

Lower Bounds on Random-Oracle-Model Signature Length

Alon Chayet*
alonchayet@mail.tau.ac.il
Tel Aviv University

Mathias Marty
mathias.marty@epfl.ch
EPFL

Iftach Haitner†
iftachh@gmail.com
Stellar Development Foundation
& Tel Aviv University

Eylon Yogev‡
eylon.yogev@biu.ac.il
Bar-Ilan University

September 30, 2026

Abstract

Hash-based signatures offer a conservative foundation for post-quantum authentication, but their large signatures impose substantial communication and storage costs. For security parameter λ , discrete-logarithm-based signatures have length $O(\lambda)$, while known hash-based constructions have quadratic signature length, up to logarithmic factors, even for one-time signing. Is this gap inherent?

We prove what are, to the best of our knowledge, the first lower bounds on the length of signature schemes in the (pure) random oracle model. Our bounds apply to schemes with non-adaptive verifiers and a natural security property called *salted soundness*: a scheme with such security is unforgeable even against an adversary that is granted a limited ability to resample and restore oracle answers. This class essentially captures all known hash-based constructions (possibly after low-cost modifications). We show that the combined public-key and signature length of a *one-time* signature scheme must be $\Omega(\lambda^2/\log \lambda)$. For *many-time* signatures, we prove the stronger conclusion that the signature length alone must satisfy the same lower bound, irrespective of the public-key length. More precisely, if the relevant length is $o(\lambda^2/\log \lambda)$, then an adversary making $2^{o(\lambda)}$ random-oracle queries breaks salted soundness with inverse-polynomial probability.

Our proof uses the high-entropy hitting lemma of Haitner, Nukrai, and Yogev (Crypto 22) to transform a short signature scheme into a scheme whose verifier makes only a few queries. We then apply the Barak and Mahmoody-Ghidary (FOCS 07) attack on one-time signatures with a low-query verifier. Balancing these two steps yields the near-quadratic lower bounds.

*The Blavatnik School of Computer Science, Tel-Aviv University.

†Stellar Development Foundation and The Blavatnik School of Computer Science, Tel Aviv University.

‡Department of Computer Science, Bar-Ilan University.

Contents

1	Introduction	1
1.1	Our Results	2
1.2	Related Work	4
2	Our Technique	5
2.1	From Short to Low-Query Verifiers	6
2.1.1	Many-Time Signatures	7
2.2	Lower Bound on Schemes with Low-Query Verifiers	8
2.3	Deriving the length bound	8
3	Preliminaries	9
3.1	Notation	9
3.2	Distribution and Entropy	9
3.3	Hitting Lemma	11
3.4	Signature Schemes in the Random Oracle Model	12
3.4.1	Discussing the Definitions	15
3.4.2	The Barak and Mahmoody-Ghidary [BMG07] Lower Bound	15
4	Short Signatures to Low-Query Verifier Signatures	16
4.1	The Low-Query Verifier Scheme	16
4.2	Security	17
4.2.1	Completeness	18
4.2.2	Soundness	20
4.3	An Improved Reduction for Many-Time Signatures	22
4.3.1	Security	24
4.3.2	Completeness	24
4.3.3	Soundness	26
5	Lower Bounds on Signature Length in the ROM	27
5.1	One-Time Signatures	27
5.2	Many-Time Signatures	29
5.2.1	Tighter Bound Assuming Improved Barak and Mahmoody-Ghidary [BMG07]	31
6	Reductions Between ROM Signature Schemes	32
6.1	From a Deterministic-Query Verifier to a Deterministic Verifier	32
6.2	Achieving Perfect Completeness	33
6.2.1	Completeness	35
6.2.2	One-Time Soundness	35
6.2.3	Many-Time Soundness	38

1 Introduction

The *length* of a digital signature is a fundamental measure of its efficiency. Signatures are communicated with every authenticated message and may be retained for as long as that message remains relevant. Their length therefore affects bandwidth, storage, latency, and the scale at which an authenticated system can operate. These costs are qualitatively different from local computation: hash evaluations are transient and typically fast, whereas signature bits must cross network boundaries and may impose a lasting storage cost.¹ This concern is especially acute in blockchain networks, where signatures are broadcast to many participants and recorded in a replicated ledger, potentially for the lifetime of the system. Every additional bit may therefore be transmitted and stored many times across the network. Accordingly, a substantial line of research has sought ever more compact signatures, both by designing new primitives and by sharpening the tradeoffs within existing ones.

The post-quantum challenge. Shor [Sho94]’s algorithm breaks the factoring and discrete-logarithm assumptions underlying essentially all short signatures in widespread use. Although cryptographically relevant quantum computers do not yet exist, recent work has substantially reduced the estimated resources needed to attack elliptic-curve cryptography on both superconducting and neutral-atom architectures [Bab+26; Cai+26]. Since such estimates can improve much faster than global systems can migrate, *post-quantum signatures*, i.e., ones that are believed to be secure against quantum attackers, must be developed and deployed well before these attacks become feasible. Candidate post-quantum signatures rely either on hash functions alone or on structured assumptions involving lattices, codes, multivariate polynomials, or isogenies. Although the latter can yield efficient signatures, their foundations continue to evolve: recent work substantially lowered the estimated security of HAWK, then a candidate in NIST’s additional signature standardization process [SW26; Bos+25]; more broadly, SIKE was broken and new quasi-polynomial attacks were found against Classic McEliece [CD23; Gho+26]. These developments illustrate the difficulty of combining high efficiency with durable confidence in a structured post-quantum assumption, and motivate the more conservative hash-based approach.

Hash-based signatures. Hash-based signatures occupy the conservative end of the post-quantum signature spectrum. Their security relies on generic properties of hash functions, without adding a number-theoretic or lattice assumption. There is no known analogue of Shor’s structural attack against the underlying hash-function properties. Thus, when minimizing cryptographic assumptions is the primary goal, hash-based signatures offer perhaps the clearest security case. This case has led to standardization of both stateful and stateless hash-based signatures [Nat24; Coo+20], and has made them prominent candidates for long-lived decentralized systems. Ethereum’s post-quantum roadmap identifies the hash-based leanXMSS construction as a prospective replacement for validator signatures [Eth26], and its key-derivation standard contains Lamport [Lam79]’s one-time signature as an emergency post-quantum bridge [Bee19]. Bitcoin has not activated a post-quantum signature scheme, but proposals and experiments in its ecosystem likewise explore constructions built from hash-based one-time signatures [Nic26; BHD24].

¹An important exception arises when signature verification is embedded in a proof system: the verifier’s computation then becomes part of the proving circuit, so its running time directly affects circuit size and prover cost.

The qualitative gap. In the pre-quantum world, signatures based on the *hardness of discrete logarithms* demonstrate that very short authentication is possible: for security parameter λ , their public keys and signatures consist of only $O(\lambda)$ bits (which is optimal up to constant factors). Post-quantum signatures paint a different picture. Specifically, hash-based signatures have signature length roughly quadratic in λ , even for one-time security [Rab78; Lam79; Mer87]. This quadratic blowup is already visible in Lamport’s scheme: a signature reveals linearly many hash preimages, each of size $O(\lambda)$. A substantial line of work has sought shorter signatures while retaining minimal assumptions, with recent constructions such as SPHINCS+C continuing to sharpen the underlying tradeoffs [Kud+23]. Despite these advances, the qualitative gap remains: $O(\lambda)$ -bit signatures in the pre-quantum setting versus roughly λ^2 bits (up to lower-order terms) for one-time signatures and roughly λ^3 bits for many-time signatures, when relying only on hash functions. This leads to the question that motivates our work:

Do hash-based signatures inherently require (near) quadratic communication?

1.1 Our Results

We give evidence that this quadratic gap in hash-based signature length is inherent, proving what are, to the best of our knowledge, the first lower bounds on the communication of signature schemes in the (pure) random oracle model (ROM): the scheme and the adversary have bounded oracle access to a random function, and the adversary is computationally unbounded. We show that secure schemes in a large and natural family of one-time signatures cannot simultaneously have a short public key and a short signature.² For many-time signatures, the lower bound applies to the signature alone, irrespective of the public-key length. The family of constructions we consider includes all constructions that have (i) a *non-adaptive verifier* and (ii) *salted soundness*.

Non-adaptive verifiers. The verifier’s random-oracle queries are determined by its public input but not by its internal coins (and can therefore all be issued at once).

Salted soundness. A signature scheme has salted soundness [CY20] if it remains unforgeable against an adversary that has limited control over the random oracle. Specifically, the adversary may *resample* the oracle answer on a given input and later restore any answer it has previously seen at that input. (It cannot, however, install an arbitrary value of its choice.) Every resampling or restoration is charged as an oracle query.

This class of signatures includes, in particular, Lamport’s scheme and, more generally, known hash-based constructions whose verification queries can be fixed in advance. It does not, however, cover constructions with long sequential hash chains, such as the *Winternitz signatures* of Merkle [Mer87]; see Section 1.2.

Theorem 1.1 (Main result, informal). *Let SIG be a ROM signature scheme with a non-adaptive verifier and salted soundness, and let λ denote its security parameter. Let $\ell_{\text{pk}} = \ell_{\text{pk}}(\lambda)$ and $\ell_{\pi} = \ell_{\pi}(\lambda)$ denote its public-key and signature lengths, respectively. Then*

$$(i) \text{ If SIG is one-time secure, then } (\ell_{\text{pk}} + \ell_{\pi}) \in \Omega\left(\frac{\lambda^2}{\log \lambda}\right).$$

²Any signature scheme can be transformed to have an $O(\lambda)$ -bit public key by publishing a hash of the original public key and including the original public key in every signature. The combined length remains $\ell_{\text{pk}} + \ell_{\pi} + O(\lambda)$, and this transformation does not strengthen our one-time lower bound.

(ii) If SIG is many-time secure, then $\ell_\pi \in \Omega\left(\frac{\lambda^2}{\log \lambda}\right)$ (independently of ℓ_{pk}).

More precisely, if the relevant length is $o(\lambda^2/\log \lambda)$, then a computationally unbounded adversary making $2^{o(\lambda)}$ salted-soundness queries forges a signature with noticeable probability.

The above requires the scheme to have query complexity polynomial in λ and to support signing messages of length λ . Both conditions are necessary for a meaningful lower bound of this form. If exponential query complexity were allowed, the Winternitz signature scheme [Mer87] with chains of length 2^λ would have signature and public-key length $O(\lambda)$. If the message space were restricted to a single bit, the Lamport one-time signature scheme [Lam79] would likewise have length $O(\lambda)$.

For one-time signatures, the bound matches, up to a $\log \lambda$ factor, the construction of Lamport [Lam79].³ For many-time signatures, our bound is off by a λ factor.⁴ Our bound is smooth: the attack uses $2^{O(\sqrt{(\ell_{\text{pk}} + \ell_\pi) \log \lambda})}$ queries, up to polynomial factors. Thus, shortening the communication continuously strengthens the attack rather than merely crossing a threshold at one special parameter choice.

Adaptive verifiers. While our results apply only to non-adaptive verifiers, they naturally yield bounds for schemes with *limited adaptivity* that can be converted to be non-adaptive while only slightly increasing the signature size. Suppose, for example, that the verifier first makes a small number of adaptive queries and then uses their answers to determine one large non-adaptive batch. The signer can append the answers to the preliminary queries to the signature. Using these claimed answers, the modified verifier determines and issues all queries at once, and then checks that the claimed answers are correct before applying the original verification rule. The resulting increase in signature length is proportional to the information needed to remove the adaptive stage. The same idea applies when only a short adaptive stage follows the main non-adaptive batch. Thus, the reduction is useful when the verifier has little adaptive depth; it does not, however, eliminate long, inherently sequential computations such as hash chains. Nevertheless, we do not know how to exploit such adaptivity to circumvent the near-quadratic bound, and no construction achieving this is known.

A conditional tight bound for many-time signatures. Our framework also yields a stronger many-time lower bound under a conjectured improvement to the query-complexity attack of Barak and Mahmoody-Ghidary [BMG07]. Specifically, if the exponent governing that attack can be reduced from linear to square-root dependence on the verifier’s query complexity, as formalized in Conjecture 5.9 with $\eta(x) = \lfloor \sqrt{x} \rfloor$, then Corollary 5.11 gives a signature-length lower bound of $\Omega(\lambda^3/\log^2 \lambda)$ under the same efficiency and security requirements as above. This conditional bound matches the length of the generic certification-tree construction with Winternitz certificates, and would therefore close the remaining gap for the class of schemes covered by our lower bound.

Theorem 1.2 (Conditional lower bound for many-time signatures). *Assume Conjecture 5.9 holds for $\eta(x) = \lfloor \sqrt{x} \rfloor$. Let SIG be a ROM signature scheme with a deterministic-query verifier, completeness at least $1/2$, query complexity (q, q_V) with $q \in \text{poly}(\lambda)$, and signature length $\ell_\pi = \ell_\pi(\lambda)$.*

³Interestingly, the adaptive Winternitz one-time signature scheme [Mer87], see Section 1.2, instantiated with chains of length λ , achieves combined public-key and signature length $O(\lambda^2/\log \lambda)$, which is our bound for non-adaptive schemes.

⁴The source of this gap is that we do not have tight bounds on the *query complexity* of many-time signatures. See the open questions in Section 1.2.

If $\ell_\pi \in o\left(\frac{\lambda^3}{\log^2 \lambda}\right)$, then SIG is not $(2^{o(\lambda)}, \Omega(1/(\lambda q_V)))$ -many-time salted sound, independently of its public-key length.

1.2 Related Work

We review prior work on hash-based signature constructions and on lower bounds for signatures and SNARGs in the ROM.

Constructions of hash-based signatures. *One-time signatures.* Hash-based one-time signatures date back to [Rab78; Lam79]; see Dods et al. [DSS05] for an early systematic treatment. Lamport’s construction has signature length $O(\lambda^2)$. Winternitz signatures trade additional, sequential hash evaluations for shorter signatures: taking polynomially long chains yields signature length $O(\lambda^2/\log \lambda)$, the best known asymptotic bound for one-time signatures in the pure ROM. WOTS⁺ improves the concrete tradeoff through a tighter security analysis [Hül13]. Other one-time and few-time constructions explore tradeoffs among signature size, public-key size, signing and verification work, and the number of safe uses of a key [BC92; Per01; RR02].

Many-time signatures. For a fixed signing capacity v , a Merkle tree authenticates v one-time verification keys [Mer87]; using Winternitz signatures gives signature length $O(\lambda^2/\log \lambda + \lambda \log v)$. The stateless many-time notion considered here instead admits the generic certification-tree approach [NY89]; a polynomial-arity tree with Winternitz certificates gives signature length $O(\lambda^3/\log^2 \lambda)$. Modern stateful schemes include XMSS and LMS/HSS [BDH11; MCF19], while practical stateless schemes include SPHINCS and SPHINCS⁺ [Ber+15; Ber+19]. SPHINCS⁺ underlies NIST’s SLH-DSA standard [Nat24], and SPHINCS+C reduces its signature size by a constant factor using proof-of-work [Kud+23]. These works improve the practical size, but the asymptotics remain the same.

Lower bounds. Prior lower bounds and optimality results for hash-based signatures concern either restricted families or efficiency measures other than general signature length. Bleichenbacher and Maurer [BM94] introduced the notion of signatures that can be implemented in a *graph-based model*. Bleichenbacher and Maurer [BM96b; BM96a] established optimality results for this family, for several efficiency measures. Biçakçı et al. [BTT03] studied a *subset-revealing* one-time signature family and derived optimal tradeoffs between signing and verification costs within that family. More recently, Zhang et al. [ZCY23] proved that constant-sum WOTS⁺ minimizes signature size within Winternitz and tree-based signature designs, whereas Khovratovich et al. [KKW25] lower-bounded the verification work of a broad class of hash-chain encodings when signature size is fixed.

Barak and Mahmoody-Ghidary [BMG07] showed that a one-time ROM signature making q oracle queries can be broken with $2^{(1+o(1))q}$ queries, essentially matching a variant of Lamport’s scheme. A variant of this attack, given in [BMG07, Theorem 6.11], has a complexity bound that accounts separately for the verifier’s query complexity and the total query complexity of the scheme. We combine this variant with our short-signature-to-low-query transformation; see Section 2.

Lower bounds on ROM SNARGs. Our proof is also related to lower bounds for succinct non-interactive arguments in the ROM. Haitner et al. [HNY22] proved an rETH-based lower bound on the proof length of salted-sound ROM-SNARGs with deterministic non-adaptive verifiers. We use

their high-entropy hitting lemma, but require a new signature-specific starting point and additional machinery for reusable signatures; see Section 2.

Open Questions

The most immediate open questions concern the two restrictions underlying our lower bounds. First, can the bounds be extended to fully adaptive verifiers, whose oracle queries may depend on answers to earlier queries? Such an extension would cover constructions with long sequential hash chains, including Winternitz signatures, for which the limited-adaptivity reduction discussed above does not suffice. Second, can the bounds be proved under standard soundness, where the adversary can only query a fixed random oracle, without the resampling and restoration operations allowed by salted soundness? Resolving both questions would establish whether the near-quadratic communication barrier holds for general signature schemes in the pure ROM.

Another central question is whether the unconditional many-time lower bound can be strengthened from $\Omega(\lambda^2/\log \lambda)$ to $\Omega(\lambda^3/\log^2 \lambda)$, matching the known construction. Our framework reduces this question to improving the query-complexity attack of Barak and Mahmoody-Ghidary [BMG07]: for a scheme making q queries in total and q_v verification queries, the attack of [BMG07, Theorem 6.11] has complexity roughly $\binom{q}{q_v} = 2^{O(q_v \log q)}$, but uses only one signing query and therefore does not exploit the additional power available in the many-time setting. As a first step, a many-time attack of complexity $2^{O(q_v)} \cdot \text{poly}(q)$ would remove the logarithmic loss and yield an $\Omega(\lambda^2)$ lower bound. The stronger conjecture in Conjecture 5.9, with $\eta(x) = \lfloor \sqrt{x} \rfloor$, would instead yield the tight $\Omega(\lambda^3/\log^2 \lambda)$ bound, as stated in Theorem 1.2. Establishing this conjecture is thus a concrete route to closing the many-time gap for the class of schemes covered by our results; see Section 5.2.1.

Paper Organization and AI Disclosure

Our proof technique is described in Section 2. Section 3 states the basic notation and notions used throughout the paper, defines ROM signature schemes along with their different properties, cites the high-entropy hitting lemma of Haitner et al. [HNY22] along with a useful corollary, and states the lower bound of Barak and Mahmoody-Ghidary [BMG07] on the verifier’s query complexity. The transformation from short signatures to signatures with few-query verifiers is described in Section 4. The one-time and many-time lower bounds are proved in Section 5. Finally, Section 6 presents derandomization, completeness amplification, and other transformations between ROM signature schemes that are used by the proofs in Section 5 and might be of independent interest.

The ideas and proofs in this paper were developed and written by the authors. AI was used only for text polishing and finding references.

2 Our Technique

Our lower bound proceeds in two steps. First, we transform a signature scheme with short signatures and salted soundness into one whose verifier makes few random-oracle queries (Section 2.1). Second, we apply the query-complexity lower bound of Barak and Mahmoody-Ghidary [BMG07, Theorem 6.11], which accounts separately for the verifier’s query complexity and the scheme’s total query complexity, to rule out the resulting scheme (Section 2.2). Together, these steps give an unconditional lower bound on signature length for the class of schemes we consider.

The short-to-low-query transformation builds on the framework of Haitner et al. [HNY22] for lower bounds on the proof length of ROM SNARGs with deterministic non-adaptive verifiers and salted soundness. Their framework uses a combinatorial *hitting lemma* to obtain a low-query verifier, then invokes the query-complexity lower bound of Chiesa and Yogev [CY20], which assumes the *randomized exponential-time hypothesis* (rETH). We use a variant of their hitting lemma and an unconditional query-complexity bound for signatures. Applying this framework to signatures requires us to remove the public-key length from the many-time bound and to handle the randomized verifier and imperfect completeness produced by the transformation. We explain these steps below.

2.1 From Short to Low-Query Verifiers

For this sketch, we assume that the original verifier is deterministic and non-adaptive; the reduction from deterministic-query verifiers to deterministic verifiers is given in Section 6.1. Our first step is to transform a signature scheme $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$ with short signatures, such a verifier, and salted soundness, into a scheme $\widehat{\text{SIG}}$ with a low-query verifier. The transformation preserves most properties of the original scheme, apart from the signer’s running time, which becomes exponential (in the security parameter), and the scheme’s completeness, which decreases to $1/\text{poly}$. Let ℓ_{all} be the overall size of the public key and signature, and let λ be the security parameter. The new scheme $\widehat{\text{SIG}}$ is defined as follows.

1. On message m , the new signer $\widehat{\text{Sign}}$ calls the original (short-signature) signer Sign on m to produce a signature π . In addition, among the queries Verify makes on input (m, π) , $\widehat{\text{Sign}}$ finds a set $\mathcal{I}$ of *important queries* of size $\ell_{\text{all}}/\lambda$ that maximizes the new verifier’s acceptance probability. It outputs $\widehat{\pi} = (\pi, \mathcal{I})$.

(This step makes the new signer inefficient in terms of running time, but does not change its query complexity.)

2. On input $(m, \widehat{\pi} = (\pi, \mathcal{I}))$, the new verifier $\widehat{\text{Verify}}$ uses the random oracle to answer the important queries indexed by $\mathcal{I}$. For each of the other queries, it samples a large, but not too large, set of candidate answers—say, $2^{\lambda/4}$ options out of the 2^λ possible outcomes. It accepts if there exists a choice of the answers it sampled by itself for the non-important queries that, together with the oracle answers for the important queries, makes the original verifier Verify accept the input (m, π) .

Clearly, $\widehat{\text{Verify}}$ makes only $\ell_{\text{all}}/\lambda$ queries to the oracle. The completeness and soundness of $\widehat{\text{SIG}}$ are argued as follows.

Completeness. We prove that if the public (i.e., verification) key and signature of the SIG scheme are short, then $\widehat{\text{SIG}}$ has $1/\text{poly}$ completeness (assuming the original scheme has constant completeness). The proof uses the hitting lemma of Haitner et al. [HNY22], via the following corollary that we prove in Section 3.3.

