Strongly Unforgeable Signatures and Hierarchical Identity-based Signatures from Lattices without Random Oracles

— Extended Version —

Markus Rückert*
rueckert@cdc.informatik.tu-darmstadt.de

Cryptography and Computeralgebra
Department of Computer Science
TU Darmstadt

February 10, 2010

Abstract. We propose a variant of Peikert's lattice-based *existentially unforgeable* signature scheme in the standard model. Our construction offers the same efficiency as Peikert's but supports the stronger notion of *strong unforgeability*. Strong unforgeability demands that the adversary is unable to produce a new message-signature pair $(\mathtt{m}, \mathtt{s})$, even if he or she is allowed to see a different signature $\mathtt{s}'$ for $\mathtt{m}$.
In particular, we provide the first treeless signature scheme that supports strong unforgeability for the post-quantum era in the standard model. Moreover, we show how to *directly* implement identity-based, and even hierarchical identity-based, signatures (IBS) in the same strong security model without random oracles. An additional advantage of this direct approach over the usual generic conversion of hierarchical identity-based encryption to IBS is that we can exploit the efficiency of ideal lattices without significantly harming security.
We equip all constructions with strong security proofs based on mild worst-case assumptions on lattices and we also propose concrete security parameters.

Keywords Post-quantum cryptography, lattice cryptography, digital signatures, identity-based cryptography, standard model

* This work was supported by CASED (www.cased.de).

1 Introduction

Digital signature schemes are the cornerstone of e-business, e-government, software security, and many more applications. Their importance is likely to grow in the future as more and more everyday tasks and processes are computerized. With identity-based signature schemes (IBS), motivated by Shamir [31], one can get rid of public-key infrastructures. The public key is replaced with a unique identifier string, such as an e-mail address, and the secret key is "extracted" by a trusted party for this identifier. In hierarchical identity-based signatures (HIBS), this concept is generalized so that each party can act as a key extraction authority for its subordinates.

There are two classes of signature schemes. The first comprises *tree-based* and stateful Merkle signature schemes [26] with a limited signature capacity. Such schemes can be solely based on the security of hash functions. The drawback is that they require an inefficient key generation phase, where all signatures need to be prepared in advance. Furthermore, its statefulness poses a synchronization problem as soon as more than one computer, or process thread, is supposed to issue signatures with the same secret key.

The second class contains *treeless* constructions that are typically more efficient and allow for an unlimited number of signatures without a complex setup phase. Currently, we mainly use schemes that fall into the second category because they are easier to handle. Most of them rely on the hardness of factoring or computing discrete logarithms. So, they are neither post-quantum, nor do they resist subexponential time attacks.

Alternatives for the post-quantum era can be based on the hardness of the decoding problem in error correcting codes, on the hardness of solving nonlinear multivariate equation systems, or on the hardness of lattice problems. Refer to [7] for an overview of each field. Basically, all three alternatives rely on the hardness of certain *average-case* problems and, at first, it is unclear how to generate hard instances of these problems. More precisely, we always need to know a "hard" distribution of keys that admits efficient key generation. Unlike with multivariate or code-based cryptography, lattice-based constructions have a "trust anchor" in the form of Ajtai's worst-case to average-case reduction [2] that is not found anywhere else in cryptography. It states that solving a certain average-case problem, which is relevant in cryptography, implies a solution to a related worst-case problem. Although this may sound purely theoretical, it is of great practical value as keys that are chosen uniformly at random already provide *worst-case* security guarantees. The hardness of this underlying worst-case problem is also plausible as the best known algorithm to solve the relevant lattice problems requires exponential time [3].

Another advantage of lattice-based cryptography over the alternatives is that there is a whole range of provably secure signature schemes. In the random oracle model there are schemes due to Gentry, Peikert, and Vaikuntanathan [16]; Stehlé, Steinfeld, Tanaka, and Xagawa [32]; and Lyubashevsky [24]. As for the standard model, there are the works of Lyubashevsky and Micciancio [25] (tree-based); Peikert [29]; and Cash, Hofheinz, and Kiltz [12].

However, there is a gap in this range because none of the above schemes is *stateless*, provably secure in the *standard model*, and *strongly unforgeable*. Strong unforgeability under chosen message attacks (SU-CMA) is stronger than existential unforgeability (EU-CMA) in the sense that the adversary is not artificially restricted by the security model. In EU-CMA, the adversary is forced to output a signature for a *fresh message* $\mathfrak{m}^*$ after seeing signature for messages $\mathfrak{m}_i \neq \mathfrak{m}^*$of his or her choice. The SU-CMA adversary is also allowed to output a *fresh signature* for one of the $\mathfrak{m}_i$.

Strong unforgeability is interesting in both, theory and practice. Consider generic tranformations from CPA to CCA2 security in the standard model, e.g., Dolev, Dwork, and Naor [13] or Boneh, Canetti, Halevi, and Katz [9]. They typically involve a strongly unforgeable signature scheme to make the ciphertext authentic and non-malleable. An EU-CMA signature of the CPA ciphertext may already provide some security against CCA1 adversaries but a CCA2 attack would certainly still succeed. Another reason is the construction of ID-based blind signatures due to Galindo, Herranz, and Kiltz [14].

As a practical example, consider an access control protocol where you may delegate certain rights to another party by signing a description for these rights with a signature $\mathfrak{s}$. You want to be able to revoke them at any time in the future via an online revocation system. The rights are revoked as soon as the online system has $\mathfrak{s}$ in its database. If the signature scheme is only EU-CMA secure, the delegee can construct another signature $\mathfrak{s}^*$ for the same set of rights and present this token instead of $\mathfrak{s}$ — the revocation mechanism breaks down.

Notice that there are generic transformations from EU-CMA to SU-CMA. They typically only apply to a certain small subclass of signature schemes, e.g., Boneh-Shen-Waters [10]. Recently, Bellare and Shoup [6] proposed an unrestricted transformation. However, all lose efficiency compared to the underlying EU-CMA scheme because they require multiple signing steps.

Our Contribution. Table 1 compares our result with the current state-of-the-art for lattice signatures, including the typical improvements with ideal lattices [30,27]. All previously known SU-CMA schemes are either stateful, causing, e.g., synchronization problems, or they require random oracles. Using random oracles is discouraged by the works of Canetti, Goldreich, and Halevi [11], as well as by the more practical work of Leurent and Nguyen [22]. The only known scheme that directly provides SU-CMA security in the standard model is stateful and has a large secret key.[1] Our construction in Section 4.1 offers the same complexity as Peikert's scheme. In our case, signing involves a simple additional linear algebra step that can be pre-computed. Thus, we achieve a stronger security notion without additional cost.

The situation is quite similar for HIBS. With the exception of Libert and Quisquater [23], previous results deal only with existentially unforgeable HIBS. They typically provide hierarchical identity-based encryption (HIBE) and then

[1] Trade-offs are possible but the key generation complexity is $n^{\mathcal{O}(1)}$ for a large polynomial in any case.

Scheme	Stateless	Standard model	SU-CMA	Public key	Secret Key	Signature
[16] with [32]	*Yes*	No	*Yes*	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n)$
[24]	*Yes*	No	*Yes*	$\widetilde{\mathcal{O}}(n)$	$\mathcal{O}(n)$	$\widetilde{\mathcal{O}}(n)$
[29], [12]	*Yes*	*Yes*	No	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n)$
[25] with [26]	No	*Yes*	*Yes*	$\mathcal{O}(n)$	$n^{\mathcal{O}(1)}$	$\widetilde{\mathcal{O}}(n)$
Section 4.1	*Yes*	*Yes*	*Yes*	$\widetilde{\mathcal{O}}(n)$	$\widetilde{\mathcal{O}}(n^2)$	$\widetilde{\mathcal{O}}(n)$

Table 1. Comparison of the properties of current lattice-based signature schemes.

apply a generic conversion, e.g., [19], to obtain an HIBS. Both, HIBE and HIBS, can be classified as selective-ID or adaptive-ID secure. Selective-ID security forces the adversary to name its target identity before seeing the public key. In the adaptive case, it may output a forgery for any identity. So far, lattice-based IBE schemes either support a hierarchy [12,29,1] or they support adaptive-ID security [16] in the random oracle model. Moreover, in contrast to our constructions, none of them gives rise to an HIBS that is provably secure against subexponential attacks when using efficient ideal lattices.

Hence, in addition to the first stateless standard-model SU-CMA signature scheme from lattices in Section 4.1, we provide the first lattice-based constructions for adaptive-ID secure and strongly unforgeably HIBS in the random oracle model in Section 3.2 and for strongly unforgeable HIBS in the standard model in Section 4.2. Our constructions in Section 4 rely on Chameleon hash functions [21]. The security proofs involve a generic transformation to SU-CMA from a slightly weaker notion, which was not explicitly known before (cf. Section 2).

2 Preliminaries

The security parameter is n. The statement $x \xleftarrow{\$} X$ means x is chosen uniformly at random from X. With $x \sim \Delta(X)$, we denote that x is chosen according to a distribution Δ over X. The concatenation of strings, vectors, and matrix columns is done via $\circ$. Furthermore, $x \sqsubset y$ means x is a prefix of y and $\varnothing$ is the empty string. Lower-case boldface identifies vectors and upper-case boldface denotes a matrix. For a given matrix $\mathbf{X} \in \mathbb{R}^{m \times m}$, we write $\tilde{\mathbf{X}}$ for its Gram-Schmidt orthogonalization.

2.1 Security Models

Throughout the paper we stick to the following notation. The length of identities is κ, the message length is λ, and ℓ is the hierarchy depth, meaning that there are $\ell+1$ levels in the hierarchy tree. With $\{x_i\}_1^m$ we denote the set $\{x_1, \ldots, x_m\}$. The subsequent paragraphs deal with the specification of strongly unforgeable signature schemes $\mathsf{DSig} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ and hierarchical identity-based signature schemes $\mathsf{HIBS} = (\mathsf{Kg}, \mathsf{Extract}, \mathsf{Sign}, \mathsf{Vf})$.

Signature Schemes. We follow the standard specification for digital signature schemes: $\mathsf{Kg}(1^n)$ outputs a private signing key sk and a public verification key pk; $\mathsf{Sign}(\mathsf{sk}, \mathfrak{m})$ outputs a signature $\mathfrak{s}$ under sk for the message $\mathfrak{m}$; $\mathsf{Vf}(\mathsf{pk}, \mathfrak{s}, \mathfrak{m})$ outputs 1 iff $\mathfrak{s}$ is a valid signature on $\mathfrak{m}$ under pk.

Most schemes are proven to be existentially unforgeable under chosen message attacks (EU-CMA), but we will consider the stronger notion of strongly unforgeability under chosen message attacks (SU-CMA) as described in the following experiment, where $\mathcal{H}$ is a family of random oracles.

