Proof-of-Forgery for Hash-based Signatures

Evgeniy Kiktenko[1,2] [a], Mikhail Kudinov[1,2] [b], Andrey Bulychev[1] and Aleksey Fedorov[1,2] [c]

[1]*Russian Quantum Center, Skolkovo, Moscow 143025, Russia*

[2]*QApp, Skolkovo, Moscow 143025, Russia*

Keywords: Hash-based Signatures, Lamport Signature, Winternitz Signature, Crypto-agility.

Abstract: In the present work, a peculiar property of hash-based signatures allowing detection of their forgery event is explored. This property relies on the fact that a successful forgery of a hash-based signature most likely results in a collision with respect to the employed hash function, while the demonstration of this collision could serve as convincing evidence of the forgery. Here we prove that with properly adjusted parameters Lamport and Winternitz one-time signatures schemes could exhibit a forgery detection availability property. This property is of significant importance in the framework of the crypto-agility paradigm since the considered forgery detection serves as an alarm that the employed cryptographic hash function becomes insecure to use and the corresponding scheme has to be replaced.

1 INTRODUCTION AND PROBLEM STATEMENT

Today, cryptography is an essential tool for protecting the information of various kinds. A particular task that is important for modern society is to verify the authenticity of messages and documents effectively. For this purpose, one can use so-called digital signatures. An elegant scheme for digital signatures is to employ one-way functions, which are one of the most important concepts for public-key cryptography. A crucial property of public-key cryptography based on one-way functions is that it provides a computationally simple algorithm for legitimate users (e.g., for key distribution or signing a document), whereas the problem for malicious agents is extremely computationally expensive. It should be noted that the very existence of one-way functions is still an open conjecture. Thus, the security of corresponding public-key cryptography tools is based on unproven assumptions about the computational facilities of malicious parties.

Assumptions on the security status of cryptographic tools may change with time. For example, breaking the RSA cryptographic scheme is at least as hard as factoring large integers (Rivest et al., 1978).

This task is believed to be extremely hard for classical computers, but it appeared to be solved in polynomial time with the use of a large-scale quantum computer using Shor's algorithm (Shor, 1997). A full-scale quantum computer that is capable of launching Shor's algorithm for realistic RSA key sizes in a reasonable time is not yet created. At the same time, there are no identified fundamental obstacles that prevent from development of quantum computers of a required scale. Thus, prudent risk management requires defending against the possibility that attacks with quantum computers will be successful.

A solution for the threat of creating quantum computers is the development of a new type of cryptographic tools that strive to remain secure even under the assumption that the malicious agent has a large-scale quantum computer. This class of quantum-safe tools consists of two distinct methods (Wallden and Kashefi, 2019). The first is to replace public-key cryptography with quantum key distribution, which is a hardware solution based on transmitting information using individual quantum objects. The main advantage of this approach is that the security relies not on any computational assumptions, but on the laws of quantum physics (Gisin et al., 2002). However, quantum key distribution technologies today face a number of important challenges such as secret key rate, distance, cost, and practical security (Diamanti et al., 2016).

Another way to guarantee the security of com-

[a] https://orcid.org/0000-0001-5760-441X

[b] https://orcid.org/0000-0002-8555-4891

[c] https://orcid.org/0000-0002-4722-3418

333

munications is to use so-called post-quantum (also known as quantum-resistant) algorithms, which use a specific class of one-way functions that are believed to be hard to invert both using classical and quantum computers (Bernstein and Lange, 2017). The main criticism of post-quantum cryptography is the fact that they are again based on computational assumptions so that there is no strict proof that they are long-term secure.

In our work, we consider a scenario, where an adversary finds a way to violate basic mathematical assumptions underlying the security of a particular cryptographic primitive. Thus, the adversary becomes able to perform successful attacks on information processing systems, which employ the vulnerable cryptographic primitive in their workflow. At the same time, it is in the interests of the adversary that the particular cryptographic primitive be in use as long as possible since its replacement with another one eliminates an obtained advantage. Thus, the preferable strategy of an attacker is to hide the fact that the underlying cryptographic primitive has been broken. It can be realized by performing attacks in such a way that their success could be explained by some other factors (e.g. user negligence, hardware faults, and etc.), but not the underlying cryptographic primitive. An illustrative example of such a strategy is hiding the information about the successes of the Enigma system cryptoanalysis during World War II.

Broadly speaking, the question we address in the present work is as follows: Is it possible to supply a new generation of post-quantum cryptographic algorithms with some kind of alarm indicating that they are broken? We argue that the answer to this question is partially positive, and the property, which we refer to as a forgery detection availability, can be realized by properly designed hash-based signatures. The intuitive idea behind this property is that a forgery of a hash-based signature most likely results in finding a collision with respect to the underlying cryptographic hash function (see Fig. 1), and so the demonstration of this collision can serve as convincing evidence of the forgery and corresponding vulnerability of the employed cryptographic hash function. We refer to the mathematical scheme for the evidence of the forgery event as a proof-of-forgery concept. We also would like to emphasize the fact that some of widespread hash functions have been compromised after their publication (Dobbertin, 1998; Black et al., 2006; Stevens et al., 2017), therefore the considered problem is more than just of academic interest.

In the present work, we illustrate the forgery detection availability property for Lamport (Buchmann et al., 2009; Lamport,) and Winternitz (Buchmann
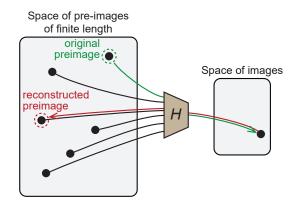

Figure 1: Demonstration of the idea behind proving the fact of the hash-based signature forgery. In order to forge the signature, an adversary finds a valid preimage for a given image of a cryptographic hash function. If the size of the preimages space is large enough then the preimage obtained adversary is most likely different from the legitimate user's one. Disclosing the colliding preimage could serve as evidence that a particular hash function is vulnerable.

et al., 2009) one-time signatures schemes. First, we consider the Lamport scheme, which is paradigmatically important: It is the first and the simplest algorithm among hash-based schemes. However, the Lamport scheme is not widely used in practice. Then we analyze the Winternitz scheme, particularly the variant presented in Ref. (Hülsing, 2013), which can be considered as a generalization of the Lamport scheme that introduced a size-performance trade-off. Variations of the Winternitz scheme are used as building blocks in a number of modern hash-based signatures, such as LMS (McGrew and Curcio, 2016), XMSS (Huelsing et al., 2018), SPHINCS (Bernstein et al., 2016) and its improved modifications (Bernstein et al., 2017; Aumasson and Endignoux, 2017), as well as applications such as IOTA distributed ledger (IOT,).

