Anamorphic Signatures With Dictator & Recipient Unforgeability for Long Messages

Amit Deo and Benoît Libert

Zama, France

Abstract. Anamorphic signatures (Kutylowski *et al.*, Crypto'23) provide a way to covertly use encryption by hiding ciphertexts inside digital signatures without a dictator noticing. Recently (Asiacrypt'24), Jaeger and Stracovsky advocated stronger security notions for the primitive. Their notion of dictator unforgeability requires a dictator's inability to produce fresh signatures that decrypt to a meaningful covert message. The notion of recipient unforgeability requires that anamorphic receivers cannot forge signatures even after having observed anamorphic signatures on messages of their choice. To date, the known schemes satisfying all these properties simultaneously rely on the "randomness replacement" technique. As a result, they are restricted to short anamorphic messages either because their anamorphic decryption mechanism involves an exhaustive search step, or because they embed the anamorphic plaintext in a public random salt (which is typically short in compatible signature schemes like RSA-PSS). In this paper, we present anamorphic signatures that depart from the randomness replacement paradigm and make it possible to encrypt longer anamorphic plaintexts. We show that (generalized) Okamoto-Schnorr signatures, as well as GQ and 2^t-root signatures all have anamorphic modes satisfying the three desired security properties. The ratio between the lengths of anamorphic plaintexts and signatures can even be very close to 1 for appropriate parameters. We also discuss an extension to Lyubashevsky's lattice-based signatures.

Keywords. Anamorphic cryptography, signatures, encryption.

1 Introduction

As introduced by Persiano, Phan and Yung [53], anamorphic cryptography is a paradigm consisting in embedding hidden communication channels inside well-known cryptographic schemes. The motivation is to allow parties to privately communicate under a dictatorship that seizes users' private keys. Covert messages should be recoverable by approved recipients, but remain undetectable to a dictator knowing users' secret keys. Persiano *et al.* [53] formalized the security of anamorphic *encryption* and demonstrated the existence of anamorphic channels in well-known public key encryption (PKE) schemes. Since its inception, anamorphic cryptography has received much attention, notably with the elaboration of more advanced properties [18,8,54].

Unfortunately, anamorphic *encryption* alone may not suffice to enable private

communications in the presence of a dictator. Indeed, one can imagine a world where encryption is entirely banned. Alternatively, we can think of a massive surveillance environment where an all-powerful authority decrypts all communication and then re-encrypts for the intended recipient.

Fortunately, anamorphic ciphertexts are not the only possible medium for covert messages in the presence of a dictator. In particular, the notion of *anamorphic signatures*, introduced by Kutylowski *et al.* [42], hides covert messages in innocent-looking digital signatures. This primitive aims at enabling secret communications in a dictatorship that forbids encryption, but still permits authentication. Informally, a signature scheme $\Pi = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver})$ is called anamorphic if there exists a so-called "anamorphic triplet" $\Sigma = (\mathsf{aGen}, \mathsf{aSig}, \mathsf{aDec})$. Algorithm aGen outputs a tuple $(\mathsf{avk}, \mathsf{ask}, \mathsf{dk})$ where: (i) avk is an anamorphic verification key, (ii) ask is an anamorphic signing key, (iii) dk is a double key that allows encrypting/decrypting covert messages. Whenever the anamorphic signing algorithm $\mathsf{aSig}(\mathsf{ask}, \mathsf{dk}, \mu, \hat{\mu})$ outputs a signature σ (on a public message μ), the anamorphic decryption procedure should recover the covert message $\hat{\mu} = \mathsf{aDec}(\mathsf{dk}, \sigma)$. Yet, a dictator that is given $(\mathsf{avk}, \mathsf{ask})$ should be unable to tell whether $(\mathsf{avk}, \mathsf{ask})$ was produced by aGen or the regular KGen algorithm, nor whether aSig is being used instead of Sig. This requirement captures the fundamental definition of anamorphic signatures [42], which we will refer to as the *stealthiness* property.

The terminology of Kutylowski *et al.* [42] distinguishes *symmetric* anamorphic signatures from *privately anamorphic* ones. In the former, anamorphic signers have to fully trust anamorphic decryptors since the double key dk can be used to generate signatures on behalf of the signer. In the latter, the double key does not provide signing capabilities. In all cases, the double key is kept away from the dictator as no security is possible otherwise.

Kutylowski *et al.* [42] showed that many existing digital signatures can be endowed with an anamorphic mode using the *randomness replacement* technique (or RRep for short). In short, it consists in encrypting the anamorphic message using a symmetric encryption scheme with pseudorandom ciphertexts. The resulting ciphertext is then used as the randomness of a probabilistic signing algorithm. If the underlying signature scheme allows extracting the signer's randomness (possibly with the help of the signer's secret key in symmetric anamorphic schemes) from signatures, the recipient can recover the pseudorandom ciphertext from which it can then obtain the anamorphic plaintext using the double key. The pseudorandomness of symmetric ciphertexts then provides stealthiness.

Kutylowski *et al.* [42] also showed that any anamorphic signature where dk is independent of the signing key ask is privately anamorphic as long as the underlying signature is unforgeable in the usual sense.

Dictator and Recipient Unforgeability. Hiding (the very existence of) the double key from the dictator is necessary to obtain stealthiness. However, beyond simply detecting the use of an anamorphic mode, a dictator might attempt to forge fake covert messages on behalf of a signer. By observing recipients' reaction, it may then infer that they are indeed in possession of a double key.

An anamorphic signature preventing this is said to be *dictator unforgeable*, as defined by Jaeger and Stracovsky [37]. This security notion says that $\mathsf{aDec}(\mathsf{dk}, \mu, \sigma) = \perp$ for any pair (μ, σ) produced by a dictator[1], even when the latter has access to an anamorphic signing and an anamorphic decryption oracle. Dictator unforgeability can be seen as a strong flavor of *robustness* in the context of anamorphic signatures. As defined by Banfi *et al.* [8] for anamorphic encryption, robustness ensures that normal encryptions are never mistakenly interpreted as carrying covert anamorphic messages.

It is also important that recipients armed with dk cannot behave maliciously. While the notion of private anamorphism [42] prevents recipients knowing only the double key from forging signatures, it does not guarantee unforgeability in the realistic setting where the recipient has access to an anamorphic signing oracle (besides the normal signing oracle). As defined by Jaeger and Stracovsky [37], the stronger notion of *recipient unforgeability* [37] requires that recipients remain unable to produce non-trivial valid signatures, even when they are given access to dk and many normal/anamorphic signatures for messages of their choice.

Jaeger and Stracovsky [37] formalized dictator and recipient unforgeability as natural strengthenings of prior robustness/private anamorphism properties [8,42]. They showed that the RRep technique is *not* recipient unforgeable in general. On the positive side, they proved that the RRep technique provides recipient unforgeability when applied to signature schemes that (besides the ability, called $\emptyset$-*recoverability*, to recover the signer's randomness from the signature without knowing the secret signing key) have the special property of being existentially unforgeable under *chosen message and randomness* attack. This notion, dubbed SUF-CRA security in [37], extends the usual definition of existential unforgeability under chosen-message attacks [31] by additionally allowing the adversary to choose the randomness used by its signing oracle. In particular, the RSA-PSS signature scheme [12] was shown [37] to satisfy this enhanced notion of SUF-CRA security and it is also $\emptyset$-recoverable.

Jaeger and Stracovsky [37] also proved that a general transformation called "randomness identification with a PRF" (or "RIdP", as initially proposed by Banfi *et al.* [8] in the encryption setting) is both stealthy and dictator unforgeable modulo some mild requirements on the underlying signature. In addition, the RIdP transform provides recipient unforgeability if the underlying signature scheme is secure in the SUF-CRA sense. So, plugging RSA-PSS signatures into the RIdP transform provides an anamorphic signature with the three desired security properties (namely, stealthiness and dictator/recipient unforgeability).

Limitations of RIdP and RRep Transforms. The RIdP transform is restricted to anamorphic messages of length $O(\log \lambda)$ (if λ is the security parameter) as the anamorphic decryption mechanism aDec involves an exhaustive search over the anamorphic message space. In addition, the aDec and aSig algorithms must share a synchronized counter. The reason is that the transform [37, Figure 5] replaces the signer's randomness by a pseudorandom function (PRF) of the anamorphic

[1] Note that the aDec syntax of Jaeger and Stracovsky takes μ as input whereas the original syntax of Kutylowski et al. does not.

message and a counter. The aDec algorithm then loops over all possible covert messages in order to determine which one is consistent with the signature.

An alternative way to obtain the three targeted properties from RSA-PSS (or any other $\emptyset$-recoverable signature that is secure in the SUF-CRA sense) is to use the RRep transform and make it dictator unforgeable by replacing the public random salt with a symmetric *authenticated* encryption ciphertext. While this approach works in theory, the length of anamorphic message is limited by the length of the random salt, which is usually very short in RSA-PSS.

In more detail, the authenticated encryption is embedded into the κ-bit random salt of RSA-PSS. In order to decrypt without a synchronized counter, the ciphertext must contain a λ-bit nonce and a λ-bit MAC/PRF for authentication, leaving $\kappa - 2\lambda$ bits to carry a covert message. If we define the bandwidth rate as the ratio between the length of anamorphic messages and the signature length, any instantiation with $\kappa > 2\lambda$ has bandwidth rate $(\kappa - 2\lambda)/|N|$ where $|N|$ is the bit-length of the RSA modulus. When $\kappa \leq 2\lambda$ (e.g., in practical deployments using a salt length of 20 or 32 bytes), there is no room for anamorphic message bits unless the security parameter is reduced.

We note that, in theory, the random salt of PSS can have up to $|N| - 2\lambda$ bits. Nevertheless, the standard specification [39,57] considers a random salt of at most 32 bytes at the 128-bit security level. Coron showed [23] that there is no motivation for using a longer salt as it does not make the security proof any tighter. The same holds for other signature schemes (e.g., [23,30]) involving a public random salt,[2] let alone with SUF-CRA security.

1.1 Contributions

We show that several well-known signature schemes admit anamorphic triplets that simultaneously satisfy the notions of dictator and recipient unforgeability in addition to the original notion of stealthiness [42]. In contrast with the RIdP transform, the anamorphic signing and decryption algorithms are stateless (and do not use a synchronized counter) and anamorphic messages are not restricted to live in a polynomial-size set. Moreover, the length of anamorphic plaintexts is not limited by that of a short salt/randomness. Our constructions are proven secure in the random oracle model under standard assumptions like the discrete logarithm, RSA/ϕ-hiding, factoring, or even lattice assumptions. They all support anamorphic messages of polynomial length. Except for the lattice-based scheme, they can all be tuned to have high bandwidth.

Specifically, we provide anamorphic triplets for Okamoto-Schnorr signatures [51] and their generalization proposed by Alwen *et al.* [6], Guillou-Quisquater (GQ) signatures [33], Ong-Schnorr [52] and similar 2^t-root signatures [47]. We also provide a construction from Lyubashevsky's lattice-based signature [45], which is based on the hardness of the Short Integer Solution (SIS) [3] problem.

In generalized Okamoto-Schnorr [6] and GQ/2^t-root anamorphic signatures

[2] Katz and Wang [41] even showed that the salt of Probabilistic Full Domain Hash can be as short as one bit.

[33,52], the bandwidth rate becomes close 1 when we increase the security parameter or other parameters of the scheme. Concretely, for GQ signatures at the 128-bit security level, it can be as large as 88%.

In our anamorphic Okamato-Schnorr signatures, the online/offline property [26,60] of the normal signing algorithm carries over to the anamorphic mode. In the case of GQ and 2^t-root signatures, the anamorphic channels can be generated as anamorphic extensions [8] of an existing normal key pair.

Conceptually, our anamorphic triplets depart from [42,37] in that they are not based on the randomness-replacement technique. Also, they do not rely on signature schemes that provide unforgeability under chosen message and randomness attacks. Indeed, none of the signature schemes we start from remain secure against an adversary that dictates the randomness of the signing oracle.

Since we are only aware of a small number of signature schemes with $\mathsf{SUF\text{-}CRA}$ security, we believe it is useful to explore a different method of endowing digital signatures with anamorphic channels. Our constructions follow such a different route and build on the observation that, besides the regular signing key, the underlying digital signatures admit some additional trapdoor information that can be leveraged by the anamorphic signing algorithm.

1.2 Technical Overview

Our constructions from Okamoto-Schnorr and GQ signatures proceed in different ways. The former encodes the anamorphic message in the response of the Σ-protocol while the latter embeds it in the first message of the prover.

