Post-quantum security of cryptographic transformations in the random oracle model

Huang, Y.

Chapter 1

Introduction

1.1 Provable Security

Cryptography has a history spanning several millennia. The cryptographic techniques were originally in rudimentary forms. They mainly consisted of ad-hoc tricks, most (if not all) of which, though becoming more sophisticated over time, did not withstand the test of time: they are eventually "broken" one way or another.

The cryptographic landscape changed with the seminal work of Shannon [Sha49], who took a scientific approach through mathematically defining and proving *information-theoretic security* of a certain cryptographic scheme. The works in late 1970s and early 1980s by Diffie and Hellman [DH76], and by Goldwasser and Micali [GM82] and others, further considered the notion of *computational security* and introduced formal security definitions, respectively. This marks the beginning of modern cryptography, where security is no longer based on the mere absence of known attacks, but on the study of rigorously defined mathematical notions — what we know today as the *provable security paradigm*.

Security definition. To be able to assess whether a given cryptographic scheme is secure, the first and immediate challenge is to formalize the right definition of the intended security notion. This may be accompanied with a *security game*, specifying in which way an adversary is allowed to interact with the cryptographic scheme, and what it means to "break" the scheme.

For a signature scheme $\mathcal{S} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ as an example, it might be natural to require in the security definition that it is hard for an attacker to forge a valid signature given the public key (only). This is formally captured in the $\mathsf{UF\text{-}NMA}$ security game[1] as in Fig. 1.1. The corresponding security

[1] The acronym stands for *(existential) unforgeability against no-message attacks.*

definition would then require that, for all (computationally bounded) attackers, it can only win the security game with a small probability.

However, an attacker in real life may fool the signer into signing certain messages. This kind of attack is not captured by the above notion of UF-NMA, and it motivates the stronger notion UF-CMA (see Fig. 1.1)[2], where an attacker is given *oracle access* to the signing procedure $\mathsf{Sign}(\mathsf{sk}, \cdot)$; in each query, it sends over a message m, and in return receives a signature $\sigma \leftarrow \mathsf{Sign}(sk, m)$ corresponding to that message. To make the attacker's task non-trivial, its goal here is to instead produce a forgery that is valid under a *fresh* message, i.e. a message that has never been queried to the oracle $\mathsf{Sign}(\mathsf{sk}, \cdot)$.

UF-NMA:	UF-CMA:
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}$	1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}$
2: $(m^*, \sigma^*) \leftarrow \mathcal{A}(\mathsf{pk})$	2: $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{pk})$
3: **return** $\mathsf{Vrfy}(\mathsf{pk}, m^*, \sigma^*) \stackrel{?}{=} 1$	3: $\{$let m_i be the ith query to $\mathsf{Sign}(\mathsf{sk}, \cdot)\}$
	4: **return** $\mathsf{Vrfy}(\mathsf{pk}, m^*, \sigma^*) \wedge m^* \notin \{m_i\}_i$

Figure 1.1: Security games UF-NMA and UF-CMA, where the attackers are denoted by $\mathcal{A}$.

Needless to say, different security games may result in differently powerful security definitions. In the above example, any UF-CMA signature scheme is also UF-NMA, but not vice versa — the Lamport signature scheme [Lam79] satisfies UF-NMA security (assuming the underlying one-way function is secure), but does not satisfy UF-CMA in that there is an explicit attack. The bottom line is: setting up proper security definitions has always been a central question in modern cryptography, which is by no means trivial: on one hand, a definition that is too weak may not rule out all relevant attacks; on the other hand, a definition that is un-necessarily strong may end up unachievable at all.

Security reduction. Once a security definition is set-up, one can proceed to prove it for a given scheme. However, a security proof of a scheme often implies resolution of the P vs NP problem, which is out of reach from the current mathematical tools. Therefore, oftentimes security proofs are conditional, of the following form: if a computational problem $\mathcal{P}$ is hard, then the considered scheme is secure (i.e. satisfying its corresponding security definition).[3]

For example, to prove security of a signature scheme, one would show that every attacker $\mathcal{A}$ winning the UF-CMA security game with non-negligible probability can be turned into a similarly efficient algorithm $\mathcal{B}$ that solves a computational problem $\mathcal{P}$ (again with non-negligible probability/advantage depending

[2]The acronym stands for *(existential) unforgeability against chosen-message attacks*

[3]Sometimes, there are multiple computational problems whose hardness is being relied on, but for the simplicity of exposition we did not bring this up here.

on $\mathcal{P}$ being a search/decision problem) that we believe to be hard. The quality of a reduction depends on the resources taken by $\mathcal{B}$ (such as the runtime $\mathsf{TIME}(\mathcal{B})$, or the number of queries that is made to the signing oracle), and the success probability $\mathbf{Adv}^{\mathcal{P}}(\mathcal{B})$ of $\mathcal{B}$ solving $\mathcal{P}$ — the less resources $\mathcal{B}$ takes and the higher probability $\mathcal{B}$ succeeds in solving $\mathcal{P}$, the stronger security is provided by such a security reduction. This is often quantitatively discussed, and the precise form may vary, such as the following hypothetical examples:

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) \leq q_S \cdot \mathbf{Adv}^{\mathcal{P}}(\mathcal{B}) \ , \quad \mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) \leq \sqrt{\mathbf{Adv}^{\mathcal{P}}(\mathcal{B})}$$
$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathcal{P}}(\mathcal{B}) + q_S \cdot \epsilon \ ,$$

where $\mathsf{TIME}(\mathcal{B}) \approx \mathsf{TIME}(\mathcal{A})$, q_S is the number of chosen-message queries $\mathcal{A}$ makes, and $\epsilon > 0$ is a quantity determined by the scheme $\mathcal{S}$. In each of these examples, there may be a gap between the theoretical guarantee obtained from the reduction, and the actual security of $\mathcal{S}$. We informally refer to such a gap as the *security loss*.