The paper is organized as follows. In Sec. 2, we give a short introduction to the scope of hash-based signatures. In Sec. 3, we provide a general scheme of detecting signature forgery event and define a property of the ε-forgery detection availability (ε-FDA). In Sec. 4, we consider the ε-FDA property for the generalized Lamport one-time signature (L-OTS). In Sec. 5, we consider the ε-FDA property for the Winternitz one-time signature (W-OTS$^+$). We summarize the results of our work in Sec. 6.

2 HASH-BASED SIGNATURES

Hash-based digital signatures (Bernstein and Lange, 2017; Bernstein et al., 2009) are one of the post-

quantum alternatives for currently deployed signature schemes, which have gained a significant deal of interest. The attractiveness of hash-based signatures is mostly due to low requirements to construct a secure scheme. Typically, a cryptographic random or pseudorandom number generator is needed, and a function with some or all of preimage, second-preimage, and collision resistance properties, perhaps, in their multi-target variety (Bernstein et al., 2017; Hülsing et al., 2016; Buchmann et al., 2009). Some schemes for hash-based signatures require a random oracle assumption (Koblitz and Menezes, 2015) to precisely compute their bit security level (Katz, 2016).

Up to date known quantum attacks based on Grover's algorithm (Grover, 1996) are capable to find a preimage and a collision with time growing subexponentially with a length of hash function output (Boyer et al., 1999; Brassard et al., 1998). Specifically, it is proven that in the best-case scenario Grover's algorithm gives a quadratic speed-up in a search problem (Grover, 1996). While this area is a subject of ongoing research and debates (Bernstein, 2009; Banegas and Bernstein, 2018; Chailloux et al., 2017), hash-based signatures are considered resilient against quantum computer attacks. Meanwhile, the overall performance of hash-based digital signatures makes them suitable for the practical use, and several algorithms have been proposed for standardization by NIST (SPHINCS$^+$ (Bernstein et al., 2017), Gravity-SPHINCS (Aumasson and Endignoux, 2017)) and IETF (LMS (McGrew and Curcio, 2016), XMSS (Huelsing et al., 2018)).

We note that the hash-based digital signature scheme can be instantiated with any suitable cryptographic hash function. In practice, standardized hash functions, such as SHA, are used for this purpose since they are presumed to satisfy all the necessary requirements. The availability of changing a core cryptographic primitive without a change in the functionality of the whole information security system fits a paradigm of crypto-agility, which is the basic principle of modern security systems development with the built-in possibility of component replacement.

3 PROVING THE FACT OF A FORGERY

Here we present a general framework for the investigation of the proof-of-forgery concept. We start our consideration by introducing a generic deterministic digital signature scheme.

Definition 1 (Deterministic Digital Signature

Scheme). *A deterministic digital signature scheme (DDSS) $S = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ is a triple of algorithms that allows performing the following tasks:*

- *$S.\mathsf{Kg}(1^n) \to (\mathsf{sk}, \mathsf{pk})$ is a probabilistic key generation algorithm that outputs a secret key sk, aimed at signing messages, and a public key pk, aimed at checking signatures validity, on input of a security parameter 1^n.*

- *$S.\mathsf{Sign}(\mathsf{sk}, M) \to \sigma$ is a <u>deterministic</u> algorithm that outputs a signature σ under secret key sk for a message M.*

- *$S.\mathsf{Vf}(\mathsf{pk}, \sigma, M) \to v$ is a verification algorithm that outputs $v = 1$ if the signature σ of the signed message M is correct under the public key pk, and $v = 0$ otherwise.*

We note that the deterministic property of the DDSS is defined by the fact that for a given pair (sk, M) the algorithm $S.\mathsf{Sign}(\mathsf{sk}, M)$ always generates the same output.

The standard security requirement for digital signature schemes is their existential unforgeability under chosen message attack (EU-CMA). The chosen message attack setting allows the adversary to choose a set of messages that a legitimate user has to sign. Then the existential unforgeability property means that the adversary should not be able to construct any valid message-signature pair $(M^\star, \sigma^\star)$, where the message $M^\star$ is not previously signed by a legitimate secret key holder. In the present work, we limit ourselves to the case of one-time signatures, so the adversary is allowed to obtain a signature for a single message only. The generalization to the many-time signature schemes is left for future research.

In the present work, we consider a stronger security requirement known as <u>strong unforgeability</u> under chosen message attack (SU-CMA). A DDSS is said to be SU-CMA if it is EU-CMA, and given signature σ on some message M, the adversary cannot even produce a new signature $\sigma^* \neq \sigma$ on the message M. We note that SU-CMA schemes are used for constructing chosen-ciphertext secure systems and group signatures (Boneh et al., 2006; Steinfeld and Wang, 2007). The security is usually proven under the assumption that the adversary is not able to solve some classes of mathematical problems, such as integer factorization, discrete logarithm problem, or inverting a cryptographic hash function. Here we consider the case where this assumption is not fulfilled.

Let us discuss the following scenario involving three parties: An honest legitimate signer $\mathcal{S}$, an honest receiver $\mathcal{R}$, and an adversary $\mathcal{A}$. At the beginning (**step 0**) we assume that $\mathcal{S}$ possesses a pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow S.\mathsf{Kg}$, while $\mathcal{R}$ and $\mathcal{A}$ have a public key

pk of S, and they have no any information about the corresponding secret key sk (see Table 1).