(Generalized) Okamoto-Schnorr Signatures. Okamoto-Schnorr signatures [51] make use of a Σ-protocol proving knowledge of $(x_1, x_2) \in \mathbb{Z}_q^2$ such that $X = g^{x_1} \cdot h^{x_2}$, where g, h generate a cyclic group $\mathbb{G}$ of prime order q. Their generalization to more than two generators [6,27] was shown to offer stronger resilience to side-channel attacks and leakage on the secret key [6,27]. For generators $(g_1, \ldots, g_{n+1}) \in \mathbb{G}^{n+1}$, the verification key consists of $\mathsf{vk} = \prod_{i=1}^{n+1} X_i \triangleq X$, where $X_i = g_i^{x_i}$ for each i and $\mathsf{sk} = (x_1, \ldots, x_{n+1}) \in \mathbb{Z}_q^{n+1}$ is the secret key. Note that the signer may store the additional information $(\omega_2, \ldots, \omega_{n+1}) \in \mathbb{Z}_q^n$ such that $g_i = g_1^{\omega_i}$ for each $i \in [2, n+1]$, although the usual signing algorithm does not use it. For a message μ, the normal signing algorithm $\mathsf{Sig}(\mathsf{sk}, \mu)$ generates a Schnorr-like proof [58] made of a tuple $(c, s_1, \ldots, s_{n+1}) \in \mathbb{Z}_q^{n+2}$ such that $c = H(R, \mu)$, where $R = X^{-c} \cdot \prod_i g_i^{s_i}$ and $H : \{0,1\}^* \to \mathbb{Z}_q$ is a hash function.

Our anamorphic triplet (Section 3.1) makes use of the trapdoor information $(\omega_i)_{i \in [2,n+1]}$. To obtain a good bandwidth rate, we embed an authenticated encryption scheme in the signature components $(s_i)_{i \in [2,n+1]}$ and we use $R \triangleq \prod_i g_i^{r_i}$ as an authenticated encryption nonce. Namely, $(s_2, \ldots, s_n)$ are used to encrypt anamorphic message components $(\hat{\mu}_i)_{i=1}^{n-1}$ as $s_i = \hat{\mu}_{i-1} + \mathsf{PRF}_K(R, i) \bmod q$ for each $i \in [2, n]$. The signature component s_{n+1} is then used to carry an authentication tag together with an additional anamorphic message slot $\hat{\mu}_n$. One detail is that s_{n+1} is split into a λ-bit tag and a $(\lceil \log(q) \rceil - \lambda)$-bit message carrying component (and therefore does not fill the entire space $\mathbb{Z}_q$). However,

a statistical argument shows that this discrepancy goes un-noticed as long as $2^{\lfloor \log(q) \rfloor} < q < 2^{\lfloor \log(q) \rfloor} + 2^{\lambda}$.

We are able to embed anamorphic plaintexts without a dictator noticing because the tuple $(R, s_2, \ldots, s_{n+1})$ is uniformly distributed for normal signatures and s_1 is uniquely defined conditionally on this tuple. In particular, given r such that $R = g_1^r$ and ω_i such that $g_i = g_1^{\omega_i}$ for each $i \in [2, n+1]$, we can write

$$s_1 = r + c \cdot \left(x_1 + \sum_{i=2}^{n+1} \omega_i x_i \right) - \sum_{i=2}^{n+1} \omega_i s_i. \tag{1}$$

The scheme thus naturally provides the stealthiness property. Anamorphic keys $(\mathsf{avk}, \mathsf{ask})$ are identical to normal secret keys. In the anamorphic mode, however, $\mathsf{tk} = (\omega_i)_{i=2}^{n+1}$ is stored as a trapdoor key while the double key held by recipients is the authenticated encryption secret key K. The anamorphic signing algorithm aSig then samples $r \hookleftarrow U(\mathbb{Z}_q)$, sets R as a nonce, computes $(s_2, \ldots, s_{n+1})$ to be an authenticated encryption of $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_{n+1})$, computes $c = H(R, \mu)$ and then computes s_1 according to (1). Dictator unforgeability is obtained via a PRF (which serves as a MAC in the encrypt-then-MAC paradigm) contained in s_{n+1}.

When it comes to proving recipient unforgeability, we show that both normal and anamorphic signatures can be simulated without the secret key component x_1 by programming the random oracle (we thus depart from the security proof of Okamoto-Schnorr signatures, where the signing oracle does not have to program the random oracle). We then show that a forging recipient can be turned into an algorithm that solves the discrete logarithm instance $X_1 = g_1^{x_1}$.

Interestingly, only the proof of recipient unforgeability relies on random oracles. The security properties against a dictator (i.e., stealthiness and dictator unforgeability) only rely on pseudorandom functions. One caveat is that recipient unforgeability is only proven for signatures on *new* messages (while [37] proves strong unforgeability). Strong recipient unforgeability can be proven using an authenticated encryption scheme in the random oracle model.

If we use a λ-bit Fiat-Shamir challenge $c = H(R, \mu)$ (instead of hashing on $\mathbb{Z}_q$), the bandwidth rate is then approximately $(n \log(q) - \lambda)/((n+1)\log(q) + \lambda)$. Since q is around $2^{2\lambda}$, the rate is approximately $1 - \frac{2}{n+1.5}$ and approaches 1 for larger values of n (e.g., in instantiations with stronger leakage resilience).

GQ AND 2^t-ROOT SIGNATURES. The verification key of GQ signatures [33] is of the form $\mathsf{vk} = (N, e, y)$ where $N = pq$ is an RSA modulus, e is a prime exponent and $y = x^e$, for some $x \hookleftarrow U(\mathbb{Z}_N^*)$, which is part of the secret key $\mathsf{sk} = (x, p, q)$. To sign a message μ, the signer computes $a = r^e \bmod N$, for a random $r \in \mathbb{Z}_N^*$, which is hashed to $c = H_{\mathsf{FS}}(a, \mu)$ before computing $z = r \cdot x^c \bmod N$ so as to obtain a signature $(c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$ satisfying $c = H_{\mathsf{FS}}(z^e \cdot y^{-c} \bmod N, \mu)$.

In anamorphic mode (Section 3.2), we use the prover message $a \in \mathbb{Z}_N^*$ to embed a random-looking authenticated encryption of a covert message $\hat{\mu}$. The covert ciphertext contains a nonce $\tilde{a}$ (in the most significant bits of a), an encrypted message $\bar{a}_0 = \hat{\mu} \oplus h(k_m, \tilde{a})$, and an authentication tag $\bar{a}_1 = H(K, \bar{a}_0, \mu, \tilde{a})$, where k_m and K are secret keys contained in dk. The signer then computes

$c = H_{\mathsf{FS}}(a, \mu)$ before computing an e-th root $z = (a \cdot y^c)^{1/e} \bmod N$.

The proof of stealthiness follows naturally from the pseudorandomness of $h(k_m, \cdot)$ and $H(K, \cdot, \cdot, \cdot)$ since the anamorphic keys $(\mathsf{avk}, \mathsf{ask})$ are identically distributed to real keys $(\mathsf{vk}, \mathsf{sk})$. In order for the proof of dictator unforgeability to work out, we crucially rely on the fact that the signed message μ is included in the input of the PRF $H(K, \cdot, \cdot, \cdot)$ that computes the authentication tag.

When it comes to proving recipient unforgeability, we need to model the functions $h(k_m, \cdot)$ and $H(K, \cdot, \cdot, \cdot)$ as random oracles. The reason is that the anamorphic signing oracles is simulated by first using the HVZK simulator of the Σ-protocol to compute $a = z^e \cdot y^{-c} \bmod N$ (and program $c = H_{\mathsf{FS}}(a, \mu)$ in the usual way). Since the simulated a is uniformly distributed over $\mathbb{Z}_N^*$, we need to program $\bar{a}_1 = H(K, \bar{a}_0, \mu, \tilde{a})$ and $\bar{a}_0 = \hat{\mu} \oplus h(k_m, \tilde{a})$ so as to make them consistent with a. We note that Jaeger and Stracovsky [37, Section 5.6] used a similar technique to achieve recipient unforgeability by introducing a random oracle in anamorphic Boneh-Boyen signatures [14].

When the regular and anamorphic signing oracles are both simulated without using the secret key, we can then prove security by adapting the usual proofs of unforgeability. The Forking Lemma [55] can be used to prove recipient unforgeability under the RSA assumption. However, we obtain a tighter security proof under the ϕ-hiding assumption [17] using ideas from [2,1].

Let $|N|$ denote the bit-length of N. The signature length is then $|N| + \lambda$ if we use a λ-bit c. Anamorphic messages can have length up to $|N| - 2\lambda - 1$ bits, leading to a bandwidth rate of $1 - (3\lambda + 1)/(|N| + \lambda)$. The latter approaches 1 as λ increases since N should have $\Omega(\lambda^3/\mathrm{polylog}(\lambda))$ bits to resist GNFS attacks.

Our anamorphic 2^t-root signatures [52] (in Appendix E) proceeds similarly since the underlying scheme is a GQ signature where the prime exponent e is replaced by a power of 2. For technical reasons, we need to restrict the scheme to use Blum integers as in [61]. One difference is that we cannot immediately obtain strong dictator/recipient unforgeability as the original 2^t-root scheme is not strongly unforgeable itself. Nevertheless, the original scheme can be made strongly unforgeable using a simple (perhaps folklore) method of restricting the prover's response z to live in $\{0, \ldots, N-1\} \cap \mathbb{Z}_N^*$ and have Jacobi symbol 1. This trick also provides strong dictator/recipient unforgeability.

We note that a similar anamorphic channel can be obtained in several variants of 2^t-root signatures, including Micali's signature scheme [47] and the GQ2 signatures included in the ISO/IEC1488-2 standard [36].

Lattice Signatures. We finally show (in Appendix B) that Lyubashevsky's signature [45] admits an anamorphic mode with the three sought-after security properties. The construction follows a similar strategy to GQ signatures in that authenticated encryption ciphertexts are embedded into the first prover message, which is fed to the random oracle before computing the prover's response. In this case, the random oracle input is a matrix-vector product $\boldsymbol{w} = \mathbf{A}\boldsymbol{y}$, where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is a flat matrix, and the vector $\boldsymbol{y}$ is sampled from a discrete Gaussian distribution with relatively small variance. Although $\boldsymbol{w}$ is not perfectly uniform, it can be computationally/statistically close to uniform in certain parameter

settings. In order to design an anamorphic mode, we need to embed lattice trapdoors [49] into $\mathbf{A}$, compute a pseudorandom $\boldsymbol{w}$ containing a nonce/authenticated encryption and then pre-image sample [29] $\boldsymbol{y}$ using a lattice trapdoor in order to obtain a properly distributed anamorphic signature.

We can prove dictator and recipient unforgeability under the SIS assumption in the random oracle model. Unfortunately, the scheme is not as efficient as previous anamorphic constructions in terms of bandwidth. In particular, the bandwidth rate declines with the inverse of the standard deviation of $\boldsymbol{y}$. It also requires somewhat larger parameters than ordinary Lyubashevsky signatures since the matrix $\mathbf{A}$ must be wide enough to accommodate a lattice trapdoor.

It remains an interesting open question to obtain more efficient anamorphic triplets for signature schemes based on standard post-quantum assumptions.

1.3 Related Work

Anamorphic cryptography was first investigated in the encryption setting (AE) by Persiano *et al.* [53]. Several works [43,62,8,18] extended their syntax and security definition to capture more advanced security properties. Catalano *et al.* [19] gave a black-box impossibility result on obtaining AE from PKE with large message space. It was further shown [20] that black-box constructions of AE from PKE are impossible when the sender/receiver do not keep a shared state.

Anamorphic signatures were subsequently suggested by Kutylowski [42] to maintain the possibility of private communication in an extreme scenario where a government simply bans the use of encryption. Their work laid the foundations of anamorphic signatures by introducing the stealthiness and private anamorphism definitions. They also introduced the randomness replacement technique and further showed that several Fiat-Shamir-based signatures (including Elgamal [25] and Schnorr [58] signatures) are amenable to constructing symmetric anamorphic signatures. They also gave a sufficient separability condition leading to privately anamorphic signatures, which is easily seen to be satisfied by Boneh-Boyen signatures [14]. They further showed that privately anamorphic signatures can be obtained from the Naor-Yung paradigm for signatures [50].