The point here is that, with security reductions, one can relate the security of a cryptographic scheme, with the hardness of better-understood computational problems, and thereby increase the confidence of the considered scheme. Moreover, the smaller security loss one has in a reduction, the better security guarantee one would get.

Random oracle model. For a cryptographic scheme that relies on a hash function H, ideally one can prove security of the scheme based on a simple computational assumption on the hash function, such as collision resistance. However, in many cases, the security requires that the hash function itself behaves as if it is a random function. Namely, every hash value looks random and unpredictable unless explicitly computed. Such an intuitive requirement is much more elusive, as it cannot be formalized as a property of H in the standard model (also known as the *plain model*).

To mitigate this difficulty, H is often idealized, treated as a random function (which we call a random oracle) chosen uniformly at the beginning of each relevant security game,[4] where an adversary is allowed to make oracle queries to H in order to learn its outputs on any inputs. This is commonly known as the *random oracle model* (ROM) today. Initially, the ROM was introduced in the context of computational complexity theory, without having any hash function in mind [BG81]. Soon after, it found relevance in cryptography, both for proving security of concrete constructions [FS87], and for establishing impossibility results [IR90]. In 1993, Bellare and Rogaway further popularized the idea of proving security of cryptographic schemes in the ROM [BR93].

[4] One may also consider a more fine-grained setting, where only a certain component of the hash function is idealized.

On one hand, the ROM facilitates provable security of many more efficient cryptographic constructions. In this model, a security reduction $\mathcal{B}$ is by default in charge of simulating the random oracle that interacts with the considered attacker $\mathcal{A}$. This is of great advantage for the reduction. For instance, $\mathcal{B}$ may observe $\mathcal{A}$'s queries to H, so, intuitively, it knows which hash values $\mathcal{A}$ knows and which ones not. On the other hand, modelling H as a random oracle is, after all, still a heuristic. For certain constructions, this heuristic is not sound. Specifically, there have been contrived separating examples that are provably secure in the ROM, but insecure for every concrete instantiation of the hash function [Bar01, GK03, CGH04]. For a period of time in the past, having a security proof (only) in the ROM was considered a major drawback.

Nevertheless, experience shows that natural schemes tend to remain secure, and since the ROM typically allows (proving security of) significantly more efficient schemes, it has remained a common methodology. Nowadays, while there is still some controversy, it is fair to say that the ROM is widely accepted: many schemes rely on the ROM, and looking ahead to the post-quantum security discussion and proposals for post-quantum secure schemes, there are very few non-random-oracle alternatives, even if one would be willing to accept worse efficiency.

A very recent work [KRS25] deserves special attention, as it demonstrates an attack against a natural scheme that is provably secure in the ROM. Again, the attack applies regardless of the concrete instantiation of the hash function. Despite ongoing controversy in the community about its impact on our confidence in the random oracle methodology, this attack still contains some aspect that does not appear in typical cryptographic schemes, such as the main focus of this thesis, signature schemes.

Post-quantum cryptography and the NIST competition. Early works of the last century [Deu85, BV93, Sim94] have provided strong evidence that the quantum model of computation is strictly more powerful than its classical counterpart. Therefore, provable security against classical attackers may not hold against quantum attackers. In fact, most of the public-key cryptographic schemes we use today, including the RSA and Diffie-Hellman families of constructions, are completely broken by Shor's (quantum) algorithm [Sho94]. Amidst global efforts to develop quantum computers, the study of cryptographic schemes that remain secure against quantum attackers, also known as *post-quantum cryptography* (PQC), is thus of undeniable importance.

Already in the late 20th century, there have been candidates that seem to resist quantum attackers, which, though, was not regarded a main feature originally. This includes early works of Lamport [Lam79], McEliece [McE78], Ajtai [Ajt96], Matsumoto and Imai [MI88],[5] and Couveignes (re-uploaded version

[5]The construction proposed in [MI88] was soon broken, but it inspired an important branch of PQC based on multivariate polynomials.

available at [Cou06]). With Shor's algorithm being a clear threat, the research greatly intensified, and security against quantum attackers turned into an explicit goal.

The search for practical post-quantum schemes further intensified in 2016, when the US National Institute of Standards and Technology (NIST) initiated a competition for selecting future standards of post-quantum cryptographic schemes, including public-key encryption (PKE)[6] and digital signature schemes. In this competition, experts around the world join forces into proposing future post-quantum standards, and assessing their security. Initially there were 82 candidates submitted to the first round; until now, 5 winners — Dilithium [LDK+22], Falcon [PFH+22], SPHINCS+ [HBD+22], Kyber [SAB+22], and HQC [AAB+22] — have been selected in the 3rd and 4th rounds; in addition, there is an extra round for signature schemes [NIST22], which has now proceeded to the second stage with 14 candidates still on the table.