Experiment $\mathsf{Exp}^{\mathsf{SU\text{-}CMA}}_{\mathcal{A},\mathsf{DSig}}(n)$
$\mathsf{H} \leftarrow \mathcal{H}(1^n)$; $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Kg}(1^n)$
$(\mathfrak{m}^*, \mathfrak{s}^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot),\mathsf{H}(\cdot)}(\mathsf{pk})$
Let $(\mathfrak{m}_i, \mathfrak{s}_i)$ be the answer returned by $\mathsf{Sign}(\mathsf{sk}, \cdot)$ on input $\mathfrak{m}_i$, for $i = 1, \ldots, k$.
Return 1 iff $\mathsf{Vf}(\mathsf{pk}, \mathfrak{m}^*, \mathfrak{s}^*) = 1$ and $(\mathfrak{m}^*, \mathfrak{s}^*) \notin \{(\mathfrak{m}_1, \mathfrak{s}_1), \ldots, (\mathfrak{m}_k, \mathfrak{s}_k)\}$.

DSig is $(t, q_\mathsf{S}, q_\mathsf{H}, \epsilon)$-strongly unforgeable if there is no t-time adversary that succeeds with probability $\geq \epsilon$ after making $\leq q_\mathsf{S}$ signature oracle queries and $\leq q_\mathsf{H}$ random oracle queries. In the standard model, we leave out H and q_H.

The difference to the EU-CMA model is that the adversary in the SU-CMA model even wins if it outputs a new signature for a message that it already knows a signature for. In the EU-CMA model, the adversary is forced to output a forgery for a "fresh" message. Instead of directly providing SU-CMA security in our main constructions in Section 4, we use the weaker notion of strong unforgeability against static message attacks. Here, the adversary submits all messages $\mathfrak{m}_1, \ldots, \mathfrak{m}_{q_\mathsf{S}}$ before seeing the public key and the corresponding signatures. Then, we use a generic transformation to achieve full security (see below).

HIBS Schemes. The specification for HIBS schemes is a straightforward generalization of that for digital signature schemes. The main difference is that there are no per-signer verification keys but rather a shared verification key for the entire system. Moreover, the signer's public key is easily computable from a unique user identification string ID over a binary alphabet. The corresponding secret signing key is "extracted" by a trusted authority using a master secret key. The hierarchy is modeled by letting identities be a concatenation of per-level identifiers with decreasing rank, i.e., $\mathsf{ID} = \mathsf{ID}_0 \circ \mathsf{ID}_1 \circ \mathsf{ID}_2$ describes an identity on level 2 with parent identity $\mathsf{ID}_0 \circ \mathsf{ID}_1$, whose parent identity is the master identity ID_0.

More formally: $\mathsf{Kg}(1^n)$ outputs a master private key msk and a master public key mpk. The master identity is the empty string $\varnothing$; $\mathsf{Extract}(\mathsf{sk}_{\mathsf{ID}^\star}, \mathsf{ID})$ outputs a secret signing key sk for ID if $\mathsf{ID}^\star \sqsubset \mathsf{ID}$, otherwise $\perp$; $\mathsf{Sign}(\mathsf{sk}_\mathsf{ID}, \mathsf{ID}, \mathfrak{m})$ outputs a signature $\mathfrak{s}$ under sk_ID for $\mathfrak{m}$; $\mathsf{Vf}(\mathsf{mpk}, \mathsf{ID}, \mathfrak{s}, \mathfrak{m})$ outputs 1 iff $\mathfrak{s}$ is a valid signature on $\mathfrak{m}$ for the given identity and master public key.

The security models for HIBS and ordinary signatures are tightly related with the exception that one has to deal with the additional key extraction mechanism. We consider two variants, selective-ID security (similar to selective-secure HIBE [8]) and the stronger notion of adaptive-ID security. In both models, the adversary can query a key extraction oracle E, a signature oracle, and an optional

random oracle. The experiment $\mathsf{Exp}^{\mathsf{SU\text{-}CMA\text{-}SelectiveID}}_{\mathcal{A},\mathsf{HIBS}}$ describes selective-ID security, where the adversary has to fix an identity ID^* before seeing the master public key. He is then forced to output a forgery for ID^*. The adversary gets secret keys for all identities that are not a prefix of ID^*. Furthermore, it can query a signature oracle $\mathsf{S}(\mathsf{ID}, \mathfrak{m})$ with arbitrary identities and messages.

Experiment $\mathsf{Exp}^{\mathsf{SU\text{-}CMA\text{-}SelectiveID}}_{\mathcal{A},\mathsf{HIBS}}(n)$
$\mathsf{H} \leftarrow \mathcal{H}(1^n)$; $(\mathsf{ID}^*, state) \leftarrow \mathcal{A}(1^n)$; $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Kg}(1^n)$
$(\mathfrak{m}^*, \mathfrak{s}^*) \leftarrow \mathcal{A}^{\mathsf{E}(\mathsf{msk},\cdot)\setminus\{\cdot \sqsubset \mathsf{ID}^*\},\mathsf{S}(\cdot,\cdot),\mathsf{H}(\cdot)}(\mathsf{mpk}, state)$
Let $\{(\mathsf{ID}_i, \mathfrak{m}_i, \mathfrak{s}_i)\}_1^k$ be the query-answer tuples for S.
Return 1 iff $\mathsf{Vf}(\mathsf{mpk}, \mathsf{ID}^*, \mathfrak{s}^*, \mathfrak{m}^*) = 1$ and $(\mathsf{ID}^*, \mathfrak{m}^*, \mathfrak{s}^*) \notin \{(\mathsf{ID}_i, \mathfrak{m}_i, \mathfrak{s}_i)\}_1^k$.

In the stronger model of adaptive-ID security, the adversary can output a forgery for any identity, for which he has never queried a prefix to the extraction oracle before.

Experiment $\mathsf{Exp}^{\mathsf{SU\text{-}CMA\text{-}AdaptiveID}}_{\mathcal{A},\mathsf{HIBS}}(n)$
$\mathsf{H} \leftarrow \mathcal{H}(1^n)$; $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Kg}(1^n)$
$(\mathsf{ID}^*, \mathfrak{m}^*, \mathfrak{s}^*) \leftarrow \mathcal{A}^{\mathsf{E}(\mathsf{msk},\cdot),\mathsf{S}(\cdot,\cdot),\mathsf{H}(\cdot)}(\mathsf{mpk})$
Let $\{(\mathsf{ID}_i, \mathfrak{m}_i, \mathfrak{s}_i)\}_1^k$ be the query-answer tuples of S;
Let $\{\mathsf{ID}_j\}_1^l$ be the query-answer pairs of E.
Return 1 iff $\mathsf{Vf}(\mathsf{mpk}, \mathsf{ID}^*, \mathfrak{s}^*, \mathfrak{m}^*) = 1$
and $(\mathsf{ID}^*, \mathfrak{m}^*, \mathfrak{s}^*) \notin \{(\mathsf{ID}_i, \mathfrak{m}_i, \mathfrak{s}_i)\}_1^k$
and $\{\mathsf{ID}_j\}_1^l \ni \mathsf{ID}_j \not\sqsubset \mathsf{ID}^*$.

HIBS is $(t, q_\mathsf{E}, q_\mathsf{S}, q_\mathsf{H}, \epsilon)$-strongly unforgeable under chosen message and selective (adaptive) identity attacks if there is no t-time adversary that succeeds with probability $\geq \epsilon$ after making $\leq q_\mathsf{E}$ extraction queries, $\leq q_\mathsf{S}$ signature oracle queries, and $\leq q_\mathsf{H}$ random oracle queries in the respective experiment. Again, we leave out H and q_H in the standard model.

In Section 4, we will provide instantiations secure against static message attacks (SMA) and then use the following transformation to achieve CMA security.

From SMA to CMA. Krawczyk and Rabin [21] proposed Chameleon hashes to be hash functions with a trapdoor and the following properties. 1) The function $\mathsf{C} : D \times E \to R$ is chosen from a family $\mathcal{C}$ of Chameleon hashes along with a secret trapdoor t. 2) In order to sample from the distribution $(d, e, \mathsf{C}(d)) \in D \times E \times R$, we can do one of two things. Either we run C on the given document d and a randomness $e \sim \Delta(E)$ (efficiently sampleable), or we apply an inversion algorithm $e \leftarrow \mathsf{C}_t^{-1}(r, d)$ on a given image $r \in R$ and a target document $d \in D$. Thus, we obtain a randomness e such that $\mathsf{C}(d) = (e, r)$. It is important that the resulting distributions are within negligible statistical distance. Note that whenever we need the Chameleon hash to map to a certain range $\neq R$, we can compose it with an arbitrary collision resistant hash function. As for their realization, Krawczyk and Rabin claim in [20] that Chameleon hash functions exist if there are claw-free trapdoor permutations. Interestingly, they can be

easily implemented with the lattice-based trapdoor function in [16] as observed in [29].

A helpful fact about Chameleon hash functions is that if they exist, then there is a generic transformation from EU-SMA to EU-CMA signatures. This was known since [21] and it is proven in [18]. We show that the transformation also works for *SU*-SMA to *SU*-CMA in Appendix A. Observe that it is also applicable to selective-ID secure HIBS schemes as it only affects the way the signature oracle is simulated for the challenge identity.

Lemma 1. *SU-SMA implies SU-CMA if Chameleon hash functions exist.*

2.2 Lattices

In this work, we deal only with full-rank q-ary lattices, i.e., lattices that represent the kernel of the linear map $\mathbf{x} \mapsto \mathbf{A}\mathbf{x} \bmod q$ for a prime modulus q and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. These lattices are denoted with $\Lambda_q^{\perp}(\mathbf{A}) := \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{A}\mathbf{v} \equiv \mathbf{0} \pmod q\}$. As with all lattices of dimension $m \geq 2$, they have infinitely many bases. A basis of $\Lambda_q^{\perp}(\mathbf{A})$ is a matrix $\mathbf{B} \in \mathbb{Z}^{m \times m}$, such that $\Lambda(\mathbf{B}) := \{\mathbf{B}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^m\}$ is equal to $\Lambda_q^{\perp}(\mathbf{A})$. The quantity $\det(\Lambda) = |\det(\mathbf{B})|$ (for any basis) is a lattice constant. The main computational problem in q-ary lattices is the "short integer solution" problem SIS. It is parameterized with positive integers $n, m = \mathsf{poly}(n), q = \mathsf{poly}(n)$, and a real norm bound ν and it is formulated as an average-case problem: Given a uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find a non-zero $\mathbf{v} \in \Lambda_q^{\perp}(\mathbf{A})$ with $\|\mathbf{v}\|_2 \leq \nu$. Ajtai showed in [2] that this problem is at least as hard as finding short vectors in *all* lattices of dimension n, i.e., solving a related worst-case problem. A recent improvement to this reduction can be formulated as follows.

Theorem 1 (Worst-case to Average-case Reduction [16] (informal)). *If there is a polynomial time algorithm that breaks* $\mathsf{SIS}(n, m, q, \nu)$ *for* $q \geq \nu\, \omega(\sqrt{n \log(n)})$, $\nu = \mathsf{poly}(n)$ *with non-negligible probability, then there is a polynomial time algorithm that finds short non-zero vectors, which are only a* $\gamma \geq \nu\widetilde{\mathcal{O}}(\sqrt{n})$ *factor longer than the shortest vector, in* all *lattices of dimension* n.

