Quantum-access Security of Hash-based Signature Schemes

Quan Yuan[1], Mehdi Tibouchi[2,3], and Masayuki Abe[2,3]

[1] The University of Tokyo, Tokyo, Japan **
[2] NTT Social Informatics Laboratories, Tokyo, Japan
[3] Kyoto University, Kyoto, Japan

Abstract. In post-quantum cryptography, hash-based signature schemes are attractive choices because of the weak assumptions. Most existing hash-based signature schemes are proven secure against post-quantum chosen message attacks (CMAs), where the adversaries are able to execute quantum computations and classically query to the signing oracle. In some cases, the signing oracle is also considered quantum-accessible, meaning that the adversaries are able to send queries with superpositions to the signing oracle. Considering this, Boneh and Zhandry [12] propose a stronger security notion called existential unforgeability under quantum chosen message attacks (EUF-qCMA). We call it quantum-access security (or Q2 security in some literature). The quantum-access security of practical signature schemes is lacking in research, especially of the hash-based ones. In this paper, we analyze the quantum-access security of hash-based signature schemes in two directions. First, we show concrete quantum chosen message attacks (or superposition attacks) on existing hash-based signature schemes, such as SPHINCS [7] and SPHINCS+ [9]. The complexity of the attacks is obviously lower than that of optimal classical chosen message attacks, implying that quantum chosen message attacks are more threatening than classical ones to these schemes. Second, we propose a simple variant of SPHINCS+ and give security proof against quantum chosen message attacks. As far as we know, it is the first practical hash-based stateless signature scheme against quantum chosen message attacks with concrete provable security.

Keywords: hash-based signatures, quantum security, post-quantum cryptography, digital signatures, superposition attacks

1 Introduction

1.1 Background

Quantum-access Security of Signature Schemes. Signature schemes [21] are essential primitives in cryptography. In the security analysis of a signature

** This work is carried out during his PhD study in Kyoto University, Japan.

scheme, one usually considers existential unforgeability under chosen message attacks (EUF-CMA). In this model, the adversary can query messages to a signing oracle. After the signing queries, the adversary is required to output a valid signature σ^* for a fresh message m^*. We say a scheme is EUF-CMA if any polynomial-time adversary succeeds with negligible probability.

Quantum attacks on cryptographic schemes are usually classified into two types [13, 24, 25]. In the first type, the adversary can use quantum computers to execute some offline algorithms or evaluate some functions and send classical queries to the online oracles. This is called post-quantum security (or Q1 security in some literature). In the second type, the queries to the oracles are also in superpositions. This is called quantum-access security (or Q2 security). In signature schemes, the above EUF-CMA security is a kind of Q1 security. In recent years, there is a large amount of research on Q2 security of various cryptographic primitives, including pseudorandom functions [36], message authentication codes [11], encryption schemes [12], signature schemes [2, 12, 16, 18] and so on.

In 2013, Boneh and Zhandry [12] propose a Q2 security notion called existential unforgeability under *quantum* chosen message attacks (EUF-qCMA). In this experiment, the adversary is required to output (q_s+1) forgeries for distinct messages after q_s quantum signing queries. They also show a separating example, implying that EUF-qCMA is a strictly stronger security notion. However, to the best of our knowledge, there is no concrete evidence to show that a Q2 attacker can be obviously stronger than a Q1 one for *practical* signature schemes (although it is intuitively true).

Hash-based Signatures. Hash-based signature (HBS) schemes are ones whose security is solely based on secure hash functions rather than mathematical hard problems. Because of the weak assumptions and resistance to quantum attacks, HBS schemes are fairly competitive in post-quantum cryptography.

The first practical stateless HBS scheme is SPHINCS [7]. It is based on HORST (Hash to Obtain Random Subsets with Trees [32]), a few-time stateless HBS scheme, where the number of signing operations is limited by a small constant. A key pair of SPHINCS can issue at most 2^{50} signatures and provide 128-bit quantum security[4]. So far, the state-of-art stateless HBS scheme is SPHINCS+ [3, 9], a variant using improved building blocks such as few-time signatures and tweakable hash functions. It behaves better in terms of the signature size, efficiency, security proof and the maximum number of signing executions (reaching to 2^{64}). Recently, SPHINCS+ is selected as one of the NIST post-quantum cryptography standardization.

In terms of the quantum-access security of HBS schemes, Boneh and Zhandry [12] prove the EUF-qCMA security of Lamport's scheme [29] and MSS [31]. The

[4] In this paper, we always focus on signature schemes with security level 5 in NIST post-quantum cryptography standardization. That is, breaking the security is at least as hard as finding a preimage of AES-256. It implies 128-bit security, which means that breaking the security requires about 2^{128} hash queries. In particular, we focus on SPHINCS-256 and SPHINCS+-256s/256f.

former is a one-time HBS scheme, and the latter is a stateful one. Hopefully, other stateful HBS schemes (such as XMSS [15]) are also EUF-qCMA since they are essentially variants of MSS and the structures are similar. However, when it comes to stateless schemes, the cases become different. (The authors proved the EUF-qCMA security of stateless MSS, but stateless MSS is far from efficient in practice.) For practical stateless HBS schemes, such as SPHINCS and SPHINCS+, the quantum-access security still lacks research.

Note that Boneh and Zhandry [12] proposed a generic construction of EUF-qCMA schemes from UUF-RMA (universally unforgeable under random message attacks) schemes by introducing a hash function modeled as a quantum random oracle. However, the construction causes large security loss, which will be especially expensive for HBS schemes, whose security are highly related to the number of issued signatures. For instance, suppose we want an EUF-qCMA scheme such that the probability of breaking the security is under 2^{-10} when $q_s \leq 2^{10}$ and $q_H \leq 2^{30}$ (number of signing queries and hash computations respectively), which is a fairly weak requirement in practice. If we use the above generic construction, we need a 2^{117}-time UU-RMA scheme with 254-bit quantum security, which far exceeds the security of SPHINCS+.

1.2 Our Contributions

In this paper, we give positive and negative results on quantum-access security of HBS schemes.

- First, we show some qCMAs on two HBS schemes, SPHINCS and SPHINCS+. The complexity of our attacks is much lower than the security in the classical setting. We thus conclude that quantum chosen message attacks are more threatening to security than classical CMAs for the two schemes.
- Second, we propose SPHINCS-FORS, a simple variant of SPHINCS+, whose quantum-access security can be proven. We give formal security proof and concrete security in sense of EUF-CMA and EUF-qCMA. As far as we know, it is the first practical stateless HBS scheme with provable security against qCMAs.

1.3 Main Techniques

Quantum Chosen Message Attacks on HBS Schemes We first give an outline of SPHINCS and SPHINCS+. Roughly speaking, SPHINCS(+) introduces a stateful signature HT and implicit 2^h instances of few-time signature key pairs (HORST in SPHINCS and FORS in SPHINCS+). HT is for signing the public keys of the few-time signatures, and the few-time signatures are for signing the messages. In each signing operation, it (pseudo-)randomly picks an index $idx \in \{0,1\}^h$ and uses the few-time signature key pairs labeled idx to sign a message. The essential idea is that the index is picked randomly, and thus, each few-time signature key pair is not used too many times in the signing operations.

The core idea of our attack is to obtain enough few-time signatures associated with a single index. In SPHINCS, the index is calculated by pseudorandom functions on the message m and included as a part of the SPHINCS signature. Although the pseudorandom functions cannot be directly evaluated by adversaries without the secret key, they can be evaluated by signing queries. Fix some index $idx^* \in \{0,1\}^h$. A quantum-access adversary can search a message m^* mapping to idx^* by Grover's algorithm with $O(2^{h/2})$ quantum queries to the signing oracle.

Thus, our quantum chosen message attack on SPHINCS runs as follows. First, try to obtain enough number (say r) of messages m_i^* mapping to an index idx^* by querying the signing oracle. This requires $O(r2^{h/2})$ quantum signing queries. Then, send m_i^* (with pure states) to the signing oracle separately, and obtain r HORST signatures associated with idx^*. This requires r signing queries. When r is large enough, the HORST secret key associated with idx^* will be completely revealed, and the adversary can forge a signature for*any* message. Finally, generate enough SPHINCS signatures associated with index idx^* to meet the requirement of one-more forgeries. Note that optimal classical chosen message attacks on SPHINCS require approximately 2^{128} hash queries (when at most 2^{50} signatures are issued). The query complexity of our attacks are much lower than that of classical attacks (see Table 1, Our Attacks (PO)).

Result 1 *For some integer r, there exists a quantum chosen message attack on SPHINCS such that*

$$q_s = O(r2^{h/2}), \quad q_H = O((1-e^{-\frac{kr}{t}})^{-\frac{k}{2}} r 2^{\frac{h}{2}}),$$

where q_s and q_H denotes the number of quantum signing queries and quantum hash queries, respectively.

The attack on SPHINCS+ is similar. The difference is that the index of a SPHINCS+ signature is not directly included. Instead, it is calculated by $h_0(z||m)$, where h_0 is a hash function mapping to $\{0,1\}^h$ and z is a (pseudo-)randomizer determined by the message m. [5] Since z is included in the signature, the index can be evaluated by a signing query (calculating z) and an additional h_0 query. Then, use Grover's algorithm to find enough FORS signatures associated with an index idx^*. Finally, to generate a forgery (m_i^*, σ_i^*), the adversary needs to find a corresponding randomizer z_i^* such that $h_0(z_i^*||m_i^*) = idx^*$. It requires $O(2^{h/2})$ quantum queries to h_0 by using Grover's algorithm for each forgery. Thus, the attack on SPHINCS+ requires more hash queries than that on SPHINCS (see Table 1, Our Attacks (PO)).

Result 2 *For some integer r, there exists a quantum chosen message attack on SPHINCS+ such that*

$$q_s = O(r2^{h/2}), \quad q_H = O((1-e^{-\frac{r}{t}})^{-\frac{k}{2}} r 2^{h}).$$

[5] SPHINCS+ has a probabilistic version where z is not determined by the message. Our attack only works on the deterministic version of SPHINCS(+).

In addition, Alagic et al. [2] propose another security notion called blind unforgeability (BU) considering quantum chosen message attacks. [6]. Informally, the (quantum-accessible) signing oracle now returns $\perp$ for messages in a ε-fraction blind subset of the message space, and it is infeasible to forge a signature for a message in this blind subset. To distinguish the two, we call the previous security model against qCMAs as PO model (Plus-one model). In our study, we also give similar attacks on SPHINCS and SPHINCS+ in the BU model, and the time complexity of the attacks is much lower than that in the PO model (see Table 1, Our Attacks(BU)).

By implementing the parameters in SPHINCS [7] and SPHINCS+ v.3 [3], we give comparisons among the attacks in the EUF-CMA model, the PO model, and the BU model. See more details in Table 1.

Scheme	CMA Security		Our Attack (PO)		Our Attack (BU)	
	$\log q_s$	$\log q_H$	$\log q_s$	$\log q_H$	$\log q_s$	$\log q_H$
SPHINCS-256 [7]	50	128	43	43	43	Small
SPHINCS+-256s [3]	64	128	48	80	43	43
SPHINCS+-256f [3]	64	128	46	80	42	42

Table 1. Comparisons between our qCMAs and the CMA security of SPHINCS(+), which implies the bounds of tolerable hash queries. q_s and q_H denote the number of (quantum) signing queries and quantum hash queries, respectively. For example, if the adversary issues 2^{43} quantum signing queries to the signing oracle and more than 2^{43} quantum hash queries, then SPHINCS-256 will be broken in the PO model. "Small" means that the attack only requires a polynomial number of queries.

A Provably Secure Stateless HBS Scheme against Quantum Chosen Message Attacks Although we show qCMAs on SPHINCS and SPHINCS+, the time complexity of the attacks does not imply the precise security levels of the schemes against qCMAs. Indeed, it only implies *upper bounds* of the security. We hope to construct a *provably* secure scheme, whose generic security against qCMAs can be guaranteed and *lower-bounded*.

We start with the security analysis of few-time signature schemes, such as HORS [32] and FORS [9]. Most the practical few-time signature schemes are related to (variants of) subset resilient hash functions (SRH) [32, 35]. Unfortunately, their quantum-access security cannot be reduced to subset resilience directly. To solve this problem, we propose a variant of subset resilience called *weak subset resilience* (wSR, see details in Appendix C). We find that the EUF-qCMA security of a randomized version of FORS (say rFORS, implicitly contained in

[6] The notion is proposed for message authentication codes but can also be extended to signature schemes.

SPHINCS+) can then be reduced to wSR and some additional common security notions for hash functions (such as multi-target collision resistance, mTCR).

Since wSR is a new notion and lack of research, the generic security needs to be evaluated. We solve this problem by the quantum query lemma [34]. As a result, the generic security of rFORS against r-time qCMAs can be bounded by

$$\mathrm{Adv}^{\mathsf{EUF\text{-}qCMA}}_{\mathsf{rFORS},r,q}(\mathcal{A}) \leq O\left(q^{2(r+1)}\left(\frac{r^2}{t}\right)^k + \frac{q^2 k t^r}{2^n}\right), \tag{1}$$

where q denotes the number of queries to the hash functions (modeled as random oracles) and k, t, n are parameters. The first term of Equation (1) comes from the generic security of wSR. The second one comes from mTCR and the reduction. They are negligible when r is a small constant number. (In practice, $t \approx 2^{\sqrt{n}}$ and $k \approx \sqrt{n}$).

(To the best of our knowledge, there is no quantum generic security bound of subset resilience. We also fill this gap, which could be independently interesting. For example, it immediately implies concrete security of the related few-time HBS schemes in terms of post-quantum EUF-CMA.)

We then turn to many-time schemes. The main reason for the insecurity against the above qCMAs is that the index is determined by the message due to the pseudorandom functions. A natural idea to resist these attacks is to introduce additional randomness into the process of choosing the index. Here, we use a simpler approach: we directly replace the index with randomness that is *independent* of the message and include it in the signature. Now, the signatures in a quantum signing response share a common few-time signature key pair in a signing query. The EUF-qCMA security is thus reduced to the EUF-qCMA security of the related few-time signature scheme, rFORS, which can be bounded by Equation (1).

Although the new variant can avoid some qCMAs, it brings other risks. Since the index is directly included in the signature, an adversary can arbitrarily choose an index in the forgery. Thus, the EUF-CMA security of the new variant needs to be re-analyzed.

Similarly, the EUF-CMA security of our new variant can be reduced to the EUF-CMA security of rFORS. The remaining work is evaluating the EUF-CMA security of rFORS. Indeed, the EUF-CMA security of rFORS can be tightly reduced to a variant of subset resilience called *extended Target Subset Resilience* (eTSR). Unfortunately, the generic security of eTSR cannot be (tightly) evaluated by the same approach. We use the adaptive reprogramming lemma [22] to solve this problem. The EUF-CMA security of rFORS is then bounded by

$$\mathrm{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{rFORS},r,q}(\mathcal{A}) \leq O\left(\sqrt{\frac{q}{2^n}} + q^2\left(\frac{r}{t}\right)^k + \frac{q^2}{2^n}\right). \tag{2}$$

The first two terms of Equation (2) comes from the generic security of eTSR, and the third one comes from mTCR.

Equipped with rFORS, the message-independent random index, and the same HT as in SPHINCS+, we get our new variant called SPHINCS-FORS. (The name is given since the framework looks like SPHINCS (not SPHINCS+) equipped with rFORS.) The outline of the security analysis of SPHINCS-FORS is summarized in Figure 1.

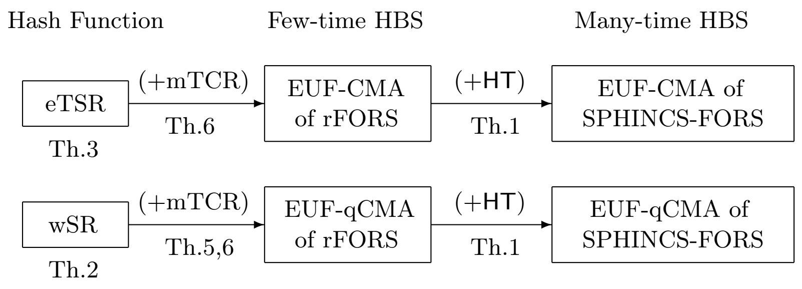

Fig. 1. The security proof sketch of SPHINCS-FORS.

Finally, the generic security of SPHINCS-FORS is evaluated as follows.

Result 3 *Let the hash functions in SPHINCS-FORS be modeled as quantum random oracles. For any adversary $\mathcal{A}$, it holds that*

$$Adv^{\textsf{EUF-CMA}}_{\textsf{SPHINCS-FORS},q_s,q_H}(\mathcal{A}) \le O\left(q_s\sqrt{\frac{q_H+q_s}{2^n}} + \frac{q_H}{2^{n/2}} + q_H^2\sum_{r=0}^{q_s} p(r,q_s)\left(\frac{r}{t}\right)^k\right),$$

$$Adv^{\textsf{EUF-qCMA}}_{\textsf{SPHINCS-FORS},q_s,q_H}(\mathcal{A}) \le O\left(\frac{q_H}{2^{n/2}} + \sum_{r=0}^{q_s} p(r,q_s)\cdot\min\left\{q_H^{2(r+1)}\left(\frac{r^2}{t}\right)^k + \frac{q_H^2 k t^r}{2^n}, 1\right\}\right),$$

where $p(r,q_s) = \min\{2^{r(\log q_s - h)+h-\log r!}, 2^h\}$.

For EUF-CMA security, we expect it to reach 128-bit security as SPHINCS+, which means that for fixed q_s (e.g., $q_s = 2^{64}$), $\mathrm{Adv}^{\textsf{EUF-CMA}}_{\textsf{SPHINCS-FORS},q_s,q_H}(\mathcal{A})$ reaches to a constant only if q_H reaches 2^{128}. When $n = 256$, the dominant term is $q_H^2 \sum_{r=0}^{q_s} p(r;q_s)(\frac{r}{t})^k$. We adapt the parameters such that $\sum_{r=0}^{q_s} p(r,q_s)(\frac{r}{t})^k = 2^{-256}$, and then the resulting scheme provides 128-bit security against CMA.

It is more complicated for EUF-qCMA security. The dominant term is also the one with the summation notion, which is larger than the above one. When r is small, $\min\{q_H^{2(r+1)}(\frac{r^2}{t})^k + \frac{q_H^2 k t^r}{2^n}, 1\}$ is negligibly small. When r becomes large, $p(r,q_s)$ becomes negligibly small and the righthand term is always bounded by 1, so the product is still small. Thus, the sum of the products remains bounded if the parameters are well stated.

We give instantiation of SPHINCS-FORS by adapting the parameters in Table 2. We observe that SPHINCS-FORS has larger signature size and running time, but can provide provable security in sense of EUF-qCMA.

Scheme	Parameters				Security		Size	Provably Secure?
	k	$\log t$	h	n	$\log q_s$	$\log q_H$		
SPHINCS+-256s	22	14	64	256	48	≤ 80	29272	✗
SPHINCS-FORS v1 (Ours)	32	17	128	256	48	≥ 80	47072	✓
SPHINCS+-256s$*$	22	14	104	256	64	≤ 128	41792	✗
SPHINCS-FORS v2 (Ours)	48	18	128	384	64	≥ 128	101424	✓

Table 2. Comparison between (deterministic) SPHINCS+ and our variants against quantum chosen message attacks. $\log q_H \leq a$ means that there exists a quantum chosen message attack with 2^a quantum hash queries. It implies an upper bound of the security level (without security proof). $\log q_H \geq a$ means that any quantum chosen message attack requires *at least* 2^a hash queries. It implies a lower bound of the security level (with security proof). Note that all the schemes in this table can provide at least 128-bit EUF-CMA security when $\log q_s \leq 64$.

1.4 Related Work

HBS schemes have a long history that begins with Lamport's one-time signature scheme [29]. SPHINCS [7] is the first practical stateless HBS using Goldreich's framework [20]. There are several variants of SPHINCS [5, 9, 27, 39]. The tight security proof is recently given in the post-quantum setting [26].

The generic security of hash functions is the core of analyzing the concrete security of HBS schemes. There is previous work proving the post-quantum generic security [1,22,26,28,34,37,38]. Especially, SPHINCS-family is related to subset resilience [4,35]. In recent concurrent work, Bouaziz–Ermann and Grilo et al. [14] shows quantum generic security of (restricted) subset resilience when the range of the "partial" hash (say t) is strictly exponential. Unfortunately, their proof fails since t is not large enough in a practical HBS scheme. (For instance, their proof once causes a term of reduction loss $O(t^{1/48})$, which is considered a negligible function when t is exponential. However, t is instantiated by 2^{14} in SPHINCS-256s, and thus the above term cannot be ignored.)