Quantum impacts on provable security. The presence of quantum attackers impacts provable security in the following three-fold manner.

First of all, if a reduction transforms an attacker $\mathcal{A}$ of the considered scheme to an algorithm solving a computational problem $\mathcal{P}$, then $\mathcal{P}$ itself must remain hard for quantum computers to provide a meaningful security guarantee. Since 1990s, candidates of $\mathcal{P}$ have been extensively investigated, including those based on lattices, isogenies, multi-variate polynomial, coding theory, and symmetric cryptography.

Second, the security definition may require non-trivial modifications in order to capture our intuition, beyond simply extending the quantification to all quantum attackers. A notable example is the *collision resistance*, a property of a hash function H that prevents an attacker to find two distinct inputs x_1 and x_2 such that $H(x_1) = H(x_2)$. Classically, this reflects the intuition that a hash value $H(x)$ uniquely determines its preimage x, at least against efficient attackers. However, the very same intuition fails against quantum attackers, because collision resistance per se does not necessarily prevent a quantum attacker to provide a hash value, and later offer one or another preimage (but not both) on its choice. This gap is addressed in [Unr16], which introduces a stronger security notion for H, the *collapsing* property, that is sufficient to capture our intuition in the quantum case. In fact, as later shown in [DS23], the collapsing property is (essentially) also necessary.

Last but not least, a security reduction must also work in the presence of quantum attackers. While sometimes the reduction carries over directly, this is generally not the case. As a transformation from one algorithm to another, a reduction is by default tailored to the underlying model of computation. Changing the model to allow quantum computation can compromise

[6] Technically speaking, the competition selects key-encapsulation mechanisms (KEMs), which are essentially equivalent to PKE schemes due to the modern KEM-DEM paradigm.

the reduction's validity, meaning it may no longer solve the underlying computational problem successfully, and thus fail to provide meaningful security guarantees.

One of the earliest such examples appeared in [BDF+11], and since then, more prominent cases have been identified, especially in the ROM. Indeed, a typical classical security proof in the ROM exploits that the reduction $\mathcal{B}$ can observe the queries that the attacker $\mathcal{A}$ makes to the random oracle. If $\mathcal{A}$ is quantum though, then these queries can be made "in superposition." Namely, they would be in a quantum state that, by fundamental properties of quantum mechanics, $\mathcal{B}$ cannot observe without causing any disturbance — in which case $\mathcal{A}$ may notice and then simply shut itself down. Therefore, more often than not, a classical security proof in the ROM does not (naively) carry over to the quantum setting.

Another concrete example where classical security proofs fall apart in the quantum realm is when a reduction uses *rewinding*. That is, when $\mathcal{B}$ undoes $\mathcal{A}$'s computation to an earlier point in time, in order to re-run it from there later. Classically, this is realized by copying the internal state of $\mathcal{A}$ at the time to which it is to be rewound. However, if $\mathcal{A}$ is quantum, then its internal state cannot be copied due to the *no-cloning theorem*. This barrier turns out to be intrinsic — [ARU14] shows that, relative to an oracle, there exists a family of cryptographic schemes whose classical security, proven via reductions that use rewinding, does not carry over to the quantum setting. More investigation is thus necessary. We refer interested readers to a line of work that has emerged over more than a decade [Unr12, CMSZ22, LMS22, CAD+24] to salvage the quantum rewinding.

The bottom line is: even with quantum computational hardness in place, there still are many cases where a security proof does not (trivially) carry over to the quantum setting.

1.2 Cryptographic Transformations

A security proof of a cryptographic scheme is typically built up from a sequence of reductions, where one first considers a simpler but weaker scheme, and then strengthens it into a more sophisticated variant with stronger security. Sometimes, certain parts of the sequence can be obtained via applying generic cryptographic transformations that are not tailored to the specific scheme at hand. For instance, one of the earliest such transformations, known as the *Goldreich-Levin construction* [GL89], allows one to enhance a public-key encryption (PKE) scheme into achieving semantic security (IND-CPA). Later works such as the *Fujisaki-Okamoto transformation* [FO99, FO13] is capable of achieving stronger security guarantee (IND-CCA) while being much more efficient.

In the context of signature schemes, most practically relevant constructions

that achieve the standard UF-CMA security follow some variant of either the Fiat-Shamir transformation, or the hash-and-sign design principle. In the context of *key-encapsulation mechanisms* (KEMs) and others, there have recently been increasing attention toward designing more conservative schemes. In that regard, cryptographic combiners may also come in handy. We will describe the above in more detail, for the rest of this section.

Hash-and-sign design principle The hash-and-sign (H&S) design principle is widely used for constructing signature schemes from a trapdoor one-way function f, while the specific requirements of f may differ. To sign a message m, one produces (with a secret trapdoor of f) a preimage $x \in f^{-1}(y)$ of the hash $y := H(m)$ of m, which can then be efficiently verified via checking $f(x) = y$.

Security of H&S has been studied in the ROM, with the earliest classical analysis traced back to the full-domain hash (FDH) signature scheme [BR93], where f is a permutation based on RSA. More than a decade afterward, [GPV08] introduce and analyze a generic framework for constructing H&S signature schemes, in which f, a preimage sampleable function as they call it, only needs to satisfy certain weaker properties. The earliest quantum analysis is treated in [BDF+11], assuming f is collision resistant, which is further relaxed by [Zha12] assuming only the one-way security of f.