At **step 1** A forces S to sign a message M of A's choice. In the result, A obtains a valid message-signature pair (M, σ). At this step, R may or may not know about the fact of signing M by S.

Then at **step 2** A performs an existential forgery by producing a new message-signature pair $(M^\star, \sigma^\star)$ with $M^\star \neq M$. Below we introduce a formal definition of the signature forgery and specify two different cases.

Definition 2 (Signature Forgery and Its Types). *A signature $\sigma^\star$ is called a forged signature of the message $M^\star$ under the public key* pk *and the signature scheme S if* $S.\mathsf{Vf}(\mathsf{pk}, \sigma^\star, M^\star) \to 1$, *where the message $M^\star$ has not been signed by the legitimate sender possessing secret key* sk *corresponding to* pk. *The following two cases are possible.*

- *A pair $(M^\star, \sigma^\star)$ is called a forgery of type I if the signature $\sigma^\star$ has been previously generated by the legitimate user a signature for some message other than $M^\star$. That is, there is a message M with* $S.\mathsf{Sign}(\mathsf{sk}, M) \to \sigma^\star$ *previously signed by a legitimate user.*

- *A pair $(M^\star, \sigma^\star)$ is called a forgery of type II if the signature $\sigma^\star$ has not been previously generated by the legitimate user. That is, there has not been a message with signature $\sigma^\star$, signed by the legitimate user.*

The type I forgery can take place if the signature algorithm $S.\mathsf{Sign}$ calculates a <u>digest</u> of an input message and then computes a signature of the corresponding digest. In this case, the adversary A may find a collision of the digest function, and then force the legitimate user to sign a first colliding message by using it as M, and automatically obtain a valid signature for the second colliding message (use it as $M^\star$).

An example of type II forgery is the reconstruction of the sk from pk using an efficient algorithm (in analogy to the use of Shor's algorithm on a quantum computer for the RSA scheme). We note that in our consideration it is assumed that the only way for the adversary A to forge the signature for $M^\star$ is to employ advanced mathematical algorithms and/or unexpectedly powerful computational resources. In other words, we do not consider any side-channel attacks or other forms of secret key "stealing", such as social engineering and others.

Coming back to the considered scenario, at the **step 2**, A sends a pair $(M^\star, \sigma^\star)$ to R claiming that $M^\star$ was originally signed by S. If the signature is successfully forged by the adversary A, then this could be the end of the story.

However, we suggest accomplishing this scenario by the following next steps. At the **step 3**, R sends a message $(M^\star, \sigma^\star)$ directly to S in order to request an additional confirmation. Then S observes a valid signature $\sigma^\star$ of the corresponding message $M^\star$, which was not generated by him. The concrete issue we address in the present work is whether S is able to <u>prove</u> the fact of a forgery event. Here we formally introduce a proof-of-forgery concept, which is mathematical evidence that someone cheats with signatures by employing computational resources or advanced mathematical algorithms.

Definition 3 (Proof-of-Forgery of Type I). *A set $E = (\mathsf{pk}, \sigma^\star, M, M^\star)$ is called a proof-of-forgery of type I (PoF-I) for a DDSS S if for $M \neq M^\star$ there is a valid signature $\sigma^\star$ for these two messages, i.e. the following relations hold:*

$$S.\mathsf{Vf}(\mathsf{pk}, \sigma^\star, M^\star) \to 1, \quad S.\mathsf{Vf}(\mathsf{pk}, \sigma^\star, M) \to 1. \quad (1)$$

Obviously, if the adversary A performs the type I forgery, then S is able to prove this fact by demonstrating M to R at **step 4**. Thus, S and R have the complete PoF-I set $E = (\mathsf{pk}, \sigma^\star, M, M^\star)$, and they are sure that someone has an ability to break SUF-CMA property (Brendel et al., 2020), which is typically beyond the consideration in standard computational hardness assumptions. Moreover, they can use the set E to prove the fact of the forgery event to any third party since E contains a public key pk. We also note that it is possible to prove the fact of a forgery of type I for any DDSS. The situation in the PoF-II case is more complicated.

Definition 4 (Proof-of-Forgery of Type II). *A set $E = (\mathsf{pk}, \widetilde{\sigma}^\star, \sigma^\star, M^\star)$ is called a proof-of-forgery of type II (PoF-II) for a DDSS S if for a message $M^\star$ there are distinct valid signatures $\widetilde{\sigma}^\star \neq \sigma^\star$, i.e. the following relations hold:*

$$S.\mathsf{Vf}(\mathsf{pk}, \widetilde{\sigma}^\star, M^\star) \to 1, \quad S.\mathsf{Vf}(\mathsf{pk}, \sigma^\star, M^\star) \to 1.$$

The ability of the adversary A to perform a forgery of type II depends on a particular deterministic signature scheme S. Suppose that A has succeeded in obtaining sk from pk (e.g. by using Shor's algorithm and RSA-like scheme), then it is impossible for S to convince R that $(M^\star, \sigma^\star)$ was not generated by S. However, if the adversary A has succeeded in obtaining a valid, but <u>different</u> secret key $\mathsf{sk}' \neq \mathsf{sk}$, then the legitimate sender S is able to construct the corresponding PoF-II set by calculating $\mathsf{Sign}(\mathsf{sk}, M^\star) \to \widetilde{\sigma}^\star$ with $\widetilde{\sigma}^\star \neq \sigma^\star$.

As we show below this scenario is the case for properly designed hash-based signatures. We consider particular examples of Lamport and Winternitz one-time signatures schemes. We show that under

Table 1: Message-signature pairs and keys available to involved parties on each step of the scenario, where the adversary $\mathcal{A}$ makes a successful CMA obtaining a signature σ for some message M, and forges a signature $\sigma^\star$ for some new message $M^\star \neq M$ under the public key pk of the signer $\mathcal{S}$. However, the signer $\mathcal{S}$ is able to construct the corresponding proof-of-forgery message E in order to convince the receiver $\mathcal{R}$ that the forgery event happened. Square brackets correspond to the optional message-signature transmission.