The definition of private anamorphism was argued [37] to be insufficient in some natural applications where anamorphic signers do not trust recipients (who can observe anamorphic signatures and not only normal ones) to not sign on their behalf. Jaeger and Stracovsky [37] strengthened private anamorphism into a notion of recipient unforgeability where the adversary is additionally granted access to an *anamorphic signing oracle*. They further showed that recipient unforgeability is achievable via the randomness replacement technique if the underlying signature is secure under chosen message and randomness attacks and proved that RSA-PSS signatures meets this requirement.

Jaeger and Stracovsky [37] additionally showed that the standard notion of unforgeability (under chosen message attacks) suffices in applications of the RRep transform if the symmetric encryption ciphertext satisfies a strong notion of simulatability in the random oracle or in the ideal cipher model. As a corollary, anamorphic Boneh-Boyen signatures can be made recipient unforgeable by

introducing a random oracle (we make a similar use of random oracles in anamorphic GQ signatures). However, simultaneously achieving dictator unforgeability is difficult as it requires to hide both the symmetric encryption nonce and its authentication tag inside the public scalar (which is typically a 256-bit exponent) contained in Boneh-Boyen signatures. Achieving dictator unforgeability in anamorphic RSA-PSS signatures raises a similar difficulty of embedding an authenticated encryption ciphertext in the short random salt.

Our anamorphic GQ signatures bears high-level resemblance (although it is different) with the swapping technique introduced by Micali and Reyzin [48] to obtain tightly secure signatures. While their swapped-GQ signatures can carry anamorphic messages in the same way as PSS, it is subject to the same length restriction as RSA-PSS. We give a comparison in Appendix D.3.

Banerjee *et al.* [7] recently considered asymmetric anamorphic signatures, where the double key consists of two distinct parts. The first part allows a signer to *send* anamorphic messages while the second part allows a receiver to *recover* anamorphic messages. As a result, the anamorphic channel is asymmetric. They proposed constructions based on BBS [13] and Waters signatures [63]. Their approach requires to split each covert ciphertext across several anamorphic signatures. In contrast, our schemes require a single anamorphic signature to send a covert ciphertext. The notions of recipient and dictator unforgeability were not considered in [7]. Instead, [7, Appendix B] discusses a notion of robustness requiring that normal signatures are unlikely to anamorphically decrypt to meaningful messages. This particular robustness notion may be seen as a weak form of dictator unforgeability.

2 Background

NOTATIONS. For integers a and b, we write $a|b$ when a divides b and $a|_1 b$ when a divides b exactly once (i.e., a divides b but a^2 does not). We denote by $\mathcal{P}_k$ the set of k-bit primes for $k \in \mathbb{N}$. If $b > a$, $[a, b]$ denotes $\{a, a+1, \ldots, b-1, b\}$.

For any integer $a \in [0, 2^\ell - 1]$, we denote by $\mathsf{bin}(a) = (a_0, \ldots, a_{\ell-1}) \in \{0,1\}^\ell$ the binary decomposition (ordered with the least significant bits on the left) of $a = \sum_{i=0}^{\ell-1} a_i \cdot 2^i$. For any ℓ-bit string $\mathbf{a} = (a_0, \ldots, a_{\ell-1}) \in \{0,1\}^\ell$, we denote by $\mathsf{bin}^{-1}(\boldsymbol{a}) = \sum_{i=0}^{\ell-1} a_i \cdot 2^i \in \mathbb{Z}$ the converse mapping.

When D is a distribution, $x \hookleftarrow D$ denotes the action of sampling x from the distribution D. The notation $x \sim D$ means that x was sampled according to the distribution D. For finite set S, $U(S)$ stands for the uniform distribution on S.

The security parameter is denoted by λ and the usual asymptotic notation Ω, O, Θ are used. A function $f : \mathbb{N} \to \mathbb{R}$ is negligible in λ (or $\mathsf{negl}(\lambda)$) if, for any constant $c > 0$, there is an integer N_c such that $|f(\lambda)| \leq \lambda^{-c}$ for all $\lambda \geq N_c$.

2.1 Hardness Assumptions

In this section, we recall the discrete logarithm, RSA and ϕ-hiding assumptions.

Definition 1. *Let a group $\mathbb{G}$ of prime order $q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$, with $g \in \mathbb{G}$. The* **Discrete Logarithm assumption** *is that, for any PPT algorithm $\mathcal{A}$, we have $\Pr[\ x \leftarrow \mathcal{A}(X) \mid x \hookleftarrow U(\mathbb{Z}_q),\ X = g^x\ ] \leq \mathsf{negl}(\lambda)$, where the probability is taken over all random coins.*

Our second construction can rely on either the RSA assumption or (with a tighter reduction) the ϕ-hiding assumption introduced by Cachin *et al.* [17].

Definition 2. *Let an RSA modulus $N = pq$, where $p, q \in \mathcal{P}_l$ are $l(\lambda)$-bit primes for some polynomial function $l : \mathbb{N} \to \mathbb{N}$. Let an integer $e \in \mathbb{N}$ such that $\gcd(e, \varphi(N)) = 1$. The* **RSA assumption** *states that, for any PPT algorithm $\mathcal{A}$, we have $\Pr[\ y = x^e \bmod N \mid y \hookleftarrow U(\mathbb{Z}_N^*),\ x \leftarrow \mathcal{A}(N, e, y)\ ] \leq \mathsf{negl}(\lambda)$.*

We call $\mathcal{RSA}_k$ the set of k-bit RSA moduli $N = pq$ such that $p, q \in \mathcal{P}_{k/2}$. For an RSA modulus $N = pq$, the order of the multiplicative group $\mathbb{Z}_N^*$ is $\varphi(N) = (p-1)(q-1)$. For a prime e, we denote by $\mathcal{RSA}_k[e \,|_1\, \varphi(N)]$ the set of k-bit RSA moduli such that $e \,|_1\, \varphi(N)$ and by $\mathcal{RSA}_k[e \not| \varphi(N)]$ the set of k-bit RSA moduli N such that $\gcd(e, \varphi(N)) = 1$. The ϕ-hiding assumption requires $|e| \ll |N|/4$ to avoid attacks based on [21].

Definition 3 ([17]). *Let integers $k, \tau \in \mathsf{poly}(\lambda)$ such that $\tau < k/4 - \lambda$. Let $\mathcal{E} \subset \mathcal{P}_\tau$ a set of τ-bit primes. The* **ϕ-hiding assumption** *for $\mathcal{E}$ states that, for any PPT algorithm $\mathcal{D}$, we have*

$$\begin{aligned}\mathbf{Adv}_{\mathcal{D}}^{\phi\text{-hide}}(\lambda) := \big|\, &\Pr\left[1 \leftarrow \mathcal{D}(N, e) \mid e \hookleftarrow \mathcal{E},\ N \leftarrow \mathcal{RSA}_k[e \,|_1\, \varphi(N)]\ \right] \\ &- \Pr\left[1 \leftarrow \mathcal{D}(N, e) \mid e \hookleftarrow \mathcal{E},\ N \leftarrow \mathcal{RSA}_k[e \not| \varphi(N)]\ \right] \big| \leq \mathsf{negl}(\lambda).\end{aligned}$$

2.2 Forking Lemma

In this section, we recall the Forking Lemma [55]. In the following, we will use the version that was proven by Bellare and Neven [10].

Lemma 1 (General Forking Lemma [10, Lemma 1]). *Let an integer $Q \geq 1$ and let a set $\mathcal{H}$ of size $h \geq 2$. Let $\mathcal{A}$ be a randomized algorithm that, on input $(x, h_1, \ldots, h_Q)$, outputs a pair comprised of an integer $J \in [0, Q]$ and a side output σ. Let IG a randomized input generator. The accepting probability acc of $\mathcal{A}$ is the probability that $J \geq 1$ in the experiment*

$$x \leftarrow \mathsf{IG};\ (h_1, \ldots, h_Q) \hookleftarrow \mathcal{H};\ (J, \sigma) \leftarrow \mathcal{A}(x, (h_1, \ldots, h_Q)).$$

The forking algorithm $F_{\mathcal{A}}(x)$ associated with $\mathcal{A}$ is described in Figure 1. If we define the probability $\mathrm{frk} := \Pr[b = 1 \ :\ x \leftarrow \mathsf{IG};\ (b, \sigma, \sigma') \leftarrow F_{\mathcal{A}}(x)]$, *then we have* $\mathrm{frk} \geq acc \cdot \left(\frac{acc}{Q} - \frac{1}{h}\right)$.

Algorithm $F_{\mathcal{A}}(x)$
Choose random coins ρ for $\mathcal{A}$
$h_1, \ldots, h_Q \hookleftarrow \mathcal{H}$
$(J, \sigma) \leftarrow \mathcal{A}(x, (h_1, \ldots, h_Q); \rho)$
If $J = 0$, then return $(0, \perp, \perp)$.
Choose $h'_J, \ldots, h'_Q \hookleftarrow \mathcal{H}$
$(J', \sigma') \leftarrow \mathcal{A}(x, (h_1, \ldots, h_{J-1}, h'_J, \ldots, h'_Q); \rho)$
If $(J = J')$ and $h'_J \neq h_J$, then return $(1, \sigma, \sigma')$
Else return $(0, \perp, \perp)$

Fig. 1. Forking algorithm $F_{\mathcal{A}}(x)$.

2.3 Definitions for Anamorphic Signatures

In Definition 4, we extend the syntax of [42,37] by allowing the aGen algorithm to additionally output a secret trapdoor tk that will only be used by the anamorphic signing algorithm. This trapdoor information will help aSig produce signatures that look like ordinary signatures (even when the key pair $(\mathsf{avk}, \mathsf{ask})$ is given). When this trapdoor tk is empty (as will be the case in our factoring-based constructions), we recover the syntax from [42,37].

Definition 4. *An anamorphic triplet* $\Sigma = (\mathsf{aGen}, \mathsf{aSig}, \mathsf{aDec})$ *for a signature scheme consists of PPT algorithms such that:*

- aGen *is a PPT algorithm that takes as input a security parameter* $\lambda \in \mathbb{N}$ *and outputs an anamorphic public key* avk*, an anamorphic secret key* ask*, a double key* dk*, and an anamorphic trapdoor key* tk *(which may be empty).*
- aSig *is a PPT algorithm that inputs* ask*,* tk *and* dk*, a real message* $\mu \in \mathcal{M}$*, and a covert message* $\hat{\mu} \in \widehat{\mathcal{M}}$*. It outputs an anamorphic signature* asig*.*
- aDec *is an anamorphic decryption algorithm that inputs* dk*,*[3] *a message* μ *and a candidate anamorphic signature* asig*. It outputs an anamorphic message* $\hat{\mu} \in \widehat{\mathcal{M}}$ *or a special symbol* $\perp \notin \widehat{\mathcal{M}}$*.*

In addition, these algorithms should satisfy the following correctness requirement: For any pair $(\mu, \hat{\mu}) \in \mathcal{M} \times \widehat{\mathcal{M}}$*, the probability that* $\mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig}) \neq \hat{\mu}$ *is negligible, where* $(\mathsf{avk}, \mathsf{ask}, \mathsf{dk}, \mathsf{tk}) \leftarrow \mathsf{aGen}(1^\lambda)$*,* $\mathsf{asig} \leftarrow \mathsf{aSig}(\mathsf{ask}, \mathsf{tk}, \mathsf{dk}, \mu, \hat{\mu})$ *and the probability is taken over the random coins of* aGen *and* aSig*.*

Stealthiness. The notion of stealthiness [42] requires that a dictator cannot distinguish the anamorphic mode of the scheme (with an anamorphic key generation and signing oracle) from its regular mode (with a normal key generation and signing oracle), even given the signer's key pair.

[3] We assume that dk contains avk for simplicity.

