RSA and ECDSA), and the hybrid-schemes obtained by combining both the post-quantum and classical ones. It does so for the NIST Security Level 1 or 2 schemes, the Security Level 3 schemes, and the Security Level 5 schemes. See Table I to III in the paper for all the combinations and variations that were implemented for the study. All the memory costs and all the computation times are plotted in various figures in the paper (see the box below for a sample).



File size (in bytes) for *certificate signing requests* (CSR) and *end-entity certificate* (CRT) for classical, post-quantum, and hybrid algorithms, achieving Security Level 5 (Fig. 2(c) in the paper).



Certificate verification time (in ms) for classical, post-Quantum, and hybrid algorithms, achieving Security Level 3 (Fig. 6(b) in the paper).

Some of their conclusions are:

- Dilithum is the fastest for key generation, CSR generation and certificate generation.
- Falcon is the fastest for certificate verification.
- Using hybrid algorithms does not have a noticeable impact over the use of only post-quantum.

Obviously, Dilithum is the recommended for time-sensitive applications (any web server-client communication) but Falcon is a good candidate for Blockchain certificate verification.

Hybrid post-quantum signatures in hardware security keys ([GKP+23])

Diana Ghinea, Fabian Kaczmarczyck, Jennifer Pullman, Julien Cretin, Stefan Kölbl, Rafael Misoczki, Jean-Michel Picod, Luca Invernizzi, and Elie Bursztein.
`https://ia.cr/2022/1225`

Summary. The paper implements a hybrid signature scheme combining ECDSA and Dilithium, in the context of hardware security keys. On the top level, the signature uses the so-called weakly nested combiner as in [BHMS17], and hence preserving the (pre-quantum and post-quantum) unforgeability from both ECDSA and Dilithium.

The Dilithium part of its implementation deviates from the reference implementation, in order to achieve practical requirements specified by the *client to authenticator protocol* (CTAP), e.g. requirement R3 regarding key and/or signature sizes. See [GKP+23, Section 5.1] for the CTAP requirements. In addition, despite such deviation (from the reference implementation), the authors argues that their implementation is side-channel resistant due to the branches being independent of the secret key.

Finally, the performance of such an implementation is evaluated. The considered metrics include running time but also stack memory usage. The implementations of ECDSA, different modes of Dilithium are compared in various dimensions, including the adopted compilation flag, whether they are in hybrid with another scheme, the signing or key generation. In addtion the performance is also compared with relevant works.

Key generation	Stack (in kB)	Runtime (ms) Pure	Runtime (ms) Hybrid	Signing	Stack (in kB)	Runtime (ms) Pure	Runtime (ms) Hybrid
ECDSA	0.3	115.7		ECDSA	3.0	188.0	
Dilithium2 (speed mode)	41.6	70.3	192.0	Dilithium2 (speed mode)	77.1	420.4	687.8
Dilithium2	14.4	82.3	207.5	Dilithium2	17.0	1053.1	1417.5
Dilithium3	19.4	142.4	258.5	Dilithium3	17.9	2077.3	2420.7
Dilithium5	21.4	271.4	393.1	Dilithium5	19.2	3305.1	3378.5

Table 3 in [GKP+23] which compares run time and stack memory of the implemented pure and hybrid schemes.

The paper concludes that the considered hybrid signatures are indeed feasible, even for the highest security mode of Dilithium, which is the most resource-consuming implementation in the paper.

Post-quantum hybrid digital signatures with hardware-support for digital twins

Saif E. Nouma, and Attila A. Yavuz

Applied Cryptography and Network Security Workshops (2023)

`https://doi.org/10.48550/arXiv.2305.12298`

Summary. This paper develops a family of the so-called hardware-assisted efficient signatures (HASES) that is designed light-weight for digital twins in the context of IoT. Tha authors implements HASES, evaluates its performance experimentally, provides pen-and-paper justification of its security, and open-sources the implementation. See summary of the achieved securities and performance comparison in [NY23, Table I, II].