In cryptography, we typically hand over $\mathbf{A}$, or a "bad" basis with long vectors, as the public key and keep a "good" (short) basis as our secret. The length of a basis is $\|\mathbf{B}\| := \max_{i=1,\ldots,m} \|\mathbf{b}_i\|_2$. This principle goes back to Ajtai. The most recent improvement for generating such a matrix $\mathbf{A}$ together with a particularly short trapdoor $\mathbf{T}$ for SIS is due to Alwen and Peikert [5].

2.3 Bonsai Trees

Peikert introduced the notion of "bonsai trees" on lattices in [29] in analogy to arboriculture. An arborist always starts with a certain amount of *undirected*, i.e., random, natural growth that he cannot control. Then, he applies his tools

and starts cultivating individual branches to achieve the desired looks via *directed* growth. The arborist is successful if the resulting tree still looks sufficiently natural to the observer. Once cultivated, a branch can easily be *extended* to form more directed growth without too much additional care. Instead of extending directed growth, the arborist can also generate a *randomized* offstrings, which can be given to another arborist that can easily cultivate them by *extending* growth. The offsprings hide the first arborist's work and the employed techniques. We formalize these concepts in the context of lattices. A (binary) bonsai tree is generated out of a root $\mathbf{A}^\star$ and branches $\mathbf{A}_i^{(b)} \in \mathbb{Z}_q^{n\times m_i}$, $b \in \{0,1\}, i \leq k \leq \mathsf{poly}(n)$, that are statistically close to uniform. The entire tree is the set $\{\mathbf{A}^\star \circ \mathbf{A}_1^{(x_1)} \circ \cdots \circ \mathbf{A}_k^{(x_k)} : \mathbf{x} \in \{0,1\}^{\leq k}\}$.

Proposition 1 (Directed Growth). *Let $\delta > 0$ be any fixed real constant and let $q \geq 3$ be odd. There is a polynomial time algorithm $\mathsf{ExtLattice}(\mathbf{A}_1, m_2)$ that, given uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n\times m_1}$ for any $m_1 \geq (1+\delta)n\log_2(q)$ and $\mathsf{poly}(n)$-bounded $m_2 \geq (4+2\delta)n\log_2(q)$, outputs $(\mathbf{A}_2 \in \mathbb{Z}_q^{n\times m_2}, \mathbf{S} \in \mathbb{Z}^{m\times m})$, where $m = m_1 + m_2$, such that $\mathbf{A} = \mathbf{A}_1 \circ \mathbf{A}_2$ is within negligible statistical distance of uniform; $\mathbf{S}$ is a basis of $\Lambda_q^\perp(\mathbf{A}_1 \circ \mathbf{A}_2)$; $\|\mathbf{S}\| \leq L = Cn\log_2(q)$ with overwhelming probability; and $\left\|\tilde{\mathbf{S}}\right\| \leq \tilde{L} = 1 + C\sqrt{(1+\delta)n\log_2(n)} \leq 1 + C\sqrt{m_1}$ with overwhelming probability.*

The proposition reflects the most recent result on lattice trapdoors from [4]. In our constructions, we will use $C = 20$, $\delta = 1$ and assume that $\mathbf{A}_2$ is uniformly random instead of within negligible statistical distance of uniform. The interpretation in terms of arboriculture is generating "directed growth" out of "undirected growth" because one starts with some random growth $\mathbf{A_1}$ and cultivates a branch $\mathbf{A_1} \circ \mathbf{A_2}$ along with a trapdoor $\mathbf{T}$, which is the arborist's journal or a trace of his work. However, the observer cannot distinguish undirected growth from directed growth.

An important observation is that knowing a trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ implies knowing a trapdoor for all $\mathbf{A'} \in \mathbb{Z}_q^{n\times m'}$, $m' \geq m$, when $\mathbf{A} \sqsubset \mathbf{A'}$. This is because one can apply the trapdoor in dimension m and then pad the resulting vector with zeros to solve SIS in dimension m'.

Proposition 2 (Extending Control). *There is polynomial time algorithm $\mathsf{ExtBasis}(\mathbf{S}_1, \mathbf{A} = \mathbf{A}_1 \circ \mathbf{A}_2)$ that takes a basis $\mathbf{S}$ of $\Lambda_q^\perp(\mathbf{A}_1)$ and a matrix $\mathbf{A}$ with $\mathbb{Z}_q^{n\times m_1} \ni \mathbf{A}_1 \sqsubset \mathbf{A} \in \mathbb{Z}_q^{n\times(m_1+m_2)}$ as input. If $m_1 \geq 2n\log_2(q)$, it outputs a basis $\mathbf{S}$ for $\Lambda_q^\perp(\mathbf{A})$ with $\left\|\tilde{\mathbf{S}}\right\| = \left\|\tilde{\mathbf{S}}_1\right\|$.*

Whenever trapdoor delegation is required, one cannot simply use extending control and hand over the resulting basis as it leaks information about the original trapdoor. Here, we can use tree propagation to obtain a *randomized* offspring with a new, random trapdoor.

Proposition 3 (Randomizing Control). *On input a basis $\mathbf{S}$ of the lattice $\Lambda_q^\perp(\mathbf{A})$ of dimension m and a Gaussian parameter $s \geq \left\|\tilde{\mathbf{S}}\right\| \omega(\sqrt{\log(n)})$, the*

polynomial time algorithm $\mathsf{RandBasis}(\mathbf{S}, s)$ *outputs a basis* $\mathbf{S}'$ *of* $\Lambda_q^\perp(\mathbf{A})$ *with* $\left\|\tilde{\mathbf{S}}'\right\| \leq s\sqrt{m}$. *The basis is independent of* $\mathbf{S}$ *in the sense that for any two bases* $\mathbf{S}_0, \mathbf{S}_1$ *of* $\Lambda_q^\perp(\mathbf{A})$ *and* $s \geq \max\{\left\|\tilde{\mathbf{S}}_0\right\|, \left\|\tilde{\mathbf{S}}_1\right\|\}\omega(\sqrt{\log(n)})$, $\mathsf{RandBasis}(\mathbf{S}_0, s)$ *is within negligible statistical distance of* $\mathsf{RandBasis}(\mathbf{S}_1, s)$.

Estimating Secure Parameters. We could use the worst-case to average-case reduction for selecting secure parameters but that might be too conservative as the reduction is quite loose with respect to the ratio $m/n = \mathcal{O}(\log(n))$. The observations of Gama and Nguyen in [15] may be more realistic. They assume that lattice reduction algorithms find short lattice vectors $\mathbf{v}$ in lattices Λ of dimension d with $\|\mathbf{v}\|_2 \leq \delta^d \det(\Lambda)^{1/d}$ for some $\delta > 0$. This value δ is supposed to "summarize" the capability of the employed lattice reduction algorithm. Nguyen and Gama analyze random lattices from a certain distribution and find that reaching $\delta < 1.01$ in high dimensions is hard.

Since the distribution of lattices in the SIS problem is different, their conjecture is not directly applicable. Furthermore, Micciancio and Regev describe an attack that works best in a sub-lattice of the usual q-ary lattice $\Lambda_q^\perp(\mathbf{A})$. Assuming an adversary with capability δ^*, they show that for given parameters n and q, the best strategy of for the adversary is to attack a sub-lattice of dimension $d^* = \sqrt{n\log(q)/\log(\delta^*)}$. In this dimension, the adversary finds vectors of Euclidean length $\nu^* = \min\{q, 2^{2\sqrt{n\log(q)\log(\delta^*)}}\}$ [28]. For our parameter choices, we will always assume that the adversary is capable of reaching $\delta^* = 1.01$, but no less. All schemes in Sections 3 and 4 are based on the hardness of SIS in q-ary lattices of dimension m for a particular norm bound ν. Given this norm bound and δ^*, we need to establish that $\nu^* \gg \nu$, e.g., with a factor of 10 between both sides. The implicit assumption is that lattice basis reduction algorithms behave the same on random q-ary sub-lattices. Whether this heuristic is completely sound, is still unknown.

3 Warm-up — Constructions with Random Oracles

In this section, we recall strongly unforgeable GPV signatures as introduced in [16]. Then, we show how to use GPV together with the Bonsai-tree concept to build strongly unforgeable hierarchical identity-based signatures in the random oracle model. The proposed scheme is also secure under adaptive-identity queries.

3.1 Strongly Unforgeable Signatures

Lattice-based strongly unforgeable signatures were first proposed by Gentry, Peikert, and Vaikuntanathan in [16]. They introduce a family of preimage sampleable functions $\mathsf{GPV} = (\mathsf{TrapGen}, \mathsf{Eval}, \mathsf{SamplePre})$ on lattices. Its parameters $q, m, \tilde{L}, s = \omega(\sqrt{\log(n)})\tilde{L}$ only depend on the security parameter n as in Proposition 1. We define the sets $D_d := \{\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\} : \|\mathbf{x}\|_2 \leq d\}$ and $R := \mathbb{Z}_q^n$.

The algorithm $\mathsf{TrapGen}(1^n)$ outputs a public description $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a secret trapdoor $\mathbf{T} \in \mathbb{Z}^{m \times m}$, $\left\|\tilde{\mathbf{T}}\right\| \leq \tilde{L}$. Evaluation of the function $f_{\mathbf{A}} : \mathbb{Z}_q^m \to R$ is performed by $\mathsf{Eval}(\mathbf{A}, \mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$. Finally, the inversion algorithm $\mathsf{SamplePre}(\mathbf{T}, s, \mathbf{y})$ samples from the set of preimages $\{\mathbf{x} \in D_{s\sqrt{m}} : f_{\mathbf{A}}(\mathbf{x}) = \mathbf{y}\}$. Preimages have a conditional min-entropy of $\omega(\log(n))$ and follow a certain Gaussian distribution that can be efficiently sampled with $\mathsf{SampleDom}$, even without the trapdoor. By construction, the function compresses the input and therefore admits collisions. Finding collisions in D_d, however, is at least as hard as solving $\mathsf{SIS}(n, q, m, 2d)$.

Now, we can define $\mathsf{DSig}^{\mathsf{GPV}}$ accordingly. Let $\mathcal{H}$ be a family of random oracles $\mathsf{H} : \{0,1\}^* \to R = \mathbb{Z}_q^n$. The key generator simply forwards the output of $\mathsf{TrapGen}$. Signing a message $\mathfrak{m}$ involves choosing $r \xleftarrow{\$} \{0,1\}^n$, computing $\mathbf{y} \leftarrow \mathsf{H}(\mathfrak{m}, r)$, and calling $\mathfrak{s} \leftarrow \mathsf{SamplePre}(\mathbf{T}, s, \mathbf{y})$. A signature $(\mathfrak{s}, r)$ for $\mathfrak{m}$ verifies if $\mathfrak{s} \in D_{s\sqrt{m}}$ and $f_{\mathbf{A}}(\mathfrak{s}) = \mathsf{H}(\mathfrak{m}, r)$.