	Signer $\mathcal{S}$	Adversary $\mathcal{A}$	Receiver $\mathcal{R}$
Step 0	sk, pk	pk	pk
Step 1	sk, pk, (M,σ)	pk, (M,σ)	pk, $[(M,\sigma)]$
Step 2	sk, pk, (M,σ)	pk, (M,σ), $(M^\star,\sigma^\star)$	pk, $[(M,\sigma)]$, $(M^\star,\sigma^\star)$
Step 3	sk, pk, (M,σ), $(M^\star,\sigma^\star)$	pk, (M,σ), $(M^\star,\sigma^\star)$	pk, $[(M,\sigma)]$, $(M^\star,\sigma^\star)$
Step 4	sk, pk, (M,σ), $(M^\star,\sigma^\star)$, E	pk, (M,σ), $(M^\star,\sigma^\star)$, $[E]$	pk, $[(M,\sigma)]$, $(M^\star,\sigma^\star)$, E

favourable circumstances $\mathcal{S}$'s signature $\widetilde{\sigma}^\star$ of the corresponding message $M^\star$ is different from $\mathcal{A}$'s signature $\sigma^\star$, and $\mathcal{S}$ can send it as part of PoF-II to $\mathcal{R}$ at **step 4**. Thus, the PoF-II set is successfully constructed, so legitimate parties are aware of the break of the used DDSS.

Here, we introduce a definition of an adversary who successfully forged, which allows proving their forgery.

Definition 5 (ε-Forgery Fetection Availability). *ε-forgery detection availability (ε-FDA) for a one-time DDSS S is defined by the following experiment.*
Experiment. $\mathrm{Exp}_{S,n}^{FDA}(\mathcal{A})$

$(\mathsf{sk}, \mathsf{pk}) \leftarrow S.\mathsf{Kg}(1^n)$

$(M^\star, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}$

Let (M, σ) *be the query-answer pair of* $\mathsf{Sign}(\mathsf{sk}, \cdot)$.

Return 1 iff $S.\mathsf{Sign}(\mathsf{sk}, M^\star) \rightarrow \sigma^\star$, $S.\mathsf{Vf}(\mathsf{pk}, \sigma^\star, M^\star) \rightarrow 1$, *and* $M^\star \neq M$.
Then the DSS scheme S has ε-FDA if there is no adversary $\mathcal{A}$ *that succeeds with probability* $\geq \varepsilon$.

Remark 1. *In our consideration, we implicitly assume that the parties are able to communicate with each other via authentic channels, e.g. when $\mathcal{R}$ sends a request to $\mathcal{S}$ at **step 3**. One can see that in order to enable the detection of the forgery event, the authenticity of the channel should be provided with some different primitives rather than employed signatures. For example, one can use message authentication codes (MACs), which can be based on information-theoretical secure algorithms and symmetric keys.*

Remark 2. *In the considered scheme honest users only become aware of the fact of forgery event. However, the scheme does not allow determining who exactly in this scenario has such powerful computational capabilities. Indeed, $\mathcal{S}$ is not sure whether the signature $\sigma^\star$ is forged by $\mathcal{R}$ or by $\mathcal{A}$. That is why it is advisable for $\mathcal{S}$ also to send evidence E to $\mathcal{A}$ as well. At the same time, $\mathcal{R}$ is not sure, who is the original*
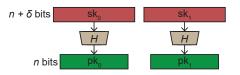

Figure 2: Basic principle of the public key construction in the (n, δ)-L-OTS scheme.

author of $\sigma^\star$. It is a possible case that $\mathcal{S}$ has forged its own signature (say, obtained two messages M and $M^\star$ with a same signature $\sigma = \sigma^\star$), and sent a message M to $\mathcal{A}$, who then just forwarded it $\mathcal{R}$. It may be in the interest of a malicious $\mathcal{S}$ to reveal $M^\star$ at the right moment and claim that it was a forgery.

4 ε-FDA FOR LAMPORT SIGNATURES

Here we start with a description of a generalized Lamport single bit one-time DDSS. Consider a cryptographic hash function $H : \{0,1\}^* \rightarrow \{0,1\}^n$. The (n, δ)-Lamport one-time signature $((n, \delta)$-L-OTS) scheme for single bit message $M \in \{0, 1\}$ has the following construction.

Key Pair Generation Algorithm: $((\mathsf{sk}, \mathsf{pk}) \leftarrow (n, \delta)$-L-OTS.Kg). The algorithm generates secret and public keys in a form $(\mathsf{sk}_0, \mathsf{sk}_1)$ and $(\mathsf{pk}_0, \mathsf{pk}_1)$, with $\mathsf{sk}_i \xleftarrow{\$} \{0,1\}^{n+\delta}$ and $\mathsf{pk}_i := H(\mathsf{sk}_i)$ (see Fig. 2). Here and after $\xleftarrow{\$}$ stands for uniformly random sampling from a given set.

Signature Algorithm: $(\sigma \leftarrow (n, \delta)$-L-OTS.Sign$(\mathsf{sk}, M))$. The algorithm outputs half of the secret key as a signature: $\sigma := sk_M$.

Verification Algorithm: $(v \leftarrow (n, \delta)$-L-OTS.Vf$(\mathsf{pk}, \sigma, M)))$. The algorithm outputs $v := 1$, if $H(s) = \mathsf{pk}_M$, and its output is 0 otherwise.

The security of the (n, δ)-L-OTS scheme is based on the fact that in order to forge a signature for a bit M it is required to invert the used one-way function H for a part of the public key pk_M, that is traditionally

assumed to be computationally infeasible.

In our work, we particularly stress the importance of inequality between space sizes of secret keys and public keys. Specifically, we demonstrate that for sufficiently large δ even if an adversary finds a correct preimage $\mathsf{sk}_M^\star$, such that $H(\mathsf{sk}_M^\star) = \mathsf{pk}_M$, the obtained value is different from the original sk_M used for calculating pk_M by the legitimate user. Then the signature of an honest user is different from a forged signature, and so the forgery event can be revealed.