Lemma 2.1 (Hitting lemma, informal). *Let T be a random variable over a set $\mathcal{T}$, let $\{Y_t\}_{t \in \mathcal{T}}$ be a family of random variables, each uniformly distributed over $\{0, 1\}^\lambda$, and let W be a Boolean*

random variable that equals 1 with constant probability. Then with constant probability over $(t, \mathbf{y}) \xleftarrow{R} (T, Y_T \mid W = 1)$, there exists a set $\mathcal{I} \subseteq [n]$ of size $\log|\mathcal{T}|/\lambda$ such that

$$\Pr_{\mathcal{Z} \leftarrow (\mathcal{P}_{2^{\lambda/4}}(\{0,1\}^\lambda))^{n-|\mathcal{I}|}} [\mathcal{Z} \cap \mathcal{G} \neq \emptyset] \geq 1/(\lambda n) \quad (1)$$

where $\mathcal{G} := \text{Supp}((Y_t)_{\mathcal{J}} \mid W = 1, T = t, (Y_t)_{\mathcal{I}} = \mathbf{y}_{\mathcal{I}})$, $\mathcal{J} = [n] \setminus \mathcal{I}$, and $\mathcal{P}_s(\mathcal{S})$ denotes the family of all s -size subsets of $\mathcal{S}$.⁵

That is, apart from the small set of important queries $\mathcal{I}$, the value of $t \xleftarrow{R} T \mid W$ is unlikely to reveal too much about the sample $\mathbf{y} \xleftarrow{R} Y_t \mid W, T = t$; after fixing $(Y_t)_{\mathcal{I}}$ according to $\mathbf{y}_{\mathcal{I}}$, the event W is not “very sensitive” to the value of the other entries of Y_t , i.e., of $(Y_t)_{\mathcal{J}}$. Namely, there are many options for these entries that cause W to happen. Note that the lemma trivially holds if one takes $\mathcal{Z} \leftarrow (\mathcal{P}_{2^{\lambda}}(\{0,1\}^\lambda))^{n-|\mathcal{I}|}$, since in this case the lemma considers *all* possible values for the entries in $(Y_t)_{\mathcal{J}}$. But for the considered parameters, for each entry the lemma only considers a $2^{-\frac{3}{4}\lambda}$ fraction of the possible values, which makes the lemma non-trivial in this case. For further intuition see Section 3.3.

The mapping from Lemma 2.1 to our setting is rather straightforward: n is the verifier query complexity, T is the public key and signature of the signer on message m , induced by a random execution of the (original) key-generation and signing algorithms, and Y_t is the vector of answers to the queries of the (original) verifier **Verify** on input m , given that the public key and signature are t . Finally, W is the event that **Verify** accepts on (m, T) . Since $\widehat{\text{Sign}}$ finds the optimal set $\mathcal{I}$ of size $\log(|\mathcal{T}|)/\lambda = \ell_{\text{all}}/\lambda$, Lemma 2.1 implies that $\widehat{\text{Verify}}$ accepts with probability $\Omega(1/(\lambda n)) \geq 1/\text{poly}(\lambda)$.

Soundness. The soundness of $\widehat{\text{SIG}}$ follows from the salted soundness of **SIG**. Given an adversary $\widehat{\mathbf{A}}$ that breaks the soundness of $\widehat{\text{SIG}}$, we present an adversary **A** that breaks the soundness of **SIG** with essentially the same probability, by mildly manipulating the oracle using the salted-soundness interface. Specifically, **A** runs $\widehat{\mathbf{A}}$ to produce an unsigned message m and a fake signature $(\pi, \mathcal{I})$. It then uses the salted-soundness interface to sample $2^{\lambda/4}$ options for each query made by **Verify** on input (m, π) that is not in $\mathcal{I}$, and checks whether there is a combination of these options with the (true) answers to the queries in $\mathcal{I}$ that makes **Verify** accept. If there is, it rewinds the queries not in $\mathcal{I}$ accordingly and outputs (m, π) . By construction, **A** makes $\text{poly}(\lambda) \cdot 2^{\lambda/4}$ queries, and makes **Verify** accept (m, π) with essentially the same probability as $\widehat{\text{Verify}}$ accepts $(m, \widehat{\pi})$. This holds since the answers **A** obtains through the different rewindings are distributed exactly as the ones $\widehat{\text{Verify}}$ samples by itself.

2.1.1 Many-Time Signatures

We show that if the original scheme **SIG** is many-time secure, then it can be modified into a new scheme $\widehat{\text{SIG}}$ whose verifier query complexity is only ℓ_{π}/λ (instead of $\ell_{\text{all}}/\lambda$ in $\widehat{\text{SIG}}$). That is, we do not pay for the public-key length. The new key-generation algorithm $\widehat{\text{Gen}}$ first calls the original key-generation algorithm **Gen** to generate a pair of secret/public keys (sk, pk) , and then appends to pk the verification query/answer pairs made by invoking the verifier **Verify**, with public key pk , on polynomially many random signatures, i.e., signatures on random messages. The modified verifier

⁵The term $\lambda/4$ above is arbitrary and can be replaced with any $\gamma \leq \lambda$, resulting in $|\mathcal{I}| = \log|\mathcal{T}|/\gamma$. This flexibility yields a tradeoff between the assumed salted soundness of **SIG** and its length.

$\widetilde{\text{Verify}}$ acts like $\widetilde{\text{Verify}}$ from the previous section, but, when available, it uses the answer from the extended public key instead of calling the oracle. The signer of $\widetilde{\text{SIG}}$ acts essentially as in $\widetilde{\text{SIG}}$, but considers important-query sets of size ℓ_π/λ . Note that the above transformation increases the public-key length, but this increase has no effect on the final bound.

Since SIG is assumed to be many-time secure, it is clear that the new scheme remains secure under this modification. Proving completeness, however, is more complicated. The rationale for the above scheme is that we would like to guarantee a *query separation* between $\widetilde{\text{Gen}}$ and $\widetilde{\text{Verify}}$: $\widetilde{\text{Verify}}$ is *unlikely to make a query asked by $\widetilde{\text{Gen}}$* . If such a query separation held, then we could ignore the queries made by $\widetilde{\text{Gen}}$, and thus ignore the public key, when bounding the size of the important query set $\mathcal{I}$. The problem is that in the new scheme we are only guaranteed that $\widetilde{\text{Verify}}$ is unlikely to make a query that was asked by the embedded key-generation algorithm $\widetilde{\text{Gen}}$. The signing calls made by $\widetilde{\text{Gen}}$ to produce the random signatures make many new queries. Since these queries are unlikely to be in the modified public key, we cannot bound the probability that $\widetilde{\text{Verify}}$ asks them.⁶

Instead, we use the following analysis: consider the variant $\widetilde{\text{SIG}}'$ of $\widetilde{\text{SIG}}$ in which the key-generation algorithm appends to the public key the query/answer pairs the embedded call to $\widetilde{\text{Gen}}$ made, instead of the verification queries appended in $\widetilde{\text{SIG}}$. For such a scheme, it holds that $\widetilde{\text{Verify}}$ never makes a query that was asked by the key-generation algorithm $\widetilde{\text{Gen}}'$ (since all queries of $\widetilde{\text{Gen}}'$ appear in the public key). One can then show that the public-key size does not affect the size of the important query set $\mathcal{I}$ guaranteed by the hitting lemma, and deduce that $\widetilde{\text{SIG}}'$ has noticeable completeness. We then prove that (1) with good probability, the public key of $\widetilde{\text{Gen}}$ contains all relevant queries appended by $\widetilde{\text{Gen}}'$, i.e., those that are likely to be asked by $\widetilde{\text{Verify}}$, and (2) the additional queries appearing in the public key of $\widetilde{\text{Gen}}$, beyond those appended by $\widetilde{\text{Gen}}'$, are harmless. Hence, we deduce that $\widetilde{\text{SIG}}$ also has noticeable completeness. See Section 4.3 for details.

2.2 Lower Bound on Schemes with Low-Query Verifiers

For this part we use the lower bound of [BMG07, Theorem 6.11] on the verifier's query complexity of ROM signatures. There is, however, a subtlety to handle. The formal treatment of [BMG07] considers deterministic verifiers with perfect completeness.⁷ The reduction discussed above yields a randomized verifier with $1/\text{poly}$ completeness. Yet, with some amplification tricks (see Section 6), we manage to transform the randomized-verifier, imperfect-completeness scheme from above into one with a deterministic verifier and perfect completeness, without hurting the parameters by too much, and we can then safely apply the bound of [BMG07, Theorem 6.11] to get the desired lower bound.

2.3 Deriving the length bound

Let $\ell_{\text{all}} = \ell_{\text{pk}} + \ell_\pi$ in the one-time setting and $\ell_{\text{all}} = \ell_\pi$ in the many-time setting. To see how the above steps yield the claimed bound, replace the candidate-set size $2^{\lambda/4}$ by 2^γ . For constant original completeness, the hitting lemma then gives a verifier making $s \approx \ell_{\text{all}}/\gamma$ queries. After the

⁶It is in fact easy to come up with a counterexample for which this query separation does not hold, regardless of how many verification queries on random messages the modified key generation appends to the public key.

⁷The text does mention bounds for randomized verifiers and constant completeness, but it is less formal there.

completeness and verifier transformations, the overall query complexity is still $w = \text{poly}(\lambda)$, while the verifier makes $O(s)$ queries. The attack of [BMG07, Theorem 6.11] therefore costs at most $\text{poly}(\lambda) \cdot w^{O(s)}$ queries. Including the cost of sampling candidate answers in the salted-soundness reduction, we obtain an attack with noticeable success probability and query complexity at most

$$\text{poly}(\lambda) \left(2^\gamma + 2^{O((\ell_{\text{all}}/\gamma+1) \log \lambda)} \right).$$

Choosing $\gamma = \Theta(\max\{\log \lambda, \sqrt{\ell_{\text{all}} \log \lambda}\})$, with constants satisfying the hitting lemma's requirements, balances these costs up to polynomial factors and yields

$$2^{O(\sqrt{\ell_{\text{all}} \log \lambda} + \log \lambda)}$$

queries. In particular, if $\ell_{\text{all}} = o(\lambda^2 / \log \lambda)$, then $\gamma = o(\lambda)$ and the attack uses only $2^{o(\lambda)}$ queries. This gives the claimed $\Omega(\lambda^2 / \log \lambda)$ lower bound on the combined public-key and signature length for one-time schemes, and on the signature length alone for many-time schemes.

3 Preliminaries

In this section, we state the basic notation and notions used throughout the paper, define ROM signature schemes along with their different properties, cite the high-entropy hitting lemma of Haitner et al. [HNY22] along with a useful corollary, and state the lower bound of Barak and Mahmoody-Ghidary [BMG07] on the verifier's query complexity.

3.1 Notation

We use calligraphic letters to denote sets and distributions, uppercase for random variables, lowercase for integers and functions, and boldface for vectors. We use $\leftarrow$ for deterministic assignment, i.e., $x \leftarrow 8$. All logarithms are base 2. For $n \in \mathbb{N}$, let $[n] := \{1, \dots, n\}$. For a vector $\mathbf{x} := (x_1, \dots, x_n)$ and $\mathcal{S} \subseteq [n]$, let $(\mathbf{x})_{\mathcal{S}} := (x_j)_{j \in \mathcal{S}}$ be the restriction of $\mathbf{x}$ to the coordinates in $\mathcal{S}$ (likewise for a tuple indexed by an arbitrary finite set). We naturally extend this notation to strings as well. For a set $\mathcal{S}$ and $s \in \mathbb{N}$, let $\mathcal{P}_s(\mathcal{S})$ denote all s -size subsets of $\mathcal{S}$. For $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ and $z \in \{0, 1\}^*$, let $f_z(x) := f(z, x)$. Let poly stand for the set of all (integer) polynomials. We sometimes abuse notation and treat poly as an arbitrary polynomial. A function f is *negligible* if for every $p \in \text{poly}$ there exists $n_0 \in \mathbb{N}$ such that $f(n) < 1/p(n)$ for all $n > n_0$. We write $f(n) \leq \text{neg}(n)$ to denote that $f(n)$ is negligible.

3.2 Distribution and Entropy

For a distribution $\mathcal{X}$, let $x \stackrel{\mathcal{R}}{\leftarrow} \mathcal{X}$ denote the process of sampling x according to $\mathcal{X}$. Similarly, for a set $\mathcal{S}$, $s \stackrel{\mathcal{R}}{\leftarrow} \mathcal{S}$ stands for the process of sampling s uniformly from $\mathcal{S}$. The entropy of a random variable is defined as follows.

Definition 3.1 (Entropy). *Let X, Y be finite random variables. The Shannon entropy of X is defined by*

$$H(X) := - \sum_x \Pr[X = x] \log \Pr[X = x].$$

The conditional Shannon entropy of X given Y is defined by

$$H(X | Y) := \sum_y \Pr[Y = y] \cdot H(X | Y = y).$$

It is well known that $H(X) \leq \log|\text{Supp}(X)|$, for $\text{Supp}(X) := \{x: \Pr[X = x] > 0\}$, with equality iff X is uniform. We use the following facts.

Proposition 3.2. *Let X be a uniform random variable and let W be a boolean random variable. Then, $H(X | W = 1) \geq H(X) - \log(1/\Pr[W = 1])$.*

Proof. Let $\rho \leftarrow \Pr[W = 1]$ and $n \leftarrow H(X)$. For $x \in \text{Supp}(X)$,

$$\begin{aligned} p(x) &:= \Pr[X = x | W = 1] = \frac{\Pr[W = 1 | X = x] \cdot \Pr[X = x]}{\Pr[W = 1]} \\ &= \frac{\Pr[W = 1 | X = x]}{2^n \cdot \rho} \leq \frac{1}{2^n \rho}. \end{aligned}$$

Hence,

$$\begin{aligned} H(X | W = 1) &\geq H_\infty(X | W = 1) := \min\{\log(1/p(x)) : x \in \text{Supp}(X | W = 1)\} \\ &\geq n + \log(\rho) = H(X) - \log(1/\Pr[W = 1]). \end{aligned}$$

□

Proposition 3.3. *Let $\mathcal{T}$ be a set, let $\{Y_t\}_{t \in \mathcal{T}}$ be a family of uniform random variables with $H(Y_t) \geq n$ for all $t \in \mathcal{T}$, let T be a random variable over $\mathcal{T}$, and let W be a non-trivial boolean random variable. Then $H(Y_T | T, W = 1) \geq n - \log(1/\Pr[W = 1]) - \log(|\mathcal{T}|)$.*

Proof. Fix $t \in \mathcal{T}$ with $\Pr[T = t | W = 1] > 0$, and apply Proposition 3.2 with respect to the boolean random variable indicating the event $\{W = 1 \wedge T = t\}$ to deduce that

$$H(Y_t | W = 1, T = t) \geq n - \log(1/\Pr[W = 1, T = t]).$$

Hence, letting $p_t \leftarrow \Pr[T = t | W = 1]$,

$$\begin{aligned} H(Y_T | T, W = 1) &= \sum_{t \in \mathcal{T}} p_t \cdot H(Y_t | W = 1, T = t) \\ &\geq n - \sum_{t \in \mathcal{T}} p_t \cdot \log \frac{1}{\Pr[W = 1] \cdot p_t} \\ &= n - \log(1/\Pr[W = 1]) - H(T | W = 1) \\ &\geq n - \log(1/\Pr[W = 1]) - \log(|\mathcal{T}|). \end{aligned}$$

□

3.3 Hitting Lemma

We make use of the following “hitting lemma” by Haitner et al. [HNY22].

Lemma 3.4 (Hitting high-entropy distributions [HNY22, Theorem 4.1]). *Let $n \in \mathbb{N}$, $\gamma \leq \lambda \in \mathbb{N}$ with $\gamma \geq 4\lceil \log n \rceil + 4$, and let Y be a random variable over $(\{0,1\}^\lambda)^n$ with $H(Y) \geq \lambda n - h$. Then with probability at least $1/2$ over $\mathbf{y} \stackrel{R}{\leftarrow} Y$, there exists a set $\mathcal{J} \subseteq [n]$ of size at least $n - (8h/\gamma + 4)$ such that*

$$\Pr_{\mathcal{Z} \leftarrow (\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda))^{| \mathcal{J} |}} [\mathcal{Z} \cap \mathcal{G} \neq \emptyset] \geq \frac{1}{32\lambda n} \quad (2)$$

for $\mathcal{G} := \text{Supp}(Y_{\mathcal{J}} \mid Y_{[n] \setminus \mathcal{J}} = \mathbf{y}_{[n] \setminus \mathcal{J}})$.

Remark 3.5. We note the following observations regarding the above set $\mathcal{G}$.

$\mathcal{G}$ is large. Lemma 3.4 states that with high probability, there exists a large set $\mathcal{J}$ of indices such that when conditioning on $Y_{[n] \setminus \mathcal{J}} = \mathbf{y}_{[n] \setminus \mathcal{J}}$, the set $\mathcal{G}$, the support of the marginal distribution of the remaining indices of Y , is large: with high probability, it intersects a “not too large” product set.

Any $\mathbf{g} \in \mathcal{G}$ “hits” $\text{Supp}(Y)$. Let $\mathbf{y} \in \text{Supp}(Y)$ and $\mathbf{g} \in \mathcal{G}$, for $\mathcal{G}$ defined with respect to $\mathbf{y}$ as in the lemma, and let $\mathbf{y}'$ be such that $\mathbf{y}'_{\mathcal{J}} = \mathbf{g}$ and $\mathbf{y}'_{[n] \setminus \mathcal{J}} = \mathbf{y}_{[n] \setminus \mathcal{J}}$. By definition, $\mathbf{y}' \in \text{Supp}(Y)$. However, only for a single $\mathbf{g} \in \mathcal{G}$ is it the case that $\mathbf{y}' = \mathbf{y}$. This very redundancy is what enables $\mathcal{G}$ to be large.

We use Lemma 3.4 via the following corollary.

Lemma 3.6 (Hitting conditional distributions). *Let n, γ, λ be as in Lemma 3.4. Let T be a random variable over a set $\mathcal{T}$, let $\{Y_t\}_{t \in \mathcal{T}}$ be a family of random variables, each uniformly distributed over $(\{0,1\}^\lambda)^n$, and let W be a non-trivial boolean random variable. Then with probability at least $1/4$ over $(t, \mathbf{y}) \stackrel{R}{\leftarrow} (T, Y_T \mid W = 1)$, there exists a set $\mathcal{J} \subseteq [n]$ of size at least $n - (16h/\gamma + 4)$, for $h \leftarrow \log(1/\Pr[W = 1]) + \log|\mathcal{T}|$, such that*

$$\Pr_{\mathcal{Z} \leftarrow (\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda))^{| \mathcal{J} |}} [\mathcal{Z} \cap \mathcal{G} \neq \emptyset] \geq \frac{1}{32\lambda n} \quad (3)$$

for $\mathcal{G} := \text{Supp}((Y_t)_{\mathcal{J}} \mid W = 1, T = t, (Y_t)_{[n] \setminus \mathcal{J}} = \mathbf{y}_{[n] \setminus \mathcal{J}})$.

That is, with good probability over $t \stackrel{R}{\leftarrow} T \mid W = 1$, the random variable $Y_t \mid (W = 1, T = t)$ behaves like the high-entropy Y in Lemma 3.4.

Proof. By Proposition 3.3, $H(Y_T \mid T, W = 1) \geq \lambda n - h$. Since $H(Y_t \mid W = 1, T = t) \leq \lambda n$ for every $t \in \mathcal{T}$, Markov’s inequality yields that for $\mathcal{H} := \{t \in \mathcal{T} : H(Y_t \mid W = 1, T = t) \geq \lambda n - 2h\}$ it holds that

$$\Pr[T \in \mathcal{H} \mid W = 1] \geq 1/2.$$

By Lemma 3.4, for every $t \in \mathcal{H}$, with probability at least $1/2$ over $\mathbf{y} \stackrel{R}{\leftarrow} (Y_t \mid W = 1, T = t)$, there exists a set $\mathcal{J}$ of size at least $n - (16h/\gamma + 4)$ satisfying Equation (3). Hence, the probability that the required set $\mathcal{J}$ exists for $(t, \mathbf{y}) \stackrel{R}{\leftarrow} (T, Y_T \mid W = 1)$ is at least $\Pr[T \in \mathcal{H} \mid W = 1] \cdot 1/2 \geq 1/4$. $\square$

3.4 Signature Schemes in the Random Oracle Model

We define signature schemes in the random oracle model (ROM). Let $\mathcal{F}_\lambda$ denote the set of all functions from $\{0, 1\}^*$ to $\{0, 1\}^\lambda$.

Definition 3.7 (ROM signature schemes). *A ROM signature scheme is a triplet $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$ of oracle-aided algorithms that, given oracle access to $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$, are of the following form:*

$\text{Gen}^f(1^\lambda, 1^\ell)$: *Outputs a pair of secret and public keys $\text{sk}, \text{pk} \in \{0, 1\}^*$.*

$\text{Sign}^f(\text{sk}, m)$: *Is a deterministic algorithm that outputs a signature $\pi \in \{0, 1\}^*$.*

$\text{Verify}^f(\text{pk}, m, \pi)$: *Outputs a bit in $\{0, 1\}$.*

Remark 3.8 (Randomized signers). *Definition 3.7 requires the signer to be deterministic. This is not a restriction, however, since a randomized signer can be transformed into a deterministic one by instructing the key-generation to append $z \xleftarrow{R} \{0, 1\}^\lambda$ to the secret key, and the signer to use $f(z, m)$ as its randomness source.*

We sometimes use $\text{Sign}_{\text{sk}}(\cdot)$ for $\text{Sign}(\text{sk}, \cdot)$ and $\text{Verify}_{\text{pk}}(\cdot)$ for $\text{Verify}(\text{pk}, \cdot)$.

Definition 3.9 (Restricted verifiers). *We consider three families of restricted verifiers.*

Deterministic. *The verifier uses no internal randomness.*

Non-adaptive. *The verifier's oracle queries are determined by its input and random coins.*

Deterministic query. *The verifier's oracle queries are determined by its input only; the final decision can additionally depend on the random coins and oracle answers.*

We use the following notation to address the properties of signature schemes.

Notation 3.10. *Let SIG be a ROM signature scheme.*

- SIG has signature length ℓ_π and public-key length ℓ_{pk} , if $|\text{pk}| \leq \ell_{\text{pk}}(\lambda)$ and $|\text{Sign}^f(\text{sk}, m)| \leq \ell_\pi(\lambda)$ for any $f \in \mathcal{F}_\lambda$, $(\text{sk}, \text{pk}) \in \text{Supp}(\text{Gen}^f(1^\lambda, 1^\ell))$, and $m \in \{0, 1\}^\ell$.
- SIG has query complexity (q, q_V) , if $\text{Gen}^f(1^\lambda, 1^\ell)$, $\text{Sign}_{\text{sk}}^f(m)$ and $\text{Verify}_{\text{pk}}^f(m, \pi)$ together make at most $q(\lambda)$ oracle queries, and $\text{Verify}_{\text{pk}}^f(m, \pi)$ makes at most $q_V(\lambda)$ queries, for any $f \in \mathcal{F}_\lambda$, $(\text{sk}, \text{pk}) \in \text{Supp}(\text{Gen}^f(1^\lambda, 1^\ell))$, $m \in \{0, 1\}^\ell$, and $\pi \in \{0, 1\}^*$.