To facilitates security proofs, H&S is often made probabilistic, where the message is hashed via $y := H(m, r)$ with a randomized salt r instead, and then r is appended as an additional part of the signature. As summarized in [KX24, Table 1], the quantum analyses of [BDF+11, Zha12] carries over to this probabilistic variant. Moreover, it enjoys tighter security reductions [KX24], as well as more freedom to choose the function f [CD20].

Fiat-Shamir transformation. The *Fiat-Shamir transformation*, proposed by Amos Fiat and Adi Shamir [FS87] in 1986, plays an important role in building signature schemes.[7] Consider a 3-round (public-coin) interactive proof system $\Sigma = (\mathcal{P}, \mathcal{V})$, commonly known as a Sigma protocol. In this protocol, a prover $\mathcal{P}$ tries to convince a verifier $\mathcal{V}$ that it knows some secret witness w of a statement x, i.e. $(x, w) \in R$ for some well-defined relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$. This protocol proceeds as spelled out in Fig. 1.2: First, $\mathcal{P}$ sends over a message r. Then $\mathcal{V}$ poses to $\mathcal{P}$ its challenge $y \leftarrow \mathcal{Y}$ which is sampled uniformly random from a finite set $\mathcal{Y}$. Then $\mathcal{P}$ responds to the challenge with z. Finally, the verifier outputs a bit $b := \mathcal{V}(x, r, y, z)$ indicating whether or not it is convinced.

[7] More generally, the Fiat-Shamir transformation can also be used to build non-interactive zero-knowledge proof (NIZK).

$\Sigma(x, w)$:
1: $(r, \mathsf{st}) \leftarrow \mathcal{P}(x, w)$
2: $y \leftarrow \mathcal{Y}$
3: $z \leftarrow \mathcal{P}(y, \mathsf{st})$
4: **return** $\mathcal{V}(x, r, y, z)$

Figure 1.2: An honest execution of the Sigma protocol $\Sigma = (\mathcal{P}, \mathcal{V})$

Every such interactive proof can be made non-interactive, by replacing the challenge y with the hash of the first message $y := H(r)$ for a hash function H; this can be turned into a signature by hashing the message m as well, i.e. setting $y := H(r, m)$. To construct a proper signature scheme, though, one additionally needs to be able to efficiently sample a "hard instance" (x, w) in R, where computing w given x is hard. Formally, we specify such a signature scheme $\mathsf{FS}[\Sigma, H]$ in Fig. 1.3, assuming that there is indeed an efficiently sampleable distribution $D_{\mathcal{R}}$ generating a hard instance (x, w) in R.

KGen:	$\mathsf{Sign}(\mathsf{sk} = (x, w), m)$:	$\mathsf{Vrfy}(\mathsf{pk} = x, \sigma = (r, z), m)$:
1: $(x, w) \leftarrow \mathcal{D}_R$	1: $(r, \mathsf{st}) \leftarrow \mathcal{P}(x, w)$	1: $y := H(r, m)$
2: $\mathsf{sk} := (x, w)$	2: $y := H(r, m)$	2: **return** $\mathcal{V}(x, r, y, z)$
3: $\mathsf{pk} := x$	3: $z \leftarrow \mathcal{P}(y, \mathsf{st})$	
4: **return** $(\mathsf{sk}, \mathsf{pk})$	4: **return** $\sigma := (r, z)$	

Figure 1.3: The Fiat-Shamir signature scheme $\mathsf{FS}[\Sigma, H] = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ constructed from a Sigma protocol $\Sigma = (\mathcal{P}, \mathcal{V})$, where H is a hash function and $\mathcal{D}_R$ efficiently samples a hard instance in R.

Security properties of the signature scheme $\mathsf{FS}[\Sigma, H]$ are typically proven in the ROM. The earliest formal analysis goes back to [PS96]. However, establishing quantum security turns out significantly more challenging. The community was facing an (improper) impossibility result [DFG13], as well as quantum attacks [ARU14] breaking certain artificial choice of Σ, which carries over to the Fiat-Shamir scheme $\mathsf{FS}[\Sigma, H]$. Facing these negative results, Dominique Unruh proposed and analyzed a variant known as the *Unruh transformation* [Unr15], and studied the plain Fiat-Shamir transformation when given statistical soundness of Σ in [Unr17]. It took several decades from the first classical analysis of the Fiat-Shamir transformation, before the generic quantum security is formally proven [DFMS19, LZ19] under a slightly stronger (computational) assumption of Σ, known as the collapsing property,[8] than what is necessary for achieving classical security.

[8] As its name suggests, the definition of Σ's collapsing property is inspired by that of a hash function.

In retrospect, it has become clear how the security properties of $\mathsf{FS}[\Sigma, H]$ in the ROM follow from those of Σ. If Σ is *knowledge sound*, which prevents $\mathcal{P}$ to convince $\mathcal{V}$ without knowing the witness w, then $\mathsf{FS}[\Sigma, H]$ is $\mathsf{UF\text{-}NMA}$ against classical attackers. In addition, whenever Σ satisfy the collapsing property, the above holds against quantum attackers as well. If Σ (1) satisfies the *honest-verifier zero-knowledge* property (HVZK), which prevents $\mathcal{V}$ to learn anything about w in an honest execution, and (2) contains high min-entropy in its first message,[9] then there is an $\mathsf{UF\text{-}CMA\text{-}to\text{-}UF\text{-}NMA}$ security reduction, i.e. $\mathsf{FS}[\Sigma, H]$ is $\mathsf{UF\text{-}CMA}$ as long as it is $\mathsf{UF\text{-}NMA}$, covering both classical and quantum attackers. Indeed, the security proof of $\mathsf{FS}[\Sigma, H]$ applies regardless of the concrete instantiation of Σ, provided the aforementioned premises are satisfied.