Quantum-access security is first considered for pseudorandom functions [36] and then generalized to message authentication codes and signatures [11, 12]. After that, other quantum-access security notions are proposed [2,19] for message authentication codes (and they can also be extended to fit signature schemes). In particular, the blind unforgeability of Lamport's scheme, WOTS, and GPV signatures has been evaluated [16, 30].

2 Preliminaries

2.1 Basic Preliminaries

Notations. For a set X, $|X|$ denotes the cardinality of X and $x \leftarrow X$ means that x is uniformly chosen from X. For integer $n \in \mathbb{N}$, denote $[n] = \{1, ..., n\}$.

We say that $\epsilon : \mathbb{N} \to \mathbb{R}$ is negligible function if for every constant $c > 0$, there exists $N_c > 0$ such that $\epsilon(n) < n^{-c}$ holds for all $n > N_c$. We denote by $||$ the concatenation operation.

Random Oracle Model. In the (classical) random oracle model [6], a random function H is uniformly chosen at the beginning and one can compute H by querying the random oracle. In quantum random oracle model [10], H is quantum-accessible, i.e., one can query $\sum_{x,y} \psi_{x,y} |x, y\rangle$ to the quantum random oracle, and obtains $\sum_{x,y} \psi_{x,y} |x, y \oplus H(x)\rangle$ in response.

Hash functions. In this paper, we frequently use hash functions mapping to k (ordered) elements of set $[t]$ (where $t = 2^\tau$ for some integer τ). We can simply sample such a function by sampling a hash function $H' : \{0,1\}^* \to \{0,1\}^{k \cdot \log t}$, splitting the output into k short strings of length τ, and then mapping the short strings into integers in $[t]$.

In the following, we denote a function $H : \{0,1\}^* \to [t]^k$ by $H = (h_1, ..., h_k)$, where $h_i : \{0,1\}^* \to [t]$ denotes the "partial" function mapping to the i-th element of the output of H. To avoid ambiguity, we always use capital letters (such as H) to denote a hash function and their corresponding small letters with a subscript (such as h_i) as the partial functions. For instance, in the case that $\mathcal{H}$ is a hash function family mapping to $[t]^k$, $(h_1, ..., h_k) \leftarrow \mathcal{H}$ means sampling $H \leftarrow \mathcal{H}$ and letting $H = (h_1, ..., h_k)$, rather than sampling k functions from $\mathcal{H}$.

2.2 Security Notions for Hash-based Signature Schemes

Let $\Gamma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme. SigO denotes the signing oracle that computes $\mathsf{Sign}(sk, m)$ where sk is the secret key. If SigO is quantum-accessible, say $|\mathsf{SigO}\rangle$,it means that

$$|\mathsf{SigO}\rangle : \sum_{m,t} \psi_{m,t} |m, t\rangle \mapsto \sum_{m,t} \psi_{m,t} |m, t \oplus \mathsf{Sign}(sk, m)\rangle .$$

Especially, if Sign is probabilistic, SigO replies $\sum_{m,t} \psi_{m,t} |m, t \oplus \mathsf{Sign}(sk, m; r)\rangle$ with a random seed r for a query $\sum_{m,t} \psi_{m,t} |m, t\rangle$.

This paper focuses on hash-based signature (HBS) schemes. The security of an HBS scheme is related to the number of hash queries from the adversary. Let q_s and q_H respectively denote the maximum number of signing queries and hash queries. The security is defined as follows.

```
Experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\Gamma,q_s,q_H}(1^n, \mathcal{A})$
  $(pk, sk) \leftarrow \mathsf{KeyGen}(1^n)$
  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{SigO}}(pk)$
  If $m^*$ has not been queried to $\mathsf{SigO}$ and $\mathsf{Ver}(pk, m^*, \sigma^*) = 1$, return 1,
otherwise return 0.
```

Experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}qCMA}}_{\Gamma,q_s,q_H}(1^n, \mathcal{A})$

$(pk, sk) \leftarrow \mathsf{KeyGen}(1^n)$
$\{(m_j, \sigma_j)\}_{j\in[r+1]} \leftarrow \mathcal{A}^{|\mathsf{SigO}\rangle}(pk)$
If m_j's are distinct and for $\forall j \in [q_s + 1]$, $\mathsf{Ver}(pk, m_j, \sigma_j) = 1$, **return** 1, otherwise **return** 0.

Definition 1. *([12]) Let Γ be a signature scheme. We say it is existentially unforgeable under chosen message attacks (EUF-CMA) (or quantum chosen message attacks (EUF-qCMA)) if for all probabilistic polynomial-time adversary $\mathcal{A}$, $\Pr[\mathit{Exp}^{\mathit{EUF\text{-}CMA}}_{\Gamma,q_s,q_H}(1^n, \mathcal{A})] \leq \boldsymbol{negl}(n)$ (or $\Pr[\mathit{Exp}^{\mathit{EUF\text{-}qCMA}}_{\Gamma,q_s,q_H}(1^n, \mathcal{A})] \leq \boldsymbol{negl}(n)$) holds, where **negl** is a negligible function.*

In addition, we introduce a security notion called blind unforgeability [2]. Let $\varepsilon : \mathbb{N} \to \mathbb{R}_{\geq 0}$ be an efficiently computable function. $B_{\varepsilon,n}$ be a subset of the message space $\mathcal{M}_n$ that is selected by placing each $m \in M_n$ independently with probability $\varepsilon(n)$. Define the blind signing oracle $B_{\varepsilon,n}\mathsf{SigO}$ as follows:

$$B_{\varepsilon,n}\mathsf{SigO} : m \mapsto \begin{cases} \mathsf{Sign}(sk, m) & (m \notin B_{\varepsilon,n}), \\ \perp, & (otherwise). \end{cases}$$

Similarly, when $B_{\varepsilon,n}\mathsf{SigO}$ is quantum-accessible, say $|B_{\varepsilon,n}\mathsf{SigO}\rangle$, it maps

$$|B_{\varepsilon,n}\mathsf{SigO}\rangle : \sum_{m,t} \psi_{m,t} |m, t\rangle \mapsto \sum_{m,t} \psi_{m,t} |m, t \oplus B_{\varepsilon,n}\mathsf{SigO}(m)\rangle .$$

Then, the experiment of blind unforgeability under quantum chosen message attacks is as follows:

Experiment $\mathsf{Exp}^{\mathsf{BU\text{-}qCMA}}_{\Gamma,q_s,q_H}(1^n, \mathcal{A})$
$(pk, sk) \leftarrow \mathsf{KeyGen}(1^n)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{|B_{\varepsilon,n}\mathsf{SigO}\rangle}(pk)$
If $m \in B_{\varepsilon,n} \wedge \mathsf{Ver}(pk, m^*, \sigma^*) = 1$, **return** 1, otherwise **return** 0.

Remark 1. In this paper, we mainly discuss EUF-qCMA security instead of BU except in Section 4.3. Apart from this subsection, we use the EUF-qCMA model in Definition 1 to evaluate the security against qCMAs by default.

We omit the security parameter 1^n in the inputs of the algorithms and experiments for simplicity.

3 Hash-based Signature Schemes

In this section, we introduce two practical stateless HBS schemes, SPHINCS [7] and SPHINCS+ [9]. Due to the complicated constructions, we follow the algorithmic descriptions in [17] by treating the building blocks as black boxes. Briefly speaking, SPHINCS(+) introduces a stateful HBS with hypertrees (denoted by HT) and a few-time HBS. We first introduce the two building blocks and then the schemes.

3.1 Building Blocks – Few-time HBS Schemes

In this section, we introduce few-time HBS schemes, which are used as building blocks in SPHINCS and SPHINCS+.

The first is Hash to Obtain Random Subsets (HORS) [32]. The outline of HORS is as follows. In the key generation algorithm, it picks a one-way function $f : \{0,1\}^{l(n)} \to \{0,1\}^n$ and an (r,k)-subset-resilient function $H = (h_1, ..., h_k) : \{0,1\}^m \to [t]^k$. Then, it picks t random strings $s_1, ..., s_t$ from $\{0,1\}^{l(n)}$ and computes $y_j = f(s_j)$ for each $j \in [t]$. Let $(s_1, ..., s_t)$ be the secret key and $(y_1, ..., y_t)$ be the public key. In the signing algorithm, it reveals k elements from $\{s_j\}_{j\in[t]}$ determined by $H(m)$. Due to (r,k)-subset resilience of H, it is hard to find a message m^* such that the secret values in the corresponding signature are covered by r (classical) queries to the signing oracle. The formal description is depicted in Figure 2.

```
$\mathsf{HORS.KeyGen}(1^\lambda)$

$F \leftarrow \mathcal{F}_n$, $H = (h_1, ..., h_k) \leftarrow \mathcal{H}$.
for $j \in [t]$, $s_j \leftarrow \{0,1\}^{l(n)}$, $y_j = f(s_j)$
$Y = (y_1, ..., y_t)$, $S = (s_1, ..., s_t)$
$pk = (Y, f, H)$, $sk = (S, f, H)$.
return $(pk, sk)$.

$\mathsf{HORS.Sig}(sk, m)$

Parse $S = (s_1, ..., s_t)$ and $H = (h_1, ..., h_k)$
for $i \in [k]$, $x_i = s_{h_i(m)}$.
return $\sigma = (x_1, ..., x_k)$.

$\mathsf{HORS.Ver}(pk, m, \sigma)$

Parse $\sigma = (x_1, ..., x_k)$ and $H = (h_1, ..., h_k)$.
if for $i \in [k]$, $y_{h_i(m)} = f(x_i)$ return 1
return 0
```

Fig. 2. Construction of Hash to Obtain Subsets (HORS).

SPHINCS [7] introduces a variant of HORS called HORST (HORS with trees). HORST compresses the public key with a (bitmarked) hash tree, and thus the signature needs to contain the corresponding authentication path. This operation does not hurt the security except that it requires a second-preimage-resistant hash function.

Furthermore, SPHINCS+ [9] introduces an improvement of HORST, which is called FORS (Forest of Random Subsets). The main differences between HORST and FORS are as follows. First, the key generation algorithm picks kt random strings from $\{0,1\}^{l(n)}$ (rather than t strings) and divides them into k groups of t strings. In the signing algorithm, instead of revealing k elements from t strings, FORS reveals *one* element from each group. Second, FORS uses a tweakable hash function $\mathbf{F}$ instead of the one-way function f, where the tweaks are the indices of the strings. Third, instead of using bitmarked hash functions in the hash tree, FORS uses a tweakable hash function $\mathbf{Th}$ in generating hash trees, where the tweaks are the addresses of the nodes. Finally, since there are k hash trees in FORS, it compresses the k roots by calling $\mathbf{Th}$ and denotes the value as the public key.

In FORS, the message is not hashed. The signing algorithm directly splits the message m into k digits with size $\log t$ and then proceeds with the following steps. Thus, the scheme is not EUF-CMA secure. (One can forge a signature on message $m_1||m_2$ given signatures on $m_1||m_2^*$ and $m_1^*||m_2$.) In practice, FORS has to be used on hashed messages to achieve EUF-CMA. We call FORS with integrated hashing as *simplified* FORS (sFORS).

```
$\mathsf{sFORS.KeyGen}(1^\lambda)$
$H = (h_1, ..., h_k) \leftarrow \mathcal{H}$
**for** $(i,j) \in [k] \times [t]$, $s_{i,j} \leftarrow \{0,1\}^{l(n)}$, $y_{i,j} = \mathbf{F}((i,j), s_{i,j})$
**for** $i \in [k]$, $y_i \leftarrow \mathsf{TreeGen}(\mathbf{Th}, t, (y_{i,1}, ..., y_{i,t}))$.
$y_0 = \mathbf{Th}(0, (y_1, ...y_k))$, $S = (s_{1,1}, ..., s_{k,t})$
$pk = (y_0, H)$, $sk = (S, H)$.
**return** $(pk, sk)$.

$\mathsf{sFORS.Sig}(sk, m)$
Parse $S = (s_{1,1}, ..., s_{k,t})$ and $H = (h_1, ..., h_k)$
**for** $i \in [k]$, $x_i = s_{i,h_i(m)}$.
**for** $(i,j) \in [k] \times [t]$, $y_{i,j} = \mathbf{F}((i,j), s_{i,j})$.
**for** $i \in [k]$, $\pi_i \leftarrow \mathsf{TreeProv}(\mathbf{Th}, t, (y_{i,1}, ..., y_{i,t}), h_i(m))$
**return** $\sigma = (x_1, ..., x_k, \pi_1, ..., \pi_k)$.

$\mathsf{sFORS.pkFromSig}(m, \sigma, H)$
Parse $\sigma = (x_1, ..., x_k, \pi_1, ..., \pi_k)$ and $H = (h_1, ..., h_k)$.
**for** $i \in [k]$, $y_i \leftarrow \mathsf{TreeVer}(\mathbf{Th}, t, h_i(m), \mathbf{F}((i,j), x_i), \pi_i)$.
**return** $y_0' = \mathbf{Th}(0, (y_1, ..., y_k))$

$\mathsf{sFORS.Ver}(pk, m, \sigma)$
Parse $pk = (y_0, H)$
**return** $[\![\mathsf{sFORS.pkFromSig}(m, \sigma, H) = y_0]\!]$
```

Fig. 3. Construction of Simplified FORS.

In SPHINCS+ [9], sFORS is never directly used. In each signing operation, it introduces a (pseudo-)randomizer z. The message is then hashed with z, and z is included in the signature. It results in a new scheme that achieves higher bit security. We call it *randomized* FORS (rFORS). The formal constructions are depicted in Figure 3 and 4.

```
$\mathsf{rFORS.KeyGen}(1^\lambda)$
$(pk, sk) \leftarrow \mathsf{sFORS.KeyGen}(1^\lambda)$
$k \leftarrow \{0,1\}^n$
**return** $(pk, (sk, k))$.

$\mathsf{rFORS.Sig}((sk, k), m)$
$z = \mathsf{PRF}(k, m)$
$\sigma \leftarrow \mathsf{sFORS.Sig}(sk', z||m)$
**return** $(z, \sigma)$.

$\mathsf{rFORS.pkFromSig}(m, (z, \sigma), H)$
**return** $\mathsf{sFORS.pkFromSig}(z||m, \sigma, H)$

$\mathsf{rFORS.Ver}(pk, m, (z, \sigma))$
Parse $pk = (y_0, H)$
**return** $[\![\mathsf{rFORS.pkFromSig}(m, (z, \sigma), H) = y_0]\!]$
```

Fig. 4. Construction of Randomized FORS.

It is clear that the security of the few-time HBS schemes is related to the security notions of the underlying hash functions. However, the concrete security is lacking research in the quantum setting. We give formal proof for the concrete Q1 and Q2 security in Appendix E.

3.2 Building Block – HT: The hypertree

HT is another building block in SPHINCS-like structure. It behaves as a stateful signature scheme, where the signing and verification algorithms additionally take as input a state $\mathsf{st} \in ST$. Note that the state space is of polynomial size. The syntax is defined as follows.

Definition 2. *A hyper tree signature scheme $\mathsf{HT} = (\mathsf{HT.KeyGen}, \mathsf{HT.Sign}, \mathsf{HT.Ver})$ consists of three polynomial-time algorithms along with an associated message space $\mathcal{M} = \{\mathcal{M}_n\}$ and a state space ST such that:*

- *The key generation algorithm KeyGen takes as input the security parameter 1^n. It outputs a pair of keys (pk, sk).*

– *For security parameter n, the signing algorithm* Sign *takes as input a secret key sk, a message $m \in \mathcal{M}_n$ and a state $\mathsf{st} \in ST$. It outputs a signature σ.*
– *For security parameter n, the verification algorithm* Ver *takes as input a public key pk, a message $m \in \mathcal{M}_n$, a signature σ and a state $\mathsf{st} \in ST$. It outputs a bit b.*

For any $(pk, sk) \leftarrow \mathsf{KeyGen}(1^n)$, $m \in \mathcal{M}_n$, $\mathsf{st} \in ST$ and $\sigma \leftarrow \mathsf{Sign}(sk, m, \mathsf{st})$, it holds that $\mathsf{Ver}(pk, m, \sigma, \mathsf{st}) = 1$.

Although SPHINCS and SPHINCS+ use different HT, the security notions for HT are the same and implicitly contained in [7,9]. The security is a stateful version of existential unforgeability under non-adaptive chosen message attacks. We call it *existential unforgeability under non-adaptive chosen message attacks with states* (EUF-sNACMA). In detail, the security experiment is defined as follows.

```
Experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}sNACMA}}_{\mathsf{HT},q_s,q_H}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$
  $(pk, sk) \leftarrow \mathsf{KeyGen}(1^n)$
  $(\{(m_i, \mathsf{st}_i)\}_{i \in [q_s]}, S) \leftarrow \mathcal{A}_1()$
  If $\mathsf{st}_i$ are not distinct, return 0
  For $i \in [q_s]$, $\sigma_i \leftarrow \mathsf{Sign}(sk, m_i, \mathsf{st}_i)$
  $(m^*, \sigma^*, \mathsf{st}^*) \leftarrow \mathcal{A}_2(pk, S, \{\sigma_i\}_{i \in [q_s]})$
  If $\mathsf{st}^* = \mathsf{st}_j$ for some $j \wedge m^* \neq m_j \wedge \mathsf{Ver}(pk, m^*, \sigma^*, \mathsf{st}^*) = 1$, return 1,
otherwise return 0.
```

Remark 2. In this paper, we do not depict the detailed construction of HT in SPHINCS or SPHINCS+, since we always use it as a black box. We only care about the security.

3.3 SPHINCS and SPHINCS+

SPHINCS and SPHINCS+ are practical stateless HBS schemes built from the above two blocks. In each signing execution, the signer first pseudorandomly picks a state from ST in HT, which authenticates the public key of the few-time HBS. A signature contains (1) a few-time signature of the message, (2) an HT signature of the public key of the corresponding few-time HBS, and (3) a (pseudo-)randomizer. Note that in SPHINCS and its variants, the few-time signatures are never explicitly verified. Instead, it uses $\mathsf{pkFromSig}$ to recover the public key from the message and signature. The security of HT guarantees that only the real public key can be verified in the next steps.

In SPHINCS, the few-time signature scheme is HORST (HORS with trees), a variant of HORS. It compresses the HORS public key by a Merkle tree structure, and the compressed public key can be generated by an algorithm $\mathsf{pkFromSig}$ from the message and the signature. In this section, we also use $\mathsf{HORS.Sig}$ and $\mathsf{HORS.pkFromSig}$ to denote the algorithms of HORS with trees.