Completeness. The completeness, i.e., acceptance probability, of a ROM signature scheme is defined as follows.

Definition 3.11 (Completeness). *A ROM signature scheme $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$ has completeness α , if for any $\lambda, \ell \in \mathbb{N}$ and for any message $m \in \{0, 1\}^\ell$:*

$$\Pr_{\substack{f \xleftarrow{R} \mathcal{F}_\lambda \\ (\text{sk}, \text{pk}) \xleftarrow{R} \text{Gen}^f(1^\lambda, 1^\ell) \\ \pi \xleftarrow{R} \text{Sign}_{\text{sk}}^f(m)}} \left[\text{Verify}_{\text{pk}}^f(m, \pi) = 1 \right] \geq \alpha(\lambda).$$

Two-part completeness. It is sometimes convenient to separate the probability that the oracle and the keys yield a good signature from the probability that the verifier accepts a good signature.

Definition 3.12 ((α_s, α_v) -completeness). A ROM signature scheme $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$ has (α_s, α_v) -completeness, if for any $\lambda, \ell \in \mathbb{N}$ and every message $m \in \{0, 1\}^\ell$:

$$\Pr_{\substack{f \xleftarrow{R} \mathcal{F}_\lambda \\ (\text{sk}, \text{pk}) \xleftarrow{R} \text{Gen}^f(1^\lambda, 1^\ell)}} \left[\Pr \left[\text{Verify}_{\text{pk}}^f(m, \text{Sign}_{\text{sk}}^f(m)) = 1 \right] \geq \alpha_v \right] \geq \alpha_s,$$

where the inner probability is over the random coins of Verify .

That is, with probability at least α_s over the oracle and keys, the verifier accepts the signature of m with probability at least α_v over its coins. Clearly, (α_s, α_v) -completeness implies $(\alpha_s \cdot \alpha_v)$ -completeness, and for a deterministic verifier, it recovers Definition 3.11 with $\alpha_v = 1$. We use the following immediate facts.

Proposition 3.13. A ROM signature scheme with (α_s, α_v) -completeness has completeness $\alpha_s \cdot \alpha_v$.

Proof. Fix $\lambda, \ell \in \mathbb{N}$ and $m \in \{0, 1\}^\ell$. Let $F \xleftarrow{R} \mathcal{F}_\lambda$, let $(\text{SK}, \text{PK}) \xleftarrow{R} \text{Gen}^F(1^\lambda, 1^\ell)$, and let $\rho := \Pr_{r \xleftarrow{R} \{0, 1\}^*} [\text{Verify}_{\text{PK}}^F(m, \text{Sign}_{\text{SK}}^F(m); r) = 1]$. By Definition 3.12, $\Pr[\rho \geq \alpha_v] \geq \alpha_s$, and thus $\mathbb{E}[\rho] \geq \alpha_s \cdot \alpha_v$. Since $\mathbb{E}[\rho]$ is the acceptance probability of Definition 3.11, this is the required completeness. $\square$

Proposition 3.14. A ROM signature scheme with completeness α has $(\alpha - \alpha_v, \alpha_v)$ -completeness, for every $\alpha_v \leq \alpha$. In particular, it has $(\alpha/2, \alpha/2)$ -completeness.

Proof. Fix λ, ℓ and m , and let ρ be as in the proof of Proposition 3.13. By Definition 3.11, $\mathbb{E}[\rho] \geq \alpha$. Since $\rho \leq 1$, it holds that $\mathbb{E}[\rho] \leq \Pr[\rho \geq \alpha_v] + \alpha_v$. Hence, $\Pr[\rho \geq \alpha_v] \geq \alpha - \alpha_v$, which is the required $(\alpha - \alpha_v, \alpha_v)$ -completeness. The second part follows by taking $\alpha_v \leftarrow \alpha/2$. $\square$

Soundness. The soundness, i.e., forging probability, of a ROM signature scheme is defined as follows.

Definition 3.15. An oracle-aided algorithm A is t -query, if on input $(1^\lambda, \cdot)$ it makes at most $t(\lambda)$ queries to each of the signing oracle and the random oracle. It is admissible, if, given oracle access to Sign , it never outputs a message on which it queried Sign .

Definition 3.16 (Soundness). A ROM signature scheme $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$ has (t, β) -soundness, if for every t -query admissible algorithm A and every $\lambda, \ell \in \mathbb{N}$:

$$\Pr_{\substack{f \xleftarrow{R} \mathcal{F}_\lambda \\ (\text{sk}, \text{pk}) \xleftarrow{R} \text{Gen}^f(1^\lambda, 1^\ell) \\ (m, \pi) \xleftarrow{R} A^{f, \text{Sign}_{\text{sk}}^f}(1^\lambda, 1^\ell, \text{pk})}} \left[\text{Verify}_{\text{pk}}^f(m, \pi) = 1 \right] \leq \beta(\lambda).$$

The scheme has (t, β) -one-time soundness, if the above holds for every A that makes at most a single query to Sign .

We sometimes refer to the unrestricted notion as many-time soundness, to contrast it with the one-time one, and likewise for the salted variants defined below.

Salted soundness. We consider signature schemes that achieve *salted soundness*, a notion first introduced by Chiesa and Yogev [CY20] for ROM-SNARGs. A signature scheme has salted soundness if its soundness holds against malicious adversaries that can *weakly program* the random oracle.

Definition 3.17 (Salted random oracle SRO). *The salted random oracle is the stateful machine SRO providing the following interface:*

Parameters: $\lambda \in \mathbb{N}$ (all algorithms implicitly receive 1^λ as input).

Init().

1. Abort if already called.
2. Initialize two key-value maps $\mathbf{t}, \mathbf{y}$, with $\mathbf{t}[w] \leftarrow \emptyset$ and $\mathbf{y}[w] \leftarrow \perp$ for all w .

Sample(w).

1. Sample $y \xleftarrow{R} \{0, 1\}^\lambda$.
2. Set $\mathbf{y}[w] \leftarrow y$ and $\mathbf{t}[w] \leftarrow \mathbf{t}[w] \cup \{y\}$.
3. Return y .

Query(w).

1. If $\mathbf{y}[w] = \perp$, return **Sample(w)**.
2. Otherwise, return $\mathbf{y}[w]$.

Rewind(w, y).

1. Return $\perp$ if $y \notin \mathbf{t}[w]$.
2. Otherwise, set $\mathbf{y}[w] \leftarrow y$ and return y .

Intuitively, the methods **Sample** and **Rewind** allow one to *weakly program* the random oracle. We next define salted soundness.

Definition 3.18 (Salted soundness). *A ROM signature scheme $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$ has (t, β) -salted soundness, if for every t -query admissible algorithm A and every $\lambda, \ell \in \mathbb{N}$:*

$$\Pr \left[\text{Verify}_{\text{pk}}^{\text{SRO}_\lambda \cdot \text{Query}}(m, \pi) = 1 \right] \leq \beta(\lambda),$$

where the probability is over the following process:

- $\text{SRO}_\lambda.\text{Init}()$.
- $(\text{sk}, \text{pk}) \xleftarrow{R} \text{Gen}^{\text{SRO}_\lambda \cdot \text{Query}}(1^\lambda, 1^\ell)$.
- $(m, \pi) \xleftarrow{R} A^{\text{SRO}_\lambda \cdot \text{All}, \text{Sign}_{\text{sk}}^{\text{SRO}_\lambda \cdot \text{Query}}}(1^\lambda, 1^\ell, \text{pk})$, letting $\text{SRO}_\lambda.\text{All}$ be a shorthand for $\{\text{SRO}_\lambda.\text{Query}, \text{SRO}_\lambda.\text{Rewind}, \text{SRO}_\lambda.\text{Sample}\}$.

Each of A 's calls to $\text{SRO}_\lambda.\text{Query}$, $\text{SRO}_\lambda.\text{Sample}$ and $\text{SRO}_\lambda.\text{Rewind}$ counts as a query. The scheme has (t, β) -one-time salted soundness, if the above holds for every A that makes at most a single query to **Sign**.

That is, the adversary can resample the oracle answers and rewind its answer to one of the previously answered values. Since, however, it is charged a query for each such call, it can only perform a limited number of them.

3.4.1 Discussing the Definitions

We make the following remarks regarding the definitions above.

Remark 3.19 (Running time). *Our definitions bound only the number of oracle queries. The running time of the scheme’s algorithms and of the adversary is unbounded. As in Barak and Mahmoody-Ghidary [BMG07], our attacks are query-efficient but not time-efficient, and our lower bounds are thus information-theoretic. Leaving the running time of the scheme unbounded only strengthens a lower bound, and it allows the transformations of Sections 4 and 6 to output inefficient schemes.*

Remark 3.20 (Independence of message length). *Our definitions require the scheme’s length and query bounds to be independent of the message length ℓ . This requirement is essentially without loss of generality for schemes that support 2λ -bit messages, by the standard hash-and-sign transformation.*

Concretely, use prefix-free domain separation to define

$$H^f(m) := f(0, m) \parallel f(1, m) \in \{0, 1\}^{2\lambda}$$

and give the original scheme access to the independently domain-separated oracle $f(2, \cdot)$. To sign an arbitrary message m , compute $m' \leftarrow H^f(m)$, sign m' with the original scheme, and output (m', π) . The new verifier checks that $m' = H^f(m)$ and verifies π as a signature of m' . Appending m' preserves non-adaptivity: the verifier can determine the original verification queries from the claimed digest m' and issue them in parallel with the two queries used to check the digest.

The transformation preserves completeness, adds 2λ bits to the signature and only constantly many oracle queries, and preserves the public-key length. The soundness reduction fails only if the forged message and a previously signed message have the same digest. For an adversary making t oracle calls, this event is bounded by $O(t^2 \cdot 2^{-2\lambda})$, and by $O(t^4 \cdot 2^{-2\lambda})$ for the salted oracle. This loss is negligible.

3.4.2 The Barak and Mahmoody-Ghidary [BMG07] Lower Bound

The main result of Barak and Mahmoody-Ghidary [BMG07] is that a sound ROM signature scheme must have large *overall* query complexity. We use the following variant of it, which accounts for the *verifier’s* query complexity, and thus rules out schemes whose verifier makes few queries even when their overall query complexity is large.

Theorem 3.21 (Lower bound on verifier query complexity, adapted from [BMG07, Theorem 6.11]). *Let SIG be a ROM signature scheme with perfect completeness, a deterministic verifier, and query complexity (q, q_V) with $q_V \leq q/2$. Then for every $\delta, \mu \in (0, 1]$ there exists $t \leq c \cdot q^2 \cdot \binom{q}{q_V} / (\mu \delta^2)$, for some universal constant c , such that SIG is not $(t, 1 - (\delta + \mu))$ -one-time sound.*

Remark 3.22 (Message length in [BMG07]). *Barak and Mahmoody-Ghidary [BMG07, Theorem 6.11] require a message space of size at least $\binom{q}{q_V} / \mu$, i.e., messages of length about $q_V \log q$. This requirement does not appear in Theorem 3.21 since by our convention a signature scheme supports any message length. (See Remark 3.20.)*

4 Short Signatures to Low-Query Verifier Signatures

In this section, we show how to transform a signature scheme with a short public key and short signatures into one with a low-query verifier. Recall (see Section 2) that in order to reduce the verifier's query complexity, the signer of the new scheme attaches a small subset of *important* verifier queries to the signature; the verifier queries the random oracle only on this small set and samples the answers to the remaining verifier queries by *itself*. The low-query verifier scheme is defined in Section 4.1, and its security is analyzed in Section 4.2.

4.1 The Low-Query Verifier Scheme

Let $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$ be a ROM signature scheme with query complexity (q, q_V) and a non-adaptive, deterministic verifier. We assume without loss of generality that Verify makes exactly q_V distinct queries (pad with dummy distinct queries otherwise), and that Gen passes the public key to Sign via the secret key. The low-query verifier variant $\widehat{\text{SIG}} = (\widehat{\text{Gen}}, \widehat{\text{Sign}}, \widehat{\text{Verify}})$ of SIG is defined as follows.

Notation 4.1 ($\text{Verify}^{\mathbf{y}}$). For $\text{pk} \in \{0, 1\}^*$ and $\mathbf{y} \in (\{0, 1\}^*)^*$, let $\text{Verify}_{\text{pk}}^{\mathbf{y}}(\cdot)$ denote the execution of $\text{Verify}_{\text{pk}}(\cdot)$ in which the j^{th} query made by Verify is answered by $\mathbf{y}_j$.

That is, the oracle answers are taken from $\mathbf{y}$, in query order.

Definition 4.2 (map). For a function $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$, a vector $\mathbf{x} \in (\{0, 1\}^*)^*$, an index set $\mathcal{J} \subseteq [|\mathbf{x}|]$, and a vector $\mathbf{z} \in (\{0, 1\}^*)^{|\mathcal{J}|}$, define $\text{map}(f, \mathbf{x}, \mathcal{J}, \mathbf{z}) \in (\{0, 1\}^*)^{|\mathbf{x}|}$ by

$$\text{map}(f, \mathbf{x}, \mathcal{J}, \mathbf{z})_j = \begin{cases} f(\mathbf{x}_j) & j \notin \mathcal{J}, \\ \mathbf{z}_{\text{idx}_{\mathcal{J}}(j)} & \text{otherwise,} \end{cases}$$

where $\text{idx}_{\mathcal{J}}(j)$ is the index of j in $\mathcal{J}$.

That is, map combines the answers from $\mathbf{z}$, to be used for the queries in $\mathbf{x}_{\mathcal{J}}$, and those from f , to be used for $\mathbf{x}_{\overline{\mathcal{J}}}$, into a single answer vector $\mathbf{y}$ (ready to be used by Verify). Finally, recall that $\mathcal{P}_{2^\gamma}(\{0, 1\}^\lambda)$ denotes the family of 2^γ -size subsets of $\{0, 1\}^\lambda$.

Algorithm 4.3 (Low-query verifier scheme $\widehat{\text{SIG}} = (\widehat{\text{Gen}}, \widehat{\text{Sign}}, \widehat{\text{Verify}})$)

Underlying scheme: $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$.

Parameters: $1^\lambda, 1^\ell$ and $\gamma, s \in \mathbb{N}$.

Oracle: $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$.

Key-generator $\widehat{\text{Gen}}^f(1^\lambda, 1^\ell)$. Same as $\text{Gen}^f(1^\lambda, 1^\ell)$.

Signing $\widehat{\text{Sign}}_{\text{sk}, \text{pk}}^f(m)$.^a

1. Let $\pi \xleftarrow{R} \text{Sign}_{\text{sk}}^f(m)$.
2. Let $\mathcal{J} \subseteq [q_V]$ be a set of size at least $q_V - s$ that maximizes $\Pr \left[\widehat{\text{Verify}}_{\text{pk}}^f(m, (\pi, \mathcal{J})) = 1 \right]$ (breaking ties arbitrarily).
3. Output $\widehat{\pi} \leftarrow (\pi, \mathcal{J})$.

Verification $\widehat{\text{Verify}}_{\text{pk}}^f(m, \widehat{\pi} = (\pi, \mathcal{J}))$.

1. Reject if $\mathcal{J} \not\subseteq [q_V]$ or $|\mathcal{J}| < q_V - s$.
2. Let $\mathbf{x}$ be the vector of oracle queries asked by $\text{Verify}_{\text{pk}}(m, \pi)$ (listed in calling order).
3. Sample $\mathcal{Z} \xleftarrow{R} \left(\mathcal{P}_{2^\gamma}(\{0, 1\}^\lambda) \right)^{|\mathcal{J}|}$.
4. Accept if $\exists \mathbf{z} \in \mathcal{Z}$ such that $\text{Verify}_{\text{pk}}^{\mathbf{y}}(m, \pi) = 1$, for $\mathbf{y} \leftarrow \text{map}(f, \mathbf{x}, \mathcal{J}, \mathbf{z})$.

^aWe assume the signer receives pk via the secret key.

For fixed (SIG, s, γ) , Algorithm 4.3 yields the following signature scheme.

Definition 4.4. For a signature scheme SIG and $s, \gamma: \mathbb{N} \rightarrow \mathbb{N}$, let $\widehat{\text{SIG}}(\text{SIG}, s, \gamma)$ be the scheme that on parameters $1^\lambda, 1^\ell$, acts according to Algorithm 4.3 with parameters $\text{SIG}, 1^\lambda, 1^\ell, s(\lambda), \gamma(\lambda)$.

4.2 Security

In this section we state and prove the security of the low-query verifier scheme $\widehat{\text{SIG}}$ described in Algorithm 4.3. For the completeness part, we also consider the case where the underlying signature scheme SIG is *query-separated*.

Definition 4.5 (Query-separated schemes). A ROM signature scheme $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$ is δ -query-separated, if for every $\lambda, \ell \in \mathbb{N}$ and $m \in \{0, 1\}^\ell$ it holds that

$$\Pr_{\substack{f \xleftarrow{R} \mathcal{F}_\lambda \\ (\text{sk}=(\cdot, \mathcal{Q}^G), \text{pk}) \xleftarrow{R} \text{Gen}^f(1^\lambda, 1^\ell) \\ \pi \xleftarrow{R} \text{Sign}_{\text{sk}}^f(m)}}} \left[\mathcal{Q}^V(\text{pk}, m, \pi) \cap \mathcal{Q}^G \neq \emptyset \right] \leq \delta(\lambda),$$

letting $\mathcal{Q}^G$ denote the queries made by Gen and $\mathcal{Q}^V(\text{pk}, m, \pi)$ the queries made by $\text{Verify}_{\text{pk}}(m, \pi)$ (which, Verify being non-adaptive with deterministic queries, do not depend on f).

That is, the verifier is unlikely to make a query asked by the key-generation algorithm. For such schemes we prove that the query complexity of the resulting verifier is *independent* of the

public-key length. In Section 4.3, we prove that for many-time signatures, assuming that SIG is query-separated is essentially without loss of generality.

Theorem 4.6 (Security of Algorithm 4.3). *Let SIG be a ROM signature scheme with a non-adaptive, deterministic verifier, query complexity (q, q_V) , public-key length ℓ_{pk} , and signature length ℓ_π . Let $s, \gamma: \mathbb{N} \rightarrow \mathbb{N}$, and let $\widehat{\text{SIG}} := \widehat{\text{SIG}}(\text{SIG}, s, \gamma)$. Then $\widehat{\text{SIG}}$ has a deterministic-query verifier with query complexity $(q + q_V + s, s)$, and the following completeness and soundness properties.*

Completeness. *Assume SIG has completeness α and that $4\lceil \log q_V \rceil + 4 \leq \gamma \leq \lambda$.⁸*

- *For $h \leftarrow \ell_{\text{pk}} + \ell_\pi - \log \alpha$, if $s \geq 16h/\gamma + 4$, then $\widehat{\text{SIG}}$ has $(\frac{\alpha}{4}, \frac{1}{32\lambda q_V})$ -completeness (Definition 3.12), and in particular completeness $\frac{\alpha}{128\lambda q_V}$.*
- *For $h \leftarrow \ell_\pi + 1 - \log(\alpha - \delta)$, if SIG is δ -query-separated and $s \geq 16h/\gamma + 4$, then $\widehat{\text{SIG}}$ has $(\frac{\alpha-\delta}{8}, \frac{1}{32\lambda q_V})$ -completeness, and in particular completeness $\frac{\alpha-\delta}{256\lambda q_V}$.*

Soundness. *Assume SIG has (t, β) -one-time salted soundness, and that $2 \leq \gamma$, $2^{\gamma+2} \cdot q_V \leq 2^\lambda$, and $\beta \geq 2^{-\lambda}$.⁹ Then $\widehat{\text{SIG}}$ has $(t - (2^{\gamma+2} + 2) \cdot q_V, 2\beta)$ -one-time soundness.¹⁰*

Proof. The query complexity of $\widehat{\text{SIG}}$ is immediate from the construction: $\widehat{\text{Verify}}$ queries f only on the verifier queries whose index lies outside the set $\mathcal{J}$, i.e., at most $q_V - |\mathcal{J}| \leq s$ queries. $\widehat{\text{Gen}}$ is Gen , and $\widehat{\text{Sign}}$ runs Sign and one Verify pass to read its queries and their answers, so the overall complexity is at most $q + s \leq q + q_V + s$, as q bounds the queries of Gen , Sign and Verify together. Completeness is proven in Claim 4.7, and in Claim 4.9 for the query-separated case. Soundness is proven in Claim 4.10. $\square$

4.2.1 Completeness

We start by proving the non-query-separated case. Throughout we assume $4\lceil \log q_V \rceil + 4 \leq \gamma \leq \lambda$.

Claim 4.7. *Assume SIG has completeness α , and that for $h \leftarrow \ell_{\text{pk}} + \ell_\pi - \log \alpha$ it holds that $s \geq 16h/\gamma + 4$. Then $\widehat{\text{SIG}}$ has $(\frac{\alpha}{4}, \frac{1}{32\lambda q_V})$ -completeness, and in particular completeness $\frac{\alpha}{128\lambda q_V}$.*

Proof. Fix $m \in \{0, 1\}^\ell$. Let $F \xleftarrow{R} \mathcal{F}_\lambda$ and $(\text{SK}, \text{PK}) \xleftarrow{R} \widehat{\text{Gen}}^F(1^\lambda, 1^\ell)$. The following random variables are defined with respect to a random execution of $\widehat{\text{Sign}}_{\text{SK}, \text{PK}}^F(m)$.

Let Π denote the value of the signature π output by Sign . Let X denote the vector of oracle queries made by $\text{Verify}_{\text{PK}}(m, \Pi)$, listed in calling order, and let J denote the value of $\mathcal{J}$. Finally, let G be the event that there exists $\mathcal{J} \subseteq [q_V]$ with $|\mathcal{J}| \geq q_V - s$ such that

$$\Pr_{Z \xleftarrow{R} (\mathcal{P}_{2^\gamma}(\{0,1\}^\lambda))^{|J|}} \left[\exists \mathbf{z} \in Z: \text{Verify}_{\text{PK}}^{\text{map}(F, X, \mathcal{J}, \mathbf{z})}(m, \Pi) = 1 \right] \geq \frac{1}{32\lambda q_V} \quad (4)$$

⁸Tuning the parameters γ and s yields a tradeoff between the scheme's completeness and its verifier's query complexity.

⁹This is the only regime of interest, as we never require security better than $2^{-\lambda}$.