Definition 5. *A signature scheme* $\Pi = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver})$ *is* **anamorphic** *if it provides unforgeability under chosen-message attacks and* **stealthiness**, *meaning there exists an anamorphic triplet* Σ *such that, for any PPT dictator* $\mathcal{D}$, *there exists a negligible function* $\nu(\lambda)$ *such that*

$$\mathbf{Adv}^{\mathsf{Stealthiness}}_{\mathcal{D},\Pi,\Sigma}(\lambda) = |\Pr[\mathsf{RealG}_{\Pi}(\lambda,\mathcal{D}) = 1] - \Pr[\mathsf{AnamorphicG}_{\Sigma}(\lambda,\mathcal{D}) = 1]| \leq \nu(\lambda)$$

where the real and ideal experiments are defined as follows:

$\mathsf{RealG}_{\Pi}(\lambda,\mathcal{D})$:
1. $(\mathsf{vk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^{\lambda})$;
2. *Return* $\mathcal{D}^{\mathcal{O}_{\mathsf{Sig}}(\mathsf{sk},\cdot,\cdot)}(\mathsf{vk},\mathsf{sk})$

where $\mathcal{O}_{\mathsf{Sig}}(\mathsf{sk},\mu,\hat{\mu}) = \mathsf{Sig}(\mathsf{sk},\mu)$

$\mathsf{AnamorphicG}_{\Sigma}(\lambda,\mathcal{D})$:
1. $((\mathsf{avk},\mathsf{ask}),\mathsf{dk},\mathsf{tk}) \leftarrow \mathsf{aGen}(1^{\lambda})$;
2. *Return* $\mathcal{D}^{\mathcal{O}_{\mathsf{aSig}}(\mathsf{ask},\cdot,\cdot)}(\mathsf{avk},\mathsf{ask})$

where $\mathcal{O}_{\mathsf{aSig}}(\mathsf{ask},\mu,\hat{\mu}) = \mathsf{aSig}(\mathsf{ask},\mathsf{tk},\mathsf{dk},\mu,\hat{\mu})$

Dictator Unforgeability. Jaeger and Stracovsky [37] introduced a property called *dictator unforgeability*. This property ensures that, in the anamorphic mode, the dictator is unable to forge a non-trivial valid signature that contains a valid anamorphic message, even if the dictator knows the secret key ask and has access to an anamorphic signing oracle and an anamorphic decryption oracle. This notion strengthens robustness [8] (as transposed in [37, Figure 4] to the signature setting) by allowing anamorphic signing queries and revealing ask.

Definition 6. *An anamorphic signature scheme* $\Pi = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver})$ *equipped with an anamorphic triplet* $\Sigma = (\mathsf{aGen}, \mathsf{aSig}, \mathsf{aDec})$ *is* **dictator unforgeable under chosen anamorphic signature attack** *(or* DUF-CASA *for short) if, for every PPT adversary* $\mathcal{A}$, *there exists a negligible function* $\nu(\lambda)$ *such that*

$$\mathbf{Adv}^{\mathsf{DUF\text{-}CASA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) := \Pr[\mathcal{G}^{\mathsf{DUF\text{-}CASA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) = 1] \leq \nu(\lambda)$$

where the experiment $\mathcal{G}^{\mathsf{DUF\text{-}CASA}}_{\mathcal{A},\Pi,\Sigma}$ *is defined as follows.*

$\mathcal{G}^{\mathsf{DUF\text{-}CASA}}_{\mathcal{A},\Pi,\Sigma}(\lambda)$:
1. $S \leftarrow \emptyset$
2. $((\mathsf{avk},\mathsf{ask}),\mathsf{dk},\mathsf{tk}) \leftarrow \mathsf{aGen}(1^{\lambda})$
3. $(\mu^{\star},\mathsf{asig}^{\star}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{aSig}},\mathcal{O}_{\mathsf{aDec}}}(\mathsf{avk},\mathsf{ask})$
4. $\hat{\mu}^{\star} \leftarrow \mathsf{aDec}(\mathsf{dk},\mu^{\star},\mathsf{asig}^{\star})$
5. *Set* $b_{\mathsf{valid}} = 1$ *if* $\hat{\mu}^{\star} \neq \perp$ *and* $b_{\mathsf{valid}} = 0$ *otherwise*
6. *Set* $b_{\mathsf{new}} = 1$ *if* $(\mu^{\star},\hat{\mu}^{\star},\mathsf{asig}^{\star}) \notin S$ *and* $b_{\mathsf{new}} = 0$ *otherwise*
7. *Return* $b_{\mathsf{valid}} \wedge b_{\mathsf{new}}$

where $\mathcal{A}$*'s oracles are defined as follows:*

$\mathcal{O}_{\mathsf{aSig}}(\mu,\hat{\mu})$:
1. $\mathsf{asig} \leftarrow \mathsf{aSig}(\mathsf{ask},\mathsf{tk},\mathsf{dk},\mu,\hat{\mu})$
2. $S \leftarrow S \cup \{(\mu,\hat{\mu},\mathsf{asig})\}$
3. *Return* asig

$\mathcal{O}_{\mathsf{aDec}}(\mu,\mathsf{asig})$:
1. $\hat{\mu} \leftarrow \mathsf{aDec}(\mathsf{dk},\mu,\mathsf{asig})$
2. *Return* $\hat{\mu}$

Definition 6 captures a notion of *strong* unforgeability in that the adversary succeeds if it outputs a new pair $(\mu^{\star},\mathsf{asig}^{\star})$ such that $(\mu^{\star},\hat{\mu}^{\star},\mathsf{asig}^{\star}) \notin S$ with $\hat{\mu}^{\star} = \mathsf{aDec}(\mathsf{dk},\mu^{\star},\mathsf{asig}^{\star}) \neq \perp$ at step 6. A relaxed notion of *weak* unforgeability

(wDUF-CASA) only requires that $\mathcal{A}$ be unable to come up with a signature on a new message. In this case, step 6 becomes "Set $b_{\mathsf{new}} = 1$ if no tuple $(\mu^\star, \hat{\mu}^\star, \cdot)$ such that $\hat{\mu}^\star = \mathsf{aDec}(\mathsf{dk}, \mu^\star, \mathsf{asig}^\star) \neq \perp$ exists in S, and $b_{\mathsf{new}} = 0$ otherwise."

Importantly, the notion of weak unforgeability allows the adversary to modify a signature obtained from $\mathcal{O}_{\mathsf{aSig}}$, but it prevents it from tampering with embedded anamorphic messages $\hat{\mu}$.

Recipient Unforgeability. Jaeger and Stracovsky [37] also formalized a notion of recipient unforgeability. Informally, it requires that a recipient knowing the double key should remain unable to forge signatures on behalf of the signer even given access to a regular signing oracle and an anamorphic signing oracle. Their definition strengthens the notion of *private anamorphism* of Kutylowski *et al.* [42] in that, in the former, the adversary is only given access to a regular signing oracle (and no anamorphic signing oracle).

Definition 7. *An anamorphic signature scheme* $\Pi = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Ver})$ *equipped with an anamorphic triplet* $\Sigma = (\mathsf{aGen}, \mathsf{aSig}, \mathsf{aDec})$ *is* **recipient unforgeable** *(*RUF-CAMA*) if, for every PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\nu(\lambda)$ *such that* $\mathbf{Adv}^{\mathsf{RUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) := \Pr[\mathcal{G}^{\mathsf{RUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) = 1] \leq \nu(\lambda)$*, where the experiment* $\mathcal{G}^{\mathsf{RUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}$ *is defined hereunder.*

$\mathcal{G}^{\mathsf{RUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}(\lambda)$:
1. $S \leftarrow \emptyset$
2. $((\mathsf{avk}, \mathsf{ask}), \mathsf{dk}, \mathsf{tk}) \leftarrow \mathsf{aGen}(1^\lambda)$
3. $(\mu^\star, \mathsf{sig}^\star) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Sig}}, \mathcal{O}_{\mathsf{aSig}}}(\mathsf{avk}, \mathsf{dk})$
4. $b_{\mathsf{valid}} = \mathsf{Ver}(\mathsf{avk}, \mu^\star, \mathsf{sig}^\star)$
5. *Set* $b_{\mathsf{new}} = 1$ *if* $(\mu^\star, \mathsf{sig}^\star) \notin S$ *and* $b_{\mathsf{new}} = 0$ *otherwise*
6. *Return* $b_{\mathsf{valid}} \wedge b_{\mathsf{new}}$

where $\mathcal{A}$*'s oracles are defined as follows:*

$\mathcal{O}_{\mathsf{Sig}}(\mu)$:
1. $\mathsf{sig} \leftarrow \mathsf{Sig}(\mathsf{ask}, \mu)$
2. $S \leftarrow S \cup \{(\mu, \mathsf{sig})\}$
3. *Return* sig

$\mathcal{O}_{\mathsf{aSig}}(\mu, \hat{\mu})$:
1. $\mathsf{asig} \leftarrow \mathsf{aSig}(\mathsf{ask}, \mathsf{tk}, \mathsf{dk}, \mu, \hat{\mu})$
2. $S \leftarrow S \cup \{(\mu, \mathsf{asig})\}$
3. *Return* asig

Definition 7 also captures strong unforgeability as the adversary succeeds if it outputs a new pair $(\mu^\star, \mathsf{asig}^\star) \notin S$ at step 5. We can relax it into a notion of *weak* unforgeability (wRUF-CAMA), which only prevents $\mathcal{A}$ from forging signatures on new messages. Then, step 5 is changed into "Set $b_{\mathsf{new}} = 1$ if no entry of the form $(\mu^\star, \cdot)$ exists in S and $b_{\mathsf{new}} = 0$ otherwise".

3 Efficient Constructions of Dictator and Recipient Unforgeable Anamorphic Signatures

We show that several Fiat-Shamir-based signatures admit anamorphic triplets that simultaneously provide stealthiness, DUF-CASA and RUF-CAMA security.

3.1 A Construction from Okamoto-Schnorr Signatures

We describe an anamorphic mode for Okamoto's variant [51] of Schnorr signatures [58] and its generalization proposed by Alwen *et al.* [6, Section 4.2]. The resulting scheme is both dictator and recipient unforgeable (albeit without strong unforgeability in the latter case) and allows instantiating the symmetric anamorphic ciphertexts without introducing additional random oracles.

This section describes the variant considered in [6,27] which uses $n+1$ generators. These signatures were devised to resist leakage on the secret key: increasing the number of generators makes it possible to maintain security when a larger fraction of the secret key is leaked, as shown in [6, Theorem 4.1]. The anamorphic variant of Okamoto's original scheme [51] (which is a special case of the construction below for $n = 1$) is described in Appendix C.1.

Increasing n allows for a better anamorphic bandwidth. In particular, the bandwidth rate approaches 1 as the number of generators grows.

KGen(1^λ)**:** Given a security parameter 1^λ,

1. Choose a cyclic group $\mathbb{G}$ of prime order q such that $2^{\ell_q} < q < 2^{\ell_q} + 2^\lambda$ with $\ell_q > 2\lambda$.
2. Choose generators $g_1 \triangleq g \hookleftarrow U(\mathbb{G})$ and $g_2, \ldots, g_{n+1} \hookleftarrow U(\mathbb{G})$.
3. Choose $x_1, \ldots, x_{n+1} \hookleftarrow U(\mathbb{Z}_q)$ and compute $X = \prod_{i=1}^{n+1} g_i^{x_i}$.
4. Choose a hash function $H : \{0,1\}^* \to \mathbb{Z}_q$.

Output $\mathsf{vk} = \big(\mathbb{G}, g_1, \ldots, g_{n+1}, X, H\big)$, $\mathsf{sk} = (x_1, \ldots, x_{n+1})$.

aGen(1^λ)**:** Given a security parameter 1^λ,

1. Choose a cyclic group $\mathbb{G}$ of prime order q such that $2^{\ell_q} < q < 2^{\ell_q} + 2^\lambda$ with $\ell_q > 2\lambda$.
2. Choose $g_1 \triangleq g \hookleftarrow U(\mathbb{G})$ and set $g_2 = g^{\omega_2}$, $g_3 = g^{\omega_3}$, $\ldots$, $g_{n+1} = g^{\omega_{n+1}}$, where $\omega_2, \ldots, \omega_{n+1} \hookleftarrow U(\mathbb{Z}_q)$.
3. Choose $x_1, \ldots, x_{n+1} \hookleftarrow U(\mathbb{Z}_q)$ and compute $X = \prod_{i=1}^{n+1} g_i^{x_i}$.
4. Choose a hash function $H : \{0,1\}^* \to \mathbb{Z}_q$.
5. Define the anamorphic message space as $\hat{\mathcal{M}} = \mathbb{Z}_q^{n-1} \times \{0,1\}^t$, where $t = \ell_q - \lambda$.
6. Choose keys $k_m \hookleftarrow U(\mathcal{K}_m)$, $\bar{k}_m \hookleftarrow U(\overline{\mathcal{K}}_m)$, $K \hookleftarrow U(\mathcal{K})$ for pseudorandom functions

$$\begin{aligned}
&\mathsf{PRF} : \mathcal{K}_m \times \big(\mathbb{G} \times [2, n]\big) \to \mathbb{Z}_q,\\
&\overline{\mathsf{PRF}} : \overline{\mathcal{K}}_m \times \mathbb{G} \to \{0,1\}^t,\\
&\mathsf{PRF}' : \mathcal{K} \times \big(\{0,1\}^t \times \mathcal{M} \times \mathbb{G} \times \mathbb{Z}_q^{n-1}\big) \to \{0,1\}^\lambda,
\end{aligned}$$

where $\mathcal{M}$ is the space of regular messages and $t = \ell_q - \lambda$.