The outlines of SPHINCS and SPHINCS+ are depicted in Figure 5. The main differences are the choices of few-time signature schemes and the ways to pick the index. In particular, the index of SPHINCS+ is calculated by a hash value of the message and the randomizer. Thus, it is not directly contained in the signature.

```
$\mathsf{SPHINCS.KeyGen}(1^\lambda)$
    $sk_{\mathsf{seed}} \leftarrow \{0,1\}^n$,
    $(pk_{\mathsf{HT}}, sk_{\mathsf{HT}}) \leftarrow \mathsf{HT.KeyGen}(1^\lambda)$
    $sk = (sk_{\mathsf{seed}}, sk_{\mathsf{HT}})$, $pk = pk_{\mathsf{HT}}$
    **Output** $(pk, sk)$.

$\mathsf{SPHINCS.Sig}(sk, m)$
    $z = \mathsf{PRF}_{\mathsf{msg}}(sk_{\mathsf{seed}}, m)$
    $idx = \mathsf{PRF}_{\mathsf{idx}}(sk_{\mathsf{seed}}, m)$
    $s_{idx} = \mathsf{PRF}_{\mathsf{seed}}(sk_{\mathsf{seed}}, idx)$
    $(pk_{\mathsf{HORS}}, sk_{\mathsf{HORS}}) \leftarrow \mathsf{HORS.KeyGen}(1^\lambda; s_{idx})$
    $\sigma_{\mathsf{HORS}} \leftarrow \mathsf{HORS.Sig}(sk_{\mathsf{HORS}}, z||m)$
    $\sigma_{\mathsf{HT}} \leftarrow \mathsf{HT.Sig}(sk_{\mathsf{HT}}, pk_{\mathsf{FORS}}, idx)$
    **return** $(idx, z, \sigma_{\mathsf{HT}}, \sigma_{\mathsf{FORS}})$.

$\mathsf{SPHINCS.Ver}(pk, m, (idx, z, \sigma_{\mathsf{HT}}, \sigma_{\mathsf{FORS}}))$
    $pk_{\mathsf{HORS}} \leftarrow \mathsf{HORS.pkFromSig}(z||m, \sigma_{\mathsf{HORS}})$
    **return** $\mathsf{HT.Ver}(pk_{\mathsf{HT}}, pk_{\mathsf{HORS}}, \sigma_{\mathsf{HT}}, idx)$
```

```
$\mathsf{SPHINCS+.KeyGen}(1^\lambda)$
    $sk_{\mathsf{seed}} \leftarrow \{0,1\}^n$, $h_0 \leftarrow \mathcal{H}_0$
    $(pk_{\mathsf{HT}}, sk_{\mathsf{HT}}) \leftarrow \mathsf{HT.KeyGen}(1^\lambda)$
    $sk = (sk_{\mathsf{seed}}, sk_{\mathsf{HT}})$, $pk = (pk_{\mathsf{HT}}, h_0)$
    **Output** $(pk, sk)$.

$\mathsf{SPHINCS+.Sig}(sk, m)$
    $z = \mathsf{PRF}_{\mathsf{msg}}(sk_{\mathsf{seed}}, m)$
    $idx = h_0(z||m)$
    $s_{idx} = \mathsf{PRF}_{\mathsf{seed}}(sk_{\mathsf{seed}}, idx)$
    $(pk_{\mathsf{FORS}}, sk_{\mathsf{FORS}}) \leftarrow \mathsf{sFORS.KeyGen}(1^\lambda; s_{idx})$
    $\sigma_{\mathsf{FORS}} \leftarrow \mathsf{sFORS.Sig}(sk_{\mathsf{FORS}}, z||m)$
    $\sigma_{\mathsf{HT}} \leftarrow \mathsf{HT.Sig}(sk_{\mathsf{HT}}, pk_{\mathsf{FORS}}, idx)$
    **return** $(z, \sigma_{\mathsf{HT}}, \sigma_{\mathsf{FORS}})$.

$\mathsf{SPHINCS+.Ver}(pk, m, (z, \sigma_{\mathsf{HT}}, \sigma_{\mathsf{FORS}}))$
    $idx = h_0(z||m)$
    $pk_{\mathsf{FORS}} \leftarrow \mathsf{sFORS.pkFromSig}(z||m, \sigma_{\mathsf{FORS}})$
    **return** $\mathsf{HT.Ver}(pk_{\mathsf{HT}}, pk_{\mathsf{FORS}}, \sigma_{\mathsf{HT}}, idx)$
```

Fig. 5. The outline of SPHINCS and SPHINCS+

Remark 3. The scheme in Figure 5 is the deterministic version of SPHINCS+. It can be converted into a probabilistic version by adding a random salt to the input of $\mathsf{PRF}_{\mathsf{msg}}$. In this paper, we mainly focus on the deterministic version and will discuss the probabilistic version in Section 5.2.

4 Quantum Chosen Message Attacks on SPHINCS(+)

4.1 Quantum Chosen Message Attacks on SPHINCS

Let q_s be the number of signing queries. The optimal attack requires approximately 2^{128} hash queries to break the EUF-CMA security of SPHINCS-256 when $q_s = 2^{50}$. Approximately the same number of hash queries are needed to break the EUF-CMA security of SPHINCS+-256s when $q_s = 2^{64}$. Our attack shows that if the signing oracle is quantum-accessible, the security will be lower.

In SPHINCS, $idx = \mathsf{PRF}_{\mathsf{idx}}(sk_{\mathsf{seed}}, \mathsf{PRF}_{\mathsf{msg}}(sk_{\mathsf{seed}}, m))$. Fix sk_{seed} and let $f(m) = \mathsf{PRF}_{\mathsf{idx}}(sk_{\mathsf{seed}}, \mathsf{PRF}_{\mathsf{msg}}(sk_{\mathsf{seed}}, m))$ be the function mapping m to idx.

Since idx is a part of the signature, f can be computed by querying the signing oracle. By using Grover's algorithm, one can search a message m mapping to any index after $O(2^{h/2})$ queries to the signing oracle.

Note that any index authenticates a key pair of the few-time signature scheme. By repeating the above steps r times, one can obtain r message-signature pairs w.r.t. the same few-time signature key pair. If the secret key of the few-time signature is used too many times, the security level will be degraded rapidly.

The formal description of the attack on SPHINCS is as follows:

1. Denote $f(m) = \mathsf{PRF}_{\mathsf{idx}}(sk_{\mathsf{seed}}, \mathsf{PRF}_{\mathsf{msg}}(sk_{\mathsf{seed}}, m))$ be the function that maps the message m to the index $idx \in \{0,1\}^h$. Randomly pick $idx^* \in \{0,1\}^h$, and denote predicate $F(m) = 1$ iff $f(m) = idx^*$. Here, f and F are quantum-computable by querying the signing oracle $|\mathsf{SigO}\rangle$ since $|\mathsf{SigO}\rangle$ maps $|m, 0^h\rangle$ to $|m, f(m)\rangle$ by discarding the signature register except the index part.
2. Run Grover's algorithm on $F(m)$. It returns a random m such that $F(m) = 1$. It implies a HORS signature labeled by idx^*. This requires $O(2^{h/2})$ quantum queries to the signing oracle. [7]
3. Repeat the previous step r times. This requires $O(r2^{h/2})$ quantum queries to the signing oracle. [8] Let S be the set of labels of which the preimages have appeared in the HORS signatures. In other words, let $m^{(1)}, ..., m^{(r)}$ be the outputs of the Grover's algorithm and $z^{(j)}$ be the pseudorandomness z in the signature of $m^{(j)}$, then $S = \{h_i(z^{(j)}||m^{(j)})\}_{i\in[k], j\in[r]}$.
   Note that for each s_j, the probability of appearing in a HORS signature is k/t. The probability of appearing in r random signatures is $1-(1-k/t)^r$. Thus, the expectation of $|S|$ is
   $$\mathbb{E}[|S|] = \left(1 - \left(1 - \frac{k}{t}\right)^r\right) \cdot t \geq (1 - e^{-\frac{kr}{t}}) \cdot t,$$
   where the inequality comes from $(1-x)^\alpha \leq e^{-\alpha x}$.
4. Denote function $G(z||m) = 1$ iff $\{h_i(z||m)\}_{i\in[k]} \subset S$. Run Grover's algorithm on G. It outputs $z^*||m^*$ whose corresponding preimages have appeared in the HORS signatures. The expected number of quantum hash queries in this step is
   $$O\left(\sqrt{\left(\frac{t}{|S|}\right)^k}\right) = O((1 - e^{\frac{kr}{t}})^{-\frac{k}{2}}).$$

[7] In this paper, we always use $O(\cdot)$ to describe the (upper-bound) number of queries due to the implement of Grover's algorithm. Note that Grover's search could fail in the case that the number of preimages is unknown (we only know the expected number). It can be solved by repeating it constant times [26]. Since the repetition only causes (at most) a constant factor on the time complexity, we ignore its effect and only focus on the complexity related to the parameters of SPHINCS(+) and r.

[8] We require the r results be distinct. It can be guaranteed by a simple modification. Let $i \in [r]$ and $f^{(i)}(m) = f(C_i||m)$, where C_i denotes the binary expression of i. Define $F^{(i)}(m) = 1$ iff $f^{(i)}(m) = idx^*$ and run Grover's algorithm on $F^{(i)}$ for each $i \in [r]$ instead. Thus, the results have distinct prefix of length $\lceil \log r \rceil$.

5. Note that the signing oracle has been queried $q_s = O(r2^{h/2})$ times. The attacker needs to return at least (q_s+1) forgeries. Step 4 needs to be repeated $(q_s + 1 - r)$ times. (It is unnecessary to compute σ_{HT} for the new forgeries since all the forgeries share a common σ_{HT}.) The total number of hash queries is

$$q_H = O(q_s + 1 - r) \cdot O((1 - e^{-\frac{kr}{t}})^{-\frac{k}{2}}) = O((1 - e^{-\frac{kr}{t}})^{-\frac{k}{2}} \cdot r2^{\frac{h}{2}}).$$

In SPHINCS-256, $k = 32$, $t = 2^{16}$ and $h = 60$. When $r = 2^{10}$, q_s and q_H are approximately 2^{40} and 2^{61}, respectively. When $r = 2^{14}$, q_s reaches 2^{43} and q_H decreases to 2^{43} as well. It is much lower than the level of EUF-CMA security, where $q_s = 2^{50}$, and q_H is expected to be 2^{128}.

4.2 Quantum Chosen Message Attacks on SPHINCS+

In SPHINCS+, the index is not directly contained in the signature. Instead, the index is computed by $idx = h_0(z||m)$, where z is a (pseudo-)randomizer and part of the signature. It is not a big issue since we can modify the condition on Grover's algorithm to find a malicious randomizer z mapping to our malicious index. In this section, we simply denote by H the hash computation of h_0 and $(h_1, ..., h_k)$. (In practice, h_0 and $(h_1, ..., h_k)$ are parts of a function H.)

Our attack on SPHINCS+ is as follows:

1. Let $z(m)$ be the function mapping from m to the corresponding z (which can be evaluated by a signing query). For some $idx^* \in \{0,1\}^h$, denote predicate $F(m) = 1$ iff $h_0(z(m)||m) = idx^*$. Similarly, F is quantum-computable by querying $|\mathsf{SigO}\rangle$ and a quantum query to h_0.
2. Run Grover's algorithm on $F(m)$. It outputs a random m such that $F(m) = 1$. It implies a sFORS signature labeled by idx^*. This requires $O(2^{h/2})$ quantum queries to the signing oracle and $O(2^{h/2})$ quantum queries to h_0.
3. Repeat the previous step r times. This requires $O(r2^{h/2})$ quantum queries to the signing oracle. For $i \in [k]$, let S_i be the set of labels of which the preimages have appeared in the i-th tree of sFORS signatures. In other words, $S_i = \{h_i(z(m^{(j)})||m^{(j)})\}_{j\in[r]}$.
   For each $s_{i,j}$, the probability of appearing in an sFORS signature is $1/t$. The probability of appearing in r random signatures is $1 - (1 - 1/t)^r$. Thus, the expectation of $|S_i|$ is

$$\mathbb{E}[|S_i|] = \left(1 - \left(1 - \frac{1}{t}\right)^r\right) \cdot t \geq (1 - e^{-\frac{r}{t}}) \cdot t,$$

   and thus

$$\mathbb{E}\left[\prod_{i\in[k]} |S_i|\right] \geq (1 - e^{-\frac{r}{t}})^k \cdot t^k.$$

4. Denote function $G(z||m) = 1$ iff $\forall i \in [k], h_i(z||m) \in S_i \wedge h_0(z||m) = idx^*$. Run Grover's algorithm on G. It outputs $z^*||m^*$ whose corresponding preimages have appeared in sFORS signatures. The expected number of quantum hash queries in this step is

$$O\left(\sqrt{\left(\frac{2^h \cdot t^k}{\prod_{i\in[k]} |S_i|}\right)}\right) = O((1-e^{-\frac{r}{t}})^{-\frac{k}{2}} \cdot 2^{\frac{h}{2}}).$$

5. The signing oracle has been queried $q_s = O(r2^{h/2})$ times. The forgery needs to contain at least (q_s+1) forgeries. Step 4 needs to be repeated (q_s+1-r) times. The total number of hash queries is

$$q_H = O(r2^{\frac{h}{2}}) + O(q_s+1-r) \cdot O((1-e^{-\frac{r}{t}})^{-\frac{k}{2}} \cdot 2^{\frac{h}{2}}) = O((1-e^{-\frac{r}{t}})^{-\frac{k}{2}} \cdot r2^h).$$

In SPHINCS+-256s, $k = 22$, $t = 2^{14}$ and $h = 64$. When $r = 2^{16}$, q_s and q_H are approximately 2^{48} and 2^{80}, respectively. It is also lower than the level of EUF-CMA security, where $q_s = 2^{64}$ and q_H is expected to be 2^{128}.

Remark 4. In a quantum algorithm, it may be problematic to maintain a quantum state with large quantum memory (especially for HBS schemes due to the large size of signatures). However, it is not an big issue since our attacks only need the index idx (or the randomizer z) in the iterations of Grover's algorithm, and main parts of the quantum responses can be discarded.

4.3 Attacks in the BU model

Note that in the above attacks, the adversary has had the ability to forge a signature for any message before the final step. Thus, if we use the BU model instead, the adversary only need to forge one signature for a message in $B_{\varepsilon,n}$, and the large number of computations in the final step can be saved.

The strategy of the attack on SPHINCS in the BU model is similar to the above one. The difference is: After the adversary obtain S, it does not directly search a $z||m$ such that $G(z||m) = 1$. Instead, the adversary first search a message $m^* \in B_{\varepsilon,n}$ by querying the blind signing oracle. Then, it search z such that $G(z||m^*) = 1$. It guarantees that the forgery is for a message in the blind region. See details in Appendix D.

5 SPHINCS-FORS: A Provably Secure HBS Scheme against Quantum Chosen Message Attacks

5.1 Generic Security of Few-time HBS Schemes

As a preparatory work, we begin with the security of the few-time HBS schemes, which are used as building blocks in SPHINCS(+). To the best of our knowledge, no quantum generic security bound is given before this work (even in the sense of EUF-CMA). We fill the gap with the following corollaries.

Corollary 1. *Let the hash functions in HORS and sFORS be modeled as quantum random oracles. It holds that*

$$Adv^{\textsf{EUF-CMA}}_{\textsf{HORS},r,q}(\mathcal{A}) \leq O\left(q^{2(r+1)}\left(\frac{rk}{t}\right)^k + \frac{q^2kt}{2^n}\right),$$

$$Adv^{\textsf{EUF-CMA}}_{\textsf{sFORS},r,q}(\mathcal{A}) \leq O\left(q^{2(r+1)}\left(\frac{r}{t}\right)^k + \frac{q^2}{2^n}\right),$$

$$Adv^{\textsf{EUF-qCMA}}_{\textsf{sFORS},r,q}(\mathcal{A}) \leq O\left(q^{2(r+1)}\left(\frac{r^2}{t}\right)^k + \frac{q^2kt^r}{2^n}\right).$$

Corollary 2. *Let the hash functions in rFORS be modeled as quantum random oracles. It holds that*

$$Adv^{\textsf{EUF-CMA}}_{\textsf{rFORS},r,q}(\mathcal{A}) \leq O\left(\sqrt{\frac{q}{2^n}} + q^2\left(\frac{r}{t}\right)^k + \frac{q^2}{2^n}\right),$$

$$Adv^{\textsf{EUF-qCMA}}_{\textsf{rFORS},r,q}(\mathcal{A}) \leq O\left(q^{2(r+1)}\left(\frac{r^2}{t}\right)^k + \frac{q^2kt^r}{2^n}\right).$$

The security is proven in two steps. First, we observe that the security is related to different variants of subset-resilient hash functions. Thus, we evaluate the generic security of variants of subset resilience in Appendix C.2. Next, we attempt to reduce the security of few-time HBS schemes to the variant of subset resilience (and other assumptions). See details in Appendix E.

5.2 Discussion: How to Avoid Quantum Attacks?

We then discuss how to construct a many-time signature scheme with provable security equipped with above few-time HBS schemes .

In the attacks in Section 4, the key idea is to search messages that map to a single index idx^*. The search is done by iteratively running a function F in Grover's algorithm. A simple improvement to avoid these attacks is making F randomized. That is, in each signing operation, the signer adds a random nonce in calculating the pseudorandomness $z = \mathsf{PRF}_{\mathsf{msg}}(sk_{\mathsf{seed}}, m)$. It is indeed the probabilistic version of SPHINCS+ [9]. Note that the nonce does not affect the security reduction of EUF-CMA, but does affect the EUF-qCMA security.

Unfortunately, we cannot give a *security proof* of EUF-qCMA security for the above variants, even if the random nonce is well sampled. Note that the security under quantum chosen message attacks of SPHINCS(+) is more complicated for the following reasons. First, in the classical setting, a response of the signing query contains only one few-time signature. Since idx may differ in superpositions in the quantum-access setting, a quantum SPHINCS(+) signature may contain many few-time signatures for *multiple* key pairs. This multi-instance case exceeds the discussion in Section 5.1. Second, a quantum SPHINCS(+) signature may also contain a large number of HT signatures σ_{HT} in superpositions. It makes the analysis even more complicated.

So how do we construct a *provably secure* hash-based signature scheme under qCMAs? Our solution is simple. The first step is to make each signing response only contain few-time signatures related to *one* key pair. For this purpose, we make the index of the few-time signature independent to the message. In each signing query, the signing algorithm randomly picks a leaf from $\{0,1\}^h$ instead of running the pseudorandom function on the message. Since the randomness of a signing query is global, the resulting signatures in superpositions share common randomness and thus a common idx, implying a common few-time signature key pair. In addition, note that σ_{HT} is the signature on the few-time signature public key. Since all superpositions share a common public key, the resulting σ_{HT} is also identical in all superpositions. The security is then reduced to the quantum-access security of the few-time signature scheme in the single-instance case, which has been evaluated in Section 5.1.

This variant can avoid the above attacks since the index is independent of the message. However, note that the random index needs to be directly contained in the signature, so an adversary can arbitrarily choose an index in the forgeries. It causes lower security than SPHINCS+, especially in the EUF-CMA model. Thus, the classical security also needs re-analyzed.

In the next subsection, we present SPHINCS-FORS, a variant of SPHINCS+ that follows the approach and provides provable EUF-qCMA security.

5.3 SPHINCS-FORS

Construction 1 *Let* $\mathsf{PRF}_{\mathsf{seed}} : \{0,1\}^n \times \{0,1\}^h \to \{0,1\}^n$ *be a pseudorandom function, and* rFORS *and* HT *be depicted in Subsection 3.1 and 3.2. SPHINCS-FORS is depicted in Figure 6.*

The difference from SPHINCS+ is as follows:

- The strategies for choosing the index are different. In SPHINCS-FORS, the index is truly random in $\{0,1\}^h$ while in SPHINCS+ it is pseudorandom related to m. The random index is directly contained in the resulting signature.
- In SPHINCS-FORS, we use rFORS as the few-time signature. A minor difference is that in SPHINCS+, the (pseudo-)randomizer z is calculated by a global function $\mathsf{PRF}_{\mathsf{msg}}(\mathsf{sk}_{\mathsf{seed}}, m)$ while in SPHINCS-FORS, sk_{seed} differs in different indices.

5.4 Security Analysis

In this subsection, we analyze the security of SPHINCS-FORS under (quantum) chosen message attacks.

At first, we need to evaluate the security for HT. As we use the same HT as SPHINCS+ and its (EUF-sNACMA) security has been evaluated in [9, 26], we omit the formal analysis of HT. The success probability of breaking the security is at most $O(q_H \cdot 2^{-n/2})$, where q_H denotes the number of hash queries.

```
SPHINCS-FORS.KeyGen($1^\lambda$)
$sk_{\mathsf{seed}} \leftarrow \{0,1\}^n$,
$(pk_{\mathsf{HT}}, sk_{\mathsf{HT}}) \leftarrow \mathsf{HT}.\mathsf{KeyGen}(1^\lambda)$
$sk = (sk_{\mathsf{seed}}, sk_{\mathsf{HT}})$, $pk = pk_{\mathsf{HT}}$
Output $(pk, sk)$.

SPHINCS-FORS.Sig($sk, m$)
$idx \leftarrow \{0,1\}^h$
$s_{idx} = \mathsf{PRF}_{\mathsf{seed}}(sk_{\mathsf{seed}}, idx)$
$(pk_{\mathsf{FORS}}, sk_{\mathsf{FORS}}) \leftarrow \mathsf{rFORS}.\mathsf{KeyGen}(1^\lambda; s_{idx})$
$(z, \sigma_{\mathsf{FORS}}) \leftarrow \mathsf{rFORS}.\mathsf{Sig}(sk_{\mathsf{FORS}}, m)$
$\sigma_{\mathsf{HT}} \leftarrow \mathsf{HT}.\mathsf{Sig}(sk_{\mathsf{HT}}, pk_{\mathsf{FORS}}, idx)$
return $(idx, z, \sigma_{\mathsf{HT}}, \sigma_{\mathsf{FORS}})$.

SPHINCS-FORS.Ver($pk, m, (idx, z, \sigma_{\mathsf{HT}}, \sigma_{\mathsf{FORS}})$)
$pk_{\mathsf{FORS}} \leftarrow \mathsf{rFORS}.\mathsf{pkFromSig}(m, (z, \sigma_{\mathsf{FORS}}))$
return $\mathsf{HT}.\mathsf{Ver}(pk_{\mathsf{HT}}, pk_{\mathsf{FORS}}, \sigma_{\mathsf{HT}}, idx)$
```

Fig. 6. The framework of SPHINCS-FORS

For a signature scheme Γ, let $\mathrm{InSec}^{*}_{\Gamma,r,q_H}(\xi)$ be the maximum of $\mathrm{Adv}^{*}_{\Gamma,r,q_H}(\mathcal{A})$ for all ξ-time adversary $\mathcal{A}$ and $* \in \{\mathsf{EUF\text{-}CMA}, \mathsf{EUF\text{-}qCMA}, \mathsf{EUF\text{-}sNACMA}\}$. The security of SPHINCS-FORS is proven as follows.