Security. Notice that the signature oracle can be efficiently simulated without the trapdoor by standard random oracle techniques. A forgery $(\mathfrak{m}^*, \mathfrak{s}^*, r^*)$ in the SU-CMA experiment implies a collision $(\mathfrak{s}^*, \mathfrak{s}_i)$ with $f_{\mathbf{A}}(\mathfrak{s}_i) = f_{\mathbf{A}}(\mathfrak{s}^*) = \mathsf{H}(\mathfrak{m}^*, r^*)$, where $\mathfrak{s}_i$ was chosen during the random oracle simulation. By the conditional min-entropy, we know that $\mathfrak{s}_i \neq \mathfrak{s}^*$ with probability $1 - n^{-\omega(1)}$. Let $T_{\mathsf{TrapGen}}(n)$, $T_{\mathsf{SampleDom}}(n)$, and $T_{\mathsf{Eval}}(n)$ be the cost functions for trapdoor generation, sampling, and trapdoor evaluation. In addition, let $T_{\mathsf{List}}(n)$ be the cost function for list processing in the simulation of the random oracle.

Theorem 2 (Strong Unforgeability [16]). *For the above choice of parameters,* $\mathsf{DSig}^{\mathsf{GPV}}$ *is* $(t, q_{\mathsf{S}}, q_{\mathsf{H}}, \epsilon)$*-strongly unforgeable in the random oracle model if* $\mathsf{SIS}(n, m, q, 2s\sqrt{m})$ *is* $(t + q_{\mathsf{S}}T_{\mathsf{List}} + q_{\mathsf{H}}(T_{\mathsf{SampleDom}}(n) + T_{\mathsf{Eval}}(n) + T_{\mathsf{List}}(n)), (\epsilon - q_{\mathsf{S}}^2/2^n)(1 - n^{-\omega(1)}))$*-hard.*

Via Theorem 1, a corollary states that a successful attacker can find the shortest vector in all lattices of dimension n up to an approximation factor $\gamma \geq \tilde{L}\widetilde{\mathcal{O}}(n) = \widetilde{\mathcal{O}}(n\sqrt{n})$.

Secure Parameters. We estimate secure paramters for $\mathsf{DSig}^{\mathsf{GPV}}$ as described in Section 2. For a given security parameter n, we choose the remaining parameters according to Proposition 1 and a prime $q \approx n^{4.5}$. For $n \geq 197$, we obtain the required hardness of the underlying SIS problem.

3.2 Strongly Unforgeable Hierarchical ID-based Signatures

Using the signature scheme from the previous section, we can can directly apply the Bonsai-tree concept to obtain a hierarchical identity-based signature scheme $\mathsf{HIBS}^{\mathsf{GPV}}$ in the random oracle model. The idea is that, based on a given ID, the secret key extraction algorithm $\mathsf{Extract}$ first uses $\mathsf{ExtBasis}$ to extend the matrix $\mathbf{A}^\star$ to $\mathbf{A}_{\mathsf{ID}}$ and the secret master key $\mathbf{T}^\star$ to $\mathbf{T}_{\mathsf{ID}}$ such that $\mathbf{A}_{\mathsf{ID}}\mathbf{T}_{\mathsf{ID}} \equiv \mathbf{0}$. Then,

in order to protect the master key, it uses $\mathsf{RandBasis}$ and outputs a randomized trapdoor $\mathbf{S}_{\mathsf{ID}}$. The individual signers use $\mathsf{DSig}^{\mathsf{GPV}}$ and we have an identity-based signature scheme that is strongly unforgeable. This concept can be transfered to the hierarchical setting, where every signer may act as a key extraction entity. Note that the number of levels in the hierarchy affects the tightness of the security proof as these randomized trapdoors are slightly worse than the master trapdoor.

We assume that all identities on all levels have length κ. The maximum number of levels, including the master-key, is $\ell+1$. We denote the basis length on level k with $\tilde{L}_k$ and the corresponding Gaussian parameter with $s_k = \omega(\sqrt{\log(n)})\tilde{L}_k$.

Master-key Generation. Let $q, \tilde{L}, m_1, m_2$ be chosen according to Proposition 1 and let $d = \tilde{L}\omega(\sqrt{\log(n)})^{\ell+1}\sqrt{m_1 + (\ell\kappa+1)m_2}^{\ell+1}$. These parameters may be excluded from the public key as they are the same for all users. Generate a description $\mathbf{A}^\star \in \mathbb{Z}_q^{n\times m_1+m_2}$ of the master lattice $\Lambda_q^\perp(\mathbf{A}^\star)$ together with a trapdoor $\mathbf{S}^\star$ such that $\left\|\tilde{\mathbf{S}}\right\| \leq \tilde{L}$ with $\mathsf{ExtLattice}$. Then, choose random matrices $\langle\mathbf{A}\rangle := \left\{(\mathbf{A}_i^{(0)}, \mathbf{A}_i^{(1)})\right\}_1^{\ell\kappa}$ from $\mathbb{Z}_q^{n\times m_2}$. The secret is $\mathbf{S}^\star$ and the public key is $(\mathbf{A}^\star, \langle\mathbf{A}\rangle)$.

Key Extraction. On input $\mathbf{S}^\star_{\mathsf{ID}^\star}$, ID with $\mathsf{ID} = \mathsf{ID}^\star \circ \mathsf{ID}'$, $|\mathsf{ID}| = l \leq \ell\kappa$, $\mathsf{ID}' = \mathsf{ID}'_1, \ldots, \mathsf{ID}'_\kappa \in \{0,1\}^\kappa$ recursively define the matrix $\mathbf{A}_{\mathsf{ID}} := \mathbf{A}_{\mathsf{ID}^\star} \circ \mathbf{A}_{l+1}^{(\mathsf{ID}'_1)} \circ \cdots \circ \mathbf{A}_{l+\kappa}^{(\mathsf{ID}'_\kappa)}$ with $\mathbf{A}_\varnothing := \mathbf{A}^\star$, and call $\mathbf{T}_{\mathsf{ID}} \leftarrow \mathsf{ExtBasis}(\mathbf{S}^\star_{\mathsf{ID}^\star}, \mathbf{A}_{\mathsf{ID}})$. Then, let $s = \left\|\tilde{\mathbf{T}}_{\mathsf{ID}}\right\| \omega(\sqrt{\log(n)})$ and output the randomized basis $\mathbf{S}_{\mathsf{ID}} \leftarrow \mathsf{RandBasis}(\mathbf{S}_{\mathsf{ID}}, s)$ with $\left\|\tilde{\mathbf{S}}_{\mathsf{ID}}\right\| \leq s\sqrt{\dim(\mathbf{S}_{\mathsf{ID}})}$. For inappropriate inputs, return $\perp$.

Signing. On input a message $\mathfrak{m} \in \{0,1\}^*$ and the trapdoor $\mathbf{S}_{\mathsf{ID}}$, the signer with identity ID on level k chooses $r \xleftarrow{\$} \{0,1\}^n$ and computes $\mathfrak{s} \leftarrow \mathsf{SamplePre}(\mathbf{S}_{\mathsf{ID}}, s_k, \mathsf{H}(\mathfrak{m}, r, \mathsf{ID}))$. It outputs $(\mathfrak{s}, r)$.

Verification. On input $(\mathbf{A}^\star, \langle\mathbf{A}\rangle)$, an identity ID, a signature $(\mathfrak{s}, r)$, and a message $\mathfrak{m}$, the algorithm outputs 1 if and only if $\mathfrak{s} \in D_d$ and $f_{\mathbf{A}_{\mathsf{ID}}}(\mathfrak{s}) = \mathsf{H}(\mathfrak{m}, r, \mathsf{ID})$.

Recall that $\mathsf{SamplePre}$ outputs a vector of length at most $s\sqrt{m}$ with $s = \omega(\sqrt{\log(n)})$ $\tilde{L}$ when called with a trapdoor of length at most $\tilde{L}$ in dimension m. The maximum dimension in the scheme is $m_1 + (\ell\kappa+1)m_2$ on level ℓ. The maximum basis length on level i is $\tilde{L}_i = s_{i-1}\sqrt{m_1 + (\ell\kappa+1)m_2}$ with $\tilde{L}_0 := \tilde{L}$. The Gaussian parameter for presampling on level i is $s_i = \omega(\sqrt{\log(n)})\tilde{L}_i$ with $s_0 = s$. This recurrence yields $\tilde{L}_\ell = \tilde{L}\omega(\sqrt{\log(n)})^\ell\sqrt{m_1 + (\ell\kappa+1)m_2}^\ell$. Signing with such a basis yields signatures of Euclidean length at most $\omega(\sqrt{\log(n)})\tilde{L}_\ell\sqrt{m_1 + (\ell\kappa+1)m_2} = d$. Thus, such signatures are accepted by the verifier. The security guarantees scale with the depth of the hierarchy as the norm bound becomes looser for increasing ℓ.

Security. We prove that $\mathsf{HIBS}^{\mathsf{GPV}}$ is secure under selective identity attacks. The second theorem shows that it is even secure under *adaptive* identity attacks,

but with a looser reduction. The latter reduction, however, is not as loose as the generic method of guessing the right identity with probability $2^{-\kappa}$ [8]. Let $T_{\mathsf{func}}(x)$ be the cost function for function func and let $T_{\mathsf{List}}(n)$ be the cost function for list processing for simulating a random oracle.

Theorem 3 (Selective Security). $\mathsf{HIBS}^{\mathsf{GPV}}$ *is* $(t, q_{\mathsf{S}}, q_{\mathsf{H}}, \epsilon)$*-strongly unforgeable under selective identity attacks in the random oracle model if* SIS *is* $(t + 2T_{\mathsf{ExtLattice}} + q_{\mathsf{E}}T_{\mathsf{Extract}} + (q_{\mathsf{H}} + q_{\mathsf{S}})T_{\mathsf{H}}, (1 - n^{-\omega(1)})(\epsilon - q_{\mathsf{S}}^2/2^n))$*-hard with norm bound* $\nu = 2\tilde{L}\omega(\sqrt{\log(n)})^{\ell+1}\sqrt{m_1 + (\ell\kappa + 1)m_2}^{\ell+1}$.

Notice the the GPV signature scheme can be efficiently simulated by a standard random oracle technique. Moreoever, the adversary will make $\leq q_{\mathsf{E}}$ extraction queries that need to be answered. We prepare for this by "knowing" a trapdoor for a prefix of all but the challenge identity. Upon an extraction query, this trapdoor can be extended to a trapdoor for the requested identity. The external challenge, the input $\mathbf{A}$ from the SIS problem, is embedded in the public key of the challenge identity. Via random oracle techniques, the reduction knows a valid signature for the output message of the adversary. However, the adversary outputs a different signature with probability $1 - n^{-\omega(1)}$ by the conditional min-entropy of the set of possible signature. The full proof is in Appendix B.

Theorem 4 (Adaptive Security). $\mathsf{HIBS}^{\mathsf{GPV}}$ *can be made* $(t, q_{\mathsf{S}}, q_{\mathsf{H}}, q_{\mathsf{G}}, \epsilon)$*-strongly unforgeable under* adaptive *identity attacks in the random oracle model if it is* $(t, q_{\mathsf{S}}, q_{\mathsf{H}}, q_{\mathsf{G}}, \epsilon/q_{\mathsf{G}})$*-strongly unforgeable under* selective *identity attacks.*

Here, we only give the idea of the conversion. We need to change the way the identities are mapped to the branches of the Bonsai tree. Instead of directly using the binary representation of ID or its hash value, we apply a random oracle G to the individual substrings of length κ first. Thus, every ID is mapped to a random position in the tree. Assume that the adversary makes q_{G} queries to this random oracle, we can prepare a randomly selected path in the tree with "undirected growth" and program the random oracle to map one of the q_{G} queries to this path. Therefore, the success probability of the reduction degrades with a factor $1/q_{\mathsf{G}}$ instead of the generic $1/2^{\kappa}$.