Output $\mathsf{avk} = \big(\mathbb{G}, g_1, \ldots, g_{n+1}, X, H\big)$, $\mathsf{ask} = (x_1, \ldots, x_{n+1})$, together with $\mathsf{dk} = (\mathsf{avk}, k_m, \bar{k}_m, K)$ and $\mathsf{tk} = (\omega_2, \ldots, \omega_{n+1})$.

Sig(sk, μ)**:** To sign $\mu \in \mathcal{M}$ using $\mathsf{sk} = (x_1, \ldots, x_{n+1})$,

1. Choose $r_1, \ldots, r_{n+1} \hookleftarrow U(\mathbb{Z}_q)$ and compute $R = \prod_{i=1}^{n+1} g_i^{r_i}$.
2. Compute $c = H(R, \mu) \in \mathbb{Z}_q$.
3. For all $i \in [1, n+1]$, compute $s_i = r_i + c \cdot x_i \bmod q$.

Output $\mathsf{sig} = (c, s_1, \ldots, s_{n+1}) \in \mathbb{Z}_q \times \mathbb{Z}_q^{n+1}$.

aSig$(\mathsf{ask}, \mathsf{tk}, \mathsf{dk}, \mu, \hat{\mu})$**:** To sign $\mu \in \mathcal{M}$ and encrypt $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_{n-1}, \hat{\mu}_n) \in \mathbb{Z}_q^{n-1} \times \{0,1\}^t$ using $\mathsf{ask} = (x_1, \ldots, x_{n+1})$, $\mathsf{dk} = (\mathsf{avk}, k_m, \bar{k}_m, K)$, and $\mathsf{tk} = (\omega_2, \ldots, \omega_{n+1})$, do the following.

1. Choose $r \hookleftarrow U(\mathbb{Z}_q)$ and compute $R = g^r$.
2. For all $i \in [2, n]$, compute $s_i = \hat{\mu}_{i-1} + \mathsf{PRF}_{k_m}(R, i) \bmod q$.
3. Let $a = \hat{\mu}_n \oplus \overline{\mathsf{PRF}}_{\bar{k}_m}(R)$ and compute

$$s_{n+1} = \mathsf{bin}^{-1}(a) + 2^t \cdot \mathsf{bin}^{-1}\big(\mathsf{PRF}'_K(a, \mu, R, s_2, \ldots, s_n)\big) \bmod q.$$

4. Compute $c = H(R, \mu) \in \mathbb{Z}_q$.
5. Compute $s_1 = r + c \cdot (x_1 + \sum_{i=2}^{n+1} \omega_i x_i) - \sum_{i=2}^{n+1} \omega_i \cdot s_i \bmod q$.

Output $\mathsf{sig} = (c, s_1, \ldots, s_{n+1}) \in \mathbb{Z}_q \times \mathbb{Z}_q^{n+1}$.

Ver$\big(\mathsf{vk}, \mu, \mathsf{sig}\big)$**:** Parse sig as $\mathsf{sig} = (c, s_1, \ldots, s_{n+1}) \in \mathbb{Z}_q \times \mathbb{Z}_q^{n+1}$ and return 1 if $c = H\big(X^{-c} \cdot \prod_{i=1}^{n+1} g_i^{s_i}, \mu\big)$. Otherwise, return 0.

aDec$\big(\mathsf{dk}, \mu, \mathsf{asig}\big)$**:** Given message μ, signature $\mathsf{sig} = (c, s_1, \ldots, s_{n+1}) \in \mathbb{Z}_q \times \mathbb{Z}_q^{n+1}$ and double key $\mathsf{dk} = (\mathsf{avk}, k_m, \bar{k}_m, K)$, do the following.

1. Return $\perp$ if $\mathsf{Ver}(\mathsf{avk}, \mu, \mathsf{asig}) = 0$. Let $R = X^{-c} \cdot \prod_{i=1}^{n+1} g_i^{s_i}$.
2. Let $a_0 = s_{n+1} \bmod 2^t$ and $a_1 = \lfloor s_{n+1}/2^t \rfloor \in \mathbb{Z}_{\lfloor q/2^t \rfloor}$. Parse $a_0 \in \mathbb{Z}_{2^t}$ as $a = \mathsf{bin}(a_0) \in \{0,1\}^t$ and return $\perp$ if $\mathsf{bin}(a_1) \neq \mathsf{PRF}'_K(a, \mu, R, s_2, \ldots s_n)$.
3. For all $i \in [2, n]$, compute $\hat{\mu}_{i-1} = s_i - \mathsf{PRF}_{k_m}(R, i) \bmod q$.
4. Compute $\hat{\mu}_n = a \oplus \overline{\mathsf{PRF}}_{\bar{k}_m}(R)$.
5. Return $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_{n-1}, \hat{\mu}_n)$.

Efficiency. The scheme can use standard parameter for Schnorr or Okamoto-Schnorr signatures. If we set $n = 1$ as in Appendix C.1, the anamorphic bandwidth rate is 1/6. For $n > 1$, since q is chosen to be around $2^{2\lambda}$, the bandwidth rate is approximately $\frac{(n-1)\log q + (1/2)\log q}{(n+2)\log q} = 1 - \frac{2.5}{n+2}$.

Just like in Schnorr signatures, it would be sufficient to use a random oracle H with output space $\{0,1\}^\lambda$ (rather than $\mathbb{Z}_q$ for $q > 2^{2\lambda}$) as the security of the scheme does not rely on the full collision-resistance of H. For $n = 1$, the bandwidth rate is then 1/5. For $n > 1$, it becomes $1 - \frac{2}{n+1.5}$.

In the anamorphic and normal signing algorithms, all exponentiations are pre-computable in an offline phase (before the message is known). The online phase only requires cheap field operations over $\mathbb{Z}_q$ and secret-key operations.

In contrast with GQ and 2^t-root signatures, the bandwidth rate can approach 1 without having to choose a larger security parameter λ. We just need to increase the signature length and compute a multi-exponentiation of larger dimension in the verification algorithm.

DICTATOR UNFORGEABILITY. We now show that the scheme provides (strong) dictator unforgeability. The proof of Theorem 1 does not use random oracles and only relies on the pseudorandomness of PRF'.

Theorem 1. *The scheme is dictator unforgeable in the* DUF-CASA *sense if* PRF' *is a pseudorandom function family. For any PPT adversary* $\mathcal{A}$ *making at most* Q_D *anamorphic decryption queries and* Q_S *signing queries, there exists a PRF distinguisher* $\mathcal{B}$ *such that* $\mathbf{Adv}^{\mathsf{DUF\text{-}CASA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) \leq \mathbf{Adv}^{\mathcal{B},\mathsf{PRF}'}_{Q_S+Q_D+1}(\lambda) + \frac{Q_D+1}{2^\lambda}$.

Proof. The proof considers a sequence of games. In each game, W_i denotes the event that the output of Game_i is 1.

Game_0**:** This is the real DUF-CASA experiment, where the adversary $\mathcal{A}$ initially receives $(\mathsf{avk}, \mathsf{ask})$ and queries oracles $\mathcal{O}_{\mathsf{aSig}}, \mathcal{O}_{\mathsf{aDec}}$. For convenience, we consider $\mathcal{A}$'s output $(\hat{\mu}^\star, \mathsf{asig}^\star)$ as an anamorphic decryption query and declare $\mathcal{A}$ successful at the first $\mathcal{O}_{\mathsf{aDec}}$-query for which aDec outputs $\hat{\mu} \neq \perp$ and asig is non-trivial. Namely, we call W_0 the event that $\mathcal{A}$ queries $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ on (μ, asig) such that $\hat{\mu} = \mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig}) \neq \perp$ and S does not contain a tuple of the form $(\mu, \hat{\mu}, \mathsf{asig})$. By definition, $\mathcal{A}$'s advantage is $\Pr[W_0]$.

Game_1**:** We modify the $\mathcal{O}_{\mathsf{aDec}}$ oracle. At the first query $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ such that $\hat{\mu} = \mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig}) \neq \perp$ and no tuple $(\mu, \hat{\mu}, \mathsf{asig})$ exists in S, the challenger halts and outputs 0 as the result of the experiment. If we call F_1 this event, it would contradict the pseudorandomness of PRF'. Lemma 2 gives a reduction $\mathcal{B}$ such that $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1] \leq \mathbf{Adv}^{\mathcal{B},\mathsf{PRF}'}_{Q_S+Q_D+1}(\lambda) + \frac{Q_D+1}{2^\lambda}$.

In Game_1, we observe that $\Pr[W_1] = 0$ since the output of the experiment is always 0, regardless of $\mathcal{A}$'s queries. This completes the proof. □

Lemma 2. *We have* $\Pr[F_1] \leq \mathbf{Adv}^{\mathcal{B},\mathsf{PRF}'}_{Q_S+Q_D+1}(\lambda) + \frac{Q_D+1}{2^\lambda}$.

Proof. For each query $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ where $\mathsf{asig} = (c, s_1, \ldots, s_{n+1}) \in \mathbb{Z}_q \times \mathbb{Z}_q^{n+1}$, we can decompose $s_{n+1} \in \mathbb{Z}_q$ as

$$a_0 = s_{n+1} \bmod 2^t, \qquad a_1 = \lfloor s_{n+1}/2^t \rfloor \in \mathbb{Z}_{\lfloor q/2^t \rfloor} \tag{2}$$

and then parse $a_0 \in \mathbb{Z}_{2^t}$ as $a = \mathsf{bin}(a_0) \in \{0,1\}^t$. If event F_1 occurs, we must have $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_{n-1}, \hat{\mu}_n) = \mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig}) \neq \perp$, which implies $\mathsf{bin}(a_1) = \mathsf{PRF}'_K(a, \mu, R, s_2 \ldots, s_n)$ where $R = X^{-c} \cdot \prod_{i=1}^{n+1} g_i^{s_i}$ and $a = \mathsf{bin}(a_0) \in \{0,1\}^t$. Moreover, F_1 implies $(\mu, \hat{\mu}, \mathsf{asig}) \notin S$, meaning that either: (i) No query $\mathcal{O}_{\mathsf{aSig}}(\mu, \hat{\mu})$ was made; or (ii) A signing query $\mathcal{O}_{\mathsf{aSig}}(\mu, \hat{\mu})$ was made but the response tuple $(c', s'_1, \ldots, s'_{n+1})$ was not $\mathsf{asig} = (c, s_1, \ldots, s_{n+1})$. We claim that, in both cases, the PRF evaluation $\mathsf{PRF}'_K(a, \mu, R, s_2, \ldots, s_{n+1})$ was never obtained by $\mathcal{A}$.

In case (i), $\mathcal{A}$ never obtained $\mathsf{PRF}'_K(a, \mu, R, s_2, \ldots, s_{n+1})$ since $(\hat{\mu}_1, \ldots, \hat{\mu}_n)$ is uniquely determined by $(a, R, s_2, \ldots, s_{n+1})$ via the equations $\hat{\mu}_n = a \oplus \overline{\mathsf{PRF}}_{\bar{k}_m}(R)$ and $\hat{\mu}_i = s_{i+1} - \mathsf{PRF}_{k_m}(R, i+1)$ for $i \in [1, n-1]$. In case (ii), we write $R' = X^{-c'} \cdot \prod_{i=1}^{n+1} g_i^{s'_i}$, where $\mathcal{O}_{\mathsf{aSig}}(\mu, \hat{\mu})$ returned $(c', s'_1, \ldots, s'_{n+1})$. If $R' = R$, we have $c = H(R, \mu) = c'$ and thus $(s'_1, \ldots, s'_{n+1}) \neq (s_1, \ldots, s_{n+1})$ as the signatures would be identical otherwise. We then consider four disjoint sub-cases of (ii):

- $R \neq R'$: It immediately follows that the value $\mathsf{PRF}'_K(a, \mu, R, s_2, \ldots, s_{n+1})$ is not used during the $\mathcal{O}_{\mathsf{aSig}}$ query on $(\mu, \hat{\mu})$.
- $R = R'$ and there is an index $j \in [2, n]$ such that $s_j \neq s'_j$: This case is impossible since $s_j = \hat{\mu}_j + \mathsf{PRF}_{k_m}(R, j)$ for each $j \in [2, n]$. So, it would contradict the hypothesis that $\hat{\mu} = \hat{\mu}'$ and $R = R'$.
- $R = R'$, $(s_2, \ldots, s_n) = (s'_2, \ldots, s'_n)$ and $s_{n+1} \neq s'_{n+1}$: In this case, we have $a' = a \iff s'_{n+1} = s_{n+1}$ because $(R, s_2, \ldots, s_n) = (R', s'_2, \ldots, s'_{n+1})$ and, if $a = a'$, the validity of signatures implies

  $$\mathsf{bin}(a_1) = \mathsf{PRF}'_K(a, \mu, R, s_2, \ldots, s_n) = \mathsf{PRF}'_K(a', \mu, R', s'_2, \ldots, s'_n) = \mathsf{bin}(a'_1).$$

  Then, we must have $a' \neq a$ and the PRF value $\mathsf{PRF}'_K(a, \mu, R, s_2, \ldots, s_{n+1})$ is not used during the $\mathcal{O}_{\mathsf{aSig}}(\mu, \hat{\mu})$ query.
- $R = R'$, $(s_2, \ldots, s_{n+1}) = (s_2, \ldots, s_{n+1})$ and $s_1 \neq s'_1$: This case never occurs since, in valid signatures, s_1 and s'_1 are uniquely determined by $(R, c, s_2, \ldots, s_{n+1})$ and $(R', c', s'_2, \ldots, s'_{n+1})$, respectively. Hence, the equality $(R, c, s_2, \ldots, s_{n+1}) = (R', c', s'_2, \ldots, s'_{n+1})$ implies $s_1 = s'_1$.