Fiat-Shamir with aborts. The *Fiat-Shamir with aborts* (FSwA) transformation is adapted from the above Fiat-Shamir transformation by Vadim Lyubashevsky [Lyu09] in 2009. It is especially useful for constructing lattice-based and isogeny-based signature schemes, many of which are relevant to the NIST PQC competition.

The main difference between FSwA and the original Fiat-Shamir transformation stems from the fact that, for many Sigma protocols $\Sigma = (\mathcal{P}, \mathcal{V})$ relevant to the post-quantum setting, the prover $\mathcal{P}$ aborts with a certain probability. Typically, the abort by $\mathcal{P}$ is introduced, because otherwise completing the protocol would leak the witness w to the verifier $\mathcal{V}$. For $\mathcal{P}(x, w)$ to convince $\mathcal{V}(x)$ that (x, w) is a valid instance in R, the entire protocol must then be repeated until $\mathcal{P}$ does not abort. This is sometimes referred to as *rejection sampling*, and it carries over when constructing an FSwA signature scheme $\mathcal{S}$, which simply replaces $\mathcal{V}$'s challenge in each repetition with a suitable hash value.

Similar to the case of the plain Fiat-Shamir transformation, analyses of FSwA are typically performed in the ROM. The first classical analysis of FSwA was treated in [Lyu09], and since then FSwA has been used in many concrete signature schemes. This includes Lyubashevsky's signature scheme [Lyu09, Lyu12], GLP [GLP12], TESLA [ABB+20], Dilithium [DKL+18], SeaSign [DG19], and HAETAE [CCD+24]. Among these, TESLA was the earliest analyzed in the quantum setting, though the proof was tailored to the specifics of the scheme. Toward achieving full-fledged quantum security of FSwA, the authors of [KLS18] provided a generic $\mathsf{UF\text{-}CMA\text{-}to\text{-}UF\text{-}NMA}$ security reduction, which, however, is shown to be flawed in our later works (see Section 1.3). They also showed that FSwA satisfies $\mathsf{UF\text{-}NMA}$ under a more stringent condition on Σ than one would ideally require. This condition was later relaxed by [DFMS19] through a generic $\mathsf{UF\text{-}NMA}$ security proof that holds for Fiat-Shamir transformations, both with and without aborts.

[9] The second requirement is mild, and easily enforceable.

Hash-and-sign with retry/aborts. Just as the original Fiat-Shamir transformation can be generalized to FSwA, the H&S design principle can likewise be extended. In particular, many post-quantum signature schemes, especially those based on coding theory and multivariate polynomials, relies on a variant known as the *hash-and-sign with retry/aborts* (HSwA) design principle. As its name suggests, in a HSwA signature scheme, the main body of the signing procedure may abort with a certain probability, and thus have to be repeated several times until it does not abort. This new design principle is adopted by various constructions, many of which are relevant to the NIST PQC competition. Concrete examples include the Hidden Field Equation (HFE) scheme [Pat96], the Unbalanced Oil and Vinegar (UOV) scheme [KPG99], the Courtois-Finiasz-Sendrier (CFS) scheme [CFS01, Dal08], GeMSS [CFMR+17], Wave [DST19], MAYO [Beu21], QR-UOV [FIKT21], and Falcon+ [GJK24] (an updated version of Falcon).

BUFF transformations. The next transformation we want to briefly introduce here is the BUFF transformation.[10] It is relatively niche compared to, e.g. the Fiat-Shamir transformation, in that its purpose is to provide certain non-standard security properties that are not relevant in typical applications of digital signature schemes. However, it has recently gained quite some attention since the call for additional post-quantum signature candidate schemes by NIST [NIST22] explicitly refers to these non-standard security properties as being "desirable," and many of the recent signature candidates refer to it. The BUFF transformation additionally gained attention due to our work which shows that the original understanding of what the BUFF transform achieves, has to be revised (see Section 1.3 for more details).

In more detail, in [CDF+21], Cremers, Düzlü, Fiedler, Fischlin, and Janson proposed the first formal definition of the aforementioned additional desirable security properties, including *exclusive ownership* [PS05], *message-bound signatures*, and *non-resignability* [JCCS19]. They act as a second layer of protection against atypical misuses of a signature scheme under a higher-level protocol — indeed, there have been real-life attacks that exploit the lack of these properties, as discussed in [CDF+21]. The authors of [CDF+21] then proposed the *BUFF transformation* as a simple and efficient generic compiler that transforms any signature scheme, into another one that (is claimed to) achieve all these additional security properties. As will be elaborated in Fig. 4.3, it simply performs an additional pre-hash to the message with the public key, signs the hash, and appends the hash as an additional part of the signature, with verification done in the obvious way.