¹⁰Actually, $\widehat{\text{SIG}}$ has salted soundness, but this is irrelevant to our lower bounds.

Assume that G occurs. As $\widehat{\text{Sign}}$ sets J to a set of size at least $q_V - s$ maximizing $\Pr[\widehat{\text{Verify}}_{\text{PK}}^F(m, (\Pi, \mathcal{J})) = 1]$, and as the set $\mathcal{J}$ guaranteed by G is a candidate for that maximization, it holds that

$$\Pr[\widehat{\text{Verify}}_{\text{PK}}^F(m, (\Pi, J)) = 1] \geq \frac{1}{32\lambda q_V} \quad (5)$$

Hence, we conclude the proof using the following claim (proven below).

Claim 4.8. $\Pr[G] \geq \alpha/4$.

Since G is determined by F and (SK, PK) , and Equation (5) holds whenever G occurs, Claim 4.8 yields the $(\frac{\alpha}{4}, \frac{1}{32\lambda q_V})$ -completeness. Completeness $\frac{\alpha}{128\lambda q_V}$ follows by Proposition 3.13. $\square$

Proving Claim 4.8.

Proof of Claim 4.8. Let $n \leftarrow q_V$. Let $T \leftarrow (\text{PK}, \Pi)$ and let $\mathcal{T} \leftarrow \text{Supp}(T)$ (hence, $\mathcal{T} \subseteq \{0, 1\}^{\ell_{\text{pk}}} \times \{0, 1\}^{\ell_\pi}$). For $t = (\text{pk}, \pi) \in \mathcal{T}$, let $\mathbf{x}(t)$ denote the queries of $\text{Verify}_{\text{pk}}(m, \pi)$, listed in calling order, and let $Y_t \leftarrow F(\mathbf{x}(t)) = (F(\mathbf{x}(t)_1), \dots, F(\mathbf{x}(t)_n))$. Since the queries are distinct, the random variable Y_t is uniform over $(\{0, 1\}^\lambda)^n$, for every t . Let

$$\mathcal{W} := \{(t = (\text{pk}, \pi), \mathbf{y}) : \text{Verify}_{\text{pk}}^Y(m, \pi) = 1\}.$$

Note that $\text{Verify}_{\text{PK}}^{Y_T}(m, \Pi) = \text{Verify}_{\text{PK}}^F(m, \Pi)$ and that $(T, Y_T) \in \mathcal{W}$ if and only if $\text{Verify}_{\text{PK}}^F(m, \Pi) = 1$. Let W be the indicator for $(T, Y_T) \in \mathcal{W}$. The completeness of SIG yields that

$$p_{\mathcal{W}} := \Pr[W = 1] \geq \alpha.$$

Since $\log(1/\Pr[W = 1]) + \log|\mathcal{T}| \leq \log(1/\alpha) + \ell_{\text{pk}} + \ell_\pi = h$, applying Lemma 3.6 with respect to $(n, \{Y_t\}, T, W)$ yields that with probability at least $1/4$ over $(t, \mathbf{y}) \xleftarrow{R} (T, Y_T) \mid W = 1$, there exists a set $\mathcal{J} \subseteq [n]$ of size at least $n - (16h/\gamma + 4) \geq n - s$ such that

$$\Pr_{\mathcal{Z} \xleftarrow{R} (\mathcal{P}_{2\gamma}(\{0,1\}^\lambda))^{|J|}} [\mathcal{Z} \cap \mathcal{G} \neq \emptyset] \geq \frac{1}{32\lambda q_V} \quad (6)$$

for $\mathcal{G} := \text{Supp}((Y_t)_{\mathcal{J}} \mid W = 1, T = t, (Y_t)_{\overline{\mathcal{J}}} = \mathbf{y}_{\overline{\mathcal{J}}})$, letting $\overline{\mathcal{J}} \leftarrow [n] \setminus \mathcal{J}$. We note that for every $\mathbf{g} \in \mathcal{G}$, if $(T, Y_T) = (t, \mathbf{y})$, then

$$\text{Verify}_{\text{PK}}^{\text{map}(F, \mathbf{x}(T), \mathcal{J}, \mathbf{g})}(m, \Pi) = 1 \quad (7)$$

Indeed, if $(T, Y_T) = (t, \mathbf{y})$ then, by definition, $\text{map}(F, \mathbf{x}(t), \mathcal{J}, \mathbf{g}) = \mathbf{y}'$ for $\mathbf{y}'$ with $\mathbf{y}'_{\mathcal{J}} = \mathbf{g}$ and $\mathbf{y}'_{\overline{\mathcal{J}}} = F(\mathbf{x}(t)_{\overline{\mathcal{J}}}) = \mathbf{y}_{\overline{\mathcal{J}}}$. By the definition of $\mathcal{G}$, $\mathbf{y}' \in \text{Supp}(Y_t \mid W = 1, T = t, (Y_t)_{\overline{\mathcal{J}}} = \mathbf{y}_{\overline{\mathcal{J}}})$; namely, $(t, \mathbf{y}') \in \mathcal{W}$, and Equation (7) holds. We conclude that with probability at least $p_{\mathcal{W}}/4 \geq \alpha/4$, there exists $\mathcal{J} \subseteq [q_V]$ with $|\mathcal{J}| \geq q_V - s$ such that

$$\begin{aligned} \Pr_{\mathcal{Z} \xleftarrow{R} (\mathcal{P}_{2\gamma}(\{0,1\}^\lambda))^{|J|}} [\exists \mathbf{z} \in \mathcal{Z} : \text{Verify}_{\text{PK}}^{\text{map}(F, \mathbf{x}(T), \mathcal{J}, \mathbf{z})}(m, \Pi) = 1] &\geq \Pr_{\mathcal{Z} \xleftarrow{R} (\mathcal{P}_{2\gamma}(\{0,1\}^\lambda))^{|J|}} [\mathcal{Z} \cap \mathcal{G} \neq \emptyset] \\ &\geq \frac{1}{32\lambda q_V}. \end{aligned}$$

The first inequality is by Equation (7) and the second by Equation (6). That is, G occurs with probability at least $\alpha/4$. $\square$

The query-separated case.

Claim 4.9. *Assume SIG has completeness α and is δ -query-separated, and that for $h \leftarrow \ell_\pi + 1 - \log(\alpha - \delta)$ it holds that $s \geq 16h/\gamma + 4$. Then $\widehat{\text{SIG}}$ has $(\frac{\alpha-\delta}{8}, \frac{1}{32\lambda q_V})$ -completeness, and in particular completeness $\frac{\alpha-\delta}{256\lambda q_V}$.*

Proof. Assume first that $\delta = 0$, i.e., $\mathcal{Q}^V(\text{PK}, m, \Pi)$ always avoids Gen's query set $\mathcal{Q}^G$. Let Θ denote Gen's transcript, i.e., the list of query/answer pairs made by $\text{Gen}^F(1^\lambda, 1^\ell)$ together with its randomness, and note that Θ determines (SK, PK) and $\mathcal{Q}^G$. Fix a value θ of Θ , and let $\text{pk} = \text{pk}(\theta)$. Conditioned on $\Theta = \theta$, the function F is still uniform outside $\mathcal{Q}^G(\theta)$. Since $\delta = 0$, the vector $\mathbf{x}(\pi) \leftarrow \mathcal{Q}^V(\text{pk}, m, \pi)$ is disjoint from $\mathcal{Q}^G(\theta)$ for every $\pi \in \text{Supp}(\Pi \mid \Theta = \theta)$, and hence, for every such π , the random variable

$$Y_\pi \leftarrow F(\mathbf{x}(\pi)) \mid \Theta = \theta$$

is uniform over $(\{0, 1\}^\lambda)^n$, for $n \leftarrow q_V$. We may thus repeat the proof of Claim 4.8 verbatim, in the probability space conditioned on $\Theta = \theta$, with respect to the family $\{Y_\pi\}$, the random variable $T \leftarrow \Pi \mid \Theta = \theta$ and $\mathcal{T} \leftarrow \text{Supp}(T) \subseteq \{0, 1\}^{\ell_\pi}$, and the indicator W of $\text{Verify}_{\text{pk}}^F(m, \Pi) = 1$. Letting $\alpha_\theta \leftarrow \Pr[W = 1 \mid \Theta = \theta]$, if $\alpha_\theta \geq \alpha/2$ then $\log(1/\alpha_\theta) + \log|\mathcal{T}| \leq \log(2/\alpha) + \ell_\pi = h$, and thus the event G of Claim 4.7 occurs with probability at least $\alpha_\theta/4$ conditioned on $\Theta = \theta$. Since $\mathbb{E}_\theta[\alpha_\theta] = \Pr[W = 1] \geq \alpha$, and the θ 's with $\alpha_\theta < \alpha/2$ contribute at most $\alpha/2$ to this expectation, averaging over the remaining θ 's yields $\Pr[G] \geq \frac{1}{4} \cdot (\alpha - \alpha/2) = \alpha/8$. Applying Equation (5) exactly as in Claim 4.7 yields the $(\frac{\alpha}{8}, \frac{1}{32\lambda q_V})$ -completeness.

For the case that $\delta > 0$, consider the variant SIG' of SIG in which Sign aborts if $\mathcal{Q}^V(\text{pk}, m, \pi) \cap \mathcal{Q}^G \neq \emptyset$.¹¹ The completeness of SIG' is at least $\alpha - \delta$, and it is 0-query-separated, since an aborting Sign produces no accepted signature. By the above, when using SIG' the scheme $\widehat{\text{SIG}}$ has $(\frac{\alpha-\delta}{8}, \frac{1}{32\lambda q_V})$ -completeness. Clearly, $\widehat{\text{SIG}}$ has at least this completeness when using SIG. $\square$

4.2.2 Soundness

Claim 4.10. *Assume SIG has (t, β) -one-time salted soundness, and that $2 \leq \gamma$, $2^{\gamma+2} \cdot q_V \leq 2^\lambda$, and $\beta \geq 2^{-\lambda}$. Then $\widehat{\text{SIG}}$ has $(t - (2^{\gamma+2} + 2) \cdot q_V, 2\beta)$ -one-time soundness.*

For proving Claim 4.10, we use the following fact.

Fact 4.11. *Let $n, k \in \mathbb{N}$ with $k \leq n$ and let $\mathcal{U}$ be a set of size n . Then*

$$\Pr_{y_1, \dots, y_{4k} \xleftarrow{R} \mathcal{U}} [|\{y_1, \dots, y_{4k}\}| \leq k] \leq (e \cdot k/n)^{3k}.$$

Proof. The event $|\{y_1, \dots, y_{4k}\}| \leq k$ implies $y_1, \dots, y_{4k} \in \mathcal{S}$ for some $\mathcal{S} \subseteq \mathcal{U}$ of size k . For a fixed $\mathcal{S}$, this occurs with probability $(k/n)^{4k}$. Hence, a union bound over the choice of $\mathcal{S}$ yields that

$$\Pr_{y_1, \dots, y_{4k} \xleftarrow{R} \mathcal{U}} [|\{y_1, \dots, y_{4k}\}| \leq k] \leq \binom{n}{k} \cdot (k/n)^{4k} \leq \left(\frac{e \cdot n}{k}\right)^k \cdot (k/n)^{4k} = e^k \cdot (k/n)^{3k} \leq (e \cdot k/n)^{3k},$$

where we used the bound $\binom{a}{b} \leq (e \cdot a/b)^b$ for the second inequality. $\square$

¹¹We assume without loss of generality that Gen appends its query/answer pairs to the secret key, and thus Sign knows $\mathcal{Q}^G$ in advance.

Proof of Claim 4.10. Let $\widehat{A}$ be an admissible adversary for the one-time salted soundness of $\widehat{\text{SIG}}$, making $\widehat{t}$ queries and forging with probability $\widehat{\beta}$. We use $\widehat{A}$ to build the adversary A of Algorithm 4.12 for the salted soundness of SIG .

Algorithm 4.12 (A: Adversary for the salted soundness of SIG)

Parameters: $1^\lambda, 1^\ell$ and $\gamma, s \in \mathbb{N}$.

Oracles: $\text{SRO}_\lambda.\text{All}$, $\text{Sign}_{\text{sk}}^{\text{SRO}_\lambda.\text{Query}}$.

Input: Public key pk .

Operation:

1. Emulate the execution of $\widehat{A}^{\text{SRO}_\lambda.\text{All}, \widehat{\text{Sign}}_{\text{sk}}^{\text{SRO}_\lambda.\text{Query}}}(\text{pk})$. Answer its signing query m to $\widehat{\text{Sign}}_{\text{sk}}$ as follows:
 - (a) Let $\pi \xleftarrow{R} \text{Sign}_{\text{sk}}^{\text{SRO}_\lambda.\text{Query}}(m)$.
 - (b) Emulate the execution of $\widehat{\text{Sign}}$ as follows:
 - i. The output of $\widehat{\text{Sign}}$ sampled in Step 1 is set to π .
 - ii. In Step 2 of $\widehat{\text{Sign}}$, answer the oracle queries made by $\text{Verify}_{\text{pk}}(m, \pi)$ using $\text{SRO}_\lambda.\text{Query}$.
 - (c) Return $\widehat{\text{Sign}}$'s output to $\widehat{A}$.
2. Let $(m, \widehat{\pi} = (\pi, \mathcal{J}))$ be the output of the above emulation, and let $\mathbf{x}$ be the vector of oracle queries asked by $\text{Verify}_{\text{pk}}(m, \pi)$ (listed in calling order). Abort if $\mathcal{J} \not\subseteq [q_V]$ or $|\mathcal{J}| < q_V - s$.
3. For each $j \in [q_V] \setminus \mathcal{J}$: let $u_j \leftarrow \text{SRO}_\lambda.\text{Query}(\mathbf{x}_j)$.
4. For each $j \in \mathcal{J}$: Make $2^{\gamma+2}$ calls to $\text{SRO}_\lambda.\text{Sample}(\mathbf{x}_j)$, and let $\mathcal{Z}_j$ be the set of the first 2^γ distinct answers to these calls. Abort if there are fewer than 2^γ distinct answers.
5. If $\exists \mathbf{z} \in \prod_{j \in \mathcal{J}} \mathcal{Z}_j$ such that $\text{Verify}_{\text{pk}}^{\mathbf{y}}(m, \pi) = 1$, for $\mathbf{y}$ defined by $\mathbf{y}_j \leftarrow \mathbf{z}_{\text{id}_{\mathcal{J}}(j)}$ for $j \in \mathcal{J}$ and $\mathbf{y}_j \leftarrow u_j$ otherwise:
 - (a) For each $j \in \mathcal{J}$: call $\text{SRO}_\lambda.\text{Rewind}(\mathbf{x}_j, \mathbf{z}_{\text{id}_{\mathcal{J}}(j)})$.
 - (b) Output (m, π) .

Let A be the adversary of Algorithm 4.12, and assume $\widehat{t} = t - (2^{\gamma+2} + 2) \cdot q_V$. Note that A queries Sign only when $\widehat{A}$ queries $\widehat{\text{Sign}}$, and on the same message. Hence, A makes at most a single signing query, and it is admissible since $\widehat{A}$ is. As for its query complexity, in addition to $\widehat{A}$'s $\widehat{t}$ queries, A makes q_V calls to $\text{SRO}_\lambda.\text{Query}$ when answering the signing query of $\widehat{A}$, and at most $(2^{\gamma+2} + 1) \cdot q_V$ calls once the emulation ends. Overall, A makes at most t queries.

A perfectly emulates $\widehat{A}$ up to Step 2. Hence,

$$\Pr \left[\widehat{\text{Verify}}_{\text{pk}}^{\text{SRO}_\lambda.\text{Query}}(m, \widehat{\pi}) = 1 \right] \geq \widehat{\beta},$$

where the probability is over $(\text{sk}, \text{pk}) \xleftarrow{R} \text{Gen}^{\text{SRO}_\lambda.\text{Query}}(1^\lambda, 1^\ell)$, $(m, \widehat{\pi} = (\pi, \mathcal{J})) \xleftarrow{R} \widehat{A}^{\text{SRO}_\lambda.\text{All}, \widehat{\text{Sign}}_{\text{sk}}^{\text{SRO}_\lambda.\text{Query}}}(\text{pk})$, and the sampling $\mathcal{Z} \xleftarrow{R} (\mathcal{P}_{2^\gamma}(\{0, 1\}^\lambda))^{| \mathcal{J} |}$ made by $\widehat{\text{Verify}}$.

For $j \in \mathcal{J}$, let Y_j be the set of answers to the $2^{\gamma+2}$ calls made for j in Step 4. We now bound the probability that Y_j contains fewer than 2^γ elements. By Fact 4.11 with respect to $\mathcal{U} \leftarrow \{0, 1\}^\lambda$

and $k \leftarrow 2^\gamma$ it holds that

$$\Pr[|Y_j| < 2^\gamma] \leq (e \cdot 2^{\gamma-\lambda})^{3 \cdot 2^\gamma}.$$

Let E be the event that $|Y_j| \geq 2^\gamma$ for every $j \in \mathcal{J}$, i.e., that $\mathbf{A}$ does not abort in Step 4. A union bound over the at most q_V indices in $\mathcal{J}$ yields $\Pr[\neg E] \leq q_V \cdot (e \cdot 2^{\gamma-\lambda})^{3 \cdot 2^\gamma}$. Assume E . Since the answers are i.i.d. uniform, $(Z_j)_{j \in \mathcal{J}}$ is distributed as $\mathcal{Z}$ in the above display. Thus, Step 5 finds an accepting $\mathbf{z}$ whenever $\widehat{\text{Verify}}$ accepts with randomness $(Z_j)_{j \in \mathcal{J}}$, and its rewinding yields $\text{Verify}_{\text{pk}}^{\text{SRO}_\lambda \cdot \text{Query}}(m, \pi) = 1$. Hence, we conclude that

$$\begin{aligned} \Pr\left[\widehat{\text{Verify}}_{\text{pk}}^{\text{SRO}_\lambda \cdot \text{Query}}(m, \widehat{\pi}) = 1\right] &\leq \Pr\left[\text{Verify}_{\text{pk}}^{\text{SRO}_\lambda \cdot \text{Query}}(m, \pi) = 1\right] + \Pr[\neg E] \\ &\leq \beta + q_V \cdot (e \cdot 2^{\gamma-\lambda})^{3 \cdot 2^\gamma} \leq \beta + 2^{-\lambda} \leq 2\beta. \end{aligned}$$

The second inequality is by the soundness of SIG and the above bound on $\Pr[\neg E]$, the third by $2 \leq \gamma$ and $2^{\gamma+2} \cdot q_V \leq 2^\lambda$, and the last by $\beta \geq 2^{-\lambda}$. $\square$

4.3 An Improved Reduction for Many-Time Signatures

We show that for many-time schemes, we get essentially the same bound as if we had query separation. Towards this end, we consider the following variant of Algorithm 4.3, in which the key-generator algorithm signs many random messages and appends the query/answer pairs the verifier makes on these signatures to the public key. For queries published in the public key, the verifier uses the published answers instead of calling the oracle (actually, it uses them in a more subtle way; see below).

Algorithm 4.13 (Low-query verifier scheme, many-time, $\widetilde{\text{SIG}} = (\widetilde{\text{Gen}}, \widetilde{\text{Sign}}, \widetilde{\text{Verify}})$)

Underlying scheme: $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$.

Parameters: $1^\lambda, 1^\ell$ and $\gamma, s, n \in \mathbb{N}$.

Oracle: $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$.

Key-generator $\widetilde{\text{Gen}}^f(1^\lambda, 1^\ell)$.

1. Let $(\text{sk}, \text{pk}) \xleftarrow{\text{R}} \text{Gen}^f(1^\lambda, 1^\ell)$.
2. Let $\mathcal{A}, \mathcal{M} \leftarrow \emptyset$.
3. For $i = 1$ to n :
 - (a) Sample $m_i \xleftarrow{\text{R}} \{0, 1\}^\ell$, and let $\mathcal{M} \leftarrow \mathcal{M} \cup \{m_i\}$.
 - (b) Let $\pi_i \xleftarrow{\text{R}} \text{Sign}_{\text{sk}}^f(m_i)$.
 - (c) Let $\mathcal{Q}_i$ be the set of oracle queries asked by $\text{Verify}_{\text{pk}}(m_i, \pi_i)$.
 - (d) Let $\mathcal{A} \leftarrow \mathcal{A} \cup \{(x, f(x)) : x \in \mathcal{Q}_i\}$.
4. Output $(\widetilde{\text{sk}} \leftarrow (\text{sk}, \text{pk}), \widetilde{\text{pk}} \leftarrow (\text{pk}, \mathcal{A}, \mathcal{M}, u))$, for $u \xleftarrow{\text{R}} \{0, 1\}^\ell$.

Signing $\widetilde{\text{Sign}}_{\widetilde{\text{sk}}=(\text{sk}, \widetilde{\text{pk}})}^f(m)$.

1. Let $m_u \leftarrow m \oplus u$.
2. Let $\pi \xleftarrow{\text{R}} \text{Sign}_{\text{sk}}^f(m_u)$.
3. Let $\mathcal{J} \subseteq [q_V]$ be a set that maximizes $\Pr \left[\widetilde{\text{Verify}}_{\widetilde{\text{pk}}}^f(m, (\pi, \mathcal{J})) = 1 \right]$.
4. Output $\widetilde{\pi} \leftarrow (\pi, \mathcal{J})$.

Verification $\widetilde{\text{Verify}}_{\widetilde{\text{pk}}=(\text{pk}, \mathcal{A}, \mathcal{M}, u)}^f(m, \widetilde{\pi} = (\pi, \mathcal{J}))$.

1. Let $m_u \leftarrow m \oplus u$. Reject if $m_u \in \mathcal{M}$.
2. Let $\mathbf{x}$ be the vector of oracle queries asked by $\text{Verify}_{\text{pk}}(m_u, \pi)$.
3. Reject if $\mathcal{J} \not\subseteq [q_V]$ or $|\{\mathbf{x}_i : i \in ([q_V] \setminus \mathcal{J}) \wedge (\mathbf{x}_i, \cdot) \notin \mathcal{A}\}| \geq s + 1$.
4. Let $f_{\mathcal{A}}$ be the function defined by $f_{\mathcal{A}}(x) = \begin{cases} y & (x, y) \in \mathcal{A} \text{ for some } y, \\ f(x) & \text{otherwise.} \end{cases}$
5. Sample $\mathcal{Z} \xleftarrow{\text{R}} \left(\mathcal{P}_{2\gamma}(\{0, 1\}^\lambda) \right)^{|\mathcal{J}|}$.
6. Accept if $\exists \mathbf{z} \in \mathcal{Z}$ such that $\text{Verify}_{\text{pk}}^{\mathbf{y}}(m_u, \pi) = 1$, for $\mathbf{y} \leftarrow \text{map}(f_{\mathcal{A}}, \mathbf{x}, \mathcal{J}, \mathbf{z})$.