The above ensures that $\mathcal{A}$ did not obtain $\mathsf{PRF}'_K(a, \mu, R, s_2, \ldots, s_n)$ during any $\mathcal{O}_{\mathsf{aSig}}$ queries in cases (i) and (ii).

Assuming that $\Pr[F_1]$ is non-negligible, we can build a PRF distinguisher $\mathcal{B}$ against PRF'. Our distinguisher $\mathcal{B}$ runs steps 1-5 of the aGen algorithm so as to generate avk and ask. It also chooses $k_m \hookleftarrow U(\mathcal{K}_m)$ and $\bar{k}_m \hookleftarrow U(\overline{\mathcal{K}}_m)$ at step 6. Instead of choosing K itself, it simulates PRF' by interacting with its pseudorandomness challenger.

In detail, $\mathcal{B}$ runs $\mathcal{A}$ on input of $(\mathsf{avk}, \mathsf{ask})$. At each query $\mathcal{O}_{\mathsf{aSig}}(\mu, \hat{\mu})$ made by $\mathcal{A}$, $\mathcal{B}$ samples $r \hookleftarrow U(\mathbb{Z}_q)$ and computes $R = g^r$. Then, it encrypts $\hat{\mu}_2, \ldots, \hat{\mu}_n$ as in the real aSig algorithm. It next encrypts $\hat{\mu}_{n+1}$ as $a = \hat{\mu}_{n+1} \oplus \overline{\mathsf{PRF}}_{\bar{k}_m}(R)$ and queries its PRF challenger on the input $(a, \mu, R, s_2, \ldots, s_n)$. The challenger returns h, which is either $\mathsf{PRF}'_K(a, \mu, R, s_2, \ldots, s_n)$ or a random element $h \sim U(\{0,1\}^\lambda)$. Using $h \in \{0,1\}^\lambda$, $\mathcal{B}$ defines

$$s_{n+1} = \mathsf{bin}^{-1}\big(a\big) + 2^t \cdot \mathsf{bin}^{-1}(h) \mod q$$

before generating (c, s_1) as in steps 4-5 of the aSig algorithm.

At each query $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ where $\mathsf{asig} = (c, s_1, \ldots, s_{n+1}) \in \mathbb{Z}_q \times \mathbb{Z}_q^{n+1}$, $\mathcal{B}$ computes a_0 and a_1 as in (2). It parses $a_0 \in \mathbb{Z}_{2^t}$ as $a = \mathsf{bin}(a_0) \in \{0,1\}^t$, computes $R = X^{-c} \cdot \prod_{i=1}^{n+1} g_i^{s_i}$ and queries its PRF challenger on the input $(a, \mu, R, s_2, \ldots, s_n)$. The challenger replies with $h_a \in \{0,1\}^\lambda$, which is either $h_a = \mathsf{PRF}'_K(a, \mu, R, s_2, \ldots, s_n)$ or a random $h_a \sim U(\{0,1\}^\lambda)$. Then, $\mathcal{B}$ checks if $\mathsf{bin}(a_1) = h_a$. If so, it halts and outputs 1. Otherwise, it keeps interacting with $\mathcal{A}$. If $\mathcal{A}$ terminates without having made a query $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ that caused $\mathcal{B}$ to halt and return 1, then $\mathcal{B}$ outputs 0.

We now claim that $\mathcal{B}$'s advantage as a PRF distinguisher against PRF' is

$$\mathbf{Adv}^{\mathcal{B},\mathsf{PRF}'}_{Q_S+Q_D+1}(\lambda) \geq \Pr[F_1] - (Q_D + 1) \cdot 2^{-\lambda}. \tag{3}$$

If $\mathcal{B}$'s challenger is always returning real PRF values at each query, $\mathcal{A}$'s view is exactly as in Game_1 and $\mathcal{B}$ outputs 1 with probability exactly $\Pr[F_1]$. In contrast,

if $\mathcal{B}$'s challenger uses a truly random function with output space $\{0,1\}^\lambda$, the probability to have $\mathsf{bin}(a_1) = h_a$ is only $1/2^\lambda$ at each anamorphic decryption query. This follows from the fact that an evaluation of PRF' causing F_1 has never been observed by $\mathcal{A}$, as argued above. Over all queries to $\mathcal{O}_{\mathsf{aDec}}$, the probability to have $\mathsf{bin}(a_1) = h_a$ is at most $(Q_D + 1)/2^\lambda$, which leads to (3). □

STEALTHINESS. The scheme also provides stealthiness in the standard model (without assuming random oracles) and while only assuming pseudorandom functions. The proof of Theorem 2 relies on Lemma 3 which was proven in [5] (and previously used in, e.g., [16]).

Lemma 3 ([5]). *If P and Q are two probability distributions such that $P(i) \geq (1-\varepsilon)\cdot Q(i)$ for all i, then the statistical distance between P and Q is at most ε.*

Theorem 2. *The scheme provides stealthiness if PRF, $\overline{\mathsf{PRF}}$ and PRF' are all pseudorandom function families. For any PPT adversary $\mathcal{A}$ making at most Q_S signing queries, there exist distinguishers $\mathcal{B}_0$, $\mathcal{B}_1$ and $\mathcal{B}_2$ against the pseudorandomness of PRF', PRF and $\overline{\mathsf{PRF}}$, respectively, for which we have*

$$\mathbf{Adv}^{\mathsf{Stealthiness}}_{\mathcal{D},\Pi,\Sigma}(\lambda) \leq \mathbf{Adv}^{\mathcal{B}_0,\mathsf{PRF}'}_{Q_S}(\lambda) + \mathbf{Adv}^{\mathcal{B}_1,\mathsf{PRF}}_{Q_S\cdot(n-1)}(\lambda) + \mathbf{Adv}^{\mathcal{B}_2,\overline{\mathsf{PRF}}}_{Q_S}(\lambda) + \tfrac{2Q_S^2+Q_S}{2^\lambda}.$$

Proof. The proof uses a sequence of games where the challenger is running the anamorphic experiment $\mathsf{AnamorphicG}_\Sigma(\lambda,\mathcal{D})$ with the adversary in the first game. In the final game, it interacts with the adversary in the same way as in the real experiment $\mathsf{RealG}_\Pi(\lambda,\mathcal{D})$.

For each index i, we let W_i denote the event that the adversary $\mathcal{A}$ outputs 1 in Game_i. We also denote by $\mathcal{O}^i_{\mathsf{aSig}}$ the anamorphic signing oracle of Game_i.

Game_0**:** This is the $\mathsf{AnamorphicG}_\Sigma(\lambda,\mathcal{D})$ experiment. The challenger generates $((\mathsf{avk},\mathsf{ask}),\mathsf{dk},\mathsf{tk}) \leftarrow \mathsf{aGen}(1^\lambda)$ and gives $(\mathsf{avk},\mathsf{ask} = (x_1,\ldots,x_{n+1}))$ to $\mathcal{A}$ but keeps $\mathsf{tk} = (\omega_2,\ldots,\omega_{n+1})$ and $\mathsf{dk} = (\mathsf{avk}, k_m, \bar{k}_m, K)$ to itself. At each query $\mathcal{O}^0_{\mathsf{aSig}}(\mathsf{ask},\mu,\hat{\mu})$, the challenger computes $\sigma \leftarrow \mathsf{aSig}(\mathsf{ask},\mathsf{dk},\mu,\hat{\mu})$ and returns the result to $\mathcal{A}$. By definition, W_0 is the event that $\mathcal{A}$ outputs 1.

Game_1**:** This game is identical to Game_0 with the difference that the challenger halts and outputs 0 in the event F_1 that the anamorphic signing oracle $\mathcal{O}^2_{\mathsf{aSig}}$ chooses $r \hookleftarrow U(\mathbb{Z}_q)$ that was used in a previous signing query. We have $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1] \leq Q_S^2/q \leq Q_S^2/2^{2\lambda}$ by a birthday bound.

Game_2**:** We modify the $\mathcal{O}^1_{\mathsf{aSig}}(\mu,\hat{\mu})$ oracle of Game_1 in the following way.

$\mathcal{O}^2_{\mathsf{aSig}}(\mathsf{ask},\mu,\hat{\mu})$: To sign $\mu \in \mathcal{M}$ and encrypt $\hat{\mu} = (\hat{\mu}_1,\ldots,\hat{\mu}_{n-1},\hat{\mu}_n)$ using $\mathsf{ask} = (x_1,\ldots,x_{n+1})$, $\mathsf{dk} = (\mathsf{avk}, k_m, \bar{k}_m, K)$ and $\mathsf{tk} = (\omega_2,\ldots,\omega_{n+1})$,

1. Choose $r \hookleftarrow U(\mathbb{Z}_q)$ and compute $R = g^r$. If r was previously used in a signing query, abort the game and output 0.
2. For all $i \in [2,n]$, compute $s_i = \hat{\mu}_{i-1} + \mathsf{PRF}_{k_m}(R,i) \bmod q$.
3. Let $a = \hat{\mu}_n \oplus \overline{\mathsf{PRF}}_{\bar{k}_m}(R)$, sample $h_R \hookleftarrow U(\{0,1\}^\lambda)$ and compute

$$s_{n+1} = \mathsf{bin}^{-1}(a) + 2^t \cdot \mathsf{bin}^{-1}(h_R) \bmod q.$$

4. Compute $c = H(R, \mu) \in \mathbb{Z}_q$.
5. Compute $s_1 = r + c \cdot (x_1 + \sum_{i=2}^{n+1} \omega_i x_i) - \sum_{i=2}^{n+1} \omega_i \cdot s_i \bmod q$.

Output $\mathsf{sig} = (c, s_1, \ldots, s_{n+1}) \in \mathbb{Z}_q \times \mathbb{Z}_q^{n+1}$.

Unless the failure event F_1 occurs (in which case both games produce identically distributed outputs), $\mathcal{O}^1_{\mathsf{aSig}}$ computes $\mathsf{PRF}'_K(a, \mu, R, \ldots)$ for distinct values of $R = g^r$. We can thus immediately rely on the pseudorandomness of PRF' and replace each $\mathsf{PRF}'_K(a, \mu)$ by a random string $h_r \sim U(\{0,1\}^\lambda)$ at step 2, which turns $\mathcal{O}^1_{\mathsf{aSig}}$ into $\mathcal{O}^2_{\mathsf{aSig}}$. A straightforward reduction builds a PRF distinguisher $\mathcal{B}_0$ for which $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}^{\mathcal{B}_0, \mathsf{PRF}'}_{Q_S}(\lambda)$.

Game_3**:** We replace $\mathcal{O}^2_{\mathsf{aSig}}(\mu, \hat{\mu})$ by the following oracle.