HASES consists of three signature schemes: PQ-HASES, LA-HASES, and HY-HASES. As their names suggest, PQ-HASES constructed from the hash-based HORS is post-quantum, LA-HASES is pre-quantum (from `Ed25519` elliptic curves), and HY-HASES is combined from the previous two, via the strongly nested combiner as described in [BHMS17].

The HASES features fast signing speed, high energy efficiency, compact signatures and secret keys, with a trade-off of larger public keys according to [NY23, Table I]. In addition, it is also claimed to be one of the few constructions with forward-security.

Remark 6. *Several relevant parts in the paper are not fully clear to us, including the precise meaning of forward-security, backward compatibility, and its security justification.*

3.3 Hybrid KEMs and Signatures

Post-quantum security for the extended access control protocol ([FvdHM+23])
Marc Fischlin, Jonas v.d. Heyden, Marian Margraf, Frank Morgner, Andreas Wallner, and Holger Bock
SSR 2023
`https://eprint.iacr.org/2023/352`

Summary. The article considers the *Extended Access Control (EAC)* protocol for authenticated key agreement, which is used to secure connections between machine-readable travel documents and inspection terminals, and it proposes a quantum-resistant version of the protocol *(PQ-EAC)*, as well as a hybrid version that uses combiners. PQ-EAC uses Dilithium3 for signing and Kyber1024 as KEM, and the proposed hybrid scheme uses XOR-then-MAC for combining the KEMs, and strong nesting for the signatures. The article offers a formal security proofs for PQ-EAC, as well as an experimental implementation of the combined PQ-EAC (without forward security), showing practical feasibility under typical circumstances.

Post-quantum hybrid KEMTLS performance in simulated and real network environments ([GdNC+23])
Alexandre Augusto Giron, João Pedro Adami do Nascimento, Ricardo Custódio, Lucas Pandolfo Perin, and Víctor Mateu
LATINCRYPT 2023
`https://eprint.iacr.org/2022/1639`

Summary. KEMTLS is an approach replacing signatures in TLS by KEM, with the purpose of obtaining smaller transcripts. This paper concerns hybrid KEMTLS, where the abovementioned KEM is then obtained via using dual-PRF combiner, and additionally it also concerns so-called KEMTLS-PDK protocols. The paper performs experimental evaluation on several relevant performance metric: handshake completion time, time-to-send-app-data, hybrid penalty, HTTPS/TLS request successes and failures, and server-side memory load, which are compared among hybrid KEMTLS, post-quantum only KEMTLS, and TLS using hybrid signatures. The paper concludes that the performance overhead posed by the considered hybrid approaches is minor.

4 Conclusion

There is some literature available on the topic of combiners and hybrid security (for KEMs and digital signature schemes), but the actual number of works that we found in our literature search is at the lower end of what we would have expected (as already mentioned earlier, we may we have missed some articles).

On the theory side, the articles mostly introduce and solve rather isolated and independent technical questions. It seems that there is no broadly accepted coherent theory of combiners (yet), nor appears there to be any milestone publication or standard reference on the topic. This is also reflected by the relatively low citation numbers; all the articles covered in the report have a citation count less than 100, and only a few have more than. There are a few established concepts, but many articles come with tailor-made ad-hoc definitions. Some even introduce and study (security) definitions whose relevance remains unclear to us.

Looking at the (more) experimental papers, most of the work has been made in sight of possible efficiency issues due to the bigger sizes to transport and the longer computational overhead. The most studied protocols for hybrid security are TLS 1.2, TLS 1.3 and SSH; other protocols have not received much attention. The main focus is on the compatibility into the current implementations of TLS and other protocols that use digital certificates. The common message seems to be that, as expected, achieving hybrid security comes with a non-negligible overhead in efficiency, and it may cause some complications in the deployment, but in the end the conclusion is typically positive.

Acknowledgements

We would like to thank Gabriele Spini (TNO) for many fruitful discussions about this report, and beyond. We would like to thank Stefan van den Berg (TNO) for useful comments on an earlier version of this report.