Before turning to the main theorem, we prove the following Lemma.

Lemma 1. *Consider a function $f : \{0,1\}^{n+\delta} \to \{0,1\}^n$ with $n \gg 1$ and $\delta \geq 0$ taken at random from the set all functions from $\{0,1\}^{n+\delta}$ to $\{0,1\}^n$. Let $y_0 = f(x_0)$ for x_0 taken uniformly at random from $\{0,1\}^{n+\delta}$. Define a set*

$$\mathrm{Inv}(y_0) := \{x \in \{0,1\}^{n+\delta} | f(x) = y_0\} \qquad (2)$$

of all preimages of y_0 under f. Consider a randomly taken preimage $X \xleftarrow{\$} \mathrm{Inv}(y_0)$. Then the probability to obtain the original preimage $X = x_0$ has the following lower and upper bounds:

a) $\Pr(X = x_0) > \exp(-2^\delta)$;

b) $\Pr(X = x_0) < 5.22 \times 2^{-\delta}$.

Proof. Let $\mathcal{N} := |\mathrm{Inv}(y_0)|$ be a number of preimages of y_0 under f. Due to the random choice of f, it is given by $\mathcal{N} = 1 + \widehat{\mathcal{N}}$, where $\widehat{\mathcal{N}}$ is a random variable having binomial distribution $\mathrm{Bin}(2^{-n}, 2^{n+\delta} - 1)$ with the success probability 2^{-n} and number of trials equal to $2^{n+\delta} - 1$. Then the corresponding probability that a randomly chosen element X from $\mathrm{Inv}(y_0)$ is equal to x_0 is as follows:

$$\Pr(X = x_0) = \sum_{N=1}^{2^{n+\delta}-1} \frac{1}{N} \Pr(\mathcal{N} = N). \qquad (3)$$

In order to obtain the lower bound for $\Pr(X = x_0)$, we consider only the first term in Eq. (3) and arrive at the following inequality:

$$\Pr(X = x_0) > \Pr(\mathcal{N} = 1) = \left(1 - 2^{-n}\right)^{2^{n+\delta}-1}$$

$$\simeq \left(1 - 2^{-n}\right)^{2^{n+\delta}} \simeq \exp(-2^\delta), \quad (4)$$

where we use the fact that $(1 - 2^{-n})^{2^n} \simeq \exp(-1)$ for $n \gg 1$. This proves part a) of Lemma 1.

In order to obtain the upper bound for $\Pr(X = x_0)$, we split the sum in Eq. (3) into following two parts:

$$\Pr(X = x_0) = \sum_{N=1}^{N_0} \frac{1}{N} \Pr(\mathcal{N} = N) +$$

$$\sum_{N=N_0+1}^{2^m-1} \frac{1}{N} \Pr(\mathcal{N} = N) \quad (5)$$

where $N_0 := k2^\delta \geq 1$ for some $k \in (0,1)$. The first part can be bounded as follows:

$$\sum_{N=1}^{N_0} \frac{1}{N} \Pr(\mathcal{N} = N) \leq \Pr(\mathcal{N} \leq N_0)$$

$$\leq \frac{(2^{n+\delta} - N_0)2^{-n}}{(2^\delta - N_0)^2} \qquad (6)$$

$$< \frac{2^{-\delta}}{(k-1)^2},$$

where we use a bound for the cumulative binomial distribution function (Feller, 1968). For the second part we consider the following bound:

$$\sum_{N=N_0+1}^{2^m-1} \frac{1}{N} \Pr(\mathcal{N} = N) < \frac{1}{N_0} \sum_{N=N_0+1}^{2^m-1} \Pr(\mathcal{N} = N)$$

$$< \frac{1}{N_0} = \frac{2^{-\delta}}{k}. \qquad (7)$$

By combining Eq. (6) with Eq. (7) and setting $k := 0.36$, which corresponds to a minimum of $\Pr(X = x_0)$, we obtain $\Pr(X = x_0) < 5.22 \times 2^{-\delta}$. This proves part b) of Lemma 1. $\qquad \square$

Remark 3. *The bound for commutative binomial distribution employed in (6) is rather rough, however, it is quite convenient for the purposes of further discussion. Tighter bound can be obtained, e.g. using the technique from Ref. (Zubkov and Serov, 2013).*

Next we assume that the number of $(n + \delta)$-bit preimages for $H(x)$ with $x \xleftarrow{\$} \{0,1\}^{n+\delta}$ behaves in the same way as for a random function from $\{0,1\}^{n+\delta}$ to $\{0,1\}^n$. The main result on the FDA property of the (n, δ)-L-OTS scheme can be formulated as follows:

Theorem 1. (n, δ)-L-OTS *scheme has ε-FDA with $\varepsilon < 5.22 \times 2^{-\delta}$.*

Proof. Consider an adversary who successfully forged a signature $\sigma^\star$ for message $M^\star$. From the construction of the signature scheme we have $H(\sigma^\star) = \mathsf{pk}_{M^\star}$. According to part b) of Lemma 1, the probability that the obtained value $\sigma^\star$ coincides with the original value $\mathsf{sk}_{M^\star}$ is bounded by $5.22 \times 2^{-\delta}$. It follows from the fact that $\mathsf{sk}_{M^\star}$ is generated uniformly randomly from the set $\{0,1\}^{n+\delta}$. Therefore, with probability at least $1 - 5.22 \times 2^{-\delta}$ the legitimate user's signature $\mathsf{sk}_{M^\star} \leftarrow (n, \delta)$-L-OTS.Sign$(\mathsf{sk}, M^\star)$ is different from the adversary's signature $\sigma^\star$ and the presence of the forgery event is then proven. $\qquad \square$

Remark 4. *We note that it is extremely important to employ true randomness in the Kg algorithm in order to provide the independence between the results of the adversary and the original value of sk. For this*

purpose, one can use, for example, certified quantum random number generators.