Theorem 1. *For any ξ-time adversary $\mathcal{A}$, it holds that*

$$\begin{aligned} Adv^{*}_{\textsf{SPHINCS-FORS},q_s,q_H}(\mathcal{A}) \leq & InSec^{\textsf{EUF-sNACMA}}_{\textsf{HT},2^h,q_H}(\xi) + InSec^{\textsf{Ind-PRF}}_{\textsf{PRF}_{\textsf{seed}},2^h}(\xi) \\ & + \sum_{r=0}^{q_s} p(r,q_s) \cdot InSec^{*}_{\textsf{rFORS},r,q_H}(\xi), \end{aligned}$$

where $p(r,q_s) = \min\{2^{r(\log q_s - h)+h-\log r!}, 2^h\}$ and $* \in \{\textsf{EUF-CMA}, \textsf{EUF-qCMA}\}$.

From Theorem 1, Corollary 2 and Lemma 4, we have

Corollary 3. *Let the hash functions in SPHINCS-FORS be modeled as quantum random oracles. It holds that*

$$Adv^{\textsf{EUF-qCMA}}_{\textsf{SPHINCS-FORS},q_s,q_H}(\mathcal{A}) \leq O\left(\frac{q_H}{2^{n/2}} + \sum_{r=0}^{q_s} p(r,q_s) \cdot \min\left\{q_H^{2(r+1)}\left(\frac{r^2}{t}\right)^k + \frac{q_H^2 k t^r}{2^n}, 1\right\}\right), \tag{3}$$

$$Adv^{\textsf{EUF-CMA}}_{\textsf{SPHINCS-FORS},q_s,q_H}(\mathcal{A}) \leq O\left(q_s\sqrt{\frac{q_H+q_s}{2^n}} + \frac{q_H}{2^{n/2}} + q_H^2 \sum_{r=0}^{q_s} p(r,q_s)\left(\frac{r}{t}\right)^k\right). \tag{4}$$

Note that here the term caused by adaptive reprogramming is $O\big(q_s\sqrt{\frac{q_H+q_s}{2^n}}\big)$ rather than $\sum_{r=0}^{q_s} p(r,q_s)\cdot\frac{3r}{2}\sqrt{\frac{q_H+r+1}{2^n}}$. It is because that the random oracle is reprogrammed at most q_s times in the reduction. For the same reason the terms caused by SM-TCR, SM-DSPR and Ind-PRF are gathered to $O(q_H\cdot 2^{-n/2})$.

5.5 Concrete Security

As a variant of SPHINCS+, SPHINCS-FORS is expected to reach (at least) the same security level as SPHINCS+. However, we note that the EUF-CMA security level of SPHINCS-FORS is lower than SPHINCS+ with the same parameters. It is because of the different strategies of choosing the index. Since the index is directly contained in the signature (just like what SPHINCS does), the adversary is able to arbitrarily choose a target index to forge a signature. In Appendix G, we show a concrete attack on SPHINCS-FORS, implying the difference in security levels with SPHINCS+.

Thus, we need to increase some parameters to reach the same level as SPHINCS+ in the sense of EUF-CMA, and meanwhile provide provable security in the sense of EUF-qCMA. As a result, the signature size and running time will become larger.

We give two instances with different parameters and security levels. See details of the parameters in Table 2.

- As is observed in Section 4.2, (deterministic) SPHINCS+-256s can provide *at most* 80-bit quantum-access security when $q_s = 2^{48}$. By adapting the parameters k, t, and h, we result in SPHINCS-FORS v1 that can provide *at least* 80-bit security in the sense of provable quantum-access security.
- If we directly increase h in SPHINCS+-256s to 104, say SPHINCS+-256s$*$, it can provide *at most* 128-bit quantum-access security (due to our attacks) when $q_s = 2^{64}$. On the other hand, by adapting the parameters, we result in SPHINCS-FORS v2 that can provide *at least* 128-bit security.

6 Conclusion and Open Questions

This paper analyzes the quantum-access security of HBS schemes in two directions. First, we show quantum chosen message attacks (or superposition attacks) on stateless HBS schemes, such as SPHINCS and SPHINCS+. The time complexity of the quantum chosen message attacks is lower than the optimal attacks in the classical setting. Next, we propose a variant of SPHINCS+ called SPHINCS-FORS. It is a provably secure HBS scheme against quantum chosen message attacks. As far as we know, it is the first practical HBS scheme with provable security against quantum chosen message attacks.

Note that our attacks do not work on the probabilistic version of SPHINCS+. Since there is no security proof, it is an open question whether probabilistic SPHINCS+ is secure against quantum chosen message attacks. In addition, our security bound of SPHINCS-FORS against quantum chosen message attacks is possibly non-tight. It shows a lower bound of the security, but we are not aware of any concrete attacks that reach this bound. It is also an open question whether we can get a tighter bound or if there exists a better attack.

7 Acknowledgement

We would like to thank Tsuyoshi Takagi for valuable suggestions and anonymous reviewers of ACISP 2023 for helpful comments. Quan Yuan is supported by JST CREST Grant Number JPMJCR2113, Japan.

References

1. S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM (JACM)*, 51(4):595–605, 2004.
2. G. Alagic, C. Majenz, A. Russell, and F. Song. Quantum-access-secure message authentication via blind-unforgeability. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 788–817. Springer, 2020.
3. J.-P. Aumasson, D. J. Bernstein, W. Beullens, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, A. Hülsing, P. Kampanakis, S. Kölbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe, and B. Westerbaan. SPHINCS+ - submission to the nist post-quantum project v.3. 2020.
4. J.-P. Aumasson and G. Endignoux. Clarifying the subset-resilience problem. 2017. Cryptology ePrint Archive, Report 2017/909.
5. J.-P. Aumasson and G. Endignoux. Improving stateless hash-based signatures. In *Topics in Cryptology - CT-RSA 2018*, volume 10808 of *LNCS*, pages 219–242. Springer, 2018.
6. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
7. D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. S. Schwabe, and Z. Wilcox-O'Hearn. SPHINCS: Practical stateless hash-based signatures. In *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 368–397. Springer, 2015.
8. D. J. Bernstein and A. Hülsing. Decisional second-preimage resistance: when does SPR imply PRE? In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 33–62. Springer, 2019.
9. D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe. The SPHINCS+ signature framework. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2129–2146. Association for Computing Machinery, 2019.
10. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In *International conference on the theory and application of cryptology and information security*, pages 41–69. Springer, 2011.

11. D. Boneh and M. Zhandry. Quantum-secure message authentication codes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 592–608. Springer, 2013.
12. D. Boneh and M. Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *Advances in Cryptology - CRYPTO 2013*, volume 8043 of *LNCS*, pages 361–379. Springer, 2013.
13. X. Bonnetain, A. Hosoyamada, M. Naya-Plasencia, Y. Sasaki, and A. Schrottenloher. Quantum attacks without superposition queries: the offline Simon's algorithm. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 552–583. Springer, 2019.
14. S. Bouaziz–Ermann, A. B. Grilo, and D. Vergnaud. Quantum security of subset cover problems. *Cryptology ePrint Archive*, 2022.
15. J. Buchmann, E. Dahmen, and A. Hülsing. XMSS - a practical forward secure signature scheme based on minimal security assumptions. In *PQCrypto 2011: Post-Quantum Cryptography*, volume 7071 of *LNCS*, pages 117–129. Springer, 2011.
16. R. Chatterjee, K.-M. Chung, X. Liang, and G. Malavolta. A note on the post-quantum security of (ring) signatures. In *IACR International Conference on Public-Key Cryptography*, pages 407–436. Springer, 2022.
17. C. Cremers, S. Düzlü, R. Fiedler, M. Fischlin, and C. Janson. Buffing signature schemes beyond unforgeability and the case of post-quantum signatures. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1696–1714. IEEE, 2021.
18. T. Gagliardoni, A. Hülsing, and C. Schaffner. Semantic security and indistinguishability in the quantum world. In *Annual international cryptology conference*, pages 60–89. Springer, 2016.
19. S. Garg, H. Yuen, and M. Zhandry. New security notions and feasibility results for authentication of quantum data. In *Annual International Cryptology Conference*, pages 342–371. Springer, 2017.
20. O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, Cambridge, UK, 2004.
21. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on computing*, 17(2):281–308, 1988.
22. A. B. Grilo, K. Hövelmanns, A. Hülsing, and C. Majenz. Tight adaptive reprogramming in the qrom. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 637–667. Springer, 2021.
23. L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
24. A. Hosoyamada and T. Iwata. 4-round luby-rackoff construction is a qprp. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 145–174. Springer, 2019.
25. A. Hosoyamada and Y. Sasaki. Cryptanalysis against symmetric-key schemes with online classical queries and offline quantum computations. In *Cryptographers' Track at the RSA Conference*, pages 198–218. Springer, 2018.
26. A. Hülsing and M. Kudinov. Recovering the tight security proof of sphincs+. In *Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part IV*, pages 3–33. Springer, 2023.
27. A. Hülsing, J. Rijneveld, and P. Schwabe. Armed SPHINCS. In *Public-Key Cryptography–PKC 2016*, pages 446–470. Springer, 2016.

28. A. Hülsing, J. Rijneveld, and F. Song. Mitigating multi-target attacks in hash-based signatures. In *Public-Key Cyptography - PKC 2016*, volume 9614 of *LNCS*, pages 387–416. Springer, 2016.
29. L. Lamport. Constructing digital signatures from a one way function. Technical report, Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.
30. C. Majenz, C. M. Manfouo, and M. Ozols. Quantum-access security of the winternitz one-time signature scheme. *arXiv preprint arXiv:2103.12448*, 2021.
31. R. C. Merkle. A certified digital signature. In G. Brassard, editor, *Advances in Cryptology - CRYPTO '89*, volume 435 of *LNCS*, pages 218–238. Springer, 1989.
32. L. Reyzin and N. Reyzin. Better than biba: Short one-time signatures with fast signing and verifying. In *ACISP 2002: Information Security and Privacy*, volume 2384 of *LNCS*, pages 144–153. Springer, 2002.
33. K. Xagawa and T. Yamakawa. (tightly) qcca-secure key-encapsulation mechanism in the quantum random oracle model. In *Post-Quantum Cryptography: 10th International Conference, PQCrypto 2019, Chongqing, China, May 8–10, 2019 Revised Selected Papers 10*, pages 249–268. Springer, 2019.
34. T. Yamakawa and M. Zhandry. Classical vs quantum random oracles. In *Advances in Cryptology - EUROCRYPT 2021*, volume 12697 of *LNCS*, pages 568–597. Springer, 2021.
35. Q. Yuan, M. Tibouchi, and M. Abe. On subset-resilient hash function families. *Designs, Codes and Cryptography*, pages 1–40, 2022.
36. M. Zhandry. How to construct quantum random functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 679–687, 2012.
37. M. Zhandry. A note on the quantum collision and set equality problems. *Quantum Information and Computation*, 15(7-8):557–567, 2015.
38. M. Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Annual International Cryptology Conference*, pages 239–268. Springer, 2019.
39. K. Zhang, H. Cui, and Y. Yu. SPHINCS-α: A compact stateless hash-based signature scheme. *Cryptology ePrint Archive*, 2022.

A Primitives

A.1 Hash Functions

Definition 3. *(Efficient function family ensemble.) A function family ensemble $\mathcal{F} = \{\mathcal{F}_n : \mathcal{D}_n \to \mathcal{R}_n\}_{n\in\mathbb{N}}$ is efficient if:*

- *$\mathcal{F}$ is samplable: there exists a probabilistic polynomial-time algorithm such that given 1^n, it outputs the description of a uniform element of $\mathcal{F}_n$.*
- *$\mathcal{F}$ can be efficiently computed: there exists a deterministic polynomial-time algorithm such that given $x \in \mathcal{D}_n$ and $f \in \mathcal{F}_n$, it outputs $f(x)$.*

Definition 4. *(One-wayness.) Let $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^{l(n)} \to \{0,1\}^n\}$ be an efficient function family ensemble. We say that $\mathcal{F}$ is a one-way function family (OWF) if for any probabilistic polynomial-time algorithm $\mathcal{A}$ with q queries of f, there exists a negligible function $\epsilon(\cdot)$ such that*

$$Adv^{\mathsf{OW}}_{\mathcal{F},q}(\mathcal{A}) \triangleq \Pr_{f\leftarrow\mathcal{F}_n, x\leftarrow\{0,1\}^{l(n)}}\left[f(x') = f(x) \middle| x' \leftarrow \mathcal{A}(1^n, f, f(x))\right] \leq \epsilon(n)$$

for large enough $n \in \mathbb{N}$.

In particular, if $\mathcal{F}$ is compressing, we say it is a hash function family. Usually, we suppose the input of a hash function is a string of arbitrary length ($\mathcal{D}_n = \{0,1\}^*$). If a hash function family is one-way, we say it is a preimage-resilient hash function family (PRE).

A tweakable hash function is a special hash function taking as input a message m together with a tweak T and a public parameter P. Especially, let $\mathcal{T}$ be the tweak space, $\mathcal{P}$ be the public parameter space and $\mathcal{M}$ be the message space, a tweakable hash function **Th** is defined as

$$\mathbf{Th} : \mathcal{P} \times \mathcal{T} \times \mathcal{M} \to \{0,1\}^n.$$

Definition 5. *(SM-TCR. [9]) Let $\boldsymbol{Th}$ be a tweakable hash function defined above. Let $p \leq |\mathcal{T}|$. For an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathcal{A}_1$ is allowed to give p queries to an oracle $\boldsymbol{Th}(P,\cdot,\cdot)$ where P is uniformly chosen from $\mathcal{P}$. Denote the set of $\mathcal{A}_1$'s queries be $Q = \{(T_i, M_i)\}_{i=1}^p$ and define the predicate $\mathsf{DIST}(\{T_i\}_{i=1}^p) = 1$ iff all tweaks are distinct. Then, $\mathcal{A}_2$ takes as input (Q, P) and the state of $\mathcal{A}_1$, and finally outputs (j, M). Define*

$$Adv^{\mathsf{SM\text{-}TCR}}_{\boldsymbol{Th},p,q}(\mathcal{A}) \triangleq \Pr_{P\leftarrow\mathcal{P}}[\boldsymbol{Th}(P, T_j, M_j) = \boldsymbol{Th}(P, T_j, M) \wedge M \neq M_j \wedge \mathsf{DIST}(\{T_i\}_{i=1}^p) = 1],$$

where q denotes the maximum number of queries to $\boldsymbol{Th}$.

We say that $\boldsymbol{Th}$ is single-function multi-target target-collision-resistant for distinct tweaks (SM-TCR) if for any polynomial-time adversary $\mathcal{A}$, there exists a negligible function $\epsilon(\cdot)$ such that $Adv^{\mathsf{SM\text{-}TCR}}_{\boldsymbol{Th},p}(\mathcal{A}) \leq \epsilon(n)$ for large enough $n \in \mathbb{N}$.

Definition 6. *(SM-DSPR. [9]) Let $\boldsymbol{Th}, \mathsf{DIST}, Q$ be as defined above. Denote a predicate $SP_{P,T}(M) = 1$ iff there exists another $M' \neq M$ such that $\boldsymbol{Th}(P,T,M) = \boldsymbol{Th}(P,T,M')$. For adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathcal{A}_1$ is allowed to give p queries to an oracle $\boldsymbol{Th}(P,\cdot,\cdot)$ where P is uniformly chosen from $\mathcal{P}$. Then, $\mathcal{A}_2$ takes as input (Q,P) and the state of $\mathcal{A}_1$, and finally outputs (j,b). Define*

$$succ \triangleq \Pr_{P \leftarrow \mathcal{P}}[\mathsf{SP}_{P,T_j}(M_j) = b \wedge \mathsf{DIST}(\{T_i\}_{i=1}^{p})],$$

$$triv \triangleq \Pr_{P \leftarrow \mathcal{P}}[\mathsf{SP}_{P,T_j}(M_j) = 1 \wedge \mathsf{DIST}(\{T_i\}_{i=1}^{p})]$$

and

$$Adv_{\boldsymbol{Th},p,q}^{\mathsf{SM\text{-}DSPR}}(\mathcal{A}) \triangleq \max\{0, succ - triv\},$$

where q denotes the maximum of queries to $\boldsymbol{Th}$.

We say that $\boldsymbol{Th}$ is single-function multi-target decisional second-preimage-resistant for distinct tweaks (SM-DSPR) if for any polynomial-time adversary $\mathcal{A}$, there exists a negligible function $\epsilon(\cdot)$ such that $Adv_{\boldsymbol{Th},p}^{\mathsf{SM\text{-}DSPR}}(\mathcal{A}) \leq \epsilon(n)$ for large enough $n \in \mathbb{N}$.

Then, we give a definition of preimage resistance for tweakable hash functions. It is a tweakable version of Open-PRE [8] and has been implicitly used in the security proof of SPHINCS+ [9].

Definition 7. *(SM-OpenPRE.) Let $\boldsymbol{Th}$ be a tweakable hash function defined above. Let $p \leq |\mathcal{T}|$. For an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, $\mathcal{A}_1$ is allowed to give p queries to an oracle $\mathcal{O}$ initialized by a random $P \in \mathcal{P}$. Taking as input $T_i \in \boldsymbol{T}$, $\mathcal{O}$ randomly choose $M_i \leftarrow \boldsymbol{M}$ and output $Y_i = \boldsymbol{Th}(P, T_i, M_i)$. Define $Q = \{(T_i, Y_i)\}_{i=1}^{p}$ be the input/output pairs of $\mathcal{O}$ and $\mathsf{DIST}(\{T_i\}_{i=1}^{p})$ as above. Then, $\mathcal{A}_2$ takes as input (Q,P) and the state of $\mathcal{A}_1$, and is given the access of an oracle $\mathsf{Open}(i) = M_i$. Let L be the list of the queries to $\mathsf{Open}(\cdot)$. Finally, $\mathcal{A}_2$ outputs (j, M). Define*

$$Adv_{\boldsymbol{Th},p,q}^{\mathsf{SM\text{-}OpenPRE}}(\mathcal{A}) = \Pr_{P \leftarrow \mathcal{P}}[\boldsymbol{Th}(P, T_j, M) = Y_j \wedge \mathsf{DIST}(\{T_i\}_{i=1}^{p}) = 1 \wedge j \notin L],$$

where q denotes the maximum number of queries to $\boldsymbol{Th}$.

We say that $\boldsymbol{Th}$ is SM-OpenPRE if for any polynomial-time adversary $\mathcal{A}$, there exists a negligible function ϵ such that $Adv_{\boldsymbol{Th},p}^{\mathsf{SM\text{-}OpenPRE}}(\mathcal{A}) \leq \epsilon(n)$ for large enough $n \in \mathbb{N}$.

The following lemma shows that the insecurity of SM-OpenPRE can be reduced to SM-TCR or SM-DSPR.

Lemma 1. *([8, 9]) Let $\boldsymbol{Th}$ be a tweakable hash function family and $\mathcal{A}$ be an adversary against SM-OpenPRE of $\boldsymbol{Th}$. Then, there exist polynomial-time reductions $\mathcal{B}^{\mathcal{A}}$ and $\mathcal{C}^{\mathcal{A}}$ such that*

$$Adv_{\boldsymbol{Th},p,q}^{\mathsf{SM\text{-}OpenPRE}}(\mathcal{A}) \leq 3 \cdot Adv_{\boldsymbol{Th},p,q}^{\mathsf{SM\text{-}TCR}}(\mathcal{B}) + Adv_{\boldsymbol{Th},p,q}^{\mathsf{SM\text{-}DSPR}}(\mathcal{C}).$$

In particular, if the tweakable hash function **Th** is modeled as a random oracle (which means that $\mathbf{Th}(P, T, M) = H(P||T||M)$ where H is a quantum random oracle), the security has been evaluated in [9, 26].