$\mathcal{O}^3_{\mathsf{aSig}}(\mathsf{ask}, \mu, \hat{\mu})$: To sign $\mu \in \mathcal{M}$ and encrypt $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_{n-1}, \hat{\mu}_n)$ using $\mathsf{ask} = (x_1, \ldots, x_{n+1})$, $\mathsf{dk} = (\mathsf{avk}, k_m, \bar{k}_m, K)$, and $\mathsf{tk} = (\omega_2, \ldots, \omega_{n+1})$,

1. Choose $r \hookleftarrow U(\mathbb{Z}_q)$ and compute $R = g^r$. If r was previously used in a signing query, abort the game and output 0.
2. For each $i \in [2, n]$, compute $s_i = \hat{\mu}_{i-1} + h_{R,i} \bmod q$ for a random $h_{R,i} \hookleftarrow U(\mathbb{Z}_q)$.
3. Let $a = \hat{\mu}_n \oplus \overline{\mathsf{PRF}}_{\bar{k}_m}(R)$, sample $h_R \hookleftarrow U(\{0,1\}^\lambda)$ and compute
$$s_{n+1} = \mathsf{bin}^{-1}\big(a\big) + 2^t \cdot \mathsf{bin}^{-1}\big(h_R\big) \bmod q.$$
4. Compute $c = H(R, \mu) \in \mathbb{Z}_q$.
5. Compute $s_1 = r + c \cdot (x_1 + \sum_{i=2}^{n+1} \omega_i x_i) - \sum_{i=2}^{n+1} \omega_i \cdot s_i \bmod q$.

Output $\mathsf{sig} = (c, s_1, \ldots, s_{n+1}) \in \mathbb{Z}_q \times \mathbb{Z}_q^{n+1}$.

Unless the failure event F_1 occurs, $\mathcal{O}^2_{\mathsf{aSig}}$ computes $\{\mathsf{PRF}_{k_m}(R, i)\}_{i \in [2,n]}$ for distinct values of R in Game_2 across different queries. On the other hand, Game_3 replaces these PRF values with uniformly chosen ones. By the same argument as in the previous game transition, we can build a distinguisher $\mathcal{B}_1$ against PRF such that $|\Pr[W_3] - \Pr[W_2]| \leq \mathbf{Adv}^{\mathcal{B}_1, \mathsf{PRF}}_{Q_S \cdot (n-1)}(\lambda)$.

Game_4**:** We replace $\mathcal{O}^3_{\mathsf{aSig}}$ by the following anamorphic signing oracle $\mathcal{O}^4_{\mathsf{aSig}}$.

$\mathcal{O}^4_{\mathsf{aSig}}(\mathsf{ask}, \mu, \hat{\mu})$: To sign $\mu \in \mathcal{M}$ and encrypt $\hat{\mu} = (\hat{\mu}_1, \ldots, \hat{\mu}_{n-1}, \hat{\mu}_n)$ using $\mathsf{ask} = (x_1, \ldots, x_{n+1})$, $\mathsf{dk} = (\mathsf{avk}, k_m, \bar{k}_m, K)$, and $\mathsf{tk} = (\omega_2, \ldots, \omega_{n+1})$,

1. Choose $r \hookleftarrow U(\mathbb{Z}_q)$ and compute $R = g^r$. If r was previously used in a signing query, abort the game and output 0.
2. For each $i \in [2, n]$, set $s_i = \hat{\mu}_{i-1} + h_{R,i} \bmod q$ where $h_{R,i} \hookleftarrow U(\mathbb{Z}_q)$.
3. Sample $\bar{h}_R \hookleftarrow \{0,1\}^t$, let $a = \hat{\mu}_n \oplus \bar{h}_R$, sample $h_R \hookleftarrow U(\{0,1\}^\lambda)$ and compute
$$s_{n+1} = \mathsf{bin}^{-1}\big(a\big) + 2^t \cdot \mathsf{bin}^{-1}\big(h_R\big) \bmod q.$$
4. Compute $c = H(R, \mu) \in \mathbb{Z}_q$.
5. Compute $s_1 = r + c \cdot (x_1 + \sum_{i=2}^{n+1} \omega_i \cdot x_i) - \sum_{i=2}^{n+1} \omega_i \cdot s_i \bmod q$.

Output $\mathsf{sig} = (c, s_1, \ldots, s_{n+1}) \in \mathbb{Z}_q \times \mathbb{Z}_q^{n+1}$.

Similarly to the previous transition, as long as F_1 does not occur, $\mathcal{O}^3_{\mathsf{aSig}}$ computes $\overline{\mathsf{PRF}}_{\bar{k}_m}(R)$ for distinct values of R and $\mathcal{O}^4_{\mathsf{aSig}}$ replaces these by fresh random values. We can then immediately build a distinguisher $\mathcal{B}_2$ against $\overline{\mathsf{PRF}}$ such that $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}^{\mathcal{B}_2,\overline{\mathsf{PRF}}}_{Q_S}(\lambda)$.

Game_5**:** We replace $\mathcal{O}^4_{\mathsf{aSig}}$ of Game_4 by the anamorphic signing oracle $\mathcal{O}^5_{\mathsf{aSig}}$. The only change between the two oracles is that in $\mathcal{O}^5_{\mathsf{aSig}}$, the values $s_2, \ldots, s_n$ at step 3 are sampled independently from the distribution $U(\mathbb{Z}_q)$. Since the distribution of $s_2, \ldots, s_n$ is already uniform in Game_4 (thanks to the uniformity of each $h_{R,i}$), this change is only conceptual and we have $\Pr[W_4] = \Pr[W_3]$

Game_6**:** We replace $\mathcal{O}^5_{\mathsf{aSig}}$ with $\mathcal{O}^6_{\mathsf{aSig}}$ as described below. The only difference between these oracles is that $\mathcal{O}^6_{\mathsf{aSig}}$ samples s_{n+1} uniformly over $\mathbb{Z}_q$. Lemma 4 shows that $|\Pr[W_6] - \Pr[W_5]| \leq Q_S/2^\lambda$.

Game_7**:** In this game, we obtain $\mathcal{O}^7_{\mathsf{aSig}}$ by modifying $\mathcal{O}^6_{\mathsf{aSig}}$ in such a way that the oracle no longer aborts the game when r happens to have been used in a previous query. By the same argument as in the transition from Game_1 to Game_2, we have $|\Pr[W_5] - \Pr[W_4]| \leq Q_S^2/2^{2\lambda}$.

Game_8**:** We now set $\mathcal{O}^8_{\mathsf{aSig}}$ to be identical to the $\mathcal{O}_{\mathsf{Sig}}$ oracle. Recall that $\mathcal{O}_{\mathsf{Sig}}(\mu, \hat{\mu})$ simply returns $\mathsf{Sig}(\mathsf{ask}, \mu)$. We describe conceptual changes to the oracle $\mathcal{O}^7_{\mathsf{aSig}}$ that do not affect its output distribution and argue that it generates signatures with exactly the same distribution as those produced by $\mathcal{O}_{\mathsf{Sig}}$. Note that r is distributed uniformly for $\mathcal{O}^7_{\mathsf{aSig}}$. Therefore, in $\mathcal{O}^7_{\mathsf{aSig}}$, we can sample $r, r_2, \ldots, r_{n+1} \hookleftarrow U(\mathbb{Z}_q)$ and set $s_i = r_i + c \cdot x_i$ for each $i \in [2, n]$, where $c = H(R, \mu)$. At this point, the distribution of $(c, r, s_2, \ldots, s_{n+1})$ is unchanged in $\mathcal{O}^7_{\mathsf{aSig}}$. Finally, s_1 is deterministically computed by $\mathcal{O}^7_{\mathsf{aSig}}$ and is equal to $s_1 = r_1 + c \cdot x_1$, where $r_1 \triangleq r - \sum_{i=2}^{n+1} \omega_i \cdot r_i$ is uniformly distributed independently of $(c, s_2, \ldots, s_{n+1})$. Conceptually, $\mathcal{O}^7_{\mathsf{aSig}}$ can equivalently sample $r_i \hookleftarrow U(\mathbb{Z}_q)$ and compute $s_i = r_i + c \cdot x_i$ for $i \in [1, n+1]$, where $c = H(R, \mu)$, for $R = g^{r_1 + \sum_{i=2}^{n+1} \omega_i \cdot r_i}$. This is precisely how real signatures are computed by $\mathsf{Sig}(\mathsf{ask}, \mu)$. Therefore, we have $\Pr[W_8] = \Pr[W_7]$.

In Game_8, the anamorphic signing oracle runs the normal signing algorithm that always returns normal signatures on μ and ignores $\hat{\mu}$ at each query. Since the distribution of $(\mathsf{vk}, \mathsf{sk})$ is identical to the distribution of $(\mathsf{avk}, \mathsf{ask})$, $\mathcal{A}$'s view in Game_8 is thus identical to its view in $\mathsf{RealG}_{\Pi}(\lambda, \mathcal{D})$. □

Lemma 4. *We have* $|\Pr[W_6] - \Pr[W_5]| \leq Q_S/2^\lambda$.

Proof. In Game_5, the distribution of s_{n+1} is $U([0, 2^{\ell_q} - 1])$ (of which the support is contained in $\mathbb{Z}_q$). In Game_6, the distribution of s_2 is $U(\mathbb{Z}_q)$. We can thus apply Lemma 3 with $P = U(\mathbb{Z}_q)$, $Q = U([0, 2^{\ell_q} - 1])$ and $\varepsilon = 2^{-\lambda}$ since, for any $x \in [0, 2^{\ell_q} - 1]$, we have $P(x)/Q(x) = 2^{\ell_q}/q > 1 - 2^{-\lambda}$ since $q < 2^{\ell_q} + 2^\lambda < 2^{\ell_q} + \frac{q}{2^\lambda}$. For any $x \in [2^{\ell_q}, q - 1]$, we have $Q(x) = 0$ and $P(x) = 1/q$.

For any $x \in [0, q - 1]$, we then have $P(x) \geq (1 - 2^{-\lambda}) \cdot Q(x)$ and Lemma 3 implies that $U([0, 2^{\ell_q} - 1])$ is within statistical distance $2^{-\lambda}$ from $U(\mathbb{Z}_q)$. □

Recipient Unforgeability. We now turn to proving recipient unforgeability in the wRUF-CAMA sense. The proof stands in the random oracle model.

Theorem 3. *In the random oracle model, the scheme is recipient weakly unforgeable under the Discrete Logarithm assumption over $\mathbb{G}$. In particular, take any PPT adversary $\mathcal{A}$ making at most Q_H queries to H and Q_S signing oracle queries in $\mathcal{G}^{\mathsf{wRUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}$. If $\epsilon \triangleq \Pr[\mathcal{G}^{\mathsf{wRUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) = 1] - 2 \cdot Q_S \cdot (Q_S + Q_H)/q$, then there is a PPT Discrete Logarithm algorithm $\mathcal{B}$ that succeeds with probability at least $\epsilon \cdot \left(\frac{\epsilon}{Q_S + Q_H} - \frac{1}{q}\right)$.*

Proof. The proof first considers a sequence of hybrid games that begins with $\mathcal{G}^{\mathsf{wRUF\text{-}CAMA}}$ and moves to a game where the signing oracles $\mathcal{O}_{\mathsf{Sig}}$ and $\mathcal{O}_{\mathsf{aSig}}$ oracles do not use the x_1-component of the secret key. After doing this, the Forking Lemma will be used to solve the discrete logarithm problem. For an adversary $\mathcal{A}$, we use W_i to denote the probability of $\mathcal{A}$ outputting a forgery (i.e., a signature on a new message) in Game_i, in which case Game_i outputs 1.

Game_0**:** This is precisely the $\mathcal{G}^{\mathsf{wRUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}(\lambda)$ game.

Game_1**:** We replace $\mathcal{O}_{\mathsf{Sig}}$ by the oracle $\mathcal{O}'_{\mathsf{Sig}}$ defined as follows:

$\mathcal{O}'_{\mathsf{Sig}}(\mu)$: Choose a random tuple $(c, s_1, \ldots, s_{n+1}) \hookleftarrow U(\mathbb{Z}_q \times \mathbb{Z}_q^{n+1})$. Program $H\big(X^{-c} \cdot \prod_{i=1}^{n+1} g_i^{s_i}, \mu\big)$ to be c. It it was already defined, abort and output 0. Otherwise, return $(c, s_1, \ldots, s_{n+1})$.

By the honest-verifier zero-knowledge property of the Σ-protocol, the output distribution of $\mathcal{O}'_{\mathsf{Sig}}$ is the same as that of $\mathcal{O}_{\mathsf{Sig}}$ as long as $\mathcal{O}'_{\mathsf{Sig}}$ does not fail. From $\mathcal{A}$'s view, Game_1 is indistinguishable from Game_0 unless the $\mathcal{O}'_{\mathsf{Sig}}$ aborts when programming H, which happens with probability $\leq Q_S \cdot (Q_S + Q_H)/q$. So, we have $|\Pr[W_1] - \Pr[W_0]| \leq Q_S \cdot (Q_S + Q_H)/q$.