Remark 5. *We see that ε-FDA property appears only for high enough values of* δ. *Meanwhile it follows from part a) of Lemma 1 that for a common case of* δ = 0 *the probability for the adversary to obtain the original value* sk_1 *is at least* $\exp(-1) \approx 0.368$ *that is non-negligible.*

5 ε-FDA FOR THE WINTERNITZ SIGNATURE SCHEME

Here we consider an extension of L-OTS scheme which allows signing messages of *L*-bit length. We base our approach on a generalization of the Winternitz one-time signature (W-OTS) scheme presented in Ref. (Hülsing, 2013), known as W-OTS$^+$, and used in XMSS (Hülsing et al., 2016), SPHINCS (Bernstein et al., 2016) and SPHINCS$^+$ (Bernstein et al., 2017). We note that W-OTS$^+$ is SU-CMA scheme (Hülsing, 2013). We refer our scheme (n, δ, L, ν)-W-OTS$^+$ and construct it as follows.

Let us introduce the parameter $\nu \in \{1, 2, \ldots\}$ defining blocks length in which a message is split during a signing algorithm, where we assume that *L* is a multiple of ν. Let us introduce the following auxiliary values: $w := 2^\nu$, $l_1 := \lceil L/\nu \rceil, l_2 := \lfloor \log_2(l_1(w-1))/\nu \rfloor + 1$, $l := l_1 + l_2$. Then we consider a family of one-way functions:

$$f_{\mathbf{r}}^{(i)} : \{0,1\}^{n+\delta(w-i)} \to \{0,1\}^{n+\delta(w-i-1)}, \quad (8)$$

where $i \in \{1, \ldots, w-1\}$ and a parameter $\mathbf{r}$ belongs to some domain $\mathcal{D}$. The employ of this parameter can correspond to XORing the result of some hash function family with a random bit-mask, as it considered in Ref. (Hülsing, 2013)). We assume that $f_{\mathbf{r}}^{(i)}$ satisfies the random oracle assumption for a uniformly randomly chosen $\mathbf{r}$ from $\mathcal{D}$.

We then introduce a chain function $F_{\mathbf{r}}^{(i)}$, which we define recursively in the following way:

$$\begin{aligned} F_{\mathbf{r}}^{(0)}(x) &= x, \\ F_{\mathbf{r}}^{(i)}(x) &= f_{\mathbf{r}}^{(i)}(F_{\mathbf{r}}^{(i-1)}(x)) \text{ for } i \in \{1, \ldots, w-1\}. \end{aligned} \quad (9)$$

The algorithms of (n, δ, L, ν)-W-OTS$^+$ scheme are the following:

<u>Key Pair Generation Algorithm:</u> $((\mathsf{sk}, \mathsf{pk}) \leftarrow (n, \delta, L, \nu)$-W-OTS$^+$.Kg). First the algorithm generates a secret key in the following form: $\mathsf{sk} := (\mathbf{r}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_l)$, with $\mathsf{sk}_i \xleftarrow{\$} \{0,1\}^{n+\delta(w-1)}$ and $\mathbf{r} \xleftarrow{\$} \mathcal{D}$ (see Fig. 3). Then a
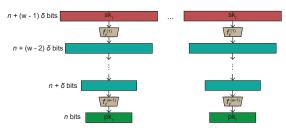

Figure 3: Basic principle of the public key construction in the (n, δ, L, ν)-W-OTS$^+$ scheme.

public key composed of the randomizing parameter $\mathbf{r}$ and results of the chain function employed to sk_i as follows: $\mathsf{pk} := (\mathbf{r}, \mathsf{pk}_1, \mathsf{pk}_2, \ldots, \mathsf{pk}_l)$ with $\mathsf{pk}_i := F_{\mathbf{r}}^{(w-1)}(\mathsf{sk}_i)$.

<u>Signature Algorithm:</u> $(\sigma \leftarrow (n, \delta, L, \nu)$-W-OTS$^+$.Sign$(\mathsf{sk}, M))$. First the algorithm computes base *w* representation of *M* by splitting it into ν-bit blocks $(M = (m_1, \ldots, m_{l_1})$, where $m_i \in \{0, \ldots, w-1\})$. We call it a message part. Then the algorithm computes a checksum $C := \sum_{i=1}^{l_1}(w-1-m_i)$ and its base *w* representation $C = (c_1, \ldots, c_{l_2})$. We call it a checksum part. Define an extended string $B = (b_1, \ldots b_l) := M \| C$ as the concatenation of message and checksum parts. Finally, the signature is generated as follows: $\sigma := (\sigma_1, \sigma_2, \ldots, \sigma_l)$ with $\sigma_i := F_{\mathbf{r}}^{(b_i)}(\mathsf{sk}_i)$.

<u>Verification Algorithm:</u> $(v \leftarrow (n, \delta, L, \nu)$-W-OTS$^+$.Vf$(\mathsf{pk}, \sigma, M)))$. The idea of the algorithm is to reconstruct a public key from a given signature σ and then to check whether it coincides with the original public key pk. First, the algorithm computes a base *w* string $B = (B_1, \ldots, B_l)$ in the same way as in the signature algorithm (see above). Then for each part of the signature σ_i the algorithm computes the remaining part of the chain as follows:

$$\mathsf{pk}_i^{\text{check}} := f_{\mathbf{r}}^{(w-1)} \circ \ldots \circ f_{\mathbf{r}}^{(b_i+1)}(\sigma_i), \quad (10)$$

where $\circ$ stands for the standard functions composition. If $\mathsf{pk}_i^{\text{check}} = \mathsf{pk}_i$ for all $i \in \{1, \ldots, l\}$, then the algorithm outputs $v := 1$, otherwise $v := 0$.

The main result on the FDA property of the (n, δ, L, ν)-W-OTS$^+$ scheme can be formulated as follows:

Theorem 2. *The (n, δ, L, ν)-W-OTS$^+$ scheme has the ε-FDA property with* $\varepsilon < 5.22 \times 2^{-\delta}$.