That is, the key-generation algorithm appends to the public key the query/answer pairs made by the verifier on n random signatures. When relevant, the verifier uses these pairs instead of calling f . The shift u makes the signed message $m \oplus u$ uniform, and thus distributed as the messages pre-signed in Step 3. For soundness, the verifier rejects the messages of $\mathcal{M}$, which the key generator has already signed.

For fixed $(\text{SIG}, s, \gamma, n)$, Algorithm 4.13 yields the following signature scheme.

Definition 4.14. For a signature scheme SIG and $s, \gamma, n: \mathbb{N} \rightarrow \mathbb{N}$, let $\widetilde{\text{SIG}}(\text{SIG}, s, \gamma, n)$ be the scheme that on parameters $1^\lambda, 1^\ell$, acts according to Algorithm 4.13 with parameters $\text{SIG}, 1^\lambda, 1^\ell, s(\lambda), \gamma(\lambda), n(\lambda)$.

4.3.1 Security

The property of $\widetilde{\text{SIG}}$ that replaces query separation is that $\mathcal{A}$ records, with high probability, the queries asked by both Gen and the verifier. Below, we sometimes abuse notation and write $x \in \mathcal{A}$ for $(x, y) \in \mathcal{A}$ for some y .

Theorem 4.15 (Security of Algorithm 4.13). *Let SIG be a ROM signature scheme with a non-adaptive, deterministic verifier, query complexity (q, q_V) , and signature length ℓ_π . Let $s, \gamma, n: \mathbb{N} \rightarrow \mathbb{N}$, and let $\widetilde{\text{SIG}} := \widetilde{\text{SIG}}(\text{SIG}, s, \gamma, n)$. Then $\widetilde{\text{SIG}}$ has a deterministic-query verifier with query complexity $((n+2) \cdot q + s, s)$, and the following completeness and soundness properties.*

Completeness. *Assume SIG has completeness α , and that $4\lceil \log q_V \rceil + 4 \leq \gamma \leq \lambda$ and $\ell \geq \lambda$. For $h \leftarrow \ell_\pi + 1 - \log \alpha$ and $\tilde{\alpha} \leftarrow \frac{\alpha}{512\lambda q_V}$, if $s \geq 16h/\gamma + 4$ and $\left\lceil \frac{4q}{\alpha} \cdot \ln\left(\frac{4q}{\alpha}\right) \right\rceil \leq n \leq \tilde{\alpha} \cdot 2^{\lambda-1}$, then $\widetilde{\text{SIG}}$ has $(\frac{\alpha}{16}, \frac{1}{32\lambda q_V})$ -completeness, and in particular completeness $\tilde{\alpha}$.*

Many-time soundness. *Assume SIG has (t, β) -many-time salted soundness, and that $2 \leq \gamma$, $2^{\gamma+2} \cdot q_V \leq 2^\lambda$, and $\beta \geq 2^{-\lambda}$. Then $\widetilde{\text{SIG}}$ has $(t/(2q_V) - 2^{\gamma+2} - n, 2\beta)$ -many-time soundness.*

Proof. On $\tilde{\pi} = (\pi, \mathcal{J})$, the verifier asks f only on the coordinates of $\mathbf{x}$ outside $\mathcal{J}$ that are not recorded in $\mathcal{A}$, i.e., at most s queries by Step 3. These are determined by (pk, m_u, π) , and since $m_u = m \oplus u$ is determined by pk and m , all its queries are determined by $(\text{pk}, m, \tilde{\pi})$. Gen makes at most $q + n \cdot q$ queries and $\widetilde{\text{Sign}}$ at most q , giving the stated complexity. Completeness is Claim 4.16 and soundness is Claim 4.18. $\square$

4.3.2 Completeness

Claim 4.16. *Assume SIG has completeness α , that $\ell \geq \lambda$, and that for $h \leftarrow \ell_\pi + 1 - \log \alpha$ and $\tilde{\alpha} \leftarrow \frac{\alpha}{512\lambda q_V}$ it holds that $s \geq 16h/\gamma + 4$ and $\left\lceil \frac{4q}{\alpha} \cdot \ln\left(\frac{4q}{\alpha}\right) \right\rceil \leq n \leq \tilde{\alpha} \cdot 2^{\lambda-1}$. Then $\widetilde{\text{SIG}}$ has $(\frac{\alpha}{16}, \frac{1}{32\lambda q_V})$ -completeness, and in particular completeness $\tilde{\alpha}$.*

For proving Claim 4.16 we use the following claim.

Claim 4.17. *Let $\tilde{\alpha}, n$ be as in Claim 4.16. Then for every $m \in \{0, 1\}^\ell$:*

$$\Pr_{\substack{f \xleftarrow{R} \mathcal{F}_\lambda \\ (\tilde{\text{sk}}, \tilde{\text{pk}} = (\text{pk}, \mathcal{A}, \mathcal{M}, u)) \xleftarrow{R} \widetilde{\text{Gen}}^f(1^\lambda, 1^\ell) \\ (\pi, \mathcal{J}) \xleftarrow{R} \widetilde{\text{Sign}}_f^{\tilde{\text{sk}}}(m)}} \left[\mathcal{Q}^V(\text{pk}, m_u, \pi) \cap \mathcal{Q}^G \subseteq \mathcal{A} \right] \geq 1 - \tilde{\alpha}/2,$$

letting $m_u \leftarrow m \oplus u$, $\mathcal{Q}^G$ denote the queries made by Gen inside $\widetilde{\text{Gen}}$, and $\mathcal{Q}^V(\text{pk}, m_u, \pi)$ those made by $\text{Verify}_{\text{pk}}(m_u, \pi)$.

That is, $\mathcal{A}$ contains, with high probability, all the queries made by both Gen and Verify . Claim 4.17 is proved at the end of this section.

Proof of Claim 4.16. Assume first that $\mathcal{M} = \emptyset$, i.e., that $\widetilde{\text{Verify}}$ never rejects in Step 1. We account for a general $\mathcal{M}$ at the end of the proof.

Let SIG^* be the signature scheme resulting from modifying the key-generation algorithm of SIG to append the query/answer pairs it makes to the public key, and modifying the signing and verification algorithms to use the recorded answers for the recorded queries. Let k be the number of queries Verify makes outside $\mathcal{Q}^G$. For ease of notation, we index the queries of Verify so that those lying in $\mathcal{Q}^G$ are its last ones. Thus, the verifier of SIG^* makes the first k queries of Verify , in this order. Clearly, SIG^* is 0-query-separated, and has the completeness and the signature length of SIG . Thus, by Theorem 4.6, $\widetilde{\text{SIG}}^* = \widetilde{\text{SIG}}(\text{SIG}^*, s, \gamma)$ has $(\frac{\alpha}{8}, \frac{1}{32\lambda q_V})$ -completeness.

We next claim that the acceptance probability of $\widetilde{\text{SIG}}$ is close to that of $\widetilde{\text{SIG}}^*$. Fix f, m, u , let $m_u \leftarrow m \oplus u$, and let $\text{pk}' = (\text{pk}, \mathcal{B})$ and $\tilde{\pi} = (\pi, \mathcal{J})$ be outputs of the key-generation algorithm and the signer of $\widetilde{\text{SIG}}^*$ on message m_u , respectively, where $\mathcal{B}$ is the set of query/answer pairs recorded by Gen . Let $\widetilde{\text{pk}} = (\text{pk}, \mathcal{A}, \mathcal{M}, u)$ be a public key generated by $\widetilde{\text{Gen}}$ using the same randomness for Gen , and let $\tilde{\mathcal{J}} \leftarrow \mathcal{J} \cap [k]$. Assume that the verifier of $\widetilde{\text{SIG}}$ does not reject in Step 3 on the public key $\widetilde{\text{pk}}$ and input $(m, \tilde{\pi} = (\pi, \tilde{\mathcal{J}}))$. We claim that for such inputs, the verifier of $\widetilde{\text{SIG}}^*$ on public key pk' and input $(m_u, \tilde{\pi})$ accepts with the same probability as the verifier of $\widetilde{\text{SIG}}$ on public key $\widetilde{\text{pk}}$ and input $(m, \tilde{\pi})$. Indeed, for a query index $j \in \tilde{\mathcal{J}}$, in both schemes the verifier samples the answers to the query from $\mathcal{P}_{2^\gamma}(\{0, 1\}^\lambda)$. For $j \leq k$ with $j \notin \mathcal{J}$, in both schemes the verifier sets the answer to $f(\mathbf{x}_j)$. For $j > k$, the verifier of SIG^* uses the recorded answer $f(\mathbf{x}_j)$, whether or not $j \in \mathcal{J}$, and so does $\widetilde{\text{Verify}}$, as $j \notin \tilde{\mathcal{J}}$ and $f_{\mathcal{A}}(\mathbf{x}_j) = f(\mathbf{x}_j)$. Since $\widetilde{\text{Sign}}$ maximizes over all sets $\mathcal{J} \subseteq [q_V]$, in particular over $\tilde{\mathcal{J}}$, we conclude that the acceptance probability of $\widetilde{\text{SIG}}$ is at least that of $\widetilde{\text{SIG}}^*$, up to the probability that $\widetilde{\text{Verify}}$ rejects in Step 3, which we bound as follows.

$$\Pr \left[\widetilde{\text{Verify}}_{\widetilde{\text{pk}}}^f(m, \tilde{\pi}) \text{ rejects in Step 3} \right] \leq \tilde{\alpha}/2 \quad (8)$$

Indeed, let $\mathbf{x}$ be the queries of $\text{Verify}_{\text{pk}}(m_u, \pi)$, and let $\mathcal{Q}^G$ be the queries of Gen . Recall that the verifier of SIG^* answers the queries in $\mathcal{Q}^G$ from the recorded pairs, and queries f only on the remaining ones. Since $|\mathcal{J}| \geq q_V - s$ by Step 2, at most s of the indices in $[k]$ lie outside $\mathcal{J}$. That is, $\left| \left\{ \mathbf{x}_i : i \notin \tilde{\mathcal{J}}, \mathbf{x}_i \notin \mathcal{Q}^G \right\} \right| \leq s$. By Step 3, $\widetilde{\text{Verify}}$ rejects only if $\left| \left\{ \mathbf{x}_i : i \notin \tilde{\mathcal{J}}, \mathbf{x}_i \notin \mathcal{A} \right\} \right| \geq s + 1$. Each such $\mathbf{x}_i$ is either not in $\mathcal{Q}^G$, and there are at most s of these, or in $\mathcal{Q}^G \setminus \mathcal{A}$. Hence, $\widetilde{\text{Verify}}$ rejects only if $\mathcal{Q}^V(\text{pk}, m_u, \pi) \cap \mathcal{Q}^G \not\subseteq \mathcal{A}$, which by Claim 4.17 happens with probability at most $\tilde{\alpha}/2$, establishing Equation (8). It follows that

$$\begin{aligned} \Pr \left[\widetilde{\text{SIG}} \text{ accepts} \right] &\geq \Pr \left[\widetilde{\text{SIG}}^* \text{ accepts} \right] - \Pr \left[\widetilde{\text{Verify}}_{\widetilde{\text{pk}}}^f(m, \tilde{\pi}) \text{ rejects in Step 3} \right] \\ &\geq \frac{\alpha}{256\lambda q_V} - \tilde{\alpha}/2, \end{aligned}$$

where the second inequality holds by Equation (8). Since the rejection bounded in Equation (8) does not depend on the coins of $\widetilde{\text{Verify}}$, the same argument yields that $\widetilde{\text{SIG}}$ has $(\alpha/8 - \tilde{\alpha}/2, \frac{1}{32\lambda q_V})$ -completeness. Finally, for a general $\mathcal{M}$, $\widetilde{\text{Verify}}$ acts exactly as described above if not rejecting in Step 1, i.e., as long as $m_u \notin \mathcal{M}$. Since $m_u = m \oplus u$ is uniform, the event $m_u \in \mathcal{M}$ happens with probability at most $n \cdot 2^{-\ell} \leq n \cdot 2^{-\lambda} \leq \tilde{\alpha}/2$. Hence, $\widetilde{\text{SIG}}$ has completeness $\frac{\alpha}{256\lambda q_V} - \tilde{\alpha} = \frac{\alpha}{512\lambda q_V}$, and $(\alpha/8 - \tilde{\alpha}, \frac{1}{32\lambda q_V})$ -completeness, which yields the claim as $\tilde{\alpha} \leq \alpha/16$. $\square$

Proving Claim 4.17.

Proof of Claim 4.17. Fix $m \in \{0, 1\}^\ell$. Since $m_u = m \oplus u$ is a uniform message independent of f , it suffices to show that for a uniform message m' , $\text{Verify}_{\text{pk}}$ queries a point of $\mathcal{Q}^G \setminus \mathcal{A}$ with probability at most $\tilde{\alpha}/2$.

Fix f and $(\text{sk}, \text{pk}, \mathcal{Q}^G) \in \text{Supp}(\text{Gen}^f(1^\lambda, 1^\ell))$. For $c \in \mathcal{Q}^G$, let

$$p_c := \Pr_{m' \xleftarrow{\text{R}} \{0, 1\}^\ell, \pi \xleftarrow{\text{R}} \text{Sign}_{\text{sk}}^f(m')} \left[c \in \mathcal{Q}^V(\text{pk}, m', \pi) \right]$$

be the probability that the verifier queries c on a uniform message. Each of the $n \geq \lceil \frac{1}{\tau} \ln \frac{1}{\tau} \rceil$ pre-signings of Step 3 repeats this experiment independently, for $\tau \leftarrow \frac{\tilde{\alpha}}{4q}$.

Heavy queries. A query $c \in \mathcal{Q}^G$ is *heavy* if $p_c > \tau$. Since the pre-signings are i.i.d., a fixed heavy c is not recorded by any of them (i.e., $c \notin \mathcal{A}$) with probability at most

$$(1 - p_c)^n < e^{-\tau n} \leq \tau,$$

as $n \geq \frac{1}{\tau} \ln \frac{1}{\tau}$. By a union bound over the at most q heavy queries, $\mathcal{A}$ omits some heavy query with probability at most $q \cdot \tau = \tilde{\alpha}/4$.

Light queries. Let $X \leftarrow |\mathcal{Q}^V(\text{pk}, m', \pi) \cap \{c \in \mathcal{Q}^G : p_c \leq \tau\}|$ be the number of light queries ($p_c \leq \tau$) in $\mathcal{Q}^G$ that the verifier makes on a uniform message. Then

$$\mathbb{E}[X] = \sum_{c: p_c \leq \tau} p_c \leq q \cdot \tau = \tilde{\alpha}/4,$$

implying that the verifier queries a light $\mathcal{Q}^G$ -point with probability $\Pr[X \geq 1] \leq \tilde{\alpha}/4$.

Since any point of $\mathcal{Q}^G \setminus \mathcal{A}$ that the verifier queries is a heavy query omitted by $\mathcal{A}$ or a light query, the two bounds give $\Pr[\mathcal{Q}^V(\text{pk}, m', \pi) \cap \mathcal{Q}^G \not\subseteq \mathcal{A}] \leq \tilde{\alpha}/4 + \tilde{\alpha}/4 = \tilde{\alpha}/2$. $\square$

4.3.3 Soundness

Claim 4.18. Assume SIG has (t, β) -many-time salted soundness, and that $2 \leq \gamma$, $2^{\gamma+2} \cdot q_V \leq 2^\lambda$, and $\beta \geq 2^{-\lambda}$. Then $\widetilde{\text{SIG}}$ has $(t/(2q_V) - 2^{\gamma+2} - n, 2\beta)$ -many-time soundness.

Proof. Let $\tilde{t} \leftarrow t/(2q_V) - 2^{\gamma+2} - n$, and let $\tilde{\text{A}}$ be an adversary making at most $\tilde{t}$ queries (oracle and signing) and forging with probability $\tilde{\beta}$. The reduction is that of Algorithm 4.12, with the following changes to A , accounting for the key generation and the shift of $\widetilde{\text{SIG}}$.

1. A generates $\mathcal{A}$ and $\mathcal{M}$ as $\widetilde{\text{Gen}}$ does, using its signing oracle and $\text{SRO}_\lambda.\text{Query}$.
2. A samples $u \xleftarrow{\text{R}} \{0, 1\}^\ell$ and emulates $\tilde{\text{A}}$ on the key $(\text{pk}, \mathcal{A}, \mathcal{M}, u)$.
 - (a) A signing query m is answered by emulating $\widetilde{\text{Sign}}$ on $m \oplus u$ (as Algorithm 4.12 emulates $\widetilde{\text{Sign}}$).
3. On the output $(m^*, (\pi, \mathcal{J}))$ of $\tilde{\text{A}}$:

- (a) Abort if $m^* \oplus u \in \mathcal{M}$ or $\mathcal{J} \not\subseteq [q_V]$.
- (b) Apply Steps 3–5 to $(m^* \oplus u, (\pi, \mathcal{J}))$, answering the recorded queries outside $\mathcal{J}$ from $\mathcal{A}$ (as $\widetilde{\text{Verify}}$ does), and rewind them to the recorded answers before outputting $(m^* \oplus u, \pi)$.

The analysis is the same as in Claim 4.10, up to the following. In addition to $\widetilde{\text{A}}$'s $\widetilde{t}$ queries, A makes $n \cdot q_V$ calls to $\text{SRO}_\lambda.\text{Query}$ when generating $\mathcal{A}$, at most q_V calls per signing query of $\widetilde{\text{A}}$, and at most $(2^{\gamma+2} + 1) \cdot q_V$ calls once the emulation ends. Overall, A makes at most $\widetilde{t} + (n + \widetilde{t} + 2^{\gamma+2} + 1) \cdot q_V \leq t$ queries. It perfectly emulates $\widetilde{\text{A}}$, since $\widetilde{\text{Gen}}$ generates $\mathcal{A}$, $\mathcal{M}$, and u in the same manner, and its rewinding to the recorded answers succeeds, since they were returned by $\text{SRO}_\lambda.\text{Query}$. Finally, $\mathcal{M}$ stores every message A signs when generating $\mathcal{A}$, and A aborts if $m^* \oplus u \in \mathcal{M}$ or $\mathcal{J} \not\subseteq [q_V]$. This loses nothing, as $\widetilde{\text{Verify}}$ rejects such outputs in Steps 1 and 3. Thus, the output message $m^* \oplus u$ is not one of them, and it differs from each message $m \oplus u$ signed for $\widetilde{\text{A}}$, as $m^* \neq m$. Hence, A is admissible, and $\widehat{\beta} \leq 2\beta$. $\square$

5 Lower Bounds on Signature Length in the ROM

In this section, we use the transformation from short signature schemes to low-query verifier schemes presented in Section 4, the reductions of Section 6, and Theorem 3.21 to prove lower bounds on the signature length of ROM signature schemes. One-time schemes are considered in Section 5.1, and many-time schemes, for which the bound depends only on the signature length, are considered in Section 5.2.

5.1 One-Time Signatures

In this section we derive the bound for signature schemes with deterministic-query verifiers and one-time salted soundness.

Theorem 5.1 (Lower bound on the soundness of short one-time signatures). *Let SIG be a ROM signature scheme with a deterministic-query verifier, completeness α , query complexity (q, q_V) , and public-key and signature lengths ℓ_{pk} and ℓ_π , respectively. Let $\delta \in (0, 1]$, and let $\gamma: \mathbb{N} \rightarrow \mathbb{N}$ be such that $\gamma(\lambda) \in [4 \cdot \lceil \log q_V \rceil + 4, \lambda - \lceil \log q_V \rceil - 2]$. Then SIG is not (t, β) -one-time salted sound, for*

- $t \leftarrow c \cdot (w^2 \binom{w}{s+1}) / (\mu \delta^2) + 2^\gamma q_V$,
- $\beta \leftarrow \alpha^3 \cdot (1 - \delta - \mu) / (c \cdot \lambda q_V)$,

where $c > 0$ is a universal constant, $s \leftarrow \lceil (16(\ell_{\text{pk}} + \ell_\pi + 1 - \log \alpha)) / \gamma \rceil + 4$, $w \leftarrow 2^{21} \cdot \lambda^2 (q + s) / \alpha^3$, and $\mu: \mathbb{N} \rightarrow [2^{-\lambda+3}, 1]$ is any function satisfying $t^2 \leq \mu(\lambda) \cdot 2^{\lambda-4}$ and $\beta \geq 2^{-\lambda}$.

That is, a scheme with good completeness cannot have a short public key and signature, and yet be salted sound. We prove Theorem 5.1 below, but first use it to derive the following corollary.

Corollary 5.2. *Let $d \in (0, 2]$ and let SIG be as in Theorem 5.1. Assume $\alpha = 1/2$, $q \in \text{poly}$, and $\ell_{\text{pk}} + \ell_\pi \in o(\lambda^d / \log \lambda)$. Then SIG is not $(2^{o(\lambda^d)}, \Omega(1/(\lambda q_V)))$ -one-time salted sound.*

Specifically, for $d = 2$, we get that a scheme with $\ell_{\text{pk}} + \ell_\pi \in o(\lambda^2 / \log \lambda)$ cannot be $(2^{o(\lambda)}, \Omega(1/(\lambda q_V)))$ -one-time salted sound (proving the one-time part of Theorem 1.1).

Proof of Corollary 5.2. We apply Theorem 5.1 with respect to $\delta = \mu = 1/4$ and $\gamma \leftarrow \max\{4\lceil \log q_V \rceil + 4, \lceil \sqrt{h \log \lambda} \rceil\}$, for $h \leftarrow 16 \cdot (\ell_{\text{all}} + 2)$ and $\ell_{\text{all}} \leftarrow \ell_{\text{pk}} + \ell_{\pi}$. Let c denote the universal constant of the theorem, and note that $s = \lceil h/\gamma \rceil + 4$ and $\beta = 1/(16c \cdot \lambda q_V)$.

Since h and q are $\text{poly}(\lambda)$, so are s and w , and thus $\log w = O(\log \lambda)$. Hence, as $h = o(\lambda^d / \log \lambda)$, it holds that

$$\log t \leq \gamma + (s + 1) \cdot \log w + O(\log \lambda) \leq O(\sqrt{h \log \lambda} + \log \lambda) = o(\lambda^{d/2}), \quad (9)$$

where the first inequality is by $\binom{w}{s+1} \leq 2^{(s+1) \cdot \log w}$, and the second by the choice of γ , which balances the two terms of the exponent (note that $(s + 1) \cdot \log w \leq h \log w / \gamma + 6 \log w$).