References

[ADK+22] Nimrod Aviram, Benjamin Dowling, Ilan Komargodski, Kenneth G. Paterson, Eyal Ronen, and Eylon Yogev, *Practical (post-quantum) key combiners from one-wayness and applications to TLS*, Cryptology ePrint Archive, Paper 2022/065, 2022, `https://eprint.iacr.org/2022/065`.

[BBD+23] Manuel Barbosa, Gilles Barthe, Christian Doczkal, Jelle Don, Serge Fehr, Benjamin Grégoire, Yu-Hsuan Huang, Andreas Hülsing, Yi Lee, and Xiaodi Wu, *Fixing and mechanizing the security proof of Fiat-Shamir with aborts and Dilithium*, Advances in Cryptology – CRYPTO 2023 (Helena Handschuh and Anna Lysyanskaya, eds.), Springer, Cham, 2023, pp. 358–389.

[BBF+19] Nina Bindel, Jacqueline Brendel, Marc Fischlin, Brian Goncalves, and Douglas Stebila, *Hybrid key encapsulation mechanisms and authenticated key exchange*, Post-Quantum Cryptography (Jintai Ding and Rainer Steinwandt, eds.), Springer, Cham, 2019, pp. 206–226.

[BBG+19] Nina Bindel, Johannes Braun, Luca Gladiator, Tobias Stöckert, and Johannes Wirth, *X.509-compliant hybrid certificates for the post-quantum transition*, Journal of Open Source Software **4** (2019), no. 40, 1606.

[BHMS17] Nina Bindel, Udyani Herath, Matthew McKague, and Douglas Stebila, *Transitioning to a quantum-resistant public key infrastructure*, Post-Quantum Cryptography (Tanja Lange and Tsuyoshi Takagi, eds.), Springer, Cham, 2017, pp. 384–405.

[CD23] Wouter Castryck and Thomas Decru, *An efficient key recovery attack on SIDH*, Advances in Cryptology – EUROCRYPT 2023 (Carmit Hazay and Martijn Stam, eds.), Springer, Cham, 2023, pp. 423–447.

[CPS19] Eric Crockett, Christian Paquin, and Douglas Stebila, *Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH*, Cryptology ePrint Archive, Paper 2019/858, 2019, `https://eprint.iacr.org/2019/858`.

[DFH22] Jelle Don, Serge Fehr, and Yu-Hsuan Huang, *Adaptive versus static multi-oracle algorithms, and quantum security of a split-key PRF*, Theory of Cryptography (Eike Kiltz and Vinod Vaikuntanathan, eds.), Springer, Cham, 2022, pp. 33–51.

[FvdHM+23] Marc Fischlin, Jonas von der Heyden, Marian Margraf, Frank Morgner, Andreas Wallner, and Holger Bock, *Post-quantum security for the extended access control protocol*, Security Standardisation Research (Felix Günther and Julia Hesse, eds.), Springer, Cham, 2023, pp. 22–52.

[FWZ+21] Jinnan Fan, Fabian Willems, Jafar Zahed, John Gray, Serge Mister, Mike Ounsworth, and Carlisle Adams, *Impact of post-quantum hybrid certificates on PKI, common libraries, and protocols*, Int. J. Secur. Networks **16** (2021), no. 3, 200–211.

[GdNC+23] Alexandre Augusto Giron, João Pedro Adami do Nascimento, Ricardo Custódio, Lucas Pandolfo Perin, and Víctor Mateu, *Post-quantum hybrid KEMTLS performance in simulated and real network environments*, Progress in Cryptology – LATINCRYPT 2023 (Abdelrahaman Aly and Mehdi Tibouchi, eds.), Springer, Cham, 2023, pp. 293–312.

[GHP18] Federico Giacon, Felix Heuer, and Bertram Poettering, *KEM combiners*, IACR International Workshop on Public Key Cryptography, Springer, 2018, pp. 190–218.