Proof. Consider a scenario of successful CMA on the (n, δ, L, ν)-W-OTS$^+$ scheme, in which an adversary first forces a legitimate user with public key $\mathsf{pk} = (\mathbf{r}, \mathsf{pk}_1, \ldots, \mathsf{pk}_l)$ to provide him a signature $\sigma = (\sigma_1, \ldots, \sigma_l)$ for some message *M*, and then generate a valid signature $\sigma^\star = (\sigma_1^\star, \ldots, \sigma_l^\star)$ for some message $M^\star \neq M$. Let $(m_1, \ldots, m_{l_1})$ and $(m_1^\star, \ldots, m_{l_1}^\star)$
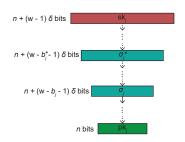
Figure 4: Illustration of the principle of PoF-II construction for the (n, δ, L, ν)-W-OTS$^+$ scheme.

be the w-base representations of M and $M^\star$ correspondingly. Consider extended w-base strings $B = (b_1^0, \ldots, b_l^0)$ and $B^\star = (b_1^\star, \ldots, b_l^\star)$ generated by adding checksum parts. It easy to see that for any distinct M and $M^\star$ there exists at least one position $j \in \{1, \ldots, l\}$ such that $b_j^\star < b_j$. Indeed, even if for all positions $i \in \{1, \ldots, l_1\}$ it happened that $m_i^\star > m_i$, from the definition of checksum it follows that there exists a position $j \in \{l_1 + 1, \ldots, l_2\}$ in checksum parts such that $b_j^\star < b_j$.

Since $\sigma^\star$ is a valid signature for $M^\star$ we have

$$f_{\mathbf{r}}^{(w-1)} \circ \ldots \circ f_{\mathbf{r}}^{(b_j^\star + 1)}(\sigma_j^\star) = \mathsf{pk}_j. \tag{11}$$

One can see that forgery event will be detected if the jth part of the legitimate user's signature of $M^\star$ is different from the forged one (see also Fig. 4), so that:

$$\widetilde{\sigma}_j^\star := F_{\mathbf{r}}^{(b_j^\star)}(\mathsf{sk}_j) \neq \sigma_j^\star. \tag{12}$$

Consider two possible cases. The first is that the condition (11) is fulfilled, but the following relation holds true:

$$f_{\mathbf{r}}^{(b_j)} \circ \ldots \circ f_{\mathbf{r}}^{(b_j^\star + 1)}(\sigma_j^\star) \neq \sigma_j. \tag{13}$$

In this case we obtain $\widetilde{\sigma}_j^\star \neq \sigma_j^\star$ with unit probability since

$$\sigma_j = f_{\mathbf{r}}^{(b_j)} \circ \ldots \circ f_{\mathbf{r}}^{(b_j^\star + 1)}(\widetilde{\sigma}_j^\star) \neq f_{\mathbf{r}}^{(b_j)} \circ \ldots \circ f_{\mathbf{r}}^{(b_j^\star + 1)}(\sigma_j^\star). \tag{14}$$

In the second case we have the following identity:

$$f_{\mathbf{r}}^{(b_j)} \circ \ldots \circ f_{\mathbf{r}}^{(b_j^\star + 1)}(\sigma_j^\star) = \sigma_j, \tag{15}$$

which automatically implies the fulfilment of Eq. (11). Consider the function

$$F := f_{\mathbf{r}}^{(b_j)} \circ \ldots \circ f_{\mathbf{r}}^{(b_j^\star + 1)} : \{0, 1\}^{n^* + \delta \Delta} \to \{0, 1\}^{n^*}, \tag{16}$$

where $\Delta := b_j^0 - b_j^\star \geq 1$ and $n^* := n + \delta(w - b_j^\star - 1)$. This function satisfies random oracle assumptions, since each of $\{f_{\mathbf{r}}^{(k)}\}_{k=b_j^\star}^{b_j}$ do. The random oracle assumption implies that the number of preimages of F is the distributes in the same way as for random random

function. Moreover, due to the fact that the values of sk_i are generated uniformly at random, the inputs for F are also uniformly distributed over the corresponding domains. So, according to part b) of Lemma 1, we have the probability of the adversary to obtain $\sigma_j^\star = \widetilde{\sigma}_j^\star$ is bounded by $\varepsilon < 5.22 \times 2^{-\delta\Delta} \leq 5.22 \times 2^{-\delta}$. □

Remark 6. *One can see that the excess of the preimage space size over the image space size, given by δ, is a crucial condition for the FDA property both for* L-OTS *and* W-OTS$^+$ *schemes. We also note that in the case of the* W-OTS$^+$ *scheme it is important to have such excess for all the elements of the employed one-way chain.*

Remark 7. *One can imagine a situation where an attacker is able to compute* $\mathsf{sk}_i^\star \neq \mathsf{sk}_i$ *for some* $i \in \{1, \ldots, l\}$*, such that* $F^{(w-1)}(\mathsf{sk}_i^\star) = \mathsf{pk}_i$*. This means that the chain collides somewhere on the way to* pk_i*. As long as an attacker reveals forgeries only above this collision, there is no way for the legitimate user to present a different signature on the same message. However, in line with our proof, the probability of such a scenario is limited by ε.*

In $WOTS^+$ scheme consider an attacker who is able to compute some value $sk_i' \neq sk_i$, such that $f^{w-1}(sk_i') = pk_i$ for one of the public-key chunks pk_i, this means that the chain collides somewhere on the way to pk_i. One can notice that as long as an attacker reveals forgeries only above this collision, there is no way for the legitimate user to present a different signature on the same message. But following our proof, this situation is limited by ε.

6 CONCLUSION

In this work, we have considered the ε-FDA property of DDSS that allows detecting a forgery event generated by advanced mathematical algorithms and/or unexpectedly powerful computational resources. We have shown that this property is fulfilled for properly-designed hash-based signatures, in particular, for L-OTS and W-OTS$^+$ schemes with properly tuned parameters. As we have noted, the probability of the successful demonstration of the DDSS forgery event depends on an excess of preimage space sizes over image space sizes and using true randomness in the generation of secret keys in hash-based DDSS. The important next step is to study this property for other types of hash-based signatures.