Candidate signature schemes in the extra round, such as Squirrels [ENST23], Racoon [dEK+23], HAWK [BBD+24], PROV [GCF+23], Vox [PCF+23], and eMLE [LZ23] have referred to BUFF in their proposal — some have incorpo-

[10] The acronym BUFF stands for Beyond-UnForgeability-Feature.

rated BUFF as a part of their design, while some others mentioned the possibility of applying BUFF to their schemes. BUFF is also relevant to two of the winners in round 3. As argued in [CDF+21], Dilithium has implicitly applied BUFF, inheriting generic security guarantees of BUFF, while Falcon does not. Nevertheless, the Falcon team has announced that they will also apply BUFF [FHK+22] to achieve the additional security properties.

Cryptographic combiners. A cryptographic combiner transforms multiple cryptographic schemes into a single (hybrid) scheme with the same or similar functionality, such that the resulting scheme remains secure as long as at least one of the component schemes is secure.

Combiners are particularly relevant in the context of post-quantum cryptography, where the security of post-quantum schemes often relies on relatively new computational assumptions, compared to well-established pre-quantum schemes based on RSA, Diffie-Hellman, and elliptic-curve cryptography. Combining a post-quantum scheme with a pre-quantum one thus provides a conservative transition to gain quantum security while retaining the classical one. This is not merely of theoretical interest — indeed, combiners can be used in standardized protocols such as Internet Key Exchange Protocol version 2 (IKEv2) [TTB+23] and Transport Layer Security 1.3 (TLS 1.3) [SFG25].[11] In the Netherlands, a nationwide initiative involving industry and academia, *Hybrid Approach for quantum-safe Public Key Infrastructure Development for Organisations* (HAPKIDO), has also been studying both practical and theoretical aspects of hybrid PKI as well as combiners. We note that, as observed in a recent literature survey [FHA23], different parties may have different opinions toward this kind of hybrid approach. Some government agencies in Europe publicly endorsed the use of combiners, such as the French Cybersecurity Agency (ANSSI) [ANSS22] and the German Federal Office for Information Security (BSI) [EHH+22], while some others hold a rather negative opinion about it, like the American National Security Agency (NSA) [NSA24].

Needless to say, the precise security notion of a combiner depends on that of the primitive being combined. In some cases, there is a straightforward construction: for instance, concatenating signatures produced by the component schemes trivially yields a combiner that preserves the standard $\mathsf{UF\text{-}CMA}$ security. In some other cases, however, a more sophisticated approach is required.

Relevant to this thesis, notable examples are combiners for *key-encapsulation mechanisms* (KEMs). In a nutshell, a KEM is a public-key primitive where, anyone with a long-term public key pk can (produce and) *encapsulate* a session key k in a ciphertext c that can later be *decapsulated* to the same k by the holder of the long-term secret key sk corresponding to pk. Securely combining KEMs is far from obvious. Indeed, the standard security of a KEM

[11] The use of combiners is not standardized in TLS 1.3, but it is supported by the IETF informational document [SFG25].

is IND-CCA, which prevents a valid ciphertext being mauled into another one that is decapsulated to the same session key. This property is in particular not preserved by the naive KEM combiner that concatenates all ciphertexts and takes the xor of the session keys, as the combined ciphertext and the combined session key respectively. Moreover, [GHP18] shows that for the similar reason, another natural construction called the xor-then-PRF combiner does not preserve IND-CCA security either.

On the positive side, constructing KEM combiners has been a topic under active research, with various concrete constructions shown to be secure. This includes combiners that are based on PRF-then-xor [GHP18], split-key PRF (skPRF) [GHP18], xor-then-MAC (XtM) [BBF+19], dual PRF [BBF+19], and the FO transformation [HV21]. Looking ahead at our contributions, our work in [DFH22], on which this thesis is based, helps establish (quantum) security of a particularly efficient construction based on skPRF. For the sake of efficiency, recent works also look into concrete constructions of hybrid KEMs [BCD+24], and KEM combiners that are not full-fledged since they rely on additional properties of the component schemes [ABK25].

1.3 Our Contributions

In this thesis, we (re)consider and study the following cryptographic transformations: the hash-and-sign and Fiat-Shamir (with Aborts), which will be covered in Chapter 3; the BUFF transformation, which will be covered in Chapter 4; and a particularly efficient KEM combiner, which will be covered in Chapter 5. For the first three cases (HSwA, FSwA, and BUFF), our results significantly contribute to the security understanding of the transformations, both when considering classical and quantum attacks. In particular, we show that there were incorrect understandings of them prior to our results, and we re-establish security (to the extent possible). For the KEM combiner, on the other hand, we provide the first quantum security proof.

Security of FSwA and HSwA. As have been discussed in the prior section, FSwA was first introduced in [Lyu09] as a variant of the Fiat-Shamir transformation, with quantum security analyzed in [KLS18, DFMS19], and is often used for constructing post-quantum signature schemes. Unexpectedly, though, there turns out to be a crucial but subtle flaw in [KLS18] and all prior UF-CMA-to-UF-NMA security reductions of FSwA signature schemes. The flaw applies both classically and quantum, and was discovered during the course of a formal verification project in [BBD+23]. It was also independently and concurrently discovered by [DFPS23], and reappeared in the context of HSwA signature schemes [KX24]. As a consequence, security proofs of a long list of (potentially hundreds of) works based on FSwA or HSwA are invalidated, and there is no known easy fix. Specifically, the upcoming NIST PQC standard,

Dilithium [DKL+18], is left with no valid security proof. We will describe this flaw in Section 3.3.