Lemma 2. *([8,9,26]) Let* $\boldsymbol{Th}$ *be constructed above mapping to* $\{0,1\}^n$. *For any quantum adversary* $\mathcal{A}$ *making at most* q *queries to the quantum random oracle, it holds that*

$$Adv^{\textsf{SM-TCR}}_{\boldsymbol{Th},p,q}(\mathcal{A}) \le 8(2q+1)^2/2^n + 16q^2/|\mathcal{P}|. \tag{5}$$

$$Adv^{\textsf{SM-DSPR}}_{\boldsymbol{Th},p,q}(\mathcal{A}) \le 32pq^2/2^n + 16q^2/|\mathcal{P}|. \tag{6}$$

Remark 5. It is conjectured that equation (6) is loose. In [9, 26], the author conjectures that $Adv^{\textsf{SM-DSPR}}_{\mathbf{Th},p,q}(\mathcal{A})$ should be bounded by $O(q^2/2^n)$ (without the factor p).

In practice, the hash function **Th** is sampled by choosing the public parameter $P \in \mathcal{P} = \{0,1\}^n$. (Thus, $16q^2/|\mathcal{P}|$ in Equation (5) and (6) is then bounded by $O(q^2/2^n)$.) Since P is public, we omit P and simply write $\mathbf{Th}(T, M)$ in the following.

An SM-TCR tweakable hash function can be used to construct a tree structure [9]. For convenience, we only show the syntax and the security notion and omit the detailed construction.

Proposition 1. *Let* τ *be an ingeter,* $(y_1, ..., y_t)$ *be a* t*-tuple of* n*-bit strings where* $t = 2^\tau$ *and* $\boldsymbol{Th} : \{0,1\}^l \times \{0,1\}^* \to \{0,1\}^n$ *be a tweakable hash function where* $l \ge \tau + 1$. *There exists a tuple of probabilistic algorithm* $\textsf{Tree} = (\textsf{TreeGen}, \textsf{TreeProv}, \textsf{TreeVer})$ *where*

- $\textsf{TreeGen}(\boldsymbol{Th}, t, (y_1, ..., y_t))$ *generates a hash tree with leaf* $(y_1, ..., y_t)$ *and function* $\boldsymbol{Th}$*, where the tweak of* $\boldsymbol{Th}$ *is the address of the node. Then, it outputs the root of the hash tree.*
- $\textsf{TreeProv}(\boldsymbol{Th}, t, (y_1, ..., y_t), i)$ *runs* $\textsf{TreeGen}(\boldsymbol{Th}, t, (y_1, ..., y_t))$*, obtains the hash tree, and outputs the authentication path* π_i *for leaf* y_i.
- $\textsf{TreeVer}(\boldsymbol{Th}, t, i, y_i, \pi_i)$ *uses* y_i *and its authentication path* π_i *to generate the root of the hash tree, and then outputs the root.*

Lemma 3. *Let* $\boldsymbol{Th} : \{0,1\}^l \times \{0,1\}^* \to \{0,1\}^n$ *and* $\textsf{Tree}$ *be depicted above. For any* $(y_1, ..., y_t)$ *and adversary* $\mathcal{A}$*, there exists a reduction* $\mathcal{M}^{\mathcal{A}}$ *such that*

$$\Pr_{\boldsymbol{Th},\mathcal{A}}\left[\begin{array}{c|c} y^* \ne y_{i^*} & y_0 \leftarrow \textsf{TreeGen}(\boldsymbol{Th}, t, (y_1, ..., y_t)) \\ y_0 = \textsf{TreeVer}(\boldsymbol{Th}, t, i^*, y^*, \pi^*) & (i^*, y^*, \pi^*) \leftarrow \mathcal{A}(\boldsymbol{Th}, t, (y_1, ..., y_t)) \end{array}\right] \le Adv^{\textsf{SM-TCR}}_{\boldsymbol{Th},t,q_H}(\mathcal{M}^{\mathcal{A}}).$$

A.2 Signature Schemes

Definition 8. *A signature scheme* $\Gamma = (\textsf{KeyGen}, \textsf{Sign}, \textsf{Ver})$ *consists of three polynomial-time algorithms along with an associated message space* $\mathcal{M} = \{\mathcal{M}_n\}$ *such that:*

- *The key generation algorithm* KeyGen *takes as input the security parameter* 1^n*. It outputs a pair of keys* (pk, sk)*, where* pk *and* sk *are called the public key and the secret key respectively.*
- *For security parameter* n*, the signing algorithm* Sign *takes as input a secret key* sk *and a message* $m \in \mathcal{M}_n$*. It outputs a signature* σ*.*
- *For security parameter* n*, the verification algorithm* Ver *takes as input a public key* pk*, a message* $m \in \mathcal{M}_n$ *and a signature* σ*. It outputs a bit* b*. If* $b = 1$*, we say* σ *is a valid signature of* m*.*

For any $(pk, sk) \leftarrow \mathsf{KeyGen}(1^n)$, $m \in \mathcal{M}_n$ and $\sigma \leftarrow \mathsf{Sign}(sk, m)$, it holds that $\mathsf{Ver}(pk, m, \sigma) = 1$.

A.3 Pseudorandom Functions

Definition 9. *Let* $l(\cdot)$ *be a polynomial. A pseudorandom function* PRF *is a function mapping from* $\{0,1\}^\kappa \times \{0,1\}^{l(n)} \to \{0,1\}^n$*.* $\{0,1\}^\kappa$ *is called the key space. Let* $\mathcal{F}_n$ *be the function family mapping from* $\{0,1\}^{l(n)}$ *to* $\{0,1\}^n$*. We say* PRF *is an indistinguishable pseudorandom function (Ind-PRF) if for all polynomial-time adversary* $\mathcal{A}$*, there exists a negligible function* ***negl*** *such that*

$$Adv^{\mathsf{Ind\text{-}PRF}}_{\mathsf{PRF},q}(\mathcal{A}) \triangleq \left| \Pr_{k \leftarrow \{0,1\}^\kappa}[\mathcal{A}^{\mathsf{PRF}(k,\cdot)}(1^n) = 1] - \Pr_{f \leftarrow \mathcal{F}_n}[\mathcal{A}^{f(\cdot)}(1^n) = 1] \right| < \boldsymbol{negl}(n),$$

where q *denotes the maximum number of queries to* PRF *or* f*.*

A pseudorandom function is usually instantiated by a hash function $\mathsf{H}(k, \cdot)$. The following lemma proves the pseudorandomness if H is modeled as a quantum random oracle.

Lemma 4. *([33], Lemma 2.2) Let* $H : \{0,1\}^\kappa \times \{0,1\}^{l(n)} \to \{0,1\}^n$ *and* $F : \{0,1\}^{l(n)} \to \{0,1\}^n$ *be two quantum random oracles. Let* $\mathcal{A}$ *be an unbounded-time adversary that can query* H q *times. It holds that*

$$\left| \Pr_{k \leftarrow \{0,1\}^\kappa}[\mathcal{A}^{H(k,\cdot),H}(1^n) = 1] - \Pr[\mathcal{A}^{F,H}(1^n) = 1] \right| \leq 2q \cdot 2^{-\kappa/2}.$$

In this paper, when we model the hash functions in a hash-based signature schemes as quantum random oracles, we also model PRF as a quantum random oracle $\mathsf{H}(k, \cdot)$. When $\kappa = n$, it can provide at least $n/2$-bit generic quantum security in the sense of pseudorandomness.

B Toolbox

In this section, we show some useful lemmas related to quantum computations and quantum random oracles.

Lemma 5. *(Quantum query complexity lemma. [34]) Let H be a random function mapping $\mathcal{X}$ to $\mathcal{Y}$, r be a positive integer, and R be a relation over $\mathcal{Y}^r$. For any quantum adversary $\mathcal{A}$ which can query H at most q times, it holds that*

$$\begin{aligned}&\Pr_H\left[x_1,...,x_r \textit{ are distinct} \wedge (H(x_1),...,H(x_r)) \in R|(x_1,...,x_r) \leftarrow \mathcal{A}^{|H\rangle}\right]\\ \leq &(2q+1)^{2r} \Pr\left[\exists \pi \in \mathsf{Perm}([r]) \textit{ s.t. } (y_{\pi(1)},...,y_{\pi(r)}) \in R|(y_1,...,y_r) \leftarrow \mathcal{Y}^r\right],\end{aligned}$$

where $\mathsf{Perm}([r])$ denotes the set of permutations of $[r]$.

Corollary 4. *Let $\mathcal{F}$ be an efficient function esemble modeled by a quantum random oracle. For any quantum adversary $\mathcal{A}$, it holds that*

$$Adv^{\mathsf{OW}}_{\mathcal{F},q}(\mathcal{A}) \leq (2q+1)^2 \cdot 2^{-n}.$$

The next lemma show that if we perform a partial measurement in the process of a quantum algorithm and the measurement obtains one of t outcomes, then the final output of the algorithm will remain unchanged with probability at least $1/t$.

Lemma 6. *[12] Let A be a probabilistic quantum algorithm. Let A' be another algorithm described as follows: A' runs as A but pauses it at an arbitrary stage of execution, performs a partial measurement that obtains one of t outcomes, and then resumes A. For any x, it holds that*

$$\Pr_{A'}[x \leftarrow A'] \geq \Pr_{A}[x \leftarrow A]/t.$$

Definition 10. *[28] Let $\mathcal{F} \triangleq \{f : \{0,1\}^m \rightarrow \{0,1\}\}$ be the collection of all boolean functions with input space $\{0,1\}^m$. Let $\lambda \in [0,1]$ be a constant. Define a family of distributions D_λ on $\mathcal{F}$ such that for $f \leftarrow D_\lambda$, $\forall x \in \{0,1\}^m$, $f(x) = 1$ with probability λ and $f(x) = 0$ with probability $1-\lambda$.*

Lemma 7. *[28] Let $\mathcal{A}$ be an algorithm $\mathcal{A}$ issuing q quantum queries to $f(\cdot)$. Define*

$$Adv^{\mathsf{Avg\text{-}Search}_\lambda}_{\mathcal{F},q}(\mathcal{A}) \triangleq \Pr_{f \leftarrow D_\lambda}[f(x) = 1|x \leftarrow \mathcal{A}^f].$$

Then, for any adversary $\mathcal{A}$, it holds that

$$Adv^{\mathsf{Avg\text{-}Search}_\lambda}_{\mathcal{F},q}(\mathcal{A}) \leq 8\lambda(q+1)^2.$$

Next we introduce the adaptive reprogramming lemma [22]. It shows that if we reprogram the random oracle in some partially random records, then an adversary is hard to tell the difference. For an oracle $\mathsf{O} : X \rightarrow Y$, $x \in X$ and $y \in Y$, denote $\mathsf{O}^{x \rightarrow y}$ as an oracle that is the same as O except that it maps x to y. The adaptive reprogramming lemma is as follows.

Lemma 8. *(Adaptive reprogramming lemma. [22]) Let X_1, X_2 and Y be finite sets, $\mathsf{O}_0 : X_1 \times X_2 \to Y$ be a random oracle and Repro_b be the adaptive reprogramming game depicted in Figure 7. Let $\mathcal{A}$ be an algorithm issuing q quantum queries to O_b and R classical queries to $\mathsf{Reprogram}$. Then, the probability of distinguishing b is at most*

$$|\Pr[\mathsf{Repro}_1^{\mathcal{A}} = 1]| - |\Pr[\mathsf{Repro}_0^{\mathcal{A}} = 1]| \leq \frac{3R}{2}\sqrt{\frac{q}{|X_1|}}.$$

```
Game Repro_b                      Reprogram(x_2)

O_1 := O_0                        (x_1, y) ← X_1 × Y
b' ← A^{|O_b⟩, Reprogram}         O_1 := O_1^{x_1||x_2 → y}
return b'                         return x_1
```

Fig. 7. Adaptive reprogramming games for $b \in \{0, 1\}$.

Lemma 9. *(Grover's Algorithm. [23]) Let $F : X \to \{0, 1\}$ be a predicate mapping an element of set X to a bit and $F^{-1}(1) = \{x : F(x) = 1\}$ is non-empty. Let $t = |F^{-1}(1)| > 0$. There is a quantum algorithm that randomly returns $x^* \in F^{-1}(1)$ with at most $O(\sqrt{\frac{|X|}{t}})$ quantum queries to F.*

We call the above quantum algorithm Grover's algorithm.

C Subset Resilience and Its variants

Subset resilience is first proposed in HORS [32], a few-time HBS scheme. In this section, we give definitions of several variants and analyze their generic security.

C.1 Definitions

Definition 11. *(Subset Cover. [32, 35]) Let $H = (h_1, ...h_k)$ be a hash function mapping $\{0, 1\}^m$ to $[t]^k$ and $r \geq 0$ be an integer. We say that $(r + 1)$-tuple $(x, x_1, ..., x_r) \in \{0, 1\}^{m(r+1)}$ is an (r, k)-subset cover of H if it holds that $\{h_i(x)\}_{i\in[k]} \subset \{h_i(x_j)\}_{i\in[k],j\in[r]}$ and $x \notin \{x_j\}_{j\in[r]}$.*

Definition 12. *(Restricted Subset Cover. [35]) Let $H = (h_1, ...h_k)$ be a hash function mapping $\{0, 1\}^m$ to $[t]^k$ and $r \geq 0$ be an integer. We say that $(r + 1)$-tuple $(x, x_1, ..., x_r) \in \{0, 1\}^{m(r+1)}$ is an (r, k)-restricted subset cover of H if it holds that $x \notin \{x_j\}_{j\in[r]}$, and for $\forall i \in [k]$, $h_i(x) \in \{h_i(x_j)\}_{j\in[r]}$.*

Next, we show a weaker statement called *weak subset cover* that is useful in the following sections.

Definition 13. *(Weak Subset Cover.) Let $H = (h_1, ...h_k)$ be a hash function mapping $\{0,1\}^m$ to $[t]^k$ and $r \geq 0$ be an integer. We say an $(r+1)$-tuple $(x_1, ..., x_{r+1}) \in \{0,1\}^{m(r+1)}$ is an (r,k)-weak subset cover of H if for $\forall i \in [k]$, it holds that $|\{h_i(x_j)\}_{j\in[r+1]}| \leq r$ and x_j's are distinct. In other words, there exists a collision in $x_1, ..., x_{r+1}$ w.r.t. each h_i.*

Corollary 5. *If $(x, x_1, ..., x_r)$ is an (r,k)-restricted subset cover of H, it is also an (r,k)-weak subset cover of H.*

If it is hard for any polynomial-time adversary to find a (restricted/weak-)subset cover, then we say that the hash function family is (restricted/weak-)subset-resilient.

Definition 14. *Let $\mathcal{H} = \{H : \{0,1\}^m \to [t]^k\}$ be a hash function family. Let $\mathcal{A}$ be an adversary that takes as input $H = (h_1, ..., h_k) \leftarrow \mathcal{H}$, runs at most q hash queries and finally outputs $(x, x_1, ..., x_r) \in \{0,1\}^{m(r+1)}$. Define*

$$Adv_{\mathcal{H},q}^{(r,k)\text{-}SR}(\mathcal{A}) \triangleq \Pr_{\mathcal{H},\mathcal{A}}\left[\{h_i(x)\}_{i\in[k]} \subset \{h_i(x_j)\}_{i\in[k],j\in[r]} \wedge x \notin \{x_j\}_{j\in[r]}\right]$$

and

$$Adv_{\mathcal{H},q}^{(r,k)\text{-}rSR}(\mathcal{A}) \triangleq \Pr_{\mathcal{H},\mathcal{A}}\left[\forall i \in [k], h_i(x) \in \{h_i(x_j)\}_{j\in[r]} \wedge x \notin \{x_j\}_{j\in[r]}\right].$$

Let $\mathcal{A}$ be an adversary that takes as input $H = (h_1, ..., h_k) \leftarrow \mathcal{H}$, runs at most q hash queries and finally outputs $(x_1, ..., x_{r+1}) \in \{0,1\}^{m(r+1)}$. Define

$$Adv_{\mathcal{H},q}^{(r,k)\text{-}wSR}(\mathcal{A}) \triangleq \Pr_{\mathcal{H},\mathcal{A}}\left[\forall i \in [k], \left|\{h_i(x_j)\}_{j\in[r+1]}\right| \leq r \wedge x_1, x_2, ..., x_{r+1} \text{ are distinct }\right].$$

We say that $\mathcal{H}$ is a secure (r,k)(-restricted/weak) subset resilient hash function family or (r,k)-SRH(/rSRH/wSRH) if $Adv_{\mathcal{H},q}^{(r,k)\text{-}SR}(\mathcal{A})/Adv_{\mathcal{H},q}^{(r,k)\text{-}\ rSR}(\mathcal{A})/Adv_{\mathcal{H},q}^{(r,k)\text{-}wSR}(\mathcal{A})$ is negligible for any probabilistic polynomial-time quantum adversary $\mathcal{A}$.

C.2 Generic Security with Quantum Queries

Theorem 2. *Let $\mathcal{H} = \{H : \{0,1\}^m \to [t]^k\}$ be a random function family and r be a positive integer. For any quantum probabilistic polynomial-time adversary $\mathcal{A}$, it holds that*

$$Adv_{\mathcal{H},q}^{(r,k)\text{-}wSR}(\mathcal{A}) \leq (2q+1)^{2(r+1)}\left(\frac{r^2}{t}\right)^k, \tag{7}$$

$$Adv_{\mathcal{H},q}^{(r,k)\text{-}\mathsf{rSR}}(\mathcal{A}) \le (2q+1)^{2(r+1)}(2r+2)\left(\frac{r}{t}\right)^k, \tag{8}$$

and

$$Adv_{\mathcal{H},q}^{(r,k)\text{-}\mathsf{SR}}(\mathcal{A}) \le (2q+1)^{2(r+1)}(2r+2)\left(\frac{rk}{t}\right)^k. \tag{9}$$

Proof. Since $\mathcal{H}$ is a random function family, the success probability of adversaries can be evaluated by Lemma 5. For $i \in [k]$, denote by $y^{(i)} \in [t]$ the i-th element of y.

1. (Proof of (7).)
For $H = (h_1, ..., h_k) : \{0,1\}^* \to [t]^k$, define $R_1 \subseteq [t]^{k(r+1)}$ as follows:

$$R_1 \triangleq \{(y_1, y_2, ..., y_{r+1}) : \forall i \in [k], \left|\{y_j^{(i)}\}_{j\in[r+1]}\right| \le r\}.$$

We analyze the size of R_1. From $(y_1, ..., y_{r+1}) \in R_1$, for every $i \in [k]$, (at least) two of $y_1^{(i)}, ..., y_{r+1}^{(i)}$ are equal. Fix an $i \in [k]$, we can traverse all the possible $(y_1^{(i)}, ..., y_{r+1}^{(i)})$ w.r.t. i as follows:
(a) Pick a pair of indices a_1, a_2 from $[r]$.
(b) Pick $y \in [t]$, let $y_{a_1} = y_{a_2} = y$.
(c) Traverse all possible values of $y_j^{(i)}$ for all $j \notin \{a_1, a_2\}$ and $j \in [r+1]$.
The numbers of choices in the three steps are $\binom{r+1}{2}$, t and t^{r-1} respectively. Thus, for all $i \in [k]$, the total number of possible values of $(y_1, ..., y_{r+1}) \in R_1$ is at most

$$\left(\binom{r+1}{2} \cdot t \cdot t^{r-1}\right)^k = \left(\frac{(r+1)r}{2} t^r\right)^k \le (r^2 t^r)^k.$$

In addition, it is not hard to see that relation R_1 is not ordered (which means that for any $\pi \in \mathsf{Perm}([r+1])$, the statement $(y_1, ..., y_{r+1}) \in R_1$ is equivalent to the statement $(y_{\pi(1)}, ..., y_{\pi(r+1)}) \in R_1$). Due to Lemma 5, we have

$$\text{Adv}_{\mathcal{H},q}^{(r,k)\text{-}\mathsf{wSR}}(\mathcal{A}) \le (2q+1)^{2(r+1)} \frac{(r^2 t^r)^k}{t^{(r+1)k}} = (2q+1)^{2(r+1)} \left(\frac{r^2}{t}\right)^k,$$

which is what we expected.
2. (Proof of (8).)
Note that in Lemma 5, the elements in a solution have to be distinct, but those in a restricted subset cover do not. (In a restricted subset cover, only $x \notin \{x_j\}_{j\in[r]}$ is demanded, and thus x_j can be equal to another $x_{j'}$.) We divide a restricted subset cover into several cases.
Fix $r, \mathcal{H}$ and $\mathcal{A}$. Recall that $H = (h_1, ..., h_k) \leftarrow \mathcal{H}$ and $(x, x_1, ..., x_r) \leftarrow \mathcal{A}(H)$. Let $1 \le s \le r$ be some integer. Define

$$f(s) \triangleq \Pr_{\mathcal{H},\mathcal{A}}\left[\forall i \in [k], h_i(x) \in \{h_i(x_j)\}_{j\in[r]} \wedge x \notin \{x_j\}_{j\in[r]} \wedge \left|\{x_j\}_{j\in[r]}\right| = s\right].$$

Then, we have

$$\mathrm{Adv}_{\mathcal{H},q}^{(r,k)\text{-}\mathsf{rSR}}(\mathcal{A}) = \sum_{s\in[r]} f(s),$$

and

$$f(r) = \Pr_{\mathcal{H},\mathcal{A}}\Big[\{h_i(x)\}_{i\in[k]} \subset \{h_i(x_j)\}_{i\in[k],j\in[r]} \wedge x, x_1, ..., x_r \text{ are distinct}\Big].$$

First, we give a bound for the case that $r = s$.