The worst-case to average-case reduction guarantees security if finding shortest vectors up to an approximation factor $\gamma \geq 2d\widetilde{\mathcal{O}}(\sqrt{n}) = \widetilde{\mathcal{O}}(\sqrt{n}^{\ell+3})$ is hard in the worst case in dimension n.

Secure Parameters. We assume identites of length $\kappa = 40$ bits, which should be sufficient in practical scenarios. The parameter q is crucial as it greatly influences the hardness of SIS and the worst-case to average-case reduction only holds for large enough q. The scheme's efficiency greatly depends on the number $\ell + 1$ of layers in the hierarchy. For larger ℓ, we are forced to increase q. Possible secure choices for (ℓ, n, q) are $(1, 467, n^6)$, $(2, 692, n^8)$, $(3, 1011, n^9)$, or $(4, 1258, n^{11})$.

4 Constructions without Random Oracles

In the following, we propose our main constructions. We start with a strongly unforgeable signature scheme and then propose a strongly unforgeable hierarchical identity-based signature scheme that is secure under selective identity attacks. Both constructions are secure in the standard model.

In the following, we let $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$ be a collision-resistant hash function that maps into the message space. Finding collisions in polynomial time is only possible with probability $\leq c$.

4.1 Strongly Unforgeable Signatures

We propose a variant of Peikert's EU-CMA signature scheme [29]. He first creates an EU-SMA secure signature scheme in the standard model and then applies a generic conversion from EU-SMA to EU-CMA, using a Chameleon hash function. We follow this line of thought and construct a signature scheme that is even secure in the SU-CMA sense. We achieve this by computing the signatures in a different way, compared to Peikert's EU-SMA scheme. In Peikert's scheme, there is a matrix $\mathbf{A}_\mathfrak{m}$ and a corresponding lattice $\Lambda_q^\perp(\mathbf{A}_\mathfrak{m})$ for which the signer can derive a trapdoor. The matrix $\mathbf{A}_\mathfrak{m}$ is formed as in Section 3.2. The signatures in Peikert's work are short lattice vectors in $\Lambda_q^\perp(\mathbf{A}_\mathfrak{m})$, i.e., short vectors $\mathfrak{s} \neq \mathbf{0}$ such that $\mathbf{A}_\mathfrak{m}\mathfrak{s} \equiv \mathbf{0}$. In our scheme, we fix a random $\mathbf{y} \in \mathbb{Z}_q^n$ and let the signer solve $\mathbf{A}_\mathfrak{m}\mathfrak{s} \equiv \mathbf{y}$ instead. The overhead is a simple linear algebra step (cf. [16]) that can done once during key generation. Surprisingly, this slight change enables us to prove *strong* unforgeability. It is important to note that, unlike Peikert, we need to be able to answer all signature queries in the security proof, i.e., even for the message $\mathfrak{m}^*$ that the adversary outputs a forgery for.

First of all, we demonstrate that the schemes in [29,12] are not strongly unforgeable by showing an attack that only works in the SU-SMA model and not in EU-SMA. The adversary in the SU-SMA experiment queries its signature oracle with a random message $\mathfrak{m}^*$ and receives a signature $\mathfrak{s}$ such that $\mathfrak{s} \in D(s\sqrt{m})$ and $\mathbf{A}_\mathfrak{m}\mathfrak{s} \equiv \mathbf{0}$. The adversary simply returns the valid forgery $\mathfrak{s}^* \leftarrow -\mathfrak{s}$.

We specify $\mathsf{DSig}^{\mathsf{su\text{-}sma}}$, where one may interpret the randomness-message pair as an identity.

Key Generation. Let $q, \tilde{L}, m_1, m_2$ be chosen according to Proposition 1 and let $s = \tilde{L}\omega(\sqrt{\log(n)})$ and $d = s\sqrt{m_1 + (\lambda+1)m_2}$. These parameters may be excluded from the public key as they are the same for all users. Use $\mathsf{ExtLattice}$ to generate a description $\mathbf{A}^\star \in \mathbb{Z}_q^{n \times m_1 + m_2}$ of the master lattice $\Lambda_q^\perp(\mathbf{A}^\star)$ together with a trapdoor $\mathbf{S}^\star$ such that $\left\|\tilde{\mathbf{S}}\right\| \leq \tilde{L}$. Furthermore, generate a set $\langle \mathbf{B} \rangle := \left\{(\mathbf{B}_i^{(0)}, \mathbf{B}_i^{(1)})\right\}_1^\lambda$ random matrices in $\mathbb{Z}_q^{n \times m_2}$. Finally, choose $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^n$ and output the secret key $\mathbf{S}^\star$ and the public key $(\mathbf{A}^\star, \langle \mathbf{B} \rangle, \mathbf{y})$.

Signing. On input a message $\mathfrak{m} \in \{0,1\}^*$ and the secret trapdoor $\mathbf{S}^\star$, the signer with identity chooses $r \xleftarrow{\$} \{0,1\}^n$ and computes $h \leftarrow \mathsf{H}(\mathfrak{m}, r)$, $\mathfrak{s} \leftarrow$

$\mathsf{SamplePre}(\mathbf{S}_h, s, \mathbf{y})$. The trapdoor $\mathbf{S}_h$ is formed via $\mathsf{ExtBasis}(\mathbf{S}^\star, \mathbf{A}_h)$, where $\mathbf{A}_h := \mathbf{A}^\star \circ \mathbf{B}_1^{(h_1)} \circ \cdots \circ \mathbf{B}_\lambda^{(h_\lambda)}$. It outputs $(\mathfrak{s}, r)$.

Verification. On input $(\mathbf{A}^\star, \langle \mathbf{B} \rangle, \mathbf{y})$, a signature $(\mathfrak{s}, r)$, and a message $\mathfrak{m}$, the algorithm outputs 1 if and only if $\mathfrak{s} \in D_d$ and $\mathbf{A}_{\mathsf{H}(\mathfrak{m},r)}(\mathfrak{s}) = \mathbf{y}$.

The scheme is complete because all signatures are generated using a basis of length $\tilde{L}$ and with the Gaussian parameter $s = \omega(\sqrt{\log(n)})\tilde{L}$. The total dimension is $m = m_1 + (\lambda+1)m_2$. Thus, $\mathsf{SamplePre}$ outputs signatures of length at most $s\sqrt{m_1 + (\lambda+1)m_2} = d$ that are accepted by Vf. In order to get the full (SU-CMA) scheme, we wrap the message with a Chameleon hash function.

Security. We prove that $\mathsf{DSig}^{\mathsf{su\text{-}sma}}$ is SU-SMA secure and then apply the black-box conversion in Lemma 1. Let $T_{\mathsf{func}}(n)$ be the cost function for the function func and let $T_{\mathsf{List}}(n)$ be the cost function for list processing, which is explained in the analysis below.

Theorem 5 (Strong Unforgeability). $\mathsf{DSig}^{\mathsf{su\text{-}sma}}$ *is* (t, ϵ) *strongly unforgeable under static message attacks (SU-SMA) if* SIS *with* $\nu = 2\tilde{L}\omega(\sqrt{\log(n)})\sqrt{m_1 + (\lambda+1)m_2}$ *is* $(t + \lambda q_{\mathsf{S}} T_{\mathsf{List}} + T_{\mathsf{ExtLattice}} + q_{\mathsf{S}}(T_{\mathsf{SamplePre}} + T_{\mathsf{ExtBasis}}), 1/2(1 - n^{-\omega(1)})(\epsilon - q_{\mathsf{S}}^2/2^n - c)/(\lambda q_{\mathsf{S}}))$*-hard.*

The idea is to separate the adversaries into two classes. One works in the EU-SMA sense and the other exploits the additional freedom of the SU-SMA setting. The reduction guesses the type of adversary before handing over the public key. If it expects an EU-SMA forger, the reduction knows $\mathbf{x}$ with $\mathbf{A}^\star\mathbf{x} \equiv \mathbf{y}$ and forces the forger to solve an inhomogeneous SIS, for which the reduction does not know the trapdoor. Together with $\mathbf{x}$, it can solve the corresponding SIS with overwhelming probability. For the SU-SMA forger, the reduction has to guess the index i^* of the signature query that will be recycled in the forgery with probability $1/q_{\mathsf{S}}$. There, it plants an $\mathbf{x}$ with $\mathbf{A}_{\mathsf{H}(\mathfrak{m}_{i^*}, r_{i^*})}\mathbf{x} \equiv \mathbf{y}$. Again, with the adversary's help, the reduction solves SIS with overwhelming probability, while being able to answer a single signature query for $\mathfrak{m}_{i^*}$ with $\mathbf{x}$. The key extraction queries are answered as described in Section 3.2. The full proof is in Appendix C.

Therefore, $\mathsf{DSig}^{\mathsf{su\text{-}sma}}$ is secure if finding shortest vectors in dimension n, within factors $\gamma \geq \widetilde{\mathcal{O}}(n\sqrt{n})$, is hard in the worst case. Via Lemma 1, we immediately get similar results for SU-CMA.

Secure Parameters. The underlying problem is SIS with $\nu = 2\omega(\sqrt{\log(n)})\tilde{L}\sqrt{m_1 + (\lambda+1)m_2}$. Let $\lambda = 160$ and consider $\log(n) = \omega(\sqrt{\log(n)})$. As before, q is chosen such that the worst-case to average-case reduction holds ($q \approx n^5$). Now, for $n \geq 247$, we obtain the required complexity of SIS.

4.2 Strongly Unforgeable Hierarchical ID-based Signatures

By adding more layers to the hierarchy in Section 4.1, we construct a hierarchical identity-based signature scheme $\mathsf{HIBS}^{\mathsf{su\text{-}sma}}$ that is strongly unforgeable. The

identities are handled as in Section 3.2 but then we use an additional Bonsai tree to sign.

The length of each sub-identity on each level is κ, the number of levels is ℓ, and messages have λ bits. $\tilde{L}_k$ is the basis length on level k. The corresponding Gaussian parameter is $s_k := \omega(\sqrt{\log(n)})\tilde{L}_k$.

Master-key Generation. Let $q, \tilde{L}, m_1, m_2$ be chosen according to Proposition 1 and let $d = \tilde{L}\omega(\sqrt{\log(n)})^{\ell+1}\sqrt{m_1 + (\ell\kappa + \lambda + 1)m_2}^{\ell+1}$. These parameters may be excluded from the public key as they are the same for all users. Use $\mathsf{ExtLattice}$ to generate a description $\mathbf{A}^\star \in \mathbb{Z}_q^{n \times m_1 + m_2}$ of the master lattice $\Lambda_q^\perp(\mathbf{A}^\star)$ together with a trapdoor $\mathbf{S}^\star$ such that $\left\|\tilde{\mathbf{T}}\right\| \leq \tilde{L}$. Furthermore, generate the sets $\langle \mathbf{A} \rangle := \left\{(\mathbf{A}_i^{(0)}, \mathbf{A}_i^{(1)})\right\}_1^{\ell\kappa}$, $\langle \mathbf{B} \rangle := \left\{(\mathbf{B}_i^{(0)}, \mathbf{B}_i^{(1)})\right\}_1^{\lambda}$ of random matrices in $\mathbb{Z}_q^{n \times m_2}$. Finally, choose $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^n$ and output the secret key $\mathbf{S}^\star$ and the public key $(\mathbf{A}^\star, \langle \mathbf{A} \rangle, \langle \mathbf{B} \rangle, \mathbf{y})$.