It remains to verify the requirements of Theorem 5.1. The condition on γ holds by its definition and since $\gamma = o(\lambda)$. As for μ , Equation (9) yields $t^2 \leq 2^{\lambda-6} = \mu \cdot 2^{\lambda-4}$. The corollary follows, as $t = 2^{o(\lambda^{d/2})}$ and $\beta = \Omega(1/(\lambda q_V))$. $\square$

Proving Theorem 5.1. In the rest of this section, we prove Theorem 5.1 via the following steps:

1. Transform SIG into a scheme SIG₁ with a non-adaptive, deterministic verifier.
2. Transform SIG₁ into a scheme SIG₂ whose deterministic-query verifier has low query complexity.
3. Transform SIG₂ into a scheme SIG₃ with a deterministic verifier.
4. Transform SIG₃ into a scheme SIG₄ with perfect completeness.
5. Prove the theorem by applying Theorem 3.21 to SIG₄.

Let SIG be as in Theorem 5.1. For the following claims we assume towards a contradiction that it has (t, β) -one-time salted soundness.

Deterministic verifier. Let SIG₁ $\leftarrow$ $\widehat{\text{SIG}}(\text{SIG}, \alpha/2)$ be as in Definition 6.2.

Claim 5.3. SIG₁ has a non-adaptive, deterministic verifier, query complexity (q, q_V) , completeness $\alpha/2$, public-key and signature lengths ℓ_{pk} and ℓ_{π} , and (t_1, β_1) -one-time salted soundness, for $t_1 \leftarrow t$ and $\beta_1 \leftarrow 2\beta/\alpha$.

Proof. The claim follows by Theorem 6.3. $\square$

Low-query verifier. Let SIG₂ $\leftarrow$ $\widehat{\text{SIG}}(\text{SIG}_1, s, \gamma)$ be as in Definition 4.4 (where s and γ are as in the theorem statement).

Claim 5.4. SIG₂ has a deterministic-query verifier, query complexity $(q + q_V + s, s)$, (α_S, α_V) -completeness for $\alpha_S \leftarrow \alpha/8$ and $\alpha_V \leftarrow 1/(32\lambda q_V)$, and (t_2, β_2) -one-time soundness, for $t_2 \leftarrow t_1 - (2^{\gamma+2} + 2) \cdot q_V$ and $\beta_2 \leftarrow 2\beta_1$.

Proof. The claim follows by Theorem 4.6, applied to SIG₁ using Claim 5.3. Its hypotheses hold by the range of γ , by $s \geq 16(\ell_{\text{pk}} + \ell_{\pi} - \log(\alpha/2))/\gamma + 4$, and by $\beta_1 \geq \beta \geq 2^{-\lambda}$. $\square$

Perfect completeness and deterministic verifier. Let $\text{SIG}_3 \leftarrow \widehat{\text{SIG}}(\text{SIG}_2, \alpha_V)$ be as in Definition 6.2, and let $\text{SIG}_4 \leftarrow \widehat{\text{SIG}}(\text{SIG}_3, \alpha_S, s+1)$ be as in Definition 6.5.

Claim 5.5. SIG_3 has a deterministic verifier, query complexity $(q + q_V + s, s)$, completeness α_S , and (t_3, β_3) -one-time soundness, for $t_3 \leftarrow t_2$ and $\beta_3 \leftarrow \beta_2/\alpha_V$.

Proof. The claim follows by Theorem 6.3 and Claim 5.4. $\square$

Claim 5.6. SIG_4 has a deterministic verifier, query complexity $(w, s+1)$, perfect completeness, and (t_4, β_4) -one-time soundness, for $t_4 \leftarrow t_3$ and $\beta_4 \leftarrow 2^{18} \cdot (\lambda q_V/\alpha^3) \cdot \beta + (4t^2 + 2) \cdot 2^{-\lambda}$.

Proof. By Theorem 6.6 and Claim 5.5, SIG_4 has a deterministic verifier, query complexity $(n \cdot ((t_G + 2) \cdot (q + q_V + s) + 1) + 2(s+1), s+1)$ for $n = \lceil 64(\lambda + 2)/\alpha \rceil$ and $t_G = \lceil 1024(\lambda + \log n + 2)/\alpha^2 \rceil$ (since $\max\{s+1, k\} = s+1$), perfect completeness, and $(t_3, 32\beta_3/\alpha_S^2 + (t_3 + 1) \cdot 2^{-\lambda} + (2t_3n + 1) \cdot 2^{-\lambda} + (t_3 + 1) \cdot 2^{-(s+1)\lambda+1})$ -one-time soundness. Note that $n \leq w \leq t \leq 2^{(\lambda-4)/2}$, and that $\log n \leq 3\lambda$ since $\alpha \geq 2^{-\lambda/3}$ by $\beta \geq 2^{-\lambda}$. Hence, the claim follows, since $n \cdot ((t_G + 2) \cdot (q + q_V + s) + 1) + 2(s+1) \leq w$, $32\beta_3/\alpha_S^2 = 2^{18} \cdot (\lambda q_V/\alpha^3) \cdot \beta$, $(t_3 + 1) + (2t_3n + 1) \leq 4tn \leq 4t^2$, and $(t_3 + 1) \cdot 2^{-(s+1)\lambda+1} \leq 2^{-\lambda}$. $\square$

Proof of Theorem 5.1. By Claim 5.6 and since $s+1 \leq w/2$, applying Theorem 3.21 to SIG_4 with parameters $(\delta/2, \mu/2)$ yields that SIG_4 is not $(t', 1 - \delta/2 - \mu/2)$ -one-time sound, for $t' \leftarrow c' \cdot w^2 \binom{w}{s+1} / ((\mu/2) \cdot (\delta/2)^2) = 8 \cdot c' \cdot w^2 \binom{w}{s+1} / (\mu\delta^2)$, where c' is a universal constant.

By taking the universal constant c used in Theorem 5.1 to be $8 \cdot c' + 2^{20}$, we have that

- $t_4 \geq t'$. Indeed, since $(2^{\gamma+2} + 2) \cdot q_V \leq 8 \cdot 2^\gamma q_V$, it holds that $t_4 = t - (2^{\gamma+2} + 2) \cdot q_V \geq c \cdot w^2 \binom{w}{s+1} / (\mu\delta^2) \geq t'$.
- $\beta_4 < 1 - \delta/2 - \mu/2$. Indeed, $\beta_4 = 2^{18} \cdot (\lambda q_V/\alpha^3) \cdot \beta + (4t^2 + 2) \cdot 2^{-\lambda}$. By the definition of β and since $c \geq 2^{20}$, the first term of the sum is at most $(2^{18}/c) \cdot (1 - \delta - \mu) \leq (1 - \delta - \mu)/4$. Since $t^2 \leq \mu \cdot 2^{\lambda-4}$ and $\mu \geq 2^{-\lambda+3}$, the second term is at most $\mu/4 + \mu/4 = \mu/2$. Hence, $\beta_4 \leq (1 - \delta - \mu)/4 + \mu/2 < 1 - \delta/2 - \mu/2$, where the last inequality holds since $\delta > 0$ and since without loss of generality $\delta + \mu \leq 1$ (otherwise the theorem trivially holds).

Thus, the attack guaranteed by Theorem 3.21 contradicts the soundness of SIG_4 . We conclude that SIG is not (t, β) -one-time salted sound. $\square$

5.2 Many-Time Signatures

In this section, we prove the improved lower bound for many-time schemes.

Theorem 5.7 (Lower bound on the soundness of short many-time signatures). *Let SIG be a ROM signature scheme with a deterministic-query verifier, completeness α , query complexity (q, q_V) , and signature length ℓ_π . Let $\delta \in (0, 1]$, and let $\gamma: \mathbb{N} \rightarrow \mathbb{N}$ be such that $\gamma(\lambda) \in [4 \cdot \lceil \log q_V \rceil + 4, \lambda - \lceil \log q_V \rceil - 2]$. Then SIG is not (t, β) -many-time salted sound, for*

- $t \leftarrow c \cdot q_V \cdot (w^2 \binom{w}{s+1} / (\mu\delta^2) + 2^\gamma)$,
- $\beta \leftarrow \alpha^3 \cdot (1 - \delta - \mu) / (c \cdot \lambda q_V)$,

where $c > 0$ is a universal constant, $s \leftarrow \lceil (16(\ell_\pi + 2 - \log \alpha))/\gamma \rceil + 4$, $n \leftarrow \left\lceil \frac{2^{12} \cdot \lambda \cdot q_V \cdot q}{\alpha} \cdot \ln \frac{2^{12} \cdot \lambda \cdot q_V \cdot q}{\alpha} \right\rceil$, $w \leftarrow 2^{27} \cdot \lambda^2((n+2) \cdot q + s)/\alpha^3$, and $\mu: \mathbb{N} \rightarrow [2^{-\lambda+3}, 1]$ is any function with $t^2 \leq \mu(\lambda) \cdot 2^{\lambda-4}$ and $\beta \geq 2^{-\lambda}$.

That is, a scheme with good completeness cannot have short signatures, regardless of its public-key length, and yet be salted sound.

Corollary 5.8. *Let $d \in (0, 2]$ and let SIG be as in Theorem 5.7. Assume $\alpha = 1/2$, $q \in \text{poly}$, and $\ell_\pi \in o(\lambda^d / \log \lambda)$. Then SIG is not $(2^{o(\lambda^{d/2})}, \Omega(1/(\lambda q_V)))$ -many-time salted sound.*

Proof. The proof is the same as that of Corollary 5.2, taking $h \leftarrow 16 \cdot (\ell_\pi + 3)$, so that $s = \lceil h/\gamma \rceil + 4$. $\square$

Specifically, for $d = 2$, we get that a scheme with $\ell_\pi \in o(\lambda^2 / \log \lambda)$ cannot be $(2^{o(\lambda)}, \Omega(1/(\lambda q_V)))$ -many-time salted sound (proving the many-time part of Theorem 1.1).

Proving Theorem 5.7. We follow the proof of Theorem 5.1, replacing the low-query transformation of Algorithm 4.3 by its many-time variant, Algorithm 4.13, whose verifier query complexity depends only on the signature length.

Proof. Assume towards a contradiction that SIG has (t, β) -many-time salted soundness. We prove the theorem via the following steps.

1. Transform SIG into a scheme SIG_1 with a non-adaptive, deterministic verifier.
2. Transform SIG_1 into a scheme SIG_2 whose deterministic-query verifier makes only s queries.
3. Transform SIG_2 into a scheme SIG_3 with a deterministic verifier.
4. Transform SIG_3 into a scheme SIG_4 with perfect completeness.
5. Apply Theorem 3.21 to SIG_4 .

Only the first two steps differ from the proof of Theorem 5.1.

In the first step, $\text{SIG}_1 \leftarrow \widehat{\text{SIG}}(\text{SIG}, \alpha/2)$ is as in Definition 6.2, and Theorem 6.3 is invoked for its many-time soundness. Hence, SIG_1 has a non-adaptive, deterministic verifier, query complexity (q, q_V) , completeness $\alpha/2$, signature length ℓ_π , and (t_1, β_1) -many-time salted soundness, for $t_1 \leftarrow t$ and $\beta_1 \leftarrow 2\beta/\alpha$.

In the second step, $\text{SIG}_2 \leftarrow \widetilde{\text{SIG}}(\text{SIG}_1, s, \gamma, n)$ is the many-time low-query scheme of Definition 4.14. Let $\tilde{\alpha} \leftarrow \alpha/(2^{10} \cdot \lambda q_V)$ be as in Theorem 4.15 with respect to the completeness $\alpha/2$ of SIG_1 , and note that, on messages of length at least λ , the hypotheses of the theorem hold by the range of γ , by the definitions of s and n , by $n \leq \tilde{\alpha} \cdot 2^{\lambda-1}$ (since $2n/\tilde{\alpha} \leq w \leq 2^{\lambda/4}$, by $t \geq w^2$ and $t^2 \leq \mu \cdot 2^{\lambda-4}$), and by $\beta_1 \geq \beta \geq 2^{-\lambda}$. Hence, SIG_2 has a deterministic-query verifier with query complexity $((n+2) \cdot q + s, s)$, (α_S, α_V) -completeness on messages of length at least λ , for $\alpha_S \leftarrow \alpha/32$ and $\alpha_V \leftarrow 1/(32\lambda q_V)$, and (t_2, β_2) -many-time soundness, for $t_2 \leftarrow t_1/(2q_V) - 2^{\gamma+2} - n$ and $\beta_2 \leftarrow 2\beta_1$. Note that here s is independent of the public-key length.

The three remaining steps are those of Theorem 5.1, applied to SIG_2 . In the third step, Theorem 6.3 is applied with respect to α_V , and in the fourth step, Theorem 6.6 is applied with respect to α_S and $k \leftarrow s + 1$ (it uses the completeness of SIG_3 only on λ -bit messages). This yields the stated w , t , and β , for a larger universal constant c . To simplify the result, we use only the one-time soundness guarantees of these theorems, which suffice, as Theorem 3.21 contradicts the one-time soundness of SIG_4 . $\square$

5.2.1 Tighter Bound Assuming Improved Barak and Mahmoody-Ghidary [BMG07]

Assuming that Barak and Mahmoody-Ghidary [BMG07]’s lower bound can be improved for many-time signatures, stated as an open question in [BMG07], we get a stronger bound on their length.

Conjecture 5.9 (Better query-complexity lower bound for many-time signatures). *Let $\eta: \mathbb{N} \rightarrow \mathbb{N}$. Let SIG be a ROM signature scheme with perfect completeness, a deterministic verifier, and query complexity (q, q_V) with $q_V \leq q/2$. Then for every $\delta, \mu \in (0, 1]$ there exists $t \leq c \cdot q^2 \cdot \binom{q}{\eta(q_V)} / (\mu\delta^2)$, for some universal constant c , such that SIG is not $(t, 1 - (\delta + \mu))$ -many-time sound.*

Namely, t is proportional to $2^{\eta(q_V) \cdot \log q}$, and not to $2^{q_V \cdot \log q}$ as in Theorem 3.21.

Theorem 5.10 (Lower bound on the soundness of short many-time signatures, under conjecture). *Assume Conjecture 5.9 holds with respect to a function η , and let SIG be as in Theorem 5.7. Then SIG is not (t, β) -many-time salted sound, for*

- $t \leftarrow c \cdot q_V \cdot \left((n's + 1) \cdot w^2 \binom{w}{\eta(s+1)} / (\mu\delta^2) + 2^\gamma \right),$
- $\beta \leftarrow \frac{\alpha^2(1-\delta-\mu)}{c \cdot \lambda q_V},$

where $c > 0$ is a sufficiently large universal constant, s, n and w are as in Theorem 5.7, $\mu: \mathbb{N} \rightarrow [2^{-\lambda+3}, 1]$ is any function with $t^2 \leq \mu(\lambda) \cdot 2^{\lambda-4}$ and $\beta \geq 2^{-\lambda}$, and $n' \leftarrow \lceil 256(\lambda + 2)/\alpha \rceil = \Theta(\lambda/\alpha)$ is the parameter n of Theorem 6.6 with respect to $\alpha_S \leftarrow \alpha/32$.

Proof. We follow the proof of Theorem 5.7, with the following differences.

1. Since Conjecture 5.9 rules out many-time soundness, the third and fourth steps use the many-time versions of Theorems 6.3 and 6.6. Hence, SIG₃ has $(t_2, \beta_2/\alpha_V)$ -many-time soundness, and SIG₄ has query complexity $(w, s+1)$ and (t_4, β_4) -many-time soundness, for $t_4 \leftarrow t_2/(n's + 1)$ and $\beta_4 \leftarrow 2^{15} \cdot (\lambda q_V/\alpha^2) \cdot \beta + (2t_2 + 2)^2 \cdot 2^{-\lambda} + (t_2 + 1)^2 \cdot 2^{-\lambda} + (t_2 + 1) \cdot 2^{-(s+1)\lambda+1}$ (since $8\beta_2/(\alpha_S \alpha_V) = 2^{15} \cdot (\lambda q_V/\alpha^2) \cdot \beta$).
2. In the last step, Conjecture 5.9 is applied to SIG₄ with parameters $(\delta/2, \mu/2)$, yielding that SIG₄ is not $(t', 1 - \delta/2 - \mu/2)$ -many-time sound, for $t' \leftarrow 8c' \cdot w^2 \binom{w}{\eta(s+1)} / (\mu\delta^2)$ and a universal constant c' .

The contradiction follows as in the proof of Theorem 5.1: $t_4 \geq (c/2 - 1) \cdot w^2 \binom{w}{\eta(s+1)} / (\mu\delta^2) \geq t'$ for $c \geq 16c' + 2$, and $\beta_4 < 1 - \delta/2 - \mu/2$. Indeed, the first term of β_4 is at most $(1 - \delta - \mu)/2$ for $c \geq 2^{16}$, and since $t_2 \leq t \leq 2^{(\lambda-4)/2}$, its remaining terms sum to less than $1/2$ for $\lambda \geq 8$. $\square$

Corollary 5.11. *Assume Conjecture 5.9 holds with respect to $\eta(x) := \lfloor \sqrt{x} \rfloor$, and let SIG be as in Theorem 5.7. Let $\alpha = 1/2$, $q \in \text{poly}$, and $\ell_\pi \in o(\lambda^3/\log^2 \lambda)$. Then SIG is not $(2^{o(\lambda)}, \Omega(1/(\lambda q_V)))$ -many-time salted sound.*

Specifically, a scheme with $\ell_\pi \in o(\lambda^3/\log^2 \lambda)$, rather than the $o(\lambda^2/\log \lambda)$ bound of the one-time case (Corollary 5.2), cannot be $(2^{o(\lambda)}, \Omega(1/(\lambda q_V)))$ -many-time salted sound; this matches, up to the $\text{poly}(\lambda, q_V)$ factors hidden in the soundness error, the length $O(\lambda^3/\log^2 \lambda)$ of the generic certification-tree construction with Winternitz certificates (see Section 1.2). That is, the conjecture above, if true with respect to $\eta(x) := \lfloor \sqrt{x} \rfloor$, would pin down the optimal many-time ROM signature length up to constant factors, rather than merely remove the $\log \lambda$ loss of Corollary 5.8.

Proof. We follow the proof of Corollary 5.8, with the following differences.

1. We set $\gamma \leftarrow \max\left\{4\lceil \log q_V \rceil + 4, \left\lceil \sqrt[3]{h \log^2 \lambda} \right\rceil\right\}$, for $h \leftarrow 16 \cdot (\ell_\pi + 3)$.
2. Since n' , q_V and w are $\text{poly}(\lambda)$ and $\left(\lfloor \frac{w}{\sqrt{s+1}} \rfloor\right) \leq 2^{\sqrt{s+1} \cdot \log w}$, it holds that $\log t \leq \gamma + \sqrt{s+1} \cdot \log w + O(\log \lambda) = O(\sqrt[3]{h \log^2 \lambda} + \log \lambda) = o(\lambda)$, where the equality is by the choice of γ , which balances the two terms of the exponent (note that $\sqrt{s+1} \cdot \log w \leq \sqrt{h/\gamma + 6} \cdot O(\log \lambda)$), and the last step is by $h = o(\lambda^3 / \log^2 \lambda)$.

The requirements of Theorem 5.10 hold as in Corollary 5.2, and the corollary follows, as $\beta = 1/(8c \cdot \lambda q_V)$. $\square$

6 Reductions Between ROM Signature Schemes

In this section, we present a sequence of transformations that, when combined, transform a ROM signature scheme with noticeable completeness and a deterministic-query verifier into one with perfect completeness and a deterministic verifier. In Section 6.1, we show how to achieve a deterministic verifier, and in Section 6.2, we show how to achieve perfect completeness, at the price of a slight increase in the soundness error.

6.1 From a Deterministic-Query Verifier to a Deterministic Verifier

Let $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$ be a ROM signature scheme with a deterministic-query verifier, and let $\alpha_V \in (0, 1]$. In the following variant $\widehat{\text{SIG}}$ of SIG , the verifier enumerates over all the coins of Verify , and accepts a signature if and only if its acceptance rate by Verify is at least α_V . Since Verify is deterministic-query, its queries are the same for all of its coins, and thus this enumeration adds no oracle queries.

Algorithm 6.1 ($\widehat{\text{SIG}} = (\widehat{\text{Gen}}, \widehat{\text{Sign}}, \widehat{\text{Verify}})$): Deterministic verifier via thresholding

Underlying scheme: $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$.

Parameters: $1^\lambda, 1^\ell, \alpha_V \in (0, 1]$.

Oracle: $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$.

Key-generator $\widehat{\text{Gen}}^f(1^\lambda, 1^\ell)$: Same as $\text{Gen}^f(1^\lambda, 1^\ell)$.

Signing $\widehat{\text{Sign}}_{\text{sk}}^f(m)$: Same as $\text{Sign}_{\text{sk}}^f(m)$.

Verification $\widehat{\text{Verify}}_{\text{pk}}^f(m, \pi)$: Accept if $\Pr[\text{Verify}_{\text{pk}}^f(m, \pi) = 1] \geq \alpha_V$, and reject otherwise.

Namely, $\widehat{\text{SIG}}$ keeps the keys and the signatures of SIG , and only replaces its verifier by a deterministic one. For fixed (SIG, α_V) , Algorithm 6.1 yields the following signature scheme.

Definition 6.2. For a signature scheme SIG and $\alpha_V: \mathbb{N} \rightarrow (0, 1]$, let $\widehat{\text{SIG}}(\text{SIG}, \alpha_V)$ be the scheme that on parameters $1^\lambda, 1^\ell$, acts according to Algorithm 6.1 with parameters $(\text{SIG}, 1^\lambda, 1^\ell, \alpha_V(\lambda))$.