Lemma 10. $f(r) \leq (2q+1)^{2(r+1)}(r+1)(\frac{r}{t})^k$.

Proof. For $H = (h_1, ..., h_k) : \{0,1\}^* \to [t]^k$, define $R_2 \subseteq [t]^{k(r+1)}$ as follows:

$$R_2 \triangleq \big\{(y, y_1, ..., y_r) : \forall i \in [k], y^{(i)} \in \{y_j^{(i)}\}_{j\in[r]}\big\}.$$

Next, we analyze the size of R_2. For convenience, we call the first element of $(y, y_1, ..., y_r) \in R_2$ as y_0.
First, there are exactly t^k possible values of y_0. Then, for any fixed $y_0 = (y_0^{(1)}, ..., y_0^{(k)})$, it holds that $y_0^{(i)} \in \{y_j^{(i)}\}_{j\in[r]}$ for each $i \in [k]$. This implies that $y_j^{(i)} = y_0^{(i)}$ for some $j \in [r]$. We can traverse all the possible value of $(y_1^{(i)}, ..., y_r^{(i)})$ for each i w.r.t. y_0 by the following steps:
(a) Pick $j \in [r]$ and let $y_j^{(i)} = y_0^{(i)}$.
(b) Traversing all the possible value of $y_{j'}^{(i)}$ for all $j' \in [r]$ and $j' \neq j$.
The number of possible values of $(y_1^{(i)}, ..., y_r^{(i)})$ w.r.t. y_0 is at most $r \cdot t^{r-1}$ for each i. Thus, considering all $i \in [k]$ and traversing all possible values of y_0, the total number of $(y_0, y_1, ..., y_r)$ is at most

$$(r \cdot t^{r-1})^k \cdot t^k = (rt^r)^k.$$

Unlike R_1, relation R_2 is ordered. Define

$$R_2^* \triangleq \{(y_1, ..., y_{r+1}) : \exists \pi \in \mathsf{Perm}([r+1]) \text{ s.t. } (y_{\pi(1)}, ..., y_{\pi(r+1)}) \in R_2\}.$$

Observe that for any $\pi \in \mathsf{Perm}([r])$, the statement $(y, y_1, ..., y_r) \in R_2$ is equivalent to the statement $(y, y_{(\pi(1))}, ..., y_{\pi(r)}) \in R_2$. This implies that the order of R_2 is only determined by the first element. Thus, we can traverse all the possible values of $(y_1, ..., y_{r+1}) \in R_2^*$ by the following steps:
(a) Pick $(y, y_1, ..., y_r) \in R_2$.
(b) Pick $j \in [r+1]$, and insert y between $(j-1)$-th element and the j-th element of $(y_1, ..., y_r)$. In other words, traverse $(y, y_1, ...y_r)$, $(y_1, y, y_2, ..., y_r)$, ..., $(y_1, ..., y_r, y)$.
Thus, we have

$$|R_2^*| \leq (r+1)|R_2| \leq (r+1)(rt^r)^k.$$

Due to Lemma 5, we have

$$f(r) \leq (2q+1)^{2(r+1)} \frac{(r+1)(rt^r)^k}{t^{(r+1)k}} = (2q+1)^{2(r+1)}(r+1)\left(\frac{r}{t}\right)^k.$$

Next, we consider the case that $s < r$.
If the adversary output an (r, k)-restricted subset cover $(x, x_1, ..., x_r)$ such that $|\{x_j\}_{j\in[r]}| = s < r$. Let $\{x_j\}_{j\in[r]} = \{x'_{j'}\}_{j'\in[s]}$ after reordering. Then, it is not hard to see that $(x, x'_1, ..., x'_s)$ is an (s, k)-restricted cover and all the elements are distinct. The probability of this event is also bounded by Lemma 10. That is, for all $1 \leq s \leq r$ it holds that

$$f(s) \leq (2q+1)^{2(s+1)}(s+1)\left(\frac{s}{t}\right)^k \leq (2q+1)^{2(s+1)}(r+1)\left(\frac{r}{t}\right)^k.$$

Thus, we have

$$\mathrm{Adv}^{(r,k)\text{-}\mathsf{rSR}}_{\mathcal{H},q}(\mathcal{A}) = \sum_{s\in[r]}(2q+1)^{2(s+1)}(r+1)\left(\frac{r}{t}\right)^k \leq (2q+1)^{2(r+1)}(2r+2)\left(\frac{r}{t}\right)^k.$$

3. (Proof of (9).)
Similarly, fix $r, \mathcal{H}, \mathcal{A}$. For $s \in [r]$, we define

$$g(s) \triangleq \Pr_{\mathcal{H},\mathcal{A}}\left[x \notin \{x_1, ..., x_r\} \wedge \{h_i(x)\}_{i\in[k]} \subset \{h_i(x_j)\}_{i\in[k],j\in[r]} \wedge \left|\{x_j\}_{j\in[r]}\right| = s\right],$$

and thus

$$\mathrm{Adv}^{(r,k)\text{-}\mathsf{SR}}_{\mathcal{H},q}(\mathcal{A}) = \sum_{s\in[r]} g(s).$$

As above, we first consider the case that $s = r$.
For $H = (h_1, ..., h_k) : \{0,1\}^* \to [t]^k$, define $R_3 \subseteq [t]^{k(r+1)}$ as follows:

$$R_3 \triangleq \{(y, y_1, ..., y_r) : \{y^{(i)}\}_{i\in[k]} \subseteq \{y_j^{(i)}\}_{i\in[k],j\in[r]}\}.$$

We divide R_3 into k subsets $R_{3,1}, ..., R_{3,k}$, where, for $m \in [k]$,

$$R_{3,m} \triangleq \{((y, y_1, ..., y_r) \in R_3 : \left|\{y^{(i)}\}_{i\in[k]}\right| = m\}.$$

Observe that $R_{3,m}$'s are disjoint and that $R_3 = \bigcup_{m\in[k]} R_{3,m}$. More precisely, the statement $(y, y_1, ..., y_r) \in R_{3,m}$ implies that $\{y^{(i)}\}_{i\in[k]}$ contains exactly m elements in $[t]$, and $\{y_j^{(i)}\}_{i\in[k],j\in[r]}$ covers them. Since there are at most rk elements in set $\{y_j^{(i)}\}_{i\in[k],j\in[r]}$, there are rk "chances" to cover the m target elements. We can traverse all the elements of $R_{3,m}$ by the following steps:
(a) Pick m distinct elements $x_1, ..., x_m$ from $[t]$. Let $X = \{x_1, ..., x_m\}$.
The number of choices in this step is $\binom{t}{m}$.
(b) Pick $y^{(1)}, ..., y^{(k)}$ from X^k such that $\{y^{(i)}\}_{i\in[k]} = X$.
This step is equivalent to the experiment of putting k different balls into m different bins such that there is at least one ball in each bin. The number of choices is $\left\{{k \atop m}\right\} \cdot m!$, where $\left\{{k \atop m}\right\}$ denotes Stirling number of the second kind.

(c) Next, we require that $\{y_j^{(i)}\}_{i\in[k],j\in[r]}$ covers $X=\{y^{(i)}\}_{i\in[k]}$. Since $|X|=m$, we only need to choose m elements of $\{y_j^{(i)}\}_{i\in[k],j\in[r]}$, make them equal to a permutation of X, and do not have any demand for the remaining $(rk-m)$ elements.
The number of choices in the two substeps are $\binom{rk}{m}$ and $m!$ respectively.

(d) Finally, since the remaining $(rk-m)$ elements have no demand, traverse all the possible $y_j^{(i)}$ that have not been assigned in the above steps.
The number of choices in this step is $t^{(rk-m)}$.

To sum up, the total number of elements in $R_{3,m}$ is

$$\begin{aligned}|R_{3,m}| =& \binom{t}{m}\left\{{k \atop m}\right\}\cdot m!\cdot\binom{rk}{m}\cdot m!\cdot t^{rk-m}\\ \le& \frac{t^m}{m!}\cdot\left\{{k \atop m}\right\}\cdot m!\cdot\binom{rk}{m}\cdot m!\cdot t^{rk-m}\\ =& \left\{{k \atop m}\right\}\cdot\binom{rk}{m}\cdot m!\cdot t^{rk}\\ =& \left\{{k \atop m}\right\}\cdot(rk)_m\cdot t^{rk},\end{aligned}$$

where $(\cdot)_m$ denotes the falling factorial:

$$(x)_m = x\cdot(x-1)\cdot\ldots\cdot(x-m+1).$$

Thus, we have

$$|R_3| = \sum_{m=1}^{k}|R_{3,m}| \le \sum_{m=1}^{k}\left\{{k \atop m}\right\}\cdot(rk)_m\cdot t^{rk} = (rk)^k\cdot t^{rk},$$

where the last equality uses the fact that $\sum_{m=1}^{k}\left\{{k \atop m}\right\}(x)_m = x^k$.
Similar to R_2, we define

$$R_3^* \triangleq \{(y_1,\ldots,y_{r+1}) : \exists\pi\in\mathsf{Perm}([r+1]) \text{ s.t. } (y_{\pi(1)},\ldots,y_{\pi(r+1)})\in R_3\},$$

and then we have

$$|R_3^*| \le (r+1)|R_3| \le (r+1)(rk)^k\cdot t^{rk}.$$

Due to Lemma 5, we have

$$g(r) \le (2q+1)^{2(r+1)}\frac{(r+1)(rk)^k\cdot t^{rk}}{t^{(r+1)k}} = (2q+1)^{2(r+1)}(r+1)\left(\frac{rk}{t}\right)^k,$$

and for $s\in[r]$,

$$g(s) \le (2q+1)^{2(s+1)}(s+1)\left(\frac{sk}{t}\right)^k \le (2q+1)^{2(s+1)}(r+1)\left(\frac{rk}{t}\right)^k.$$

Thus,

$$\mathrm{Adv}_{\mathcal{H},q}^{(r,k)\text{-}\mathsf{SR}}(\mathcal{A}) = \sum_{s\in[r]}(2q+1)^{2(s+1)}(r+1)\left(\frac{rk}{t}\right)^k \le (2q+1)^{2(r+1)}(2r+2)\left(\frac{rk}{t}\right)^k.$$

C.3 Target Subset Resilience

Target subset resilience (TSR) [32] is a variant of subset resilience. In (r,k)-TSR experiment, the adversary is given a hash function $H=(h_1,...,h_k)$ and r random targets $x_1,...x_r$. Then, the adversary is required to output a single element x such that $\{h_i(x)\}_{i\in[k]} \subset \{h_i(x_j)\}_{i\in[k],j\in[r]}$. It is not hard to see that (r,k)-TSR is a weaker notion than (r,k)-SR.

In this section we propose a target version of restricted subset resilience, which is called *extended target subset resilience* (eTSR). Unlike TSR, the adversary in eTSR can adaptively control the target to some extent. In detail, the adversary is able to adaptively query a (classical) oracle Box. For a query x_j, Box randomly chooses $z_j \in \{0,1\}^n$ and returns $(z_j, H(z_j||x_j))$. After r queries, the adversary is required to output (x,z) such that for each $i\in[k]$, $h_i(z||x) \in \{h_i(z_j||x_j)\}_{j\in[r]}$ and $(x,z) \notin \{(x_j,z_j)\}_{j\in[r]}$ hold. Note that (r,k)-eTSR is a weaker notion than than (r,k)-rSR.

Definition 15. *(Extended Target Subset Resilience.) Let $\mathcal{H}=\{H=(h_1,...,h_k):\{0,1\}^{m+n}\to[t]^k\}$ be a hash function family. Let $\mathcal{A}^{\mathsf{Box}}$ be an adversary that queries Box at most r times, computes H at most q times and then outputs $(x,z)\in\{0,1\}^{m+n}$. Define*

$$Adv_{\mathcal{H},q}^{(r,k)\text{-}\mathsf{eTSR}}(\mathcal{A}) \triangleq \Pr_{\mathsf{Box},\mathcal{H},\mathcal{A}}\left[\forall i\in[k], h_i(z||x)\in\{h_i(z_j||x_j)\}_{j\in[r]} \wedge (x,z)\notin\{(x_j,z_j)\}_{j\in[r]}\right].$$

We say that hash function family $\mathcal{H}$ is an (r,k)-extended target-subset-resilient hash function family ((r,k)-eTSRH) if $Adv_{\mathcal{H},q}^{(r,k)\text{-}\mathsf{eTSR}}(\mathcal{A})$ is negligible for any probabilistic polynomial-time quantum adversary $\mathcal{A}$.

Here we model $\mathcal{H}$ as a quantum-accessible $\mathsf{H}:\{0,1\}^{m+n}\to[t]^k$ and $\mathsf{h}_1,...,\mathsf{h}_k:\{0,1\}^{m+n}\to[t]$ be the partial oracle. Then, the experiment of eTSR is depicted as **Game 0** in Figure 8.

Theorem 3. *Let $\mathcal{H}$ be modeled as a quantum-accessible random oracle H and r be a positive integer. For any quantum probabilistic polynomial-time adversary $\mathcal{A}$, it holds that*

$$Adv_{\mathsf{H},q}^{(r,k)\text{-}\mathsf{eTSR}}(\mathcal{A}) \le \frac{3r}{2}\sqrt{\frac{q+r+1}{2^n}} + 8(q+r+2)^2\left(\frac{r}{t}\right)^k.$$

Proof. We use the technique in the security proof of (multi-target) extended collision resistance in [28] and [22]. We prove this theorem by showing the following games:

- **Game 0** is the original experiment of eTSR.
- **Game 1** differs from **Game 0** in that Box is replaced by Box'. Every time $\mathcal{A}$ queries x_j to the oracle Box', Box' randomly picks z_j, randomly chooses $y_j^{(1)},...,y_j^{(k)}\in[t]$ and reprograms $\mathsf{h}_i(z_j||x_j):=y_j^{(i)}$ for each $i\in[k]$. (In other

```
Game 0
    $(x, z) \leftarrow \mathcal{A}^{|\mathsf{H}\rangle, \mathsf{Box}}()$
    if $\forall i \in [k], \mathsf{h}_i(z||x) \in \{\mathsf{h}_i(z_j||x_j)\}_{j\in[r]}$
        if $(x, z) \notin \{(x_j, z_j)\}_{j\in[r]}$ return 1
    return 0

$\mathsf{Box}(x_j)$
    $z_j \leftarrow \{0,1\}^n$
    return $(z_j, \mathsf{H}(z_j||x_j))$

Game 1
    $(x, z) \leftarrow \mathcal{A}^{|\mathsf{H}\rangle, \mathsf{Box'}}()$
    if $\forall i \in [k], \mathsf{h}_i(z||x) \in \{y_j^{(i)}\}_{j\in[r]}$
        if $(x, z) \notin \{(x_j, z_j)\}_{j\in[r]}$ return 1
    return 0

$\mathsf{Box'}(x_j)$
    $z_j \leftarrow \{0,1\}^n$
    $y_j := y_j^{(1)}||...||y_j^{(k)} \leftarrow [t]^k$
    $\mathsf{H} := \mathsf{H}^{z_j||x_j \to y_j}$
    return $(z_j, y_j)$
```

Fig. 8. Hybrid arguments in the proof of Theorem 3.

words, it reprograms $\mathsf{H}(z_j||x_j) := y_j = y_j^{(1)}||...||y_j^{(k)}$.) Then, it returns (z_j, y_j) to the adversary. See details in Figure 8.

Next, we show that the probabilities of **Game 0** and **Game 1** are negligible close by Lemma 8. If not, the adversary $\mathcal{A}$ can be used to distinguish Repro_0 and Repro_1 in Figure 7. In **Game 0**, H is simulated by O_0 and Box is simulated by $\mathsf{Reprogram}_0$ with additional classical query to O_0. In **Game 1**, $\mathsf{Box'}$ is simulated by $\mathsf{Reprogram}_1$ with additional classical query to O_1. In total, it issues $(q+r+1)$ queries to O_b (q for simulating H, r for simulating $\mathsf{Box}/\mathsf{Box'}$ and 1 for the final verification). Due to Lemma 8, we have

$$|\Pr[\textbf{Game 0}] - \Pr[\textbf{Game 1}]| \leq \frac{3r}{2}\sqrt{\frac{q+r+1}{2^n}}. \tag{10}$$

– In **Game 1**, the adversary outputs (x, z) such that for all $i \in [k]$, $\mathsf{h}_i(z||x)$ is covered by $\{y_j^{(i)}\}_{j\in[r]}$. Define $S = \{y_{a_1}^{(1)}||...||y_{a_k}^{(k)}\}_{(a_1,...,a_k)\in[r]^k}$. In other words, S contains all $y = y^{(1)}||...||y^{(k)}$ where $y^{(i)} \in \{y_j^{(i)}\}_{j\in[r]}$ for each i. Thus, the adversary is to output (x, z) such that $H(z||x) \in S$ and (x, z) is not equal to any (x_j, z_j).

  Note that $|S| \leq r^k$. Without loss of generality, we suppose $|S| = r^k$. (If $|S| < r^k$, the success probability is obviously smaller. Here our purpose is to find an upper bound of the probability.) Reorder $S = \{y'_1, ..., y'_{r^k}\}$.

  Next, we use an adversary succeeding in **Game 1** to construct a reduction breaking $\mathsf{Avg\text{-}Search}_\lambda$ in Lemma 7.

  Let $f \leftarrow D_\lambda : \{0,1\}^{m+n} \to \{0,1\}$ and $\lambda = (\frac{r}{t})^k$. Let $I : \{0,1\}^{m+n} \to [r^k]$ and $g : \{0,1\}^{m+n} \to [t]^k \backslash S$ be random functions. Construct $\tilde{\mathsf{H}} : \{0,1\}^{m+n} \to [t]^k$ as follows: for any $(z||x) \in \{0,1\}^{m+n}$, define:

$$\tilde{\mathsf{H}}(z||x) = \begin{cases} y_j, & x = x_j \wedge z = z_j, \\ y'_{I(z||x)}, & f(z||x) = 1, \\ g(x), & otherwise. \end{cases}$$

Note that the outputs of $\tilde{\mathsf{H}}$ distributes uniformly. Thus, an adversary in **Game 2** finds (x, z) such that $f(z||x) = 1$. Due to Lemma 7, we have

$$\Pr[\textbf{Game 1}] \leq 8(q + r + 1 + 1)^2 \left(\frac{r}{t}\right)^k = 8(q + r + 2)\left(\frac{r}{t}\right)^k. \qquad (11)$$

From equation (10) and (11), we complete the proof.

Interleaved target subset resilience (ITSR) [9] is another variant of target resilience and has been used in constructing SPHINCS+. It can be considered an extended version of eTSR. We introduce another hash function $h_0 : \{0,1\}^{m+n} \rightarrow \{0,1\}^h$ (where h is a constant) and modify the oracle Box to an interleaved version iBox. Given x_j as input, iBox picks random z_j and outputs $(z_j, h_0(z_j||x_j), H(z_j||x_j))$. After queries, the adversary outputs (x, z) that, for each $i \in [k]$, $(h_i(z||x), h_0(z||x)) \in \{(h_i(z_j||x_j), h_0(z_j||x_j))\}_{j\in[r]}$ and $(x, z) \notin \{(x_j, z_j)\}_{j\in[r]}$ hold. Note that if $h = 0$, ITSR becomes the same as eTSR.