Key Extraction. On input $\mathbf{S}^\star_{\mathsf{ID}^\star}$, ID with $\mathsf{ID} = \mathsf{ID}^\star \circ \mathsf{ID}'$, $|\mathsf{ID}| = l \leq \ell\kappa$, $\mathsf{ID}' = \mathsf{ID}'_1, \ldots, \mathsf{ID}'_\kappa \in \{0,1\}^\kappa$ recursively define the matrix $\mathbf{A}_{\mathsf{ID}} := \mathbf{A}_{\mathsf{ID}^\star} \circ \mathbf{A}_{l+1}^{(\mathsf{ID}'_1)} \circ \cdots \circ \mathbf{A}_{l+\kappa}^{(\mathsf{ID}'_\kappa)}$ with $\mathbf{A}_\varnothing := \mathbf{A}^\star$, and call $\mathbf{T}_{\mathsf{ID}} \leftarrow \mathsf{ExtBasis}(\mathbf{S}^\star_{\mathsf{ID}^\star}, \mathbf{A}_{\mathsf{ID}})$. Then, let $s = \left\|\tilde{\mathbf{T}}\right\| \omega(\sqrt{\log(n)})$ and output the randomized basis $\mathbf{S}_{\mathsf{ID}} \leftarrow \mathsf{RandBasis}(\mathbf{T}_{\mathsf{ID}}, s)$ with $\left\|\tilde{\mathbf{S}}_{\mathsf{ID}}\right\| \leq s\sqrt{\dim(\mathbf{T}_{\mathsf{ID}})}$. For inappropriate inputs, return $\perp$.

Signing. On input a message $\mathfrak{m} \in \{0,1\}^*$ and a secret trapdoor $\mathbf{S}_{\mathsf{ID}}$, the signer with identity ID on level k chooses $r \xleftarrow{\$} \{0,1\}^n$ and computes $\mu \leftarrow \mathsf{H}(\mathfrak{m}, r)$. Now μ is used to form $\mathbf{A}_{\mathsf{ID}\circ\mu} := \mathbf{A}_{\mathsf{ID}} \circ \mathbf{B}_1^{(\mu_1)} \circ \cdots \circ \mathbf{B}_\lambda^{(\mu_\lambda)}$. Then, the signer extends its secret basis by calling $\mathbf{S}_{\mathsf{ID}\circ\mu} \leftarrow \mathsf{ExtBasis}(\mathbf{S}_{\mathsf{ID}}, \mathbf{A}_{\mathsf{ID}\circ\mu})$. Finally, it outputs $\mathfrak{s} \leftarrow \mathsf{SamplePre}(\mathbf{S}_{\mathsf{ID}\circ\mu}, s_k, \mathbf{y})$ and r.

Verification. On input $(\mathbf{A}^\star, \langle \mathbf{A} \rangle, \langle \mathbf{B} \rangle, \mathbf{y})$, a signature $(\mathfrak{s}, r)$, and a message $\mathfrak{m}$, the algorithm outputs 1 if and only if $\mathfrak{s} \in D_d$ and $\mathbf{A}_{\mathsf{ID}\circ\mathsf{H}(\mathfrak{m},r)}(\mathfrak{s}) = \mathbf{y}$.

Notice that the scheme is complete by a similar argument as in Section 3.2. The maximum trapdoor length (level ℓ) is $\tilde{L}_\ell = \tilde{L}\omega(\sqrt{\log(n)})^\ell\sqrt{m_1 + (\ell\kappa + \lambda + 1)m_2}^\ell$. This basis is extended once before signing, which is length-preserving. Then, signing is done via $\mathsf{SamplePre}$ that outputs a vector of length at most $\omega(\sqrt{\log(n)})\tilde{L}_\ell\sqrt{m_1 + (\ell\kappa + \lambda + 1)m_2} \leq d$. Thus, the verification algorithm accepts.

Unforgeability. We prove that $\mathsf{HIBS}^{\mathsf{su\text{-}sma}}$ is SU-SMA secure under a selective-identity attack. Then, we can apply the transform in Lemma 1 to make it SU-CMA secure. For the transformation to be identity-based as well, the Chameleon hash function needs to be published as part of mpk.

Theorem 6 (Selective Security). *$\mathsf{HIBS}^{\mathsf{su\text{-}sma}}$ is $(t, q_\mathsf{S}, \epsilon)$ SU-SMA secure under selective identity attacks if SIS is $(t + 3T_{\mathsf{ExtLattice}} + q_\mathsf{E}T_{\mathsf{Extract}} + \lambda q_\mathsf{S}T_{\mathsf{List}} + q_\mathsf{S}(T_{\mathsf{SamplePre}} + T_{\mathsf{ExtBasis}}), 1/2(1 - n^{-\omega(1)})(\epsilon - q_\mathsf{S}^2/2^n - c)/(\lambda q_\mathsf{S}))$-hard with norm bound $\nu = 2\tilde{L}\omega(\sqrt{\log(n)})^{\ell+1}\sqrt{m_1 + (\ell\kappa + \lambda + 1)m_2}^{\ell+1}$.*

Proof (Sketch). We assume that there is a successful adversary $\mathcal{A}$ against unforgeability of $\mathsf{HIBS}^{\mathsf{su\text{-}sma}}$ and construct a reduction $\mathcal{B}$ that solves SIS. The setup is as in Theorem 3 in order to be able to simulate the key extraction queries for all identities that are not a prefix of the challenge identity I. However, the input $\mathbf{A}$ of the reduction needs to be wider, namely in $\mathbb{Z}_q^{m_1+(\ell\kappa+\lambda+1)m_2}$ in order to simulate the static signature queries for identity I. Signature queries for identities $I' \neq I$ can be simulated with a known trapdoor. The overhead of the reduction is $2T_{\mathsf{ExtLattice}} + q_{\mathsf{E}}T_{\mathsf{Extract}}$ for setting up and running the extraction oracle. The overhead for signing is $\lambda q_{\mathsf{S}}T_{\mathsf{List}} + T_{\mathsf{ExtLattice}} + q_{\mathsf{S}}(T_{\mathsf{SamplePre}} + T_{\mathsf{ExtBasis}})$. As in Theorem 5, the reduction sets $\mathbf{y}$ such that it knows a preimage $\mathbf{x}$ either for EU-SMA or for SU-SMA forgers. In both cases, the reduction solves SIS. The probability that the adversary outputs a signature that yields a solution to SIS is as in Theorem 5 because we use the same signature scheme here and we can prepare for all extraction queries.

The full proof is a combination of the techniques that are used in Theorems 3 and 5. □

In consequence, $\mathsf{HIBS}^{\mathsf{su\text{-}sma}}$ is secure as long as finding shortest lattice vectors up to approximation factors $\gamma \geq 2d\widetilde{\mathcal{O}}(\sqrt{n}) = \widetilde{\mathcal{O}}(n\sqrt{n}^{\ell+3})$ is hard in the worst case in dimension n.

Secure Parameters. We let $\kappa = 40$ and $\lambda = 160$ and estimate secure parameters for $\ell = 1, 2, 3, 4$, i.e., for $2, 3, 4, 5$ levels. For (ℓ, n, q), we obtain $(1, 478, n^7)$, $(2, 730, n^9)$, $(3, 1082, n^{10})$, or $(4, 1362, n^{12})$.

5 Conclusions

We have shown three results. As a warm-up, we have constructed an adaptive-ID secure, strongly unforgeable hierarchical identity-based signature scheme in the random oracle model. Then, we have shown the first lattice-based strongly unforgeable signature scheme without Merkle trees [26] in the standard model. And, finally, we provide the first strongly unforgeable hierarchical identity-based signatures scheme in the standard model from lattices. For each construction, we have proposed a set of secure parameters based on today's knowledge about lattice reduction algorithms. Another benefit is that all of our constructions can be transferred to ideal lattices that admit short keys and efficient operations. Moreover, we can use mild, worst-case hardness assumptions in lattices as the basis of security in all constructions.

Acknowledgments

The author would like to thank Benoît Libert for a helpful discussion on HIBS. He also thanks Pierre-Louis Cayrel and Dominique Schröder for reviewing parts of this work. Furthermore, the author thanks the anonymous reviewers of PQC

2010 for their constructive comments. On of the reviewers suggested a simplification for demonstrating that [29] is not SU-CMA secure. This part of Section 4 is now much easier to understand.

References

1. Shweta Agrawal and Xavier Boyen. Identity-based encryption from lattices in the standard model. Manuscript, July 2009. Available at `http://www.cs.stanford.edu/~xb/ab09/`.
2. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108. ACM, 1996.
3. Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610. ACM, 2001.
4. Joel Alwen and Chris Peikert. Generating shorter bases for hard random lattices. Cryptology ePrint Archive, Report 2008/521, 2008. `http://eprint.iacr.org/`.
5. Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In Susanne Albers and Jean-Yves Marion, editors, *STACS*, volume 09001 of *Dagstuhl Seminar Proceedings*, pages 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2009.
6. Mihir Bellare and Sarah Shoup. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 201–216. Springer, 2007.
7. Daniel J. Bernstein, Johannes A. Buchmann, and Erik Dahmen, editors. *Post-Quantum Cryptography*. Springer, 2008.
8. Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *Eurocrypt*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.
9. Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.
10. Dan Boneh, Emily Shen, and Brent Waters. Strongly unforgeable signatures based on computational diffie-hellman. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 229–240. Springer, 2006.
11. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.
12. David Cash, Dennis Hofheinz, and Eike Kiltz. How to delegate a lattice basis. Cryptology ePrint Archive, Report 2009/351, 2009. `http://eprint.iacr.org/`.
13. Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.
14. David Galindo, Javier Herranz, and Eike Kiltz. On the generic construction of identity-based signatures with additional properties. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 178–193. Springer, 2006.
15. Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *LNCS*, pages 31–51. Springer, 2008.
16. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *STOC*, pages 197–206. ACM, 2008.