The main technical contribution of Chapter 3 is to restore confidence toward FSwA and HSwA signature schemes from the aforementioned flaw. Below, we briefly explain how we achieve this.

To begin with, as a conceptual contribution, we notice the structural resemblance of FSwA and HSwA, and put together (in Section 3.2.1) a unified framework capturing both, which we refer to as *generalized Fiat-Shamir with aborts* signature schemes. In Section 3.4, we then provide a new, fixed, and generic UF-CMA-to-UF-NMA security proof in the ROM, which applies both classically and quantum, for all such (generalized) FSwA signature schemes $\mathcal{S}$. Concretely, we show that as long as $\mathcal{S}$ satisfies what we call the *accepting honest-verifier zero-knowledge* (acHVZK) property, together with a mild requirement about min-entropy, and if it is UF-NMA, then it is also UF-CMA. The acHVZK property that we premise is essentially the minimal: it ensures that an "honestly generated signature" leaks nothing about the secret key. In particular, it is weaker than the naHVZK property premised in the prior flawed security proof by [KLS18].

More formally, in Section 3.4, based on our work in [BBD+23], we show that every UF-CMA attacker $\mathcal{A}$ making at most q_H quantum queries to the random oracle H, and at most q_S classical queries to the signing oracle, can be transformed into a similarly efficient UF-NMA attacker $\mathcal{B}$ such that

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathcal{S}}(\mathcal{B}) + O\left(\frac{q_H\sqrt{q_S}}{1-p} + \frac{q_S\sqrt{q_H}}{(1-p)^2}\right)\sqrt{\epsilon} + q_S\zeta \ ,$$

for parameters p, ϵ, ζ dependent on the scheme,[12] where $\epsilon, \zeta > 0$ are negligibly small, and $0 \leq p < 1$, which we call the "abort rate," is not too close to 1. In the case where queries to H are restricted to be classical, we have a correspondingly tighter bound

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathcal{S}}(\mathcal{B}) + O\left(\frac{q_S q_H}{1-p} + \frac{q_S^2}{(1-p)^2}\right)\epsilon + q_S\zeta \ .$$

In Section 3.5, based on our follow-up work in [FFH25], we further improve the generic quantum bound to

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathcal{S}}(\mathcal{B}) + O\left(\frac{q_S\sqrt{q_H}}{1-p}\right)\sqrt{\epsilon} + q_S\zeta \ .$$

Note that, most of the time q_H is much larger than q_S, with $q_S\zeta$ being non-dominating. In this case, the quantum security loss is brought down from $O\left(\frac{q_H\sqrt{q_S}}{1-p}\right)\sqrt{\epsilon}$ to $O\left(\frac{q_S\sqrt{q_H}}{1-p}\right)\sqrt{\epsilon}$. Concretely, this amounts to a factor of

[12] In general, these parameters may be key-dependent, but we ignore it here for simplicity.

roughly $\sqrt{q_H/q_S} \approx 2^{32}$, for the NIST security level 2 where q_S and q_H are up to 2^{64} and 2^{128} respectively. This, as well as the classical bound, is (to some extent) as tight as one can hope for. Specifically, in all relevant regimes,[13] the obtained additive security loss, up to a constant factor, matches that of the standard, well-studied Fiat-Shamir transformation without aborts (see [GHHM21, Theorem 3]).[14] Therefore, any further improvements, should they exist, also need to carry over to this well-understood setting without aborts.

Naively applying our generic bound to Dilithium, however, does not provide sufficient concrete security. This is mainly due to our generic bounds still being worse compared to those of the flawed analyses in [KLS18]. The bounds in [KLS18] are independent of q_S and q_H, while our losses scale along with q_S and q_H. Therefore, we provide a more elaborated concrete analysis of Dilithium in Section 3.6, which is partially computer-aided, and recover its full security level.

(In)security of BUFF. Recall that, the BUFF transformation is introduced in [CDF+21], and is claimed to achieve several additional security properties that they formally defined: exclusive ownership, message-bound signatures, and non-resignability (NR). However, in Chapter 4, which is based on our works in [DFHS24, DFH+24, FHK25], we show that the actual situation for the case of NR is much more subtle.

In Section 4.3, we show that the NR property, as defined in [CDF+21], is essentially impossible to achieve. More precisely, we show that, when considering a signature scheme that has sufficient computational min-entropy (HILL entropy) within a randomly chosen message given its signature, there is an explicit attack breaking NR, both in the plain model and the ROM. In particular, for every signature scheme obtained via applying the BUFF transformation, there is a concrete attack breaking its NR security, provided that the underlying hash function is sufficiently compressing, as is typically the case. At the opposite extreme, if a message can be efficiently recovered from its signature, then as noted by [CDF+21], the scheme is trivially not NR. While there is still a small, artificial gap that is not covered, all natural schemes fall into either of the above two categories.

Our generic attack against NR contradicts and invalidates the claimed security of BUFF in [CDF+21], for which we then identify the crucial flawed step in the security proof. We observe that the flawed analysis reduces the NR security of the BUFF transformation, to a particular security of the underlying hash function called Φ-*non-malleability* (Φ-NM), where the prefix Φ specifies a suitable class of functions of relevance to the security definition. Then, to complete the full security justification of BUFF, a claim in [BFS11] that the

[13] If the abort rate p is very close to 1, which practically never happens, then our security bounds degrade slightly.