Definition 16. *(Interleaved Target Subset Resilience.) Let $\mathcal{H} = \{H = (h_1, ..., h_k) : \{0,1\}^{m+n} \rightarrow [t]^k\}$ and $\mathcal{H}_0 = \{h_0 : \{0,1\}^{m+n} \rightarrow \{0,1\}^h\}$ be hash function families. Let $\mathcal{A}^{\mathsf{iBox}}$ be an adversary that queries iBox at most r times, computes (H, h_0) at most q times and then outputs $(x, z) \in \{0,1\}^{m+n}$. Define*

$$Adv^{(r,k)\text{-}\mathsf{ITSR}}_{\mathcal{H},\mathcal{H}_0,q}(\mathcal{A}) \triangleq \Pr_{\mathsf{iBox},\mathcal{H},\mathcal{H}_0,\mathcal{A}}\left[\begin{array}{c}\forall i \in [k], (h_i(z||x), h_0(z||x)) \in \{(h_i(z_j||x_j), h_0(z_j||x_j))\}_{j\in[r]} \\ (x, z) \notin \{(x_j, z_j)\}_{j\in[r]}\end{array}\right].$$

We say that hash function family pair $(\mathcal{H}, \mathcal{H}_0)$ is an (r, k)-interleaved target subset resilient hash function family ((r, k)-ITSRH) if $Adv^{(r,k)\text{-}\mathsf{ITSR}}_{\mathcal{H},\mathcal{H}_0,q}(\mathcal{A})$ is negligible for any probabilistic polynomial-time quantum adversary $\mathcal{A}$.

Remark 6. In practice (e.g., in SPHINCS+ [9]), $\mathcal{H}_0$ and $\mathcal{H}$ are instantiated by a separation of a single hash function family. That is, pick a hash function H_{msg} mapping to $\{0,1\}^{h+k\log t}$ and let $H_{\mathsf{msg}}(z||x) = (\mathrm{MD}||idx)$. H denotes the map from $(z||x)$ to MD and h_0 denotes the map to idx.

In [9], the authors give an attack on ITSR and conjecture a bound for the security. Here we give a concrete lower bound of the security in the quantum-accessible random oracle model. The idea mainly follows [9] except using Lemma 8.

Game 0

```
$(x, z) \leftarrow \mathcal{A}^{|\mathsf{h}_0\rangle, |\mathsf{H}\rangle, \mathsf{iBox}}()$
if $\forall i \in [k], (\mathsf{h}_0(z||x), \mathsf{h}_i(z||x)) \in \{(\mathsf{h}_0(z||x), \mathsf{h}_i(z_j||x_j))\}_{j\in[r]}$
    if $(x, z) \notin \{(x_j, z_j)\}_{j\in[r]}$ return 1
return 0
```

$\mathsf{Box}(x_j)$

```
$z_j \leftarrow \{0,1\}^n$
return $(z_j, \mathsf{h}_0(z_j||x_j), \mathsf{H}(z_j||x_j))$
```

Game 1

```
$(x, z) \leftarrow \mathcal{A}^{|\mathsf{h}_0\rangle, |\mathsf{H}\rangle, \mathsf{iBox'}}()$
if $\forall i \in [k], (\mathsf{h}_0(z||x), \mathsf{h}_i(z||x)) \in \{(y_j^{(0)}, y_j^{(i)})\}_{j\in[r]}$
    if $(x, z) \notin \{(x_j, z_j)\}_{j\in[r]}$ return 1
return 0
```

$\mathsf{iBox'}(x_j)$

```
$z_j \leftarrow \{0,1\}^n$
for $\forall i \in [k]$, $y_j^{(i)} \leftarrow [t]$, $y_j^{(0)} \leftarrow \{0,1\}^h$
$y_j := y_j^{(1)}||...||y_j^{(k)}$
$\mathsf{H} := \mathsf{H}^{z_j||x_j \rightarrow y_j}$, $\mathsf{h}_0 := \mathsf{h}_0^{z_j||x_j \rightarrow y_j^{(0)}}$
return $(z_j, y_j^{(0)}, y_j)$
```

Fig. 9. Hybrid arguments in the proof of Theorem 4.

Theorem 4. *Let H and h_0 be modeled as quantum-accessible random oracles H and h_0 respectively and r be a positive integer. For any quantum probabilistic polynomial-time adversary $\mathcal{A}$, it holds that*

$$Adv^{(r,k)\text{-}\mathsf{ITSR}}_{\mathsf{H},\mathsf{h}_0,q}(\mathcal{A}) \leq \frac{3r}{2}\sqrt{\frac{q+r+1}{2^n}} + 8(q+r+2)^2\Gamma,$$

where $\Gamma = \sum_\gamma \left(1-\left(1-\frac{1}{t}\right)^\gamma\right)^k \binom{r}{\gamma}\left(1-\frac{1}{2^h}\right)^{r-\gamma}\frac{1}{2^{h\gamma}}$.

Proof. We prove the statement by the following games, which is very similar to Theorem 3. We write $\mathsf{H}'(z||x) = (\mathsf{H}(z||x), \mathsf{h}_0(z||x))$.

- **Game 0** is the original experiment of ITSR as depicted in Figure 9.
- **Game 1** differs from **Game 0** expect that iBox is replaced by iBox' in Figure 9. Every time iBox' is queried with x_j, it randomly samples $y_j^{(0)}$,...,$y_j^{(k)}$ where $y_j^{(0)} \in \{0,1\}^h$ and $y_j^{(i)} \in [t]$ for other $i \in [k]$, and does reprogramming. As in Theorem 3, we have
$$|\Pr[\textbf{Game 0}] - \Pr[\textbf{Game 1}]| \leq \frac{3r}{2}\sqrt{\frac{q+r+1}{2^n}}. \tag{12}$$
- We give a bound for the probability of **Game 1**. Let $\mathbf{y} = (y_j^{(0)},...,y_j^{(k)})_{j\in[r]}$ and $\mathbf{z} = (z_j)_{j\in[r]}$ be all the choices of $y_j^{(i)}$'s and z_j's in Box' respectively. For $y \in \{0,1\}^h$, let J_y be the set of index j that $h_0(z_j||x_j) = y$. That is
$$J_y \triangleq \{j : h_0(z_j||x_j) = y\}.$$
Then, define
$$S(\mathbf{y}) \triangleq \bigcup_{y\in\{0,1\}^h : J_y \neq \emptyset} \{(y_{a_1}^{(1)}||...||y_{a_k}^{(k)}, y)\}_{(a_1,...,a_k)\in J_y^k}.$$
The adversary succeed if and only if it finds an (x,z) such that $H'(z||x) \in S(\mathbf{y})$ and x is not equal to any x_j. The success probability is taken over the choice of $\mathbf{y}$, $\mathbf{z}$, H' and the randomness of $\mathcal{A}$. That is
$$\Pr[\textbf{Game 2}] \leq \Pr_{\mathbf{y},\mathbf{z},\mathsf{H}',\mathcal{A}}\left[\mathsf{H}'(z||x) \in S(\mathbf{y})\right] = \mathbb{E}_{\mathbf{y}} \Pr_{\mathbf{z},\mathsf{H}',\mathcal{A}}\left[\mathsf{H}'(z||x) \in S(\mathbf{y})\right].$$
Again, we use the adversary in **Game 1** to construct a reduction breaking $\mathsf{Avg\text{-}Search}_\lambda$ in Lemma 7. Fix $\mathbf{y}$ (and also $S(\mathbf{y})$). Let $f \leftarrow D_\lambda : \{0,1\}^{m+n} \to \{0,1\}$ and $\lambda = |S|/t^k$. Let $I : \{0,1\}^{m+n} \to |S|$ and $g : \{0,1\}^{m+n} \to [t]^k \times \{0,1\}^h \backslash S(\mathbf{y})$ be random functions. Reorder $S(\mathbf{y}) = \{y'_1, ..., y'_{|S(\mathbf{y})|}\}$. Construct $\tilde{\mathsf{H}'} : \{0,1\}^{m+n} \to [t]^k \times \{0,1\}^h$ as follows: for any $(z||x) \in \{0,1\}^{m+n}$, define
$$\tilde{\mathsf{H}'}(z||x) = \begin{cases} y_j & (x = x_j \wedge z = z_j),\\ y'_{I(z||x)} & (f(z||x) = 1),\\ g(x) & (otherwise).\end{cases}$$

If $\mathcal{A}$ succeeds in **Game 1**, then the reduction succeeds in Avg-Search with the same probability. From Lemma 7, we have

$$\Pr_{\mathbf{z},\mathsf{H}',\mathcal{A}}\left[\mathsf{H}'(z||x) \in S(\mathbf{y})\right] \leq 8(q+r+2)^2 \frac{|S(\mathbf{y})|}{2^h t^k}.$$

From the analysis in [9], for any $y' \in [t]^k \times \{0,1\}^h$, the probability of $y' \in S$ (over the choice of $\mathbf{y}$) is at most Γ. Accordingly the expectation of $|S(\mathbf{y})|/(2^h t^k)$ over the choice of $\mathbf{y}$ is at most Γ. Thus, we have

$$\Pr[\textbf{Game 1}] \leq \mathbb{E}_{\mathbf{y}}\left[8(q+r+2)^2 \frac{|S(\mathbf{y})|}{2^h t^k}\right] \leq 8(q+r+2)^2 \Gamma. \tag{13}$$

From equation (12) and (13), we complete the proof.

Remark 7. We give an exact bound for the generic security of ITSR. Compared to the conjecture in [9], we have an additional term $\frac{3r}{2}\sqrt{\frac{q+r+1}{2^n}}$ here. This term does not have an essential role if a mild security argument based on the notion of security level is sufficient. Note that in HBS schemes, the security level is defined as the complexity of q_H, making the probability of breaking the security reach a constant. For example, in SPHINCS+-256s ($r = 2^{64}$ and $n = 256$), this additional term will cause 128-bit security, which is the same as the original security level of SPHINCS+-256s. Thus, this additional term has a small impact on the security level.

D Attacks in the BU model

In the following, we omit the parameter n and use B_ϵ to denote $B_{\epsilon,n}$. The strategy of attacking SPHINCS in the BU model is as follows.

1. Denote $f(m) = \mathsf{PRF}_{\mathsf{idx}}(sk_{\mathsf{seed}}, \mathsf{PRF}_{\mathsf{msg}}(sk_{\mathsf{seed}}, m))$ be the function that maps the message m to the index $idx \in \{0,1\}^h$. For some $idx^* \in \{0,1\}^h$, denote predicate $F(m) = 1$ iff $f(m) = idx^*$ and $m \notin B_\varepsilon$. Here, F is quantum-computable by querying the signing oracle $B_\varepsilon\mathsf{SigO}$.
2. Run Grover's algorithm on $F(m)$. This requires $O\big(\sqrt{\frac{2^h}{1-\varepsilon}}\big)$ queries to $B_\varepsilon\mathsf{SigO}$.
3. Repeat the last step r times and denote S as above. The expectation of $|S|$ is also $(1 - e^{-\frac{kr}{t}}) \cdot t$.
4. Denote predicate $F'(m) = 1$ iff $B_\varepsilon\mathsf{SigO}(m) = \perp$. It outputs a message $m^* \in B_\varepsilon$. This requires approximately $O(\varepsilon^{-1/2})$ signing queries.
5. Denote function $G(z) = 1$ iff $\{h_i(z||m^*)\}_{i\in[k]} \subset S$. Run Grover's algorithm on G. It outputs z^* such that the preimages corresponding to (z^*, m^*) have appeared in S. The expected number of quantum hash queries in this step is also $O((1 - e^{\frac{kr}{t}})^{-\frac{k}{2}})$.
   Then the adversary successfully generates a forgery $\sigma^* = (idx^*, z^*, \sigma^*_{\mathsf{HT}}, \sigma^*_{\mathsf{HORS}})$ for $m^* \in B_\varepsilon$, where σ^*_{HT} is obtained by an additional signing query and σ^*_{HORS} is obtained by S.

The total number of required (quantum) queries is respectively

$$q_s = O(r2^{\frac{h}{2}}(1-\varepsilon)^{-\frac{1}{2}}) + O(\varepsilon^{-\frac{1}{2}}), \quad q_H = O((1-e^{-\frac{kr}{t}})^{-\frac{k}{2}}).$$

Note that q_s is slightly larger than the original attack (increased by a polynomial $\sqrt{\varepsilon}$) and q_H decreases to $1/r2^{h/2}$ of the original one.

The above attack also works on SPHINCS+, but we have another one that requires less queries. The main idea is similar. First, we find a message m^* in the blind region. Then, to sign a message for m^*, we need an sFORS signature associated with some index idx^*. Note that the sFORS signature includes k elements. We directly use Grover's algorithm to search k messages (outside of the blind region) that respectively map to the k target elements. It can be done by quantum signing queries. Finally, the sFORS signature of m^* is covered by the k signatures, and a forgery is generated. The attack is as follows.

1. Find m^* such that $B_\varepsilon\mathsf{SigO}(m^*) = \perp$. This requires $O(\varepsilon^{-\frac{1}{2}})$ quantum hash queries to $B_\varepsilon\mathsf{SigO}$.
2. Pick $z^* \in \{0,1\}^n$ and let $idx^* = h_0(z^*||m^*)$. Let function $z(m)$ be the map from m to the corresponding z. For $i \in [k]$, denote predicate $F_i(m) = 1$ iff (1) $z(m) \neq \perp$, (2) $h_0(z(m)||m) = idx^*$, and (3) $h_i(z(m)||m) = h_i(z^*||m^*)$. F_i is quantum-computable by querying $B_\varepsilon\mathsf{SigO}$ and a quantum query to h_0, h_i. Run Grover's algorithm on F_i. The expected number of F_i computations is $O(\sqrt{(1-\varepsilon)^{-1} \cdot 2^h \cdot t})$.
3. After that, the secret values in sFORS signature on m^* is covered the k signatures. A forgery is then generated.

In total, the number of required (quantum) queries is respectively

$$q_s = O(k2^{\frac{h+\log t}{2}}(1-\varepsilon)^{-\frac{1}{2}}) + O(\varepsilon^{-\frac{1}{2}}), \quad q_H = O(k2^{\frac{h+\log t}{2}}(1-\varepsilon)^{-\frac{1}{2}}).$$

With the parameters in SPHINCS+-256s, q_s and q_H are both approximately 2^{43}, which is lower than our attack in the PO model.

The concrete complexity of our attacks is summarized in Table 1.

E Security Analysis of Few-time HBS Schemes

E.1 Proof of Corollary 1

In the security analysis, we use insecurity functions to show the maximum probability of breaking the security notions. For $* \in \{\mathsf{SR}, \mathsf{rSR}, \mathsf{wSR}, \mathsf{eTSR}, \mathsf{OW}\}$ and hash function family $\mathcal{H}$, $\mathrm{InSec}^{*}_{\mathcal{H},q_H}(\xi)$ denotes the maximum number of $Adv^{*}_{\mathcal{H},q_H}(\mathcal{A})$ for all ξ-time adversary $\mathcal{A}$. In addition, for $* \in \{\mathsf{SM\text{-}TCR}, \mathsf{SM\text{-}DSPR}\}$ and tweakable hash function $\mathbf{Th}$, $\mathrm{InSec}^{*}_{\mathbf{Th},q_1,q_2}(\xi)$ denotes the maximum of $Adv^{*}_{\mathcal{H},q_1,q_2}(\mathcal{A})$ for all ξ-time adversary $\mathcal{A}$.

First we show the EUF-CMA security of HORS and sFORS. The reduction is straightforward and has been given in previous work [32, 35].

Lemma 11. *For any ξ-time adversary $\mathcal{A}$, it holds that*

$$\begin{aligned} Adv^{\textsf{EUF-CMA}}_{\textsf{HORS},r,q_H}(\mathcal{A}) &\le InSec^{(r,k)\text{-}\textsf{SR}}_{\mathcal{H},q_H}(\xi) + kt \cdot InSec^{\textsf{OW}}_{\mathcal{F},q_H}(\xi), \\ Adv^{\textsf{EUF-CMA}}_{\textsf{sFORS},r,q_H}(\mathcal{A}) &\le InSec^{\textsf{SM-TCR}}_{\textbf{Th},rkt,q_H}(\xi) + InSec^{(r,k)\text{-}\textsf{rSR}}_{\mathcal{H},q_H}(\xi) \\ &\quad + 3InSec^{\textsf{SM-TCR}}_{\textbf{F},kt,q_H}(\xi) + InSec^{\textsf{SM-DSPR}}_{\textbf{F},kt,q_H}(\xi), \end{aligned}$$

Next, we focus on EUF-qCMA security of sFORS.

Theorem 5. *For any ξ-time adversary $\mathcal{A}$, it holds that*

$$\begin{aligned} Adv^{\textsf{EUF-qCMA}}_{\textsf{sFORS},r,q_H}(\mathcal{A}) \le InSec^{\textsf{SM-TCR}}_{\textbf{Th},rkt,q_H}(\xi) + InSec^{(r,k)\text{-}\textsf{wSR}}_{\mathcal{H},q_H}(\xi) \\ + kt^r\big(3InSec^{\textsf{SM-TCR}}_{\textbf{F},t,q_H}(\xi) + InSec^{\textsf{SM-DSPR}}_{\textbf{F},t,q_H}(\xi)\big). \end{aligned}$$

Proof. Our proof follows the idea of proving the EUF-qCMA security of Lamport's scheme in [12]. The outline is as follows. Assume $\mathcal{H}$ is weak subset-resilient and the adversary eventually outputs forgeries for (distinct) $m_1, ..., m_{r+1}$. There must exist some $i^* \in [k]$ such that $h_{i^*}(m_1), ..., h_{i^*}(m_{r+1})$ are distinct. As a hybird argument, we measure the values of h_{i^*} in the signing queries [9]. After the measurement, the signing oracle will only return *one* of the preimages at position i^* in each signing query, but the adversary is required to output one more of them, which can be used to break OpenPRE. From Lemma 6, each partial measurement only causes security loss of t, a polynomial number. Since the partial measurements are performed r times, a constant number, the overall security loss is still polynomial.

Let $\mathcal{A}$ be a quantum adversary, we reduce $\text{Adv}^{\textsf{EUF-qCMA}}_{\textsf{sFORS},r,q_H}(\mathcal{A})$ by the following hybrid arguments.

- **Game 0** is the original EUF-qCMA experiment of sFORS for $\mathcal{A}$.
- **Game 1** differs from **Game 0** as follows. In **Game 1**, the challenger stores all the $y_{i,j}$ in key generation algorithm. **Game 1** returns 0 if $\mathcal{A}$ outputs a $(m, \sigma) = (m, (x_1, ..., x_k, \pi_1, ..., \pi_k))$ such that $f(x_i) \ne y_{i,h_i(m)}$ for some $i \in [k]$.
  The probabilities of **Game 0** and **Game 1** differ only if the adversary generates a different hash tree of which the root collides with the real one. By Lemma 3, this implies a reduction to SM-TCR of **Th**. That is, there exists a reduction $\mathcal{M}_1$ such that
  $$|\Pr[\textbf{Game 0}] - \Pr[\textbf{Game 1}]| \le Adv^{\textsf{SM-TCR}}_{\textbf{Th},rkt,q_H}(\mathcal{M}_1). \tag{14}$$
- **Game 2** differs from **Game 1** as follows. Given forgeries $(m_j, \sigma_j)_{j\in[r+1]}$, **Game 2** outputs 0 if for $\forall i \in [k]$, $|\{h_i(m_j)\}_{j\in[r+1]}| \le r$ holds. If the output of the game differs, it implies that the adversary outputs a weak subset cover

[9] We cannot learn from the final forgeries about which i^* should be targeted, since the measurements should be performed on-line. Instead, we guess it from $[k]$ initially.

of $H = (h_1, ..., h_k)$, and thus succeeds in attacking the weak subset resilience of $\mathcal{H}$. There exists a reduction $\mathcal{M}_2$ such that

$$|\Pr[\textbf{Game 1}] - \Pr[\textbf{Game 2}]| \leq \mathrm{Adv}^{(r,k)\text{-wSR}}_{\mathcal{H},q_H}(\mathcal{M}_2). \tag{15}$$

If $\mathcal{A}$ succeeds in **Game 2**, there exists $i^* \in [k]$ such that $|\{h_{i^*}(m_j)\}_{j\in[r+1]}| = r+1$ holds, which means that $h_{i^*}(m_j)$'s are distinct for $j \in [r+1]$. In other words, $\mathcal{A}$ outputs $(r+1)$ preimages of $\{y_{i^*,j}\}_{j\in[t]}$ after only r queries to the signing oracle.