Our observation is important in the view of the crypto-agility paradigm. Indeed, the considered forgery detection serves as an alarm that the employed

cryptographic hash function has a critical vulnerability and it has to be replaced. We note that a similar concept has been recently considered for detecting brute-force attacks on cryptocurrency wallets in the Bitcoin network (Kiktenko et al., 2019). Namely, it was considered the alarm system that detects the case of stealing coins by finding a secret-public key pair for standard elliptic curve digital signature algorithm (ECDSA) used in the Bitcoin system, such that a public key hash of adversary equals a public key hash of a legitimate user. This kind of alarm system can be of particular importance in view of the development of quantum computing technologies (Fedorov et al., 2018).

ACKNOWLEDGEMENTS

REFERENCES

IOTA project website: www.iota.org.

Aumasson, J.-P. and Endignoux, G. (2017). Improving stateless hash-based signatures.

Banegas, G. and Bernstein, D. (2018). Low-communication parallel quantum multi-target preimage search. In Selected Areas in Cryptography – SAC 2017, page 325.

Bernstein, D. (2009). Cost analysis of hash collisions: Will quantum computers make sharcs obsolete. In Proceedings of Workshop Record of SHARCS 2009: Special-Purpose Hardware for Attacking Cryptographic Systems, page 51.

Bernstein, D., Buchmann, J., and Dahmen, E. (2009). Post-Quantum Cryptography. Springer-Verlag, Berlin Heidelberg.

Bernstein, D., D. Hopwood, A. H., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., and Wilcox-O'Hear, Z. (2016). Sphincs: Practical stateless hash-based signatures. Lect. Notes Comp. Sci, 9056:368.

Bernstein, D., Dobraunig, C., Eichlseder, M., Fluhrer, S., Gazdag, S.-L., Hülsing, A., Kampanakis, P., Kölbl, S., Lange, T., Lauridsen, M., Mendel, F., Niederhagen, R., Rechberger, C., Rijneveld, J., and Schwabe, P. (2017). SPHINCS+.

Bernstein, D. and Lange, T. (2017). Post-quantum cryptography. Nature, 549:188.

Black, J., Cochran, M., and Highland, T. (2006). A study of the md5 attacks: Insights and improvements. Lect. Notes Comp. Sci., 4047:262.

Boneh, D., Shen, E., and Waters, B. (2006). Unforgeable signatures are used for constructing chosen-ciphertext

secure systems and group signatures. Lect. Notes Comp. Sci., 3958:229.

Boyer, M., Brassard, G., Hoeyer, P., and Tapp, A. (1999). Tight bounds on quantum searching. Fortschr. Phys., 46:493.

Brassard, G., Høyer, P., and Tapp, A. (1998). Quantum cryptanalysis of hash and claw-free functions. Lect. Notes Comp. Sci., 1380:163.

Brendel, J., Cremers, C., Jackson, D., and Zhao, M. (2020). The Provable Security of Ed25519: Theory and Practice.

Buchmann, J., Dahmen, E., and Szydlo, M. (2009). Hash-based digital signature schemes. In Post-Quantum Cryptography. Springer-Verlag Berlin Heidelberg.

Chailloux, A., Naya-Plasencia, M., and Schrottenloher, A. (2017). An Efficient Quantum Collision Search Algorithm and Implications on Symmetric Cryptography.

Diamanti, E., Lo, H.-K., and Yuan, Z. (2016). Practical challenges in quantum key distribution. npj Quant. Inf., 2:16025.

Dobbertin, H. (1998). Cryptanalysis of md4. J. Crypt., 11:253.

Fedorov, A., Kiktenko, E., and Lvovsky, A. (2018). Quantum computers put blockchain security at risk. Nature, 563:465.

Feller, W. (1968). An Introduction to Probability Theory and Its Applications. Wiley, USA, 3rd edition.

Gisin, N., Ribordy, G., Tittel, W., and Zbinden, H. (2002). Quantum cryptography. Rev. Mod. Phys., 74:145.

Grover, L. (1996). A fast quantum mechanical algorithm for database search. In Proceedings of 28th Annual ACM Symposium on the Theory of Computing, volume 212, page 261, New York, USA.

Hülsing, A. (2013). W-ots+ – shorter signatures for hash-based signature schemes. In Progress in Cryptology - AFRICACRYPT 2013, volume 7918, page 173–188. Springer.

Hülsing, A., Rijneveld, J., and Song, F. (2016). Mitigating multi-target attacks in hash-based signatures. Lect. Notes Comp. Sci., 9614:387.

Huelsing, A., Butin, D., Gazdag, S.-L., Rijneveld, J., and Mohaisen, A. (2018). eXtended Merkle Signature Scheme. RFC 8391, Internet-Draft.

Katz, J. (2016). Analysis of a proposed hash-based signature standard. Lect. Notes Comp. Sci., 10074:261.

Kiktenko, E., Kudinov, M., and Fedorov, A. (2019). Detecting brute-force attacks on cryptocurrency wallets. In Business Information Systems Workshops. BIS, page 373. Lecture Notes in Business Information Processing.

Koblitz, N. and Menezes, A. (2015). The random oracle model: a twenty-year retrospective. Des. Codes Cryptogr., 77:587.

Lamport, L. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory.

McGrew, D. and Curcio, M. (2016). Hash-Based Signatures. RFC 8554, Internet-Draft.

Rivest, R., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21.

Shor, P. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput., 26:1484.

Steinfeld, R. and Wang, J. P. H. (2007). How to strengthen any weakly unforgeable signature into a strongly unforgeable signature. Lect. Notes Comp. Sci., 4377:357.

Stevens, M., Bursztein, E., Karpman, P., Albertini, A., and Markov, Y. (2017). The first collision for full sha-1. Lect. Notes Comp. Sci., 10401:570.

Wallden, P. and Kashefi, E. (2019). Cyber security in the quantum era. Commun. ACM, 62:120.

Zubkov, A. and Serov, A. (2013). A complete proof of universal inequalities for the distribution function of the binomial law. Theory Probab. Appl., 57:539.