Game_2**:** This is the same as Game_1 apart from that the signing oracle $\mathcal{O}_{\mathsf{aSig}}$ is replaced by $\mathcal{O}'_{\mathsf{aSig}}$, which is defined below:

$\mathcal{O}'_{\mathsf{aSig}}(\mu, \hat{\mu})$:

1. Define $X_1 \triangleq X \cdot \prod_{i=2}^{n+1} g^{-\omega_i \cdot x_i}$.
2. Sample $(c, s'_1) \hookleftarrow U(\mathbb{Z}_q \times \mathbb{Z}_q)$ and set $R_1 = g^{s'_1} \cdot X_1^{-c}$
3. For each $i \in [2, n+1]$, sample $r_i \hookleftarrow U(\mathbb{Z}_q)$, set $R_i = g^{\omega_i \cdot r_i}$ and define $R \triangleq R_1 \cdot R_2 \cdot \cdots \cdot R_{n+1}$.
4. Program $H(R, \mu)$ to equal c. Abort and output 0 if H was already defined for (R, μ).
5. For $i \in [2, n]$, compute $s_i = \hat{\mu}_{i-1} + \mathsf{PRF}_{k_m}(R, i) \bmod q$.
6. Compute $a = \hat{\mu}_n \oplus \overline{\mathsf{PRF}}_{\bar{k}_m}(R)$ and set

$$s_{n+1} = \mathsf{bin}^{-1}(a) + 2^t \cdot \mathsf{bin}^{-1}\left(\mathsf{PRF}'_K(a, \mu, R, s_2, \ldots, s_n)\right) \mod q.$$

7. Set $s_1 = s'_1 + \sum_{i=2}^{n+1} \omega_i \cdot (r_i + c x_i) - \sum_{i=1}^{n+1} \omega_i \cdot s_i$.
8. Return $(c, s_1, \ldots, s_{n+1})$.

The distribution of $(s_2, \ldots, s_{n+1})$ is *exactly* as in Game_1 (this is necessary since $\mathcal{A}$ knows $\mathsf{dk} = (\mathsf{avk}, k_m, \bar{k}_m, K)$ and can recompute $(s_2, \ldots, s_{n+1})$ from $(\mu, \hat{\mu})$ and R). As for the distribution of s_1, we note that the simulated $R \in \mathbb{G}$ at step 3 can be written $R = g^r$ with $r = s'_1 - c \cdot x_1 + \sum_{i=2}^{n+1} \omega_i \cdot r_i$, which is uniformly distributed in $\mathbb{Z}_q$ thanks to the uniformity of s'_1. Then, we have $s'_1 + \sum_{i=2}^{n+1} \omega_i \cdot r_i = r + c \cdot x_1$ and the simulated s_1 of step 6 satisfies

$$s_1 = r + c \cdot \Big(x_1 + \sum_{i=2}^{n+1} \omega_i \cdot x_i\Big) - \sum_{i=2}^{n+1} \omega_i \cdot s_i$$

as in an output of $\mathcal{O}_{\mathsf{aSig}}$ in Game_1 (note that s_1 is uniquely determined by X, R, c and $(s_i)_{i=2}^{n+1}$). Therefore signatures $(c, s_1, \ldots, s_{n+1})$ produced by $\mathcal{O}'_{\mathsf{aSig}}$ are identically distributed to those generated by $\mathcal{O}_{\mathsf{aSig}}$ in Game_1. The two games are thus indistinguishable unless $\mathcal{O}'_{\mathsf{aSig}}$ aborts because of a collision at step 4. This implies $|\Pr[W_2] - \Pr[W_1]| \leq Q_S \cdot (Q_S + Q_H)/q$.

By the triangle inequality, we have $\Pr[W_2] \geq \Pr[W_0] - 2 \cdot Q_S \cdot (Q_S + Q_H)/q$. In Game_2, the signing oracles $\mathcal{O}'_{\mathsf{Sig}}$ and $\mathcal{O}'_{\mathsf{aSig}}$ do not use x_1. This allows embedding a discrete logarithm challenge into X_1. $\mathcal{A}$'s forgeries can then be exploited to extract the solution x_1 via the Forking Lemma, as shown in Lemma 5. The proof of the theorem follows by combining the success probability in Lemma 5 with the lower bound on $\Pr[W_2]$ given above. □

Lemma 5. *Let* $\Pr[W_2]$ *the probability of* Game_2 *outputting 1 for some PPT adversary* $\mathcal{A}$*. Further, suppose that* $\mathcal{A}$ *makes* Q_H *queries to* H *and* Q_S *queries to its signing oracles. Then, there exists a PPT algorithm* $\mathcal{B}$ *that solves the discrete logarithm problem in* $\mathbb{G}$ *with probability at least* $\Pr[W_2] \cdot \Big(\frac{\Pr[W_2]}{Q_S + Q_H + 1} - 1/q\Big)$. (The proof is given in Appendix C.2.)

On Strong Unforgeability. An adversary $\mathcal{A}$ that can compute distinct $\{s'_i\}_{i=1}^{n+1}$ and $\{s_i\}_{i=1}^{n+1}$ such that $s'_1 + \sum_{i=2}^{n+1} \omega_i \cdot s'_i = s_1 + \sum_{i=2}^{n+1} \omega_i \cdot s_i$ can re-randomize signatures by finding a different representation of $R \cdot X^c$. The ability to do so implies the ability to find a non-trivial discrete logarithm relation among $(g_i)_{i=1}^{n+1}$, which is as hard as computing discrete logarithms. However, in the proof of Lemma 5, we cannot rule out this case by relying on the discrete logarithm assumption because it would require to embeded a discrete logarithm instance in $(\omega_i)_{i=2}^{n+1}$ and the reduction $\mathcal{B}$ needs $(\omega_i)_{i=2}^{n+1}$ to simulate $\mathcal{O}'_{\mathsf{aSig}}$. This prevents us from proving strong unforgeability although we do not see any obvious way for an adversary to defeat it.

On the use of tk. We include the discrete logarithms $(\omega_i)_{i=2}^{n+1}$ in the anamorphic signing trapdoor tk. They could alternatively be part of the secret keys sk and ask without any change in the security proofs. We chose to include them in tk because the signer can obliviously sample the generators $(g_i)_{i=2}^{n+1}$ without knowing their discrete logarithms in the normal key generation algorithm.

3.2 A Construction from GQ Signatures

We now show that GQ signatures [33] provide anamorphic signatures that are simultaneously dictator and recipient unforgeable in the random oracle model.

As in [33], the real signing algorithm does not need the factorization of the RSA modulus N (only the e-th root of y is necessary). While the factorization is used in the anamorphic signing algorithm, we will show that a dictator cannot distinguish real signatures from anamorphic ones *even* given the factorization of $N = pq$. Moreover, as required by the dictator unforgeability property, it remains unable to forge anamorphic signatures even knowing the factorization.

In the description, we include the factorization of N in sk and ask[4] because (in contrast with Okamoto-Schnorr signatures and the discrete logarithm setting) there is no way to obliviously generate $N = pq$ without knowing its prime factors. Since the signer has to choose p and q at some step, it makes sense to give it to the dictator in the proofs of stealthiness and dictator unforgeability.

$\mathsf{KGen}(1^\lambda)$: Given a security parameter 1^λ,

1. Choose an RSA modulus $N = pq$ such that $2^{\nu(\lambda)-1} < N < 2^{\nu(\lambda)}$, where p, q are large primes, for some polynomial $\nu : \mathbb{N} \to \mathbb{N}$. Choose a prime $e \hookleftarrow U(\mathcal{P}_\lambda)$ conditionally on $\gcd(e, \varphi(N)) = 1$.
2. Choose $x \hookleftarrow U(\mathbb{Z}_N^*)$ and compute $y = x^e \bmod N$.
3. Choose a hash function $H_{\mathsf{FS}} : \{0,1\}^* \to \mathbb{Z}_e$ modeled as a random oracle.

Output $\mathsf{vk} = \big(N, e, y, H_{\mathsf{FS}}\big)$, $\mathsf{sk} = (x, p, q)$.

$\mathsf{aGen}(1^\lambda)$: Given a security parameter 1^λ,

1. Choose an RSA modulus $N = pq$ such that $2^{\nu(\lambda)-1} < N < 2^{\nu(\lambda)}$, where p, q are large primes, for some polynomial $\nu : \mathbb{N} \to \mathbb{N}$. Choose a prime $e \hookleftarrow U(\mathcal{P}_\lambda)$ conditionally on $\gcd(e, \varphi(N)) = 1$.
2. Choose $x \hookleftarrow U(\mathbb{Z}_N^*)$ and compute $y = x^e \bmod N$.
3. Choose a hash function $H_{\mathsf{FS}} : \{0,1\}^* \to \mathbb{Z}_e$ modeled as a random oracle.
4. Define an anamorphic message length $\ell_m \leq \nu - 2\lambda - 1$ and define the anamorphic message space as $\widehat{\mathcal{M}} = \{0,1\}^{\ell_m}$.
5. Choose random keys $k_m, K \hookleftarrow U(\{0,1\}^\lambda)$ for hash functions

$$h : \{0,1\}^\lambda \times \mathbb{Z}_{\lfloor N/2^\ell \rfloor} \to \{0,1\}^{\ell_m},$$
$$H : \{0,1\}^\lambda \times \{0,1\}^{\ell_m} \times \mathcal{M} \times \mathbb{Z}_{\lfloor N/2^\ell \rfloor} \to \{0,1\}^\lambda.$$

Output $\mathsf{avk} = \big(N, e, y, H_{\mathsf{FS}}\big)$, $\mathsf{ask} = (x, p, q)$, $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, h, H)$ and $\mathsf{tk} = \emptyset$.

$\mathsf{Sig}(\mathsf{sk}, \mu)$: To sign $\mu \in \mathcal{M}$,

1. Choose $r \hookleftarrow U(\mathbb{Z}_N^*)$ and compute $a = r^e \bmod N$.
2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_e$.
3. Compute $z = r \cdot x^c \bmod N$.

[4] We could also include (p, q) in tk without any change in the security proofs.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

aSig$(\mathsf{ask}, \mathsf{tk}, \mathsf{dk}, \mu, \hat{\mu})$**:** To sign $\mu \in \mathcal{M}$ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, h, H)$, let $\ell = \ell_m + \lambda$. Then, conduct the following steps.

1. Choose $a' \hookleftarrow U(\mathbb{Z}_N)$ and compute $\tilde{a} = \lfloor a'/2^{\ell} \rfloor \in \mathbb{Z}_{\lfloor N/2^{\ell} \rfloor}$.
2. Compute $\bar{a}_0 = \hat{\mu} \oplus h(k_m, \tilde{a}) \in \{0,1\}^{\ell_m}$ and then

$$\begin{aligned} a &= \mathsf{bin}^{-1}\big(\bar{a}_0 \;||\; H(K, \bar{a}_0, \mu, \tilde{a}) \;||\; \mathsf{bin}(\tilde{a})\big) \\ &= \mathsf{bin}^{-1}(\bar{a}_0) + 2^{\ell_m} \cdot \mathsf{bin}^{-1}(H(K, \bar{a}_0, \mu, \tilde{a})) + 2^{\ell} \cdot \tilde{a} \end{aligned}$$

3. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_e$.
4. If $\gcd(a \cdot y^c \bmod N, N) \neq 1$, return $\perp$. Otherwise, compute the e-th root $z = (a \cdot y^c)^{1/e} \bmod N$ using p and q.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

Ver$(\mathsf{vk}, \mu, \mathsf{sig})$**:** Parse the candidate signature sig as $(c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$ and return 1 if $c = H_{\mathsf{FS}}(z^e \cdot y^{-c} \bmod N, \mu)$. Otherwise, return 0.

aDec$(\mathsf{dk}, \mu, \mathsf{asig})$**:** Given μ, $(c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, h, H)$, let $\ell = \ell_m + \lambda$. Then, do the following.

1. Return $\perp$ if $\mathsf{Ver}(\mathsf{avk}, \mu, \mathsf{asig}) = 0$. Otherwise, let $a = z^e \cdot y^{-c} \bmod N$.
2. Compute $\bar{a} = a \bmod 2^{\ell}$ and $\tilde{a} = \lfloor a/2^{\ell} \rfloor \in \mathbb{Z}_{\lfloor N/2^{