[14] In the (atypical) regime where q_S is much larger than q_H, our quantum bound is even tighter.

random oracle satisfy Φ-NM is recycled. Unfortunately, this particular claim is false. As shown in Section 4.3, for any hash function that is sufficiently compressing (which is typically so), there is an explicit attack breaking its Φ-non-malleability.

We observe, however, that there are certain aspects of our attack that do not appear in real-world scenarios. Therefore, rather than being viewed as a practical threat, our attack shows that the definition of NR security as in [CDF+21] is not the "right" one. To recover from this negative state of affairs, it is necessary to explore alternative definitions of NR, and to investigate whether they can be achieved. That is exactly what we do for the rest of Chapter 4.

In Section 4.4, we introduce a slightly weaker yet still meaningful definition of non-resignability in the ROM, which we call $\mathsf{NR}^{H,\perp}$. With $\mathsf{NR}^{H,\perp}$ in place, the generic attack no longer applies. Whether or not BUFF satisfies $\mathsf{NR}^{H,\perp}$ is far from obvious. Therefore, to begin with, we introduce a salted variant of BUFF, \$-BUFF, and show that it satisfies $\mathsf{NR}^{H,\perp}$, covering both classical and quantum attackers, if the entropy requirement in the definition of $\mathsf{NR}^{H,\perp}$ is statistical. On the other hand, if the entropy requirement is computational, then there is a counter example, for which applying \$-BUFF (or BUFF) would result in a scheme that does not satisfy $\mathsf{NR}^{H,\perp}$. Again, the attack here still contains a certain level of contrivedness, which does not seem very realistic either.

In Section 4.5, we introduce yet another variant of NR in the ROM, which we call $\mathsf{sNR}^{H,\perp}$. One motivation doing so is to mitigate the negative result for $\mathsf{NR}^{H,\perp}$ under the computational entropy requirement, which no longer applies to $\mathsf{sNR}^{H,\perp}$. Moreover, it also serves as a proxy for analyzing the $\mathsf{NR}^{H,\perp}$ property of the original, unsalted BUFF, since under the statistical entropy requirement, any scheme that satisfies $\mathsf{sNR}^{H,\perp}$ also satisfies $\mathsf{NR}^{H,\perp}$.[15] As the main technical contribution of this section, we then show that the (unsalted) BUFF satisfies $\mathsf{sNR}^{H,\perp}$, which implies that it also satisfies $\mathsf{NR}^{H,\perp}$. This confirms our intuition that BUFF indeed satisfies a meaningful notion of NR, when the underlying hash function is idealized as a random oracle.

In Section 4.6, we further investigate the security of BUFF, when using a typical iterative hash function, e.g. SHA-2, SHA-3, or SHAKE. Specifically, we consider a more fine-grained setting, where, instead of idealizing the entire hash function as a random oracle, only the round function is idealized. To our surprise, in this fine-grained setting, the BUFF transformation is no longer secure! We show that, if the underlying hash function of BUFF is an iterative hash function, then there is an explicit attack breaking the (fine-grained version of) $\mathsf{sNR}^{H,\perp}$ for any scheme obtained from applying the BUFF transformation.

To recover from the above (fine-grained) negative result, we introduce the third variant of BUFF in Section 4.7, which we call the Sandwich BUFF trans-

[15] On the other hand, under the computational entropy requirement, $\mathsf{sNR}^{H,\perp}$ and $\mathsf{NR}^{H,\perp}$ become incomparable.

formation (sBUFF). Recall that the original BUFF works via hashing the message with the public key, signing the hash, and then appending the hash to the signature. As its name suggests, the Sandwich BUFF is almost identical, except that in the hashing step, the public key is sandwiched between two copies of the message, rather than concatenated after it. We then show that, if the Sandwich BUFF transformation $\mathsf{sBUFF}[\mathcal{S}, \mathsf{MD}]$ uses the Merkle-Damgård hash function MD^H, where the compression function is modelled as a random oracle H, then it satisfies $\mathsf{sNR}^{H,\perp}$.[16]

Finally, we conclude Chapter 4 with Section 4.8, where we show that all positive results in terms of $\mathsf{sNR}^{H,\perp}$ (and its fine-grained version) carries over, when the underlying entropy requirements are computational.

Quantum security of a KEM combiner In [GHP18], Giacon, Heuer and Poettering showed that any *split-key PRF* (skPRF) gives rise to a secure KEM combiner. In more detail, they show that if an skPRF is used in the (rather) obvious way as a key-derivation function in a KEM combiner, then the resulting hybrid KEM is IND-CCA secure if at least one of the component KEMs is IND-CCA secure. They also suggest a few candidates for skPRFs. The most efficient of the proposed constructions is a hash-based skPRF, which is proven secure in [GHP18] in the random-oracle model, considering classical attackers. However, in the context of a quantum attack, which is in particular relevant in the above example application of a combiner, it is crucial to prove security when quantum attackers are in place, i.e. when attacker can query the random oracle in quantum superposition. In Chapter 5, which is based on our work in [DFH22], we close this gap by proving post-quantum security of the hash-based skPRF construction as mentioned above.

[16] For simplicity, we do not cover the case of sBUFF using a Sponge function, but it is treated in our work [FHK25].