17. Shai Halevi, editor. *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *LNCS*. Springer, 2009.
18. Susan Hohenberger and Brent Waters. Short and stateless signatures from the rsa assumption. In Halevi [17], pages 654–670.
19. Eike Kiltz and Gregory Neven. Identity-based signatures. In M. Joye and G. Neven, editors, *Cryptology and Information Security Series*, volume 2, pages 31–44. IOS Press, 2008.
20. Hugo Krawczyk and Tal Rabin. Chameleon hashing and signatures. Cryptology ePrint Archive, Report 1998/010, 1998. `http://eprint.iacr.org/`.
21. Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS*. The Internet Society, 2000.
22. Gaëtan Leurent and Phong Q. Nguyen. How risky is the random-oracle model? In Halevi [17], pages 445–464.
23. Benoît Libert and Jean-Jacques Quisquater. The exact security of an identity based signature and its applications. Cryptology ePrint Archive, Report 2004/102, 2004. `http://eprint.iacr.org/`.
24. Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures, 2009. Asiacrypt 2009, to appear.
25. Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. In Ran Canetti, editor, *TCC*, volume 4948 of *LNCS*, pages 37–54. Springer, 2008.
26. Ralph C. Merkle. A certified digital signature. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *LNCS*, pages 218–238. Springer, 1989.
27. Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. Prelim. in FOCS 2002.
28. Daniele Micciancio and Oded Regev. Lattice-based cryptography. In Bernstein et al. [7], pages 147–191.
29. Chris Peikert. Bonsai trees (or, arboriculture in lattice-based cryptography), 2010. Eurocrypt 2010, to appear.
30. Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *LNCS*, pages 145–166. Springer, 2006.
31. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.
32. Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635. Springer, 2009.

A Proof of Lemma 1

Proof. Let $\mathcal{C}$ be a Chameleon hash function family. Let $\mathsf{DSig}^{\mathsf{SU\text{-}SMA}}$ be an SU-SMA secure signature scheme with $\mathsf{DSig}^{\mathsf{SU\text{-}SMA}} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$. We define $\mathsf{DSig}^{\mathsf{SU\text{-}CMA}}$ as follows.

Key Generation. Run $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{DSig}^{\mathsf{SU\text{-}SMA}}.\mathsf{Kg}(1^n)$ and $(\mathsf{C}, t) \leftarrow \mathcal{C}(1^n)$. Output the secret key (sk, t) and the public key $(\mathsf{pk}, \mathsf{C})$.

Signing. On input a secret key sk and a message $\mathfrak{m} \in \{0,1\}^*$. Choose a random $r \leftarrow \{0,1\}^n$ Call $\mathfrak{m}' \leftarrow \mathsf{C}(\mathfrak{m}, r) \in \mathcal{M}$ and $\mathfrak{s} \leftarrow \mathsf{DSig}^{\text{SU-SMA}}.\mathsf{Sign}(\mathsf{sk}, \mathfrak{m}')$. Output $(\mathfrak{s}, r)$.

Verification. On input a verification key pk, a signature $(\mathfrak{s}, r)$, and a message $\mathfrak{m}$, compute $\mathfrak{m}' \leftarrow \mathsf{C}(\mathfrak{m}, r)$ and output the result of $\mathsf{DSig}^{\text{SU-SMA}}.\mathsf{Vf}(\mathsf{pk}, \mathfrak{s}, \mathfrak{m}')$.

Observe that the modified scheme is complete.

Let T_{Sign} be the cost function for signing and T_{Inv} be the cost of calling C^{-1}. We prove that it is indeed $(t, q_{\mathsf{S}}, \epsilon)$-secure (SU-CMA) if $\mathsf{DSig}^{\text{su-sma}}$ is $(t + q_{\mathsf{S}} T_{\mathsf{Inv}}, q_{\mathsf{S}}, \epsilon)$-secure (SU-SMA) and $\mathcal{C}$ is $(t + q_{\mathsf{S}} T_{\mathsf{Sign}}, \epsilon)$-collision-resistant. Towards contradiction, assume that there is an adversary $\mathcal{A}$ against SU-CMA security of $\mathsf{DSig}^{\text{su-cma}}$ that runs in time t, makes q_{S} signature queries, and has success probability ϵ. Assume that the adversary queries the messages $\mathfrak{m}_1, \ldots, \mathfrak{m}_{q_{\mathsf{S}}}$ to its signature oracle and receives $(\mathfrak{s}_1, r_1), \ldots, (\mathfrak{s}_{q_{\mathsf{S}}}, r_{q_{\mathsf{S}}})$. Now, a *successful* adversary can be classified into one of three types, depending on its output $(\mathfrak{m}^*, \mathfrak{s}^*, r^*)$.

1. $\exists i\, \mathsf{C}(\mathfrak{m}^*, r^*) = \mathsf{C}(\mathfrak{m}_i, r_i)$
   (a) $r^* \neq r_i$ or $\mathfrak{m}^* \neq \mathfrak{m}_i$: we have a collision under C.
   (b) $r^* = r_i$ and $\mathfrak{m}^* = \mathfrak{m}_i$: we have $\mathfrak{s}^* \neq \mathfrak{s}_i$, an SU-SMA forgery.
2. $\forall i\, \mathsf{C}(\mathfrak{m}^*, r^*) \neq \mathsf{C}(\mathfrak{m}_i, r_i)$: we have an SU-SMA (even EU-SMA) forgery.

Type-1a adversaries find collisions under C, therefore the reduction has to play against collision resistance of the family $\mathcal{C}$. Type-1b adversaries find a forgery in the strong sense, i.e., the reduction plays against SU-SMA security of $\mathsf{DSig}^{\text{su-sma}}$. Type-2 adversaries $\mathcal{A}$ always output an existential forgery that refutes SU-SMA, and even EU-SMA, security of $\mathsf{DSig}^{\text{su-sma}}$. Whenever we expect the adversary to be of type 1a, we simulate the environment with the secret signing key. Otherwise, we receive the public verification key from the SU-SMA experiment and simulate the signature oracle with a trapdoor for the Chameleon has. However, the adversary's view in both reductions is indistinguishable. We describe both reductions. W.l.o.g., we assume that the type is known in advance.

Type-1a. We describe a reduction that refutes collision resistance of the Chameleon hash function family $\mathcal{C}$. The reduction receives a random element of the family $\mathcal{C}$.

Setup. Receive C and run $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{DSig}^{\text{SU-SMA}}.\mathsf{Kg}(1^n)$. Then, execute $\mathcal{A}$ on input $(\mathsf{pk}, \mathsf{C})$.

Signature Queries. On input $\mathfrak{m}$, choose a random $r \xleftarrow{\$} \{0,1\}^n$. Then, compute $c \leftarrow \mathsf{C}(\mathfrak{m}, r)$ and return the result of $\mathsf{DSig}^{\text{SU-SMA}}.\mathsf{Sign}(\mathsf{sk}, c)$ and r.

Output. The adversary $\mathcal{A}$ outputs $(\mathfrak{m}^*, \mathfrak{s}^*, r^*)$. Output the collision $(\mathfrak{m}, r), (\mathfrak{m}^*, r^*)$.

Analysis. Observe that the environment of $\mathcal{A}$ is perfectly simulated. By definition, a type-1a forger outputs a signature $(\mathfrak{s}^*, r^*)$ and a message $\mathfrak{m}^*$ such that $\mathsf{C}(\mathfrak{m}^*, r^*) = \mathsf{C}(\mathfrak{m}_i, r_i)$ for some i but with $\mathfrak{m}^* \neq \mathfrak{m}_i$ or $r^* \neq r_i$. Therefore, the output is a valid collision under C and the reduction is successful whenever $\mathcal{A}$. The overhead of the reduction is dominated by q_{S} calls of Sign.

Type-1b, Type-2. We describe a reduction that refutes SU-SMA security of $\mathsf{DSig}^{\mathsf{SU\text{-}SMA}}$ The reduction has access to an external signature oracle during the setup phase.

Setup. Choose q_{S} message $\{\mathfrak{m}_i\}_1^{q_{\mathsf{S}}}$ and send them to the signature oracle. Receive $\{\mathfrak{s}_i\}_1^{q_{\mathsf{S}}}$ and pk. Choose $(\mathsf{C}, t) \leftarrow \mathcal{C}(1^n)$ and execute $\mathcal{A}$ on input $(\mathsf{pk}, \mathsf{C})$. Set up a counter $\imath \leftarrow 0$.

Signature Queries. On input $\mathfrak{m}$, increment $\imath$ and compute $r_i \leftarrow \mathsf{C}^{-1}(\mathfrak{m}, \mathfrak{m}_i)$. Return $(r_i, \mathfrak{s}_i)$.

Output. The adversary $\mathcal{A}$ outputs $(\mathfrak{m}^*, \mathfrak{s}^*, r^*)$. Output the forgery $\mathsf{C}(\mathfrak{m}^*, r^*), \mathfrak{s}^*)$.

Analysis. Observe that the environment of $\mathcal{A}$ is perfectly simulated. By definition, a type-1b forger outputs a signature $(\mathfrak{s}^*, r^*)$ and a message $\mathfrak{m}^*$ such that $\mathsf{C}(\mathfrak{m}^*, r^*) = \mathsf{C}(\mathfrak{m}_i, r_i)$ for some i and $\mathfrak{m}^* = \mathfrak{m}_i$, $r^* = r_i$. Since it is a forgery, we have that $\mathfrak{s}^* \neq \mathfrak{s}_i$. Therefore, $\mathfrak{s}^*$ is a forgery for $\mathsf{C}(\mathfrak{m}_i, r_i)$ in the strong sense. A type-2 forger outputs a pair $(\mathfrak{m}^*, r^*)$ such that $\mathsf{C}(\mathfrak{m}^*, r^*) \neq \mathsf{C}(\mathfrak{m}_i, r_i)$ for all $i = 1, \ldots, q_{\mathsf{S}}$. Thus, the reduction outputs a forgery in the existential sense. In either case, the reduction is successful whenever $\mathcal{A}$ is. The overhead is dominated by calling C^{-1}. □

B Proof of Theorem 3

Proof. We assume that there is a successful adversary $\mathcal{A}$ against unforgeability of $\mathsf{HIBS}^{\mathsf{GPV}}$ and construct a reduction $\mathcal{B}$ that solves SIS.

Setup. On input $\mathbf{A} = \mathbf{A}^\star \circ \mathbf{U}_1 \circ \cdots \circ \mathbf{U}_{\ell\kappa} \in \mathbb{Z}_q^{m_1+(\ell\kappa+1)}$, invoke the adversary and receive the challenge identity $I = I_1 \circ \ldots \circ I_k$, $l_I \leq \ell\kappa$. Setup matrices $\mathbf{A}_i^{(I_i)} \leftarrow \mathbf{U}_i$ for $i = 1, \ldots, l_I$ and matrices $\mathbf{A}_i^{(1-I_i)} \leftarrow \mathsf{ExtLattice}(\mathbf{A}^\star \circ \mathbf{A}_1^{(I_1)} \circ \cdots \circ \mathbf{A}_{i-1}^{(I_{i-1})}, m_2)$ for $i = 1, \ldots, l_I$. If $l_I < \ell\kappa$, set $\mathbf{A}_{l_I+1}^{(0)} \leftarrow \mathsf{ExtLattice}(\mathbf{A}^\star \circ \mathbf{A}_1^{(I_1)} \circ \cdots \circ \mathbf{A}_{l_I}^{(l_I)}, m_2)$ and $\mathbf{A}_{l_I+1}^{(1)} \leftarrow \mathsf{ExtLattice}(\mathbf{A}^\star \circ \mathbf{A}_1^{(I_1)} \circ \cdots \circ \mathbf{A}_{l_I}^{(l_I)}, m_2)$. The remaining matrices $\mathbf{A}_i^{(0)}, \mathbf{A}_i^{(1)}$, $l_I + 1 < i \leq \ell\kappa$ are chosen uniformly at random. The master public key is $\langle \mathbf{A} \rangle := \left\{(\mathbf{A}_i^{(0)}, \mathbf{A}_i^{(1)})\right\}_1^{\ell\kappa}$. Set up a list of random oracle queries L_{H} and run $\mathcal{A}$ on input $\langle \mathbf{A} \rangle$.