– **Game 3** differs from **Game 2** as follows. Before running $\mathcal{A}$, **Game 3** randomly guesses $i' \in [k]$. **Game 3** outputs 1 if and only if **Game 2** outputs 1 and $i' = i^*$. We have

$$\Pr[\textbf{Game 2}] \leq k \cdot \Pr[\textbf{Game 3}]. \tag{16}$$

Recall that in each signing query, the signing oracle can be operated as follows:

1. Yield $(k \log t + |\sigma|)$ qbits with initial state $\bigotimes_k |0\rangle_j \otimes |0\rangle_\sigma$, where the first k registers (with $\log t$ qubit for each[10]) compute and record $h_i(m)$ for $i \in [k]$ (say i-th j-register), and the last register computes the signature (say σ-register).
2. Let $\mathbf{U}_{h_i}$ be the unitary operation of h_i. Perform $\mathbf{U}_{h_i}$ from m-register to i-th j-register.
3. Perform the remaining operations of the signing algorithm from j-registers to σ-register, computing the signatures on σ-register.
4. Perform XOR operation from σ-register to t-register.

– **Game 4** differs from **Game 3** as follows. Every time $\mathcal{A}$ queries to the signing oracle, perform a partial measurement on the i^*-th j-register after the second step[11]. Since each measurement has at most t outcomes and there are at most r measurements, due to Lemma 6 we have

$$\Pr[\textbf{Game 3}] \leq t^r \cdot \Pr[\textbf{Game 4}]. \tag{17}$$

– In **Game 4**, i^*-th j-register is measured and collapsed to a pure state $j \in [t]$ after the second steps. Thus, the σ-register in position $s_{i^*,j}$ is also in a pure state. The adversary can only obtain information about r elements of $\{s_{i^*,j}\}_{j\in[t]}$, but it is required to output $(r+1)$ of them in the forgeries. The forgeries must contain an x_{i^*,u^*} where $u^* \in [t]$ is never the outcome of the partial measurements on h_{i^*} (and thus s_{i^*,u^*} is never revealed), but x_{i^*,u^*} is a preimage of y_{i^*,u^*}.

Given a successful adversary in **Game 4**, we construct a reduction $\mathcal{M}$ to attack SM-OpenPRE of $\mathbf{F}$. First, $\mathcal{M}$ queries all $(i,j) \in [k] \times [t]$ to oracle $\mathcal{O}$ and obtains $y_{i,j} = \mathbf{F}((i,j), s_{i,j})$ for $(i,j) \in [k] \times [t]$ where $s_{i,j}$'s are random

[10] We always assume that t is a power of 2.

[11] Note that this operation can affect the states of all registers, such as m, t and i-th j-register for $i \neq i^*$.

strings picked by $\mathcal{O}$. In the second stage, $\mathcal{M}$ obtains parameter P of $\mathbf{F}$ and uses P and $y_{i,j}$'s to generate a public key of sFORS. Then, it queries the oracle Open with all $(i,j) \in [k] \times [t]$ except $i = i^*$. In response, it obtains all $s_{i,j}$'s expect $i = i^*$ and sends the public key to $\mathcal{A}$. When $\mathcal{A}$ sends a quantum signing query, $\mathcal{M}$ performs a partial measurement after computing h_{i^*} and suppose the outcome is u. It sends (i^*, u) to Open and obtains $s_{i^*,u}$ in response. It then simulates the signing oracle in **Game 4** and sends the corresponding signature to $\mathcal{A}$. Finally, when $\mathcal{A}$ successfully returns $(r+1)$ forgeries, there must be an x_{i^*,u^*} such that (i^*, u^*) is never sent to Open and is a preimage of y_{i^*,u^*}. Sending x_{i^*,u^*} to the challenger completes the SM-OpenPRE attack on $\mathbf{F}$. We thus have

$$\Pr[\textbf{Game 4}] \leq \mathrm{Adv}^{\textsf{SM-OpenPRE}}_{\mathbf{F},t,q_H}(\mathcal{M}).$$

By Lemma 1, there exist $\mathcal{M}_3$ and $\mathcal{M}_4$ such that

$$\Pr[\textbf{Game 4}] \leq 3\mathrm{Adv}^{\textsf{SM-TCR}}_{\mathbf{F},t,q_H}(\mathcal{M}_3) + \mathrm{Adv}^{\textsf{SM-DSPR}}_{\mathbf{F},t,q_H}(\mathcal{M}_4). \tag{18}$$

From equation (14), (15), (16), (17) and (18), we complete the proof.

By introducing Theorem 5, Corollary 4, Lemma 2 and the conjectured bound of SM-DSPR in Remark 5, we complete the proof of Corollary 1.

E.2 Proof of Corollary 2

We first show the reduction to security notions for hash functions as follows.

Theorem 6. *For any ξ-time adversary $\mathcal{A}$, it holds that*

$$\begin{aligned} Adv^{\textsf{EUF-CMA}}_{\textsf{rFORS},r,q_H}(\mathcal{A}) \leq\ & InSec^{\textsf{SM-TCR}}_{\mathbf{Th},rkt,q_H}(\xi) + InSec^{(r,k)\textsf{-eTSR}}_{\mathcal{H},q_H}(\xi) + InSec^{\textsf{Ind-PRF},r}_{\textsf{PRF}}(\xi) \\ & +3InSec^{\textsf{SM-TCR}}_{\mathbf{F},kt,q_H}(\xi) + InSec^{\textsf{SM-DSPR}}_{\mathbf{F},kt,q_H}(\xi), \end{aligned}$$

$$\begin{aligned} Adv^{\textsf{EUF-qCMA}}_{\textsf{rFORS},r,q_H}(\mathcal{A}) \leq\ & InSec^{\textsf{SM-TCR}}_{\mathbf{Th},rkt,q_H}(\xi) + InSec^{(r,k)\textsf{-wSR}}_{\mathcal{H},q_H}(\xi) \\ & + kt^r\big(3InSec^{\textsf{SM-TCR}}_{\mathbf{F},t,q_H}(\xi) + InSec^{\textsf{SM-DSPR}}_{\mathbf{F},t,q_H}(\xi)\big). \end{aligned}$$

Proof. 1. (Proof of EUF-CMA.)
The reduction of rFORS is similar to sFORS. The main difference is that rFORS is not reduced to rSR. Instead, it is reduced to eTSR and the security of PRF. Since the signing algorithm is deterministic, we suppose the adversary does not require repeated messages in the experiment. Let $(m^*, (z^*, \sigma^*))$ be a forgery that $\mathcal{A}$ outputs.
The hybrid argument is as follows.
– **Game 0** is the original EUF-CMA experiment of rFORS.
– **Game 1** diffes from **Game 0** in that it returns 0 if σ^* contains a different hash tree with the real one. As shown in the proof of Theorem 5, there exists a reduction $\mathcal{M}_1$ such that

$$|\Pr[\textbf{Game 0}] - \Pr[\textbf{Game 1}]| \leq Adv^{\textsf{SM-TCR}}_{\mathbf{Th},rkt,q_H}(\mathcal{M}_1) \tag{19}$$

- **Game 2** differs from **Game 1** in that it returns 0 if σ^* contains a preimage which has not been revealed in the signing queries. Also as shown in the proof of Theorem 5, a difference of success probability with **Game 1** implies a breaker of SM-PRE of $\mathbf{F}$. Then, there exist reduction $\mathcal{M}_2$ and $\mathcal{M}_3$ such that

$$|\Pr[\textbf{Game 1}] - \Pr[\textbf{Game 2}]| \leq 3\mathrm{Adv}^{\mathsf{SM\text{-}TCR}}_{\mathbf{F},t,q_H}(\mathcal{M}_2) + \mathrm{Adv}^{\mathsf{SM\text{-}DSPR}}_{\mathbf{F},t,q_H}(\mathcal{M}_3). \tag{20}$$

- **Game 3** differs from **Game 2** in that the signing oracle does not calculate pseudorandom functions. Instead, it uses a truly random function (which can be instantiated by querying a random oracle). If the probability differs, there exists a distingusher $\mathcal{M}_4$ breaking the security of PRF:

$$|\Pr[\textbf{Game 3}] - \Pr[\textbf{Game 2}]| \leq Adv^{\mathsf{Ind\text{-}PRF}}_{\mathsf{PRF},r}(\mathcal{M}_4). \tag{21}$$

- In **Game 3**, the adversary succeeds if $h_i(z^*||m^*) \in \{h_i(z_j||m_j)\}_{j\in[r]}$ holds for $\forall i \in [k]$. We construct a reduction $\mathcal{M}_5$ that breaks eTSR of $\mathcal{H}$. Given challenge $\mathcal{H}$, it generates the key pair of rFORS and give the public key to $\mathcal{A}$. When the adversary queries m_j to the signing oracle, $\mathcal{M}_5$ queries m_j to the oracle Box and obtains $(z_j, H(z_j))$ in response. Then, $\mathcal{M}_5$ generates the corresponding signature and gives it to $\mathcal{A}$. Finally, $\mathcal{M}_5$ returns (m^*, z^*) to the challenger. If $\mathcal{A}$ succeeds, then $\mathcal{M}_5$ succeeds with the same probability. We thus have

$$\Pr[\textbf{Game 3}] \leq Adv^{(r,k)\text{-}\mathsf{eTSR}}_{\mathcal{H},q_H}(\mathcal{M}_5). \tag{22}$$

From equation (19), (20), (21) and (22), we complete the proof.

2. (Proof of EUF-qCMA.)
We claim that rFORS is at least as secure as sFORS (if we suppose that the input of $\mathcal{H}$ is of arbitrary length). That is, if there exists an adversary $\mathcal{A}$ breaks EUF-qCMA security of rFORS, then we construct $\mathcal{M}$ breaking EUF-qCMA security of sFORS.
Given the public key pk of sFORS as the challenge, $\mathcal{M}$ randomly picks $k \in \{0,1\}^n$ and then sends pk to $\mathcal{A}$. Every time $\mathcal{A}$ queries $\sum_{m,t}|m,t\rangle$, $\mathcal{M}$ computes $\sum_{m,t}|m, \mathsf{PRF}_k(m), t\rangle$ and queries $\sum_{m,t}\big|\big(\mathsf{PRF}_k(m)||m\big), t\big\rangle$ to the signing oracle. Then, $\mathcal{M}$ sends to $\mathcal{A}$ what it receives from the signing oracle. Finally, when $\mathcal{A}$ returns $\{(m_j^*, (z_j^*, \sigma_j^*))\}_{j\in[r+1]}$, $\mathcal{M}$ returns $\{(z_j^*||m_j^*, \sigma_j^*)\}_{j\in[r+1]}$. It is a set of valid forgeries of sFORS if $\mathcal{A}$ succeeds. We thus have

$$\mathrm{Adv}^{\mathsf{EUF\text{-}qCMA}}_{\mathsf{rFORS},r}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{EUF\text{-}qCMA}}_{\mathsf{sFORS},r}(\mathcal{M}).$$

By Theorem 5, we complete the proof.

By incorporating Theorem 3 to Theorem 6, we obtain the generic security of rFORS in the random oracle model in Corollary 2.

E.3 Concrete Security

In this subsection, we give an example of the concrete security of few-time signatures by implementing the parameters used in SPHINCS-256 [7] ($t = 2^{16}$, $k = 32$, $n = 256$). The concrete security of few-time signature schemes is summarized in Figure 3. Here the security levels denote the logarithm of q_H such that the probabilities depicted in Corollary 1 reach a constant. It implies lower bounds of the generic security of the schemes.

In addition, note that an attack on restricted-subset resilience immediately implies a chosen message attack on the corresponding sFORS. In [35], a quantum generic attack is shown on (r, k)-restricted subset resilience with $O(kt^{\frac{1}{2}(1-\frac{1}{2^{k+1}-1})})$ quantum queries to H. It implies an upper bound of the generic security of sFORS (and HORS) against CMA (and qCMA, respectively). The comparison is also depicted in Table 3.

r	CMA-HORS	CMA-sFORS	qCMA-sFORS	Attacks [23, 35]
1	87.0	116.5	108.5	128.0
2	52.3	79.0	73.6	114.8
3	36.6	56.6	50.3	87.2
4	27.8	43.8	37.4	67.0
5	22.1	35.4	29.2	60.2
6	18.2	29.6	23.7	52.6
7	15.3	25.3	19.7	44.9

Table 3. Concrete security of r-time HBS schemes derived from Corollary 1 with concrete parameters in SPHINCS-256. The three columns in the left hand show lower bounds of the logarithm of hash queries needed in breaking the generic security, and the rightmost column shows upper bounds.

F Proof of Theorem 1

We only show the proof for EUF-qCMA security here. The proof for EUF-CMA security is very similar and thus omitted.

As mentioned in Section 5.2, all the signatures contained in a response from the signing oracle share a common index and thus a common rFORS key pair. In each superposition, idx and σ_{HT} are identical. The only "quantum part' of a response is $(z, \sigma_{\mathsf{FORS}})$, the signature of rFORS. It implies that the EUF-qCMA security of SPHINCS-FORS is reduced to the EUF-qCMA security of rFORS and the classical security of HT.

The statement can be proven by the following hybrid arguments.

- **Game 0** is the original EUF-qCMA experiment of SPHINCS-FORS.

– **Game 1** differs from **Game 0** in that, in the signing oracle, s_{idx} is calculated by $\mathsf{TRF}(idx)$ where $\mathsf{TRF} : \{0,1\}^h \to \{0,1\}^\lambda$ is a truly random function. If the success probability differs, it implies a reduction distinguishing $\mathsf{PRF}_{\mathsf{seed}}$ and TRF. Note that there are at most 2^h calls to PRF. We have

$$|\Pr[\textbf{Game 1}] - \Pr[\textbf{Game 0}]| \leq \mathrm{InSec}^{\mathsf{Ind\text{-}PRF}}_{\mathsf{PRF}_{\mathsf{seed}},2^h}(\xi).$$

– **Game 2** differs from **Game 1** as follows. After $\mathcal{A}$ outputs (q_s+1) message-signature pairs, check whether there exists a forgery $(m^*, \Sigma^*) = (m^*, (idx^*, z, \sigma^*_{\mathsf{FORS}}, \sigma^*_{\mathsf{HT}}))$ such that $pk^*_{\mathsf{FORS}} \neq pk_{\mathsf{FORS}}$, where $pk^*_{\mathsf{FORS}} \leftarrow \mathsf{rFORS.pkFromSig}(m^*, (z, \sigma^*_{\mathsf{FORS}}))$, $s_{idx^*} = \mathsf{TRF}(idx^*)$ and $pk_{\mathsf{FORS}} \leftarrow \mathsf{rFORS.KeyGen}(1^n; s_{idx^*})$. If so, it returns 0.

Game 2 differs from **Game 1** only if the adversary generates a HT signature for a "fake" pk^*_{FORS} which is not consistent to the real one. It implies a reduction attacking the EUF-sNACMA security of HT. At the beginning, the reduction generates the rFORS public keys w.r.t. all the indices in $\{0,1\}^h$ and sends them with the corresponding indices to the challenger. Then, it obtains the HT signatures and the public key pk_{HT} from the challenger. When signing a message m from the adversary, it picks a random $idx \in \{0,1\}^h$, generates the corresponding rFORS signature $(z, \sigma_{\mathsf{FORS}})$. Then, it replies with $(idx, z, \sigma_{\mathsf{FORS}})$ and the corresponding σ_{HT} from the challenger. Finally, the adversary outputs a pk^*_{FORS} that is different from the real one. It implies a forgery of HT with state idx^*. We have

$$\Pr[\textbf{Game 1}] \leq \Pr[\textbf{Game 2}] + Adv^{\mathsf{EUF\text{-}sNACMA}}_{\mathsf{HT},2^h,q_H}(\mathcal{A}).$$

In **Game 2**, the adversary wins only if it generates (q_s+1) rFORS forgeries (of multiple instances) after q_s signing queries. Note that in each signing query, the signing oracle picks one $idx \in \{0,1\}^h$ and signs the message by the rFORS key pair associated with idx. Due to the pigeonhole principle, there must exist a special $idx^* \in \{0,1\}^h$ that has been used r times in signing queries and is used at least $(r+1)$ times in the forgeries for some $r \geq 0$.

– **Game 3** differs from **Game 2** in that it guesses $idx' \in \{0,1\}^h$ at the beginning of the experiment and outputs 0 if $idx' \neq idx^*$. We have

$$\Pr[\textbf{Game 2}] \leq 2^h \cdot \Pr[\textbf{Game 3}].$$

– In **Game 3**, $\mathcal{A}$ wins if it generates $(r+1)$ forgeries for the rFORS key pair associated with idx' conditioned that it is chosen r times in q_s signing queries. It breaks the EUF-qCMA security of rFORS with r signing queries. In addition, let $\mathbf{E}_r$ be the event that a leaf is chosen r times. The probability of $\mathbf{E}_r$ is $\binom{q_s}{r}(1-2^{-h})^{q_s-r}(2^{-h})^r < 2^{r(\log q_s - h) - \log r!}$. Thus, we have

$$\begin{aligned}\Pr[\textbf{Game 3}] &= \sum_{r=0}^{q_s} \Pr[\textbf{Game 3}|\mathbf{E}_r] \cdot \Pr[\mathbf{E}_r] \\ &\leq \sum_{r=0}^{q_s} \mathrm{InSec}^{\mathsf{EUF\text{-}qCMA}}_{FORS,r,q_H}(\xi) \cdot \min\{2^{r(\log q_s - h) - \log r!}, 1\}.\end{aligned}$$

This completes the proof.

G An Attack on SPHINCS-FORS

In this section, we show a chosen message attack on SPHINCS-FORS focusing on breaking eTSR. It is similar to the attack on SPHINCS+ focusing on ITSR, but there are some slight differences. This implies the difference in security levels between SPHINCS-FORS and SPHINCS+ in the EUF-CMA model and that the EUF-CMA security bound of SPHINCS-FORS is tight.

First, the adversary queries arbitrary q_s messages to the signing oracle and obtains the corresponding sigantures. Let S_j be the set of labels of preimages contained in the signature of m_j, that is, $S_j = \{(idx_j, i, h_i(z_j||m_j))\}_{i\in[k]}$. After that, the adversary uses Grover's algorithm to find (z^*, m^*) such that $\exists idx^* \in \{0,1\}^h : \{(idx^*, i, h_i(z^*||m^*))\}_{i\in[k]} \subset \bigcup_{j=1}^{q_s} S_j$ and that m^* has not been queried. Then, $(m^*, (idx^*, z^*, \sigma^*_{\mathsf{HT}}, \sigma^*_{\mathsf{FORS}}))$ is a forgery where σ^*_{HT} and σ^*_{FORS} can be calculated by the signatures received from the signing oracle.

Next, we evaluate the complexity of hash queries needed in the above attack. For $idx \in \{0,1\}^h$, let Y_{idx} be the set of $y = y_1||...||y_k$ such that $\{(idx, i, y_i)\}_{i\in[k]} \subset S = \bigcup_{j=1}^{q_s} S_j$. Let $\mathbf{E}_{idx,\gamma}$ be the event that idx is picked γ times. We have $\Pr[\mathbf{E}_{idx,\gamma}] = \binom{q_s}{\gamma}(1-\frac{1}{2^h})^{q_s-\gamma}\frac{1}{2^{h\gamma}}$ and

$$\mathbb{E}[|Y_{idx}|] = \left(1-\left(1-\frac{1}{t}\right)^{\gamma}\right)^{k}\binom{q_s}{\gamma}\left(1-\frac{1}{2^h}\right)^{q_s-\gamma}\frac{t^k}{2^{h\gamma}}.$$

Thus,

$$\mathbb{E}\Big[\sum_{idx}|Y_{idx}|\Big] = \left(1-\left(1-\frac{1}{t}\right)^{\gamma}\right)^{k}\binom{q_s}{\gamma}\left(1-\frac{1}{2^h}\right)^{q_s-\gamma}\frac{t^k}{2^{\gamma}}.$$

Let $\Gamma = \big(1-\big(1-\frac{1}{t}\big)^{\gamma}\big)^{k}\binom{q_s}{\gamma}\big(1-\frac{1}{2^h}\big)^{q_s-\gamma}\frac{1}{2^{h\gamma}}$. We have $\mathbb{E}\big[\sum_{idx}|Y_{idx}|\big] = t^k 2^h \Gamma$. In the average case, the probability of the above Grover's search is $\mathbb{E}\big[\sum_{idx}|Y_{idx}|\big]/t^k = 2^h\Gamma$. The number of quantum hash queries is approximately $O(2^{-\frac{h}{2}}\Gamma^{-\frac{1}{2}})$.

Note that in SPHINCS+, the quantum hash query complexity needed in breaking ITSR is $O(\Gamma^{-\frac{1}{2}})$. Thus, with the same parameters, the quantum bit security of SPHINCS-FORS is $h/2$ lower than SPHINCS+.