Theorem 6.3 (Security of Algorithm 6.1). *Let SIG be a ROM signature scheme with a deterministic-query verifier, query complexity (q, q_V) , public-key length ℓ_{pk} and signature length ℓ_π . Let $\alpha_S, \alpha_V: \mathbb{N} \rightarrow (0, 1]$, and let $\widehat{\text{SIG}} := \widehat{\text{SIG}}(\text{SIG}, \alpha_V)$. Then $\widehat{\text{SIG}}$ has a non-adaptive, deterministic verifier, query complexity (q, q_V) , public-key length ℓ_{pk} and signature length ℓ_π , and the following completeness and soundness properties.*

Completeness. *Assume SIG has (α_S, α_V) -completeness (Definition 3.12). Then $\widehat{\text{SIG}}$ has completeness α_S . In particular, if SIG has completeness α , then $\widehat{\text{SIG}}(\text{SIG}, \alpha/2)$ has completeness $\alpha/2$.*

Soundness. *Assume SIG has (t, β) -one-time [resp., many-time] soundness. Then $\widehat{\text{SIG}}$ has $(t, \beta/\alpha_V)$ -one-time [resp., many-time] soundness. The same holds for salted soundness.*

Proof. Since Verify is deterministic-query, the queries of $\text{Verify}_{\text{pk}}^f(m, \pi)$ are determined by (pk, m, π) , and are the same for all of its coins. Hence, enumerating the coins adds no query, and $\widehat{\text{Verify}}$ is a non-adaptive, deterministic verifier making at most q_V queries. The keys, the signatures and the queries of $\widehat{\text{Gen}}$ and $\widehat{\text{Sign}}$ are those of SIG . As for completeness, (α_S, α_V) -completeness means that for every message m , with probability at least α_S over f and $(\text{sk}, \text{pk}) \xleftarrow{R} \text{Gen}^f(1^\lambda, 1^\ell)$, the acceptance rate of $\text{Sign}_{\text{sk}}^f(m)$ by $\text{Verify}_{\text{pk}}^f$ is at least α_V , in which case $\widehat{\text{Verify}}$ accepts it. The second part follows, as by Proposition 3.14 completeness α implies $(\alpha/2, \alpha/2)$ -completeness. As for soundness, let $\widehat{A}$ be an admissible adversary for the one-time [resp., many-time] soundness of $\widehat{\text{SIG}}$ that makes t queries and forges with probability $\widehat{\beta}$. Let A be the adversary for SIG that emulates $\widehat{A}$, answering its oracle and signing queries using its own oracles. Then A is admissible and makes t queries, and since a forgery accepted by $\widehat{\text{Verify}}$ is accepted by Verify with probability at least α_V , it forges with probability at least $\widehat{\beta} \cdot \alpha_V$. The same holds for salted soundness, with A forwarding also the $\text{SRO}_\lambda.\text{Sample}$ and $\text{SRO}_\lambda.\text{Rewind}$ calls of $\widehat{A}$. $\square$

6.2 Achieving Perfect Completeness

Let $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$ be a ROM signature scheme with a deterministic verifier and completeness α . The modified scheme $\widehat{\text{SIG}}$ uses SIG with message length λ . Its key generator resamples the keys until Verify accepts the signatures of most messages, and in addition samples a random trapdoor and publishes its image under the oracle. As Verify is deterministic, the signer can test its candidate signatures, and it resamples the signed message until Verify accepts. If all its attempts fail, it exposes the trapdoor, which the verifier accepts unconditionally. Forging via the trapdoor thus requires inverting the random oracle. For $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ and $u \in \{0, 1\}^\lambda$, let $f_0(x) := f(0, (0)_\lambda, x)$, $f_1(x) := f(0, (1)_\lambda, x)$, and $f_u(x) := f(1, u, x)$, for $(i)_\lambda$ being the λ -bit binary representation of i . In particular, the oracles f_0 , f_1 and f_u , used below, are evaluated on disjoint inputs.

Algorithm 6.4 ($\widehat{\text{SIG}} = (\widehat{\text{Gen}}, \widehat{\text{Sign}}, \widehat{\text{Verify}})$): Perfect completeness)

Underlying scheme: $\text{SIG} = (\text{Gen}, \text{Sign}, \text{Verify})$.

Parameters: $1^\lambda, 1^\ell, \alpha \in (0, 1], k \in \mathbb{N}$. Let $n \leftarrow \lceil 8(\lambda + 2)/\alpha \rceil$ and $t_G \leftarrow \lceil 16(\lambda + \log n + 2)/\alpha^2 \rceil$.

Oracle: $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$.

Key-generator $\widehat{\text{Gen}}^f(1^\lambda, 1^\ell)$:

1. Sample $z \xleftarrow{R} \{0, 1\}^{k\lambda}$ and let $y \leftarrow (f_1(1, z), \dots, f_1(k, z))$.
2. For $j = 1$ to n :
 - (a) Sample $u \xleftarrow{R} \{0, 1\}^\lambda$ distinct from previous values.
 - (b) Let $(\text{sk}, \text{pk}) \xleftarrow{R} \text{Gen}^{f_u}(1^\lambda, 1^\lambda)$.
 - (c) Sample $m_1, \dots, m_{t_G} \xleftarrow{R} \{0, 1\}^\lambda$.
 - (d) For every $i \in [t_G]$: let $\pi_i \leftarrow \text{Sign}_{\text{sk}}^{f_u}(m_i)$.
 - (e) If $\text{Verify}_{\text{pk}}^{f_u}(m_i, \pi_i) = 1$ for at least an $\alpha/2$ fraction of the $i \in [t_G]$, output $(\widehat{\text{sk}} \leftarrow (u, \text{sk}, \text{pk}, z), \widehat{\text{pk}} \leftarrow (u, \text{pk}, y))$.
3. Output $(\widehat{\text{sk}} \leftarrow (u, \text{sk}, \text{pk}, z), \widehat{\text{pk}} \leftarrow (u, \text{pk}, y))$, for the last sampled u and (sk, pk) .

Signing $\widehat{\text{Sign}}_{\widehat{\text{sk}}=(u, \text{sk}, \text{pk}, z)}^f(m)$:

1. For $i = 1$ to n :
 - (a) Sample $r \xleftarrow{R} \{0, 1\}^\lambda$ distinct from previous values.
 - (b) Let $m' \leftarrow f_0(m, r)$ and $\pi \leftarrow \text{Sign}_{\text{sk}}^{f_u}(m')$.
 - (c) If $\text{Verify}_{\text{pk}}^{f_u}(m', \pi) = 1$, output $\widehat{\pi} \leftarrow (0, r, \pi)$.
2. Output $\widehat{\pi} \leftarrow (1, z)$.

Verification $\widehat{\text{Verify}}_{\widehat{\text{pk}}=(u, \text{pk}, y)}^f(m, \widehat{\pi})$:

1. If $\widehat{\pi} = (1, z')$: accept if $(f_1(1, z'), \dots, f_1(k, z')) = y$.
2. If $\widehat{\pi} = (0, r, \pi)$: accept if $\text{Verify}_{\text{pk}}^{f_u}(f_0(m, r), \pi) = 1$.

That is, the scheme adds to the secret key a trapdoor that trades completeness error for soundness error. The new signer is randomized, but by Remark 3.8 it can be transformed to be deterministic with no significant cost. The adaptivity of the new verifier increases by one, since $\widehat{\text{Verify}}$ determines its queries to f_u only after learning $m' \leftarrow f_0(m, r)$. Yet, by appending m' to the signature and having $\widehat{\text{Verify}}$ check that $f_0(m, r) = m'$, it can be transformed to maintain the same adaptivity as the original verifier, at the price of λ additional bits (see Remark 3.20). For fixed (SIG, α, k) , Algorithm 6.4 yields the following signature scheme.

Definition 6.5. For a signature scheme SIG , $\alpha: \mathbb{N} \rightarrow (0, 1]$ and $k: \mathbb{N} \rightarrow \mathbb{N}$, let $\widehat{\text{SIG}}(\text{SIG}, \alpha, k)$ be the scheme that on parameters $1^\lambda, 1^\ell$, acts according to Algorithm 6.4 with parameters $(\text{SIG}, 1^\lambda, 1^\ell, \alpha(\lambda), k(\lambda))$.

Theorem 6.6 (Security of Algorithm 6.4). Let SIG be a ROM signature scheme with a deterministic verifier and query complexity (q, q_V) . Let $\alpha: \mathbb{N} \rightarrow (0, 1]$ and $k: \mathbb{N} \rightarrow \mathbb{N}$, let $n \leftarrow \lceil 8(\lambda + 2)/\alpha \rceil$ and $t_G \leftarrow \lceil 16(\lambda + \log n + 2)/\alpha^2 \rceil$, and let $\widehat{\text{SIG}} := \widehat{\text{SIG}}(\text{SIG}, \alpha, k)$. Then $\widehat{\text{SIG}}$ has a deterministic verifier with query complexity $(n \cdot ((t_G + 2) \cdot q + 1) + k + \max\{q_V + 1, k\}, \max\{q_V + 1, k\})$, and the following completeness and soundness properties.

Completeness. Assume that $n \leq 2^\lambda$. Then $\widehat{\text{SIG}}$ has perfect completeness.

One-time soundness. Assume SIG has completeness α and (t, β) -one-time soundness, and that $n \leq 2^\lambda$. Then $\widehat{\text{SIG}}$ has $(\hat{t}, \hat{\beta})$ -one-time soundness for $\hat{t} = t$ and $\hat{\beta} = 32\beta/\alpha^2 + (t+1) \cdot 2^{-\lambda} + (2tn+1) \cdot 2^{-\lambda} + (t+1) \cdot 2^{-k\lambda+1}$.

Many-time soundness. Assume SIG has completeness α and (t, β) -many-time soundness, and that $n \leq 2^\lambda$. Then $\widehat{\text{SIG}}$ has $(\hat{t}, \hat{\beta})$ -many-time soundness for $\hat{t} = t/(n \cdot q_V + 1)$ and $\hat{\beta} = 8\beta/\alpha + (2t+2)^2 \cdot 2^{-\lambda} + (t+1)^2 \cdot 2^{-\lambda} + (t+1) \cdot 2^{-k\lambda+1}$.

Proof of Theorem 6.6. First, note that $\widehat{\text{Verify}}$ is deterministic (since Verify is) with query complexity $\max\{q_V + 1, k\}$. As for the overall complexity, $\widehat{\text{Gen}}$ makes the k queries $f_1(i, z)$ in Step 1, and $\widehat{\text{Gen}}$ and each signing iteration of Step 2 make at most q queries, so $\widehat{\text{Gen}}$ makes at most $n \cdot (1 + t_G) \cdot q + k$ queries. In each of $\widehat{\text{Sign}}$'s iterations in Step 1, it invokes Sign and Verify and makes a single query to f_0 , and it therefore makes at most $n \cdot (q + 1)$ queries. Completeness is proven in Claim 6.7, and the one-time and many-time soundness are proven in Claims 6.8 and 6.15. $\square$

6.2.1 Completeness

Claim 6.7. Assume that for $n \leftarrow \lceil 8(\lambda + 2)/\alpha \rceil$ it holds that $n \leq 2^\lambda$. Then $\widehat{\text{SIG}}$ has perfect completeness.

Proof. By construction. If (the deterministic) Verify rejects all the signatures produced in Step 1, $\widehat{\text{Sign}}$ exposes the trapdoor in Step 2, which $\widehat{\text{Verify}}$ accepts. $\square$

6.2.2 One-Time Soundness

Claim 6.8. Assume SIG has completeness α and (t, β) -one-time soundness, and that for $n \leftarrow \lceil 8(\lambda + 2)/\alpha \rceil$ it holds that $n \leq 2^\lambda$. Then $\widehat{\text{SIG}}$ has $(\hat{t}, \hat{\beta})$ -one-time soundness for $\hat{t} = t$ and $\hat{\beta} = 32\beta/\alpha^2 + (t+1) \cdot 2^{-\lambda} + (2tn+1) \cdot 2^{-\lambda} + (t+1) \cdot 2^{-k\lambda+1}$.

To prove Claim 6.8, we use the following fact.

Proposition 6.9 (Rejection sampling). Let P be a distribution over $\mathcal{U}$, let $W: \mathcal{U} \rightarrow [0, 1]$, and let Q be the distribution induced by the following process:

1. $x \xleftarrow{R} P$.
2. Output x with probability $W(x)$. Otherwise, go back to Step 1.

Assume $\exists \mathcal{S} \subseteq \mathcal{U}$ such that

1. $\Pr_{x \leftarrow P}[x \in \mathcal{S}] \geq p$, and
2. $\forall x \in \mathcal{S}: W(x) \geq w$.

Then $\forall x \in \mathcal{U}: \Pr_Q[x] \leq \Pr_P[x]/(pw)$.

Proof. Let E be the event that a single iteration of the process outputs a value. By the stated conditions, $\Pr[E] \geq pw$. Hence, for all $x \in \mathcal{U}$:

$$\Pr_Q[x] / \Pr_P[x] = \sum_{i=1}^{\infty} (1 - \Pr[E])^{i-1} \cdot W(x) \leq \sum_{i=1}^{\infty} (1 - pw)^{i-1} = 1/(pw).$$

□

Proof of Claim 6.8. Let $\hat{\mathbf{A}}$ be an admissible adversary for the one-time soundness of $\widehat{\text{SIG}}$, making $\hat{t}$ queries and forging with probability $\hat{\beta}$. We use $\hat{\mathbf{A}}$ to build the adversary $\mathbf{A}$ of Algorithm 6.10 for the one-time soundness of SIG .

Algorithm 6.10 ($\mathbf{A}$: adversary for the one-time soundness of SIG)

Oracles: $f, \text{Sign}_{\text{sk}}^f$.

Input: Public key pk .

Operation:

1. Sample $u \xleftarrow{\mathcal{R}} \{0, 1\}^\lambda$ and $y \xleftarrow{\mathcal{R}} \{0, 1\}^{k\lambda}$.
2. Emulate $\hat{\mathbf{A}}$ on the key (u, pk, y) , answering its queries as follows:

Oracle queries. Answer calls to f_u using f , and other calls by fresh randomness (consistently with previous answers).

Signing queries. On the (single) query m to $\widehat{\text{Sign}}$:

 - (a) Sample $r \xleftarrow{\mathcal{R}} \{0, 1\}^\lambda$, let $m' \leftarrow f_0(m, r)$ and $\pi \leftarrow \text{Sign}_{\text{sk}}^f(m')$.
 - (b) Return $(0, r, \pi)$.
3. Let $(m^*, \hat{\pi}^*)$ be the output of $\hat{\mathbf{A}}$. If $\hat{\pi}^* = (0, r^*, \pi^*)$, output $(f_0(m^*, r^*), \pi^*)$.

To argue about the success probability of $\mathbf{A}$, we compare the output distributions of $\widehat{\text{Gen}}$ and $\widehat{\text{Sign}}$ to those of Gen and of Sign on a uniform message, and bound the probability that $\widehat{\text{Sign}}$ exposes the trapdoor. For a function $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ and a key pair $(\text{sk}, \text{pk}) \in \text{Supp}(\text{Gen}^f(1^\lambda, 1^\lambda))$, let $g^f(\text{sk}, \text{pk}) := \Pr_{m \xleftarrow{\mathcal{R}} \{0, 1\}^\lambda} [\text{Verify}_{\text{pk}}^f(m, \text{Sign}_{\text{sk}}^f(m)) = 1]$ denote the fraction of messages whose signature is accepted. Let

$$\hat{\mathcal{G}} := \{(f, \hat{\text{sk}}, \hat{\text{pk}}) : f \in \mathcal{F}_\lambda, (\hat{\text{sk}} = (u, \text{sk}, \cdot, \cdot), \hat{\text{pk}} = (\cdot, \text{pk}, \cdot)) \in \text{Supp}(\widehat{\text{Gen}}^f(1^\lambda, 1^\ell)), g^{f_u}(\text{sk}, \text{pk}) \geq \alpha/4\}.$$

Since each iteration of Step 2 outputs a key with probability at least $\alpha/8$, a Chernoff bound yields that

$$\Pr_{f \xleftarrow{\mathcal{R}} \mathcal{F}_\lambda, (\hat{\text{sk}}=(u, \text{sk}, \cdot, \cdot), \hat{\text{pk}}=(\cdot, \text{pk}, \cdot)) \xleftarrow{\mathcal{R}} \widehat{\text{Gen}}^f(1^\lambda, 1^\ell)} [(f, \hat{\text{sk}}, \hat{\text{pk}}) \notin \hat{\mathcal{G}}] \leq 2^{-\lambda-2} \quad (10)$$

Claim 6.11 ($\widehat{\text{Gen}}$'s output distribution). *For every $(f, \hat{\text{sk}} = (u, \text{sk}, \cdot, \cdot), \hat{\text{pk}} = (\cdot, \text{pk}, \cdot)) \in \hat{\mathcal{G}}$.*¹²

$$\frac{\Pr_{f' \xleftarrow{\mathcal{R}} \mathcal{F}_\lambda} [f' = f_u \wedge (\text{sk}', \text{pk}') = (\text{sk}, \text{pk})]}{(\text{sk}', \text{pk}') \xleftarrow{\mathcal{R}} \text{Gen}^{f'}(1^\lambda, 1^\lambda)} \geq \frac{\alpha}{8} \cdot \frac{\Pr_{f' \xleftarrow{\mathcal{R}} \mathcal{F}_\lambda} [f'_{u'} = f_u \wedge (\text{sk}', \text{pk}') = (\text{sk}, \text{pk})]}{((u', \text{sk}', \cdot, \cdot), (\cdot, \text{pk}', \cdot)) \xleftarrow{\mathcal{R}} \widehat{\text{Gen}}^{f'}(1^\lambda, 1^\ell)}$$

¹²Here and below, since $\mathcal{F}_\lambda$ is infinite, the probability of an outcome that contains a function refers to its density with respect to the uniform distribution over $\mathcal{F}_\lambda$.

Proof. Let P be the distribution of $(f', \text{sk}', \text{pk}')$, for $f' \xleftarrow{R} \mathcal{F}_\lambda$ and $(\text{sk}', \text{pk}') \xleftarrow{R} \text{Gen}^{f'}(1^\lambda, 1^\lambda)$, and let W map such a triple to the probability that it passes the test of Step 2e. Since the values of u sampled in Step 2 are distinct, the functions f'_u of the different iterations are independent, and the iterations are i.i.d. samples of P . Let $\mathcal{S} \leftarrow \left\{ (f', \text{sk}', \text{pk}') : g^{f'}(\text{sk}', \text{pk}') \geq 3\alpha/4 \right\}$. By the completeness of $\widehat{\text{SIG}}$, it holds that $\mathbb{E}_P[g] \geq \alpha$, and since $g \leq 1$, $\Pr_P[\mathcal{S}] \geq \alpha/4$. Moreover, the test threshold being $\alpha/2$, a Chernoff bound over the t_G messages yields $W \geq 1/2$ on $\mathcal{S}$. Hence, each iteration outputs a key with probability $e \geq \alpha/8$, and for $x \leftarrow (f_u, \text{sk}, \text{pk})$, the probability that $\widehat{\text{Gen}}^{f'}$ outputs a key with $(f'_{u'}, \text{sk}', \text{pk}') = x$ is

$$\Pr_P[x] \cdot \left(W(x) \cdot \sum_{i=1}^n (1-e)^{i-1} + (1-W(x)) \cdot (1-e)^n \right) \leq \Pr_P[x] \cdot \frac{W(x) + (1-W(x))}{e} \leq \frac{8}{\alpha} \cdot \Pr_P[x],$$

the second summand accounting for Step 3. $\square$

Claim 6.12 ($\widehat{\text{Sign}}$'s output distribution). *For every $(f, \widehat{\text{sk}} = (u, \text{sk}, \cdot, \cdot), \widehat{\text{pk}}) \in \widehat{\mathcal{G}}$, $m \in \{0, 1\}^\ell$, and $(m', \pi) \in \{0, 1\}^\lambda \times \{0, 1\}^*$:*

$$\Pr_{\substack{f' \xleftarrow{R} \mathcal{F}_\lambda : f'_u = f_u \\ \widehat{\pi}' \xleftarrow{R} \widehat{\text{Sign}}_{\widehat{\text{sk}}}^{f'}(m)}} \left[\widehat{\pi}' = (0, r, \pi') \wedge (f'_0(m, r), \pi') = (m', \pi) \right] \leq \frac{4}{\alpha} \cdot \Pr_{m'' \xleftarrow{R} \{0, 1\}^\lambda} \left[(m'', \text{Sign}_{\text{sk}}^{f_u}(m'')) = (m', \pi) \right].$$

Proof. Apply Proposition 6.9 with respect to the distribution P of $(m'', \text{Sign}_{\text{sk}}^{f_u}(m''))$ for $m'' \xleftarrow{R} \{0, 1\}^\lambda$, and W that maps such a pair to the indicator of the test of Step 1c. Since the values of r sampled in Step 1 are distinct and f'_0 is a uniform function, the messages $f'_0(m, r)$ of the different iterations are i.i.d. uniform, so the iterations are i.i.d. samples of P , and Q is the distribution of the pair $(f'_0(m, r), \pi')$ for the output $(0, r, \pi')$ of $\widehat{\text{Sign}}_{\widehat{\text{sk}}}^{f'}(m)$, given it is not $(1, z)$. Taking $\mathcal{S}$ to be the pairs passing the test, the assumption $(f, \widehat{\text{sk}}, \widehat{\text{pk}}) \in \widehat{\mathcal{G}}$ yields $\Pr_P[\mathcal{S}] = g^{f_u}(\text{sk}, \text{pk}) \geq \alpha/4$ and $W \equiv 1$ on $\mathcal{S}$. $\square$

Claim 6.13 (Trapdoor exposure). *For every $(f, \widehat{\text{sk}} = (u, \text{sk}, \cdot, z), \widehat{\text{pk}}) \in \widehat{\mathcal{G}}$ and $m \in \{0, 1\}^\ell$:*

$$\Pr_{\substack{f' \xleftarrow{R} \mathcal{F}_\lambda : f'_u = f_u \\ \widehat{\pi}' \xleftarrow{R} \widehat{\text{Sign}}_{\widehat{\text{sk}}}^{f'}(m)}} \left[\widehat{\pi}' = (1, z) \right] \leq (1 - \alpha/4)^n \leq 2^{-2\lambda-4}.$$

Proof. Since the values of r sampled in Step 1 are distinct and f'_0 is a uniform function, the messages m' sampled in Step 1b are i.i.d. uniform over $\{0, 1\}^\lambda$, so the iterations are independent and each of them outputs a signature with probability $g^{f_u}(\text{sk}, \text{pk}) \geq \alpha/4$. The second inequality holds by the choice of n . $\square$

Claim 6.14 (Hitting hidden values). *Let $\widehat{\mathbf{A}}$ be a $\hat{t}$ -query adversary attacking $\widehat{\text{SIG}}$. Then in its attack, the following hold.*

1. $\widehat{\mathbf{A}}$ makes a query that $\widehat{\text{Gen}}$ made, other than to f_u and f_1 , with probability at most $\hat{t}n \cdot 2^{-\lambda}$.
2. $\widehat{\mathbf{A}}$ makes a query that $\widehat{\text{Sign}}$ made when answering a given signing query m , other than to f_u and, after receiving the answer $(0, r, \pi)$, the query $f_0(m, r)$, with probability at most $\hat{t}n \cdot 2^{-\lambda}$.

3. $\widehat{\mathbf{A}}$ queries $f_1(i, z')$ for some $i \in [k]$, or outputs $(m^*, (1, z'))$, for some z' with $(f_1(1, z'), \dots, f_1(k, z')) = y$, while no signing query is answered by $(1, z)$, with probability at most $(\widehat{t} + 1) \cdot 2^{-k\lambda+1}$.

Proof. The queries of the first two items are to $f_{u'}$ and $f_0(m, r')$ for u' and r' sampled in Step 2 of $\widehat{\mathbf{Gen}}$ and Step 1 of $\widehat{\mathbf{Sign}}$, respectively. Since these values are uniform and hidden from $\widehat{\mathbf{A}}$, and there are at most n of each, the bounds follow. As for the third item, each of the at most $\widehat{t} + 1$ values z' that $\widehat{\mathbf{A}}$ queries or outputs equals z , or satisfies $(f_1(1, z'), \dots, f_1(k, z')) = y$, with probability at most $2^{-k\lambda+1}$. $\square$

Back to the proof. Let m be the signing query of $\widehat{\mathbf{A}}$ and $\widehat{\pi}$ the answer it received. Assume first that none of the events of Claim 6.14 occurs.¹³ In particular, y and r are uniform and independent of the rest of the view of $\widehat{\mathbf{A}}$, and if $\widehat{\mathbf{A}}$ forges, its output is of the form $(m^*, (0, r^*, \pi^*))$. We say that $\widehat{\mathbf{A}}$ *strongly forges* a signature if $\widehat{\pi} = (0, r, \pi)$ and $\widehat{\mathbf{A}}$ outputs $(m^*, (0, r^*, \pi^*))$ with $f_0(m^*, r^*) \neq f_0(m, r)$. By Equation (10), the probability that in the attack of $\widehat{\mathbf{A}}$ on $\widehat{\mathbf{SIG}}$ the generated keys $(f, \widehat{\mathbf{sk}}, \widehat{\mathbf{pk}}) \notin \widehat{\mathcal{G}}$ is at most $2^{-\lambda-2}$. By Claim 6.13, for $(f, \widehat{\mathbf{sk}}, \widehat{\mathbf{pk}}) \in \widehat{\mathcal{G}}$, the probability that $\widehat{\pi} = (1, z)$ is at most $2^{-2\lambda-4}$. Since f_0 is a random function, the probability that $\widehat{\pi} = (0, r, \pi)$ and $f_0(m^*, r^*) = f_0(m, r)$ is at most $(\widehat{t} + 1) \cdot 2^{-\lambda}$. Hence,

$$\Pr[(f, \widehat{\mathbf{sk}}, \widehat{\mathbf{pk}}) \in \widehat{\mathcal{G}} \wedge \widehat{\mathbf{A}} \text{ strongly forges}] \geq \widehat{\beta}',$$

for $\widehat{\beta}' \leftarrow \widehat{\beta} - (\widehat{t} + 1) \cdot 2^{-\lambda} - 2^{-\lambda-2} - 2^{-2\lambda-4}$. Assume for ease of notation that $\widehat{\mathbf{A}}$ is deterministic. By the above assumption, the view of $\widehat{\mathbf{A}}$ is determined by the *transcript* $\tau = (u, f_u, \mathbf{sk}, \mathbf{pk}, y, r, f_0(m, r))$ and by uniform answers to its remaining queries, both in its attack on $\widehat{\mathbf{SIG}}$ and in its emulation by $\mathbf{A}$. By construction, if τ describes a successful strong forgery attack of $\widehat{\mathbf{A}}$ on $\widehat{\mathbf{SIG}}$, then it describes a successful forgery attack of $\mathbf{A}$ on $\mathbf{SIG}$, and by Claims 6.11 and 6.12, if $(f, \widehat{\mathbf{sk}}, \widehat{\mathbf{pk}}) \in \widehat{\mathcal{G}}$, then τ occurs in the attack of $\mathbf{A}$ with probability at least $\frac{\alpha}{8} \cdot \frac{\alpha}{4}$ times its probability in the attack of $\widehat{\mathbf{A}}$. Moreover, since $\widehat{\mathbf{Verify}}$ accepts the output of $\widehat{\mathbf{A}}$, $\mathbf{Verify}$ accepts the pair output by $\mathbf{A}$. We conclude that $\mathbf{A}$ forges with probability at least $\widehat{\beta}' \cdot \frac{\alpha}{8} \cdot \frac{\alpha}{4}$. Finally, $\mathbf{A}$ makes at most $\widehat{t}$ oracle queries and a single signing query, and by the definition of strong forgery its output message differs from the message it signed, so it is admissible. Hence, the assumed one-time soundness of $\mathbf{SIG}$ yields $\widehat{\beta}' \cdot \frac{\alpha^2}{32} \leq \beta$, and thus $\widehat{\beta} \leq 32\beta/\alpha^2 + (\widehat{t} + 1) \cdot 2^{-\lambda} + 2^{-\lambda-2} + 2^{-2\lambda-4}$. It is left to remove the assumption that none of the events of Claim 6.14 occurs, which by that claim fails with probability at most $2\widehat{t}n \cdot 2^{-\lambda} + (\widehat{t} + 1) \cdot 2^{-k\lambda+1}$. Adding this term and using $\widehat{t} = t$ yields $\widehat{\beta} \leq 32\beta/\alpha^2 + (t + 1) \cdot 2^{-\lambda} + (2tn + 1) \cdot 2^{-\lambda} + (t + 1) \cdot 2^{-k\lambda+1}$, proving the claim. $\square$

6.2.3 Many-Time Soundness

Claim 6.15. Assume $\mathbf{SIG}$ has completeness α and (t, β) -many-time soundness, and that for $n \leftarrow \lceil 8(\lambda + 2)/\alpha \rceil$ it holds that $n \leq 2^\lambda$. Then $\widehat{\mathbf{SIG}}$ has $(\widehat{t}, \widehat{\beta})$ -many-time soundness for $\widehat{t} = t/(n \cdot q_V + 1)$ and $\widehat{\beta} = 8\beta/\alpha + (2t + 2)^2 \cdot 2^{-\lambda} + (t + 1)^2 \cdot 2^{-\lambda} + (t + 1) \cdot 2^{-k\lambda+1}$.

Proof. Let $\widehat{\mathbf{A}}$ be an admissible adversary for the many-time soundness of $\widehat{\mathbf{SIG}}$, making $\widehat{t}$ queries and forging with probability $\widehat{\beta}$. We use $\widehat{\mathbf{A}}$ to build the adversary $\mathbf{A}$ of Algorithm 6.16 for the many-time soundness of $\mathbf{SIG}$.

¹³Formally, $\widehat{\mathbf{Gen}}$ and $\widehat{\mathbf{Sign}}$ use an oracle independent of that of $\widehat{\mathbf{A}}$, except for f_u and, after the signing query, $f_0(m, r)$. This changes the attack only if one of these events occurs.

Algorithm 6.16 (A: adversary for the many-time soundness of SIG)

Oracles: $f, \text{Sign}_{\text{sk}}^f$.

Input: Public key pk .

Operation:

1. Sample $u \xleftarrow{\text{R}} \{0, 1\}^\lambda$ and $y \xleftarrow{\text{R}} \{0, 1\}^{k\lambda}$.
2. Emulate $\widehat{\text{A}}$ on the key (u, pk, y) , answering its queries as follows:

Oracle queries. Answer calls to f_u using f , and other calls by fresh randomness (consistently with previous answers).

Signing queries. On a query m to $\widehat{\text{Sign}}$:

 - (a) For $i = 1$ to n :
 - i. Sample $r \xleftarrow{\text{R}} \{0, 1\}^\lambda$ distinct from previous values.
 - ii. Let $m' \leftarrow f_0(m, r)$ and $\pi \leftarrow \text{Sign}_{\text{sk}}^f(m')$.
 - iii. If $\text{Verify}_{\text{pk}}^f(m', \pi) = 1$, return $\widehat{\pi} \leftarrow (0, r, \pi)$.
 - (b) Abort.
3. Let $(m^*, \widehat{\pi}^*)$ be the output of $\widehat{\text{A}}$. If $\widehat{\pi}^* = (0, r^*, \pi^*)$, output $(f_0(m^*, r^*), \pi^*)$.

That is, A acts as in the one-time case, but in the signing phase it *perfectly* mimics the signing oracle $\widehat{\text{Sign}}$, except that it aborts where $\widehat{\text{Sign}}$ exposes the trapdoor. In addition to $\widehat{\text{A}}$'s $\widehat{t}$ queries, A makes q_V queries for the test of Step 1c in each of the n iterations it performs per signing query of $\widehat{\text{A}}$, and it thus makes at most $\widehat{t} \cdot (n \cdot q_V + 1) \leq t$ queries. Using the argument of the one-time case, without paying for the difference in the signer's output distribution, we deduce that A forges with probability at least $\widehat{\beta}' \cdot \frac{\alpha}{8}$, for $\widehat{\beta}' \leftarrow \widehat{\beta} - (2\widehat{t} \cdot n + 2)^2 \cdot 2^{-\lambda} - \widehat{t}n \cdot 2^{-\lambda} - \widehat{t}^2 n \cdot 2^{-\lambda} - (\widehat{t} + 1) \cdot 2^{-k\lambda+1} - 2^{-\lambda-2} - \widehat{t} \cdot 2^{-2\lambda-4}$. The second term bounds the probability that $\widehat{\text{Sign}}$ evaluates f_0 twice on the same input, or that two of the f_0 evaluations in the attack collide, which guarantees that A is admissible, the third to fifth terms bound the probability of the events of Claim 6.14, the second one for each of the $\widehat{t}$ signing queries, the sixth term bounds, by Equation (10), the probability that $(f, \widehat{\text{sk}}, \widehat{\text{pk}}) \notin \widehat{\mathcal{G}}$, and the last term bounds, by Claim 6.13, the probability that $\widehat{\text{Sign}}$ exposes the trapdoor on one of the $\widehat{t}$ signing queries. Hence, the assumed many-time soundness of SIG yields $\widehat{\beta}' \cdot \frac{\alpha}{8} \leq \beta$, and the claim follows as $\widehat{t} \cdot n \leq t$ and $\widehat{t} \leq t$. $\square$

Acknowledgments

Alon Chayet and Iftach Haitner are supported by ISF grant 836/23. Eylon Yogev is supported by the Israel Science Foundation (Grant No. 2302/22), and the European Union (ERC, CRYPTO-PROOF, 101164375). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

- [Bab+26] Ryan Babbush, Adam Zalcman, Craig Gidney, Michael Broughton, Tanuj Khattar, Hartmut Neven, Thiago Bergamaschi, Justin Drake, and Dan Boneh. *Securing Elliptic Curve Cryptocurrencies against Quantum Vulnerabilities: Resource Estimates and Mitigations*. 2026. arXiv: [2603.28846](https://arxiv.org/abs/2603.28846) [quant-ph]. URL: <https://arxiv.org/abs/2603.28846> (cit. on p. 1).
- [BC92] Jurjen N. Bos and David Chaum. “Provably Unforgeable Signatures”. In: *Annual International Cryptology Conference (CRYPTO)*. 1992, pp. 1–14. DOI: [10.1007/3-540-48071-4_1](https://doi.org/10.1007/3-540-48071-4_1) (cit. on p. 4).
- [BDH11] Johannes Buchmann, Erik Dahmen, and Andreas Hülsing. “XMSS—A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions”. In: *Post-Quantum Cryptography*. 2011, pp. 117–129. DOI: [10.1007/978-3-642-25405-5_8](https://doi.org/10.1007/978-3-642-25405-5_8) (cit. on p. 4).
- [Bee19] Carl Beekhuizen. *ERC-2333: BLS12-381 Key Generation*. 2019. URL: <https://eips.ethereum.org/EIPS/eip-2333> (cit. on p. 1).
- [Ber+15] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O’Hearn. “SPHINCS: Practical Stateless Hash-Based Signatures”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. 2015, pp. 368–397. DOI: [10.1007/978-3-662-46800-5_15](https://doi.org/10.1007/978-3-662-46800-5_15) (cit. on p. 4).
- [Ber+19] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. “The SPHINCS⁺ Signature Framework”. In: *ACM SIGSAC Conference on Computer and Communications Security*. 2019, pp. 2129–2146. DOI: [10.1145/3319535.3363229](https://doi.org/10.1145/3319535.3363229) (cit. on p. 4).
- [BHD24] Hunter Beast, Ethan Heilman, and Isabel Foxen Duke. *BIP 360: Pay-to-Merkle-Root (P2MR)*. Draft Bitcoin Improvement Proposal. 2024. URL: <https://github.com/bitcoin/bips/blob/master/bip-0360.mediawiki> (cit. on p. 1).
- [BM94] Daniel Bleichenbacher and Ueli M. Maurer. “Directed Acyclic Graphs, One-Way Functions and Digital Signatures”. In: *Annual International Cryptology Conference (CRYPTO)*. 1994, pp. 75–82. DOI: [10.1007/3-540-48658-5_9](https://doi.org/10.1007/3-540-48658-5_9) (cit. on p. 4).
- [BM96a] Daniel Bleichenbacher and Ueli M. Maurer. “On the Efficiency of One-Time Digital Signatures”. In: *Annual International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*. 1996, pp. 145–158. DOI: [10.1007/BFb0034843](https://doi.org/10.1007/BFb0034843) (cit. on p. 4).
- [BM96b] Daniel Bleichenbacher and Ueli M. Maurer. “Optimal Tree-Based One-Time Digital Signature Schemes”. In: *Symposium on Theoretical Aspects of Computer Science*. 1996, pp. 363–374. DOI: [10.1007/3-540-60922-9_30](https://doi.org/10.1007/3-540-60922-9_30) (cit. on p. 4).
- [BMG07] Boaz Barak and Mohammad Mahmoody-Ghidary. “Lower bounds on signatures from symmetric primitives”. In: *Annual Symposium on Foundations of Computer Science (FOCS)*. 2007, pp. 680–688 (cit. on pp. 3–5, 8, 9, 15, 31).

- [Bol65] Béla Bollobás. “On Generalized Graphs”. In: *Acta Mathematica Academiae Scientiarum Hungaricae* 16.3–4 (1965), pp. 447–452. DOI: [10.1007/BF01904851](https://doi.org/10.1007/BF01904851).
- [Bos+25] Joppe W. Bos, Olivier Bronchain, Léo Ducas, Serge Fehr, Yu-Hsuan Huang, Thomas Pornin, Eamonn W. Postlethwaite, Thomas Prest, Ludo N. Pulles, and Wessel van Woerden. *HAWK*. Specification, version 1.1. 2025. URL: <https://hawk-sign.info/hawk-spec.pdf> (cit. on p. 1).
- [BTT03] Kemal Biçakçı, Gene Tsudik, and Brian Tung. “How to Construct Optimal One-Time Signatures”. In: *Computer Networks* 43.3 (2003), pp. 339–349. DOI: [10.1016/S1389-1286\(03\)00285-8](https://doi.org/10.1016/S1389-1286(03)00285-8) (cit. on p. 4).
- [Cai+26] Madelyn Cain, Qian Xu, Robbie King, Lewis R. B. Picard, Harry Levine, Manuel Endres, John Preskill, Hsin-Yuan Huang, and Dolev Bluvstein. *Shor’s Algorithm Is Possible with as Few as 10,000 Reconfigurable Atomic Qubits*. 2026. arXiv: [2603.28627](https://arxiv.org/abs/2603.28627) [quant-ph]. URL: <https://arxiv.org/abs/2603.28627> (cit. on p. 1).
- [CD23] Wouter Castryck and Thomas Decru. “An Efficient Key Recovery Attack on SIDH”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. 2023, pp. 423–447. DOI: [10.1007/978-3-031-30589-4_15](https://doi.org/10.1007/978-3-031-30589-4_15). URL: <https://eprint.iacr.org/2022/975> (cit. on p. 1).
- [Coo+20] David A. Cooper, Daniel Apon, Quynh H. Dang, Michael S. Davidson, Morris J. Dworkin, and Carl A. Miller. *Recommendation for Stateful Hash-Based Signature Schemes*. NIST Special Publication 800-208. National Institute of Standards and Technology, 2020. DOI: [10.6028/NIST.SP.800-208](https://doi.org/10.6028/NIST.SP.800-208). URL: <https://doi.org/10.6028/NIST.SP.800-208> (cit. on p. 1).
- [CY20] Alessandro Chiesa and Eylon Yogev. “Barriers for Succinct Arguments in the Random Oracle Model”. In: *Theory of Cryptography (TCC)*. 2020, pp. 47–76. DOI: [10.1007/978-3-030-64381-2_2](https://doi.org/10.1007/978-3-030-64381-2_2) (cit. on pp. 2, 6, 14).
- [DSS05] Chris Dods, Nigel P. Smart, and Martijn Stam. “Hash-Based Digital Signature Schemes”. In: *Cryptography and Coding*. 2005, pp. 96–115. DOI: [10.1007/11586821_8](https://doi.org/10.1007/11586821_8) (cit. on p. 4).
- [Eth26] Ethereum Foundation Post-Quantum Team. *Post-Quantum Ethereum*. Living roadmap, last updated June 27, 2026. 2026. URL: <https://pq.ethereum.org/> (cit. on p. 1).
- [Gho+26] Ashrujit Ghoshal, Yuval Ishai, Aayush Jain, and Nuozhou Sun. *Quasipolynomial Cryptanalysis of the McEliece Cryptosystem (or: PIR Meets McEliece)*. Cryptology ePrint Archive, Paper 2026/1630. 2026. URL: <https://eprint.iacr.org/2026/1630> (cit. on p. 1).
- [HNY22] Iftach Haitner, Daniel Nukrai, and Eylon Yogev. “Lower Bound on SNARGs in the Random Oracle Model”. In: *Annual International Cryptology Conference (CRYPTO)*. 2022, pp. 97–127. DOI: [10.1007/978-3-031-15982-4_4](https://doi.org/10.1007/978-3-031-15982-4_4) (cit. on pp. 4–6, 9, 11).
- [Hül13] Andreas Hülsing. “W-OTS⁺—Shorter Signatures for Hash-Based Signature Schemes”. In: *Progress in Cryptology—AFRICACRYPT 2013*. 2013, pp. 173–188. DOI: [10.1007/978-3-642-38553-7_10](https://doi.org/10.1007/978-3-642-38553-7_10) (cit. on p. 4).

- [KKW25] Dmitry Khovratovich, Mikhail A. Kudinov, and Benedikt Wagner. “At the Top of the Hypercube—Better Size-Time Tradeoffs for Hash-Based Signatures”. In: *Annual International Cryptology Conference (CRYPTO)*. 2025, pp. 93–123. DOI: [10.1007/978-3-032-01887-8_4](https://doi.org/10.1007/978-3-032-01887-8_4). URL: <https://eprint.iacr.org/2025/889> (cit. on p. 4).
- [Kud+23] Mikhail A. Kudinov, Andreas Hülsing, Eyal Ronen, and Eylon Yogev. “SPHINCS+C: Compressing SPHINCS+ With (Almost) No Cost”. In: *IEEE Symposium on Security and Privacy*. 2023, pp. 1435–1453. DOI: [10.1109/SP46215.2023.10179381](https://doi.org/10.1109/SP46215.2023.10179381). URL: <https://eprint.iacr.org/2022/778> (cit. on pp. 2, 4).
- [Lam79] Leslie Lamport. *Constructing digital signatures from a one way function*. Tech. rep. SRI-CSL-98. SRI International Computer Science Laboratory, 1979 (cit. on pp. 1–4).
- [MCF19] David McGrew, Michael Curcio, and Scott Fluhrer. *Leighton–Micali Hash-Based Signatures*. RFC 8554. RFC Editor, 2019. DOI: [10.17487/RFC8554](https://doi.org/10.17487/RFC8554). URL: <https://www.rfc-editor.org/rfc/rfc8554> (cit. on p. 4).
- [Mer87] Ralph C. Merkle. “A Digital Signature Based on a Conventional Encryption Function”. In: *Advances in Cryptology—CRYPTO ’87*. Springer, 1987, pp. 369–378. DOI: [10.1007/3-540-48184-2_32](https://doi.org/10.1007/3-540-48184-2_32) (cit. on pp. 2–4).
- [Nat24] National Institute of Standards and Technology. *Stateless Hash-Based Digital Signature Standard*. Federal Information Processing Standards Publication 205. National Institute of Standards and Technology, 2024. DOI: [10.6028/NIST.FIPS.205](https://doi.org/10.6028/NIST.FIPS.205). URL: <https://doi.org/10.6028/NIST.FIPS.205> (cit. on pp. 1, 4).
- [Nic26] Jonas Nick. *OP_CHECKSHRINCS: A Hash-Based Signature Opcode for Post-Quantum Bitcoin*. 2026. URL: https://blog.blockstream.com/op_checkshrincs-a-hash-based-signature-opcode-for-post-quantum-bitcoin/ (cit. on p. 1).
- [NY89] Moni Naor and Moti Yung. “Universal One-Way Hash Functions and their Cryptographic Applications”. In: *Annual ACM Symposium on Theory of Computing (STOC)*. 1989, pp. 33–43 (cit. on p. 4).
- [Per01] Adrian Perrig. “The BiBa one-time signature and broadcast authentication protocol”. In: *Proceedings of the 8th ACM conference on Computer and Communications Security*. ACM. 2001, pp. 28–37 (cit. on p. 4).
- [Rab78] Michael O. Rabin. “Digitalized Signatures”. In: *Foundations of Secure Computation*. Ed. by Richard A. DeMillo, David P. Dobkin, Anita K. Jones, and Richard J. Lipton. Academic Press, 1978, pp. 155–168. URL: <https://cir.nii.ac.jp/crid/1573105973881933824> (cit. on pp. 2, 4).
- [RR02] Leonid Reyzin and Natan Reyzin. “Better than BiBa: Short one-time signatures with fast signing and verifying”. In: *Australasian Conference on Information Security and Privacy*. 2002, pp. 144–153 (cit. on p. 4).
- [Sho94] Peter W. Shor. “Polynomial time algorithms for discrete logarithms and factoring on a quantum computer”. In: *Algorithmic Number Theory*, 1994, p. 289 (cit. on p. 1).
- [SW26] Žygimantas Straznickas and Stephen A. Weis. *HAWK- n Key Recovery Reduces to SVP in Dimension $n/2 + 1$* . 2026. URL: https://www.anthropic.com/document/hawk_key_recovery.pdf (cit. on p. 1).

- [ZCY23] Kaiyi Zhang, Hongrui Cui, and Yu Yu. “Revisiting the Constant-Sum Winternitz One-Time Signature with Applications to SPHINCS+ and XMSS”. In: *Annual International Cryptology Conference (CRYPTO)*. 2023, pp. 455–483. URL: <https://eprint.iacr.org/2023/850> (cit. on p. 4).