Key Extraction Oracle Queries. On input ID, the oracle acts as in the original scheme. However, the oracle answers $\perp$ if ID is a prefix of I.

Random Oracle Queries. On input $\mathfrak{m}, r, \mathsf{ID}$ search the list L_{H}. If there is a corresponding entry $(\mathfrak{s}, \mathfrak{m}, r, \mathsf{ID}, h)$, return h. If not, use $\mathsf{SampleDom}$ in dimension $m_1 + (k\kappa + 1)m_2$ with basis length $\tilde{L}_k$ to sample $\mathfrak{s}$, where $k \geq 0$ is the level of ID. Return $h = \mathsf{Eval}(\mathbf{A}_{\mathsf{ID}}, \mathfrak{s})$ and store $(\mathfrak{s}, \mathfrak{m}, r, \mathsf{ID}, h)$.

Signature Queries. On input $\mathfrak{m}, \mathsf{ID}$, choose $r \xleftarrow{\$} \{0,1\}^n$ and call $h \leftarrow \mathsf{H}(\mathfrak{m}, r, \mathsf{ID})$ and return $\mathfrak{s}$ from the corresponding entry in L_{H}.

Output. When $\mathcal{A}$ halts, it outputs a forgery $\mathfrak{m}^*, \mathfrak{s}^*, r^*$ for the challenge identity I. W.l.o.g., there already is a tuple $(\mathfrak{s}, \mathfrak{m}, r, I, h)$ in list L_H. The reduction computes $\mathbf{v} = \mathfrak{s}^* - \mathfrak{s}$ and outputs a vector $\mathbf{w} \in \Lambda_q^\perp(\mathbf{A})$ by suitably rearranging and paddding the entries of $\mathbf{v}$.

Analysis. First of all, observe that all oracles are efficiently simulated. If $\mathcal{A}$ runs in time t, then the reduction runs in time $t' = t + 2T_{\mathsf{ExtLattice}} + q_{\mathsf{E}}T_{\mathsf{Extract}} + (q_{\mathsf{H}} + q_{\mathsf{S}})T_{\mathsf{H}}$. Next, notice that all oracle queries are answered as expected. In particular, the signature oracle can answer all queries for the challenge identity without knowing the secret key and key extraction outputs matrices from the correct distribution. There is only one deviation. In the case that the honest signer would sign the hash $\mathsf{H}(\mathfrak{m}, r, \mathsf{ID})$ twice, the reduction acts stateful and does not return different signatures. However, this happens with negligible probability $\leq q_{\mathsf{S}}^2/2^n$. As for key extraction, notice that the reduction knows a trapdoor for all prefixes of all identities except for the challenge identity. Thus, it can simulate key extraction.

The output $(\mathfrak{m}^*, \mathfrak{s}^*, r^*)$ and the entry $(\mathfrak{s}, \mathfrak{m}, r, I, h)$ in L_H are used to find a vector $\mathbf{v} = \mathfrak{s}^* - \mathfrak{s}$ with $\mathbf{A}_I\mathbf{v} \equiv \mathbf{0}$ and $\|\mathbf{v}\|_2 \leq 2d$. Now, $\mathbf{A}_I$ is a concatenation of certain blocks in $\mathbf{A} = \mathbf{A}^\star \circ \mathbf{U}_1 \circ \cdots \circ \mathbf{U}_{\ell\kappa}$. Therefore, we can build $\mathbf{w}$ by padding and rearranging $\mathbf{v}$ such that $\mathbf{A}\mathbf{w} \equiv \mathbf{0}$. The probability that $\mathbf{w} = \mathbf{0}$ is at most $n^{-\omega(1)}$. Thus, $\mathcal{B}$ solves SIS with $\nu = 2d$ with non-negligible probability. □

C Proof of Theorem 5

Proof. We assume that there is a successful adversary $\mathcal{A}$ against unforgeability of $\mathsf{DSig}^{\mathsf{su\text{-}sma}}$ and construct a reduction $\mathcal{B}$ that solves SIS. The reduction receives a list of messages and returns a list of signatures along with the public key. Then, the adversary either outputs a weak forgery (EU-SMA) or a strong forgery (SU-SMA). The reduction guesses the type of adversary beforehand and solves SIS is both cases. For the EU-SMA forger, the reduction knows $\mathbf{x}$ with $\mathbf{A}^\star\mathbf{x} \equiv \mathbf{y}$ and for the SU-SMA forger, it prepares $\mathbf{A}_{h^*}\mathbf{x} \equiv \mathbf{y}$ by guessing the right message with probability $1/q_{\mathsf{S}}$.

Setup. On input $\mathbf{A} = \mathbf{A}^\star \circ \mathbf{U}_1^{(0)} \circ \mathbf{U}_1^{(1)} \circ \cdots \circ \mathbf{U}_\lambda^{(0)} \circ \mathbf{U}_\lambda^{(1)} \in \mathbb{Z}_q^{m_1+(2\lambda+1)m_2}$, invoke the adversary and receive a list of messages $\mathfrak{m}_1, \ldots, \mathfrak{m}_{q_{\mathsf{S}}}$. It flips a coin $a \xleftarrow{\$} \{0,1\}$ (0=EU, 1=SU) and $i^* \xleftarrow{\$} \{1, \ldots, q_{\mathsf{S}}\}$. Choose $r_1, \ldots, r_{q_{\mathsf{S}}} \xleftarrow{\$} \{0,1\}^n$ and compute $h^{(i)} \leftarrow \mathsf{H}(\mathfrak{m}_i, r_i)$ for all i. If there is a collision under H, choose a fresh set of r_i's. Let $\langle\pi\rangle := \{\pi_i\}_1^p$ be the set of all strings $\pi \in \{0,1\}^\lambda$ such that $\pi \not\sqsubset h^{(j)}$ for $j \in \{1, \ldots, q_{\mathsf{S}}\} \setminus \{ai^*\}$, and $\pi_i \not\sqsubset \pi_j$ for all distinct pairs (π_i, π_j) in $\langle\pi\rangle$. The set $\langle\pi\rangle$ contains at most λq_{S} elements. Now, randomly select an element $\pi \xleftarrow{\$} \langle\pi\rangle$, which represents the challenge subtree. Let $|\pi| = l_\pi$.

Setup matrices $\mathbf{B}_i^{(\pi_i)} \leftarrow \mathbf{U}_i^{(0)}$ for $i = 1, \ldots, l_\pi$ and matrices $\mathbf{B}_i^{(b)} \leftarrow \mathbf{U}_i^{(b)}$ for $b \in \{0,1\}$ and $i = l_\pi + 1, \ldots, \lambda$.

Then, prepare the public key for signing by computing $\mathbf{B}_i^{1-\pi_i}$ and $\mathbf{S}_i$ via $\mathsf{ExtLattice}(\mathbf{A}^\star \circ \mathbf{B}_1^{(\pi_1)} \circ \cdots \circ \mathbf{B}_{i-1}^{(\pi_{i-1})}, m_2)$ for $i = 1, \ldots, l_\pi$.

If $a = 0$, use $\mathsf{SampleDom}$ with $s = \omega(\sqrt{\log(n)})\tilde{L}$ to sample a vector $\mathbf{x} \in \mathbb{Z}^{m_1+m_2}$ and compute $\mathbf{y} \leftarrow \mathbf{A}^\star\mathbf{x} \bmod q$. For $a = 1$, sample $\mathbf{x} \in \mathbb{Z}^{m_1+(\lambda+1)m_2}$, and set $\mathbf{y} \leftarrow \mathbf{A}_{h^{(i^*)}}\mathbf{x} \bmod q$. For each hash value $h^{(i)}$, let j be the smallest

index with $h_j^{(i)} \neq \pi_j$. Since $\left\|\tilde{\mathbf{S}}_i\right\| \leq \tilde{L}$, we let $s = \omega(\sqrt{\log(n)})\tilde{L}$ and compute the signature $\mathfrak{s}_i \leftarrow \mathsf{SamplePre}(\mathsf{ExtBasis}(\mathbf{S}_j, \mathbf{A}_{h^{(i)}}), s, \mathbf{y})$.

The public key comprises $\mathbf{A}^\star$, $\mathbf{y}$, and $\langle \mathbf{B} \rangle := \left\{(\mathbf{B}_i^{(0)}, \mathbf{B}_i^{(1)})\right\}_1^\lambda$ and the reduction returns the public key and the list of signatures to $\mathcal{A}$.

Output. Eventually, $\mathcal{A}$ outputs its forgery $(\mathfrak{m}^*, \mathfrak{s}^*, r^*)$. If $a = 0$, the reduction pads $\mathbf{x}$ to form $\mathbf{x}' \in \mathbb{Z}^{m_1+(\lambda+1)m_2}$ and outputs $\mathbf{x}' - \mathfrak{s}^*$. In case $a = 1$, the reduction outputs $\mathbf{x} - \mathfrak{s}^*$. In either case, the output needs to be suitably padded and rearranged to solve SIS for $\mathbf{A}$.

Analysis. First of all, the setup phase is efficient. As in [29], the set $\langle \pi \rangle$ is of polynomial size in n. Finding each of the strings π in the set costs T_{List}. If $\mathcal{A}$ runs in time t, then the reduction runs in time $t + \lambda q_{\mathsf{S}} T_{\mathsf{List}} + T_{\mathsf{ExtLattice}} + q_{\mathsf{S}}(T_{\mathsf{SamplePre}} + T_{\mathsf{ExtBasis}})$ plus some minor overhead. Next, notice that the setup phase outputs valid signatures and a public key that is indistinguishable from uniform. The only deviation from the real scenario is that there might be a collision under H, which happens with probability $\leq c$. It is also possible that the signer chooses the same r when queried with a message $\mathfrak{m}$ twice. This happens with probability at most $q_{\mathsf{S}}^2/2^n$.

As for the output $(\mathfrak{m}^*, \mathfrak{s}^*, r^*)$ of $\mathcal{A}$, one of two things happens with probability $1/2$: There is an index i with $h^* = \mathsf{H}(\mathfrak{m}^*, r^*) = h^{(i)}$. Then, the reduction must have guessed $i^* = i$ correctly (probability $1/q_{\mathsf{S}}$) and $\mathcal{A}$ has to output a signature $\mathfrak{s}^* \neq \mathbf{x}$ (probability $1 - n^{-\omega(1)}$). If there is no such i, the reduction works if $\pi \sqsubset h^*$ (probability $1/(\lambda q_{\mathsf{S}})$) and the padded $\mathbf{x}' \neq \mathfrak{s}^*$ (probability $1 - n^{-\omega(1)}$). Therefore, the total success probability is $\epsilon' = 1/2(\epsilon - q_{\mathsf{S}}^2/2^n - c)/(\lambda q_{\mathsf{S}})(1 - n^{-\omega(1)})$. $\square$