[GKP+23] Diana Ghinea, Fabian Kaczmarczyck, Jennifer Pullman, Julien Cretin, Stefan Kölbl, Rafael Misoczki, Jean-Michel Picod, Luca Invernizzi, and Elie Bursztein, *Hybrid post-quantum signatures in hardware security keys*, Applied Cryptography and Network Security Workshops (Jianying Zhou, Lejla Batina, Zengpeng Li, Jingqiang Lin, Eleonora Losiouk, Suryadipta Majumdar, Daisuke Mashima, Weizhi Meng, Stjepan Picek, Mohammad Ashiqur Rahman, Jun Shao, Masaki Shimaoka, Ezekiel Soremekun, Chunhua Su, Je Sen Teh, Aleksei Udovenko, Cong Wang, Leo Zhang, and Yury Zhauniarovich, eds.), Springer, Cham, 2023, pp. 480–499.

[GM22] Brian Goncalves and Atefeh Mashatan, *Tightly secure PKE combiner in the quantum random oracle model*, Cryptography **6** (2022), no. 2, 15.

[HDV21] Loïs Huguenin-Dumittan and Serge Vaudenay, *FO-like combiners and hybrid post-quantum cryptography*, International Conference on Cryptology and Network Security, Springer, 2021, pp. 225–244.

[KL07] Jonathan Katz and Yehuda Lindell, *Introduction to modern cryptography: principles and protocols*, Chapman and hall/CRC, 2007.

[KPDG18] Panos Kampanakis, Peter Panburana, Ellie Daw, and Daniel Van Geest, *The viability of post-quantum X.509 certificates*, Cryptology ePrint Archive, Paper 2018/063, 2018, `https://eprint.iacr.org/2018/063`.

[Lyt21] John Lytle, *Performance of hybrid signatures for public key infrastructure certificates*, Master's thesis, Naval Postgraduate School, California, USA, 2021.

[NY23] Saif E. Nouma and Attila A. Yavuz, *Post-quantum hybrid digital signatures with hardware-support for digital twins*, arXiv preprint arXiv:2305.12298 (2023).

[RCW+21] Manohar Raavi, Pranav Chandramouli, Simeon Wuthier, Xiaobo Zhou, and Sang-Yoon Chang, *Performance characterization of post-quantum digital certificates*, 30th International Conference on Computer Communications and Networks, ICCCN 2021, Athens, Greece, July 19-22, 2021, IEEE, 2021, pp. 1–9.

[SBM23] Tudor Soroceanu, Nicolas Buchmann, and Marian Margraf, *On multiple encryption for public-key cryptography*, Cryptography **7** (2023), no. 4, 49.

[SFG23] Douglas Stebila, Scott Fluhrer, and Shay Gueron, *Hybrid key exchange in TLS 1.3*, Internet-Draft draft-ietf-tls-hybrid-design-09, Internet Engineering Task Force, September 2023, Work in Progress.

[Sho94] Peter W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings 35th Annual Symposium on Foundations of Computer Science, IEEE, 1994, pp. 124–134.

[TLG+18] Alexander Truskovsky, Philip Lafrance, Daniel Van Geest, Scott Fluhrer, Panos Kampanakis, Mike Ounsworth, and Serge Mister, *Multiple public-key algorithm X.509 certificates*, Internet-Draft draft-truskovsky-lamps-pq-hybrid-x509-00, IETF Secretariat, March 2018, `http://www.ietf.org/internet-drafts/draft-truskovsky-lamps-pq-hybrid-x509-00.txt`.

[TTB+23] C. Tjhai, M. Tomlinson, G. Bartlett, Scott Fluhrer, Daniel Van Geest, Oscar Garcia-Morchon, and Valery Smyslov, *Multiple key exchanges in the internet key exchange protocol version 2 (IKEv2)*, RFC 9370, May 2023.

[XGL+21] Jia Xu, Yiwen Gao, Hoon Wei Lim, Hongbing Wang, and Ee-Chien Chang, *Stateful KEM: Towards optimal robust combiner for key encapsulation mechanism*, Cryptology ePrint Archive, Paper 2021/989, 2021, `https://eprint.iacr.org/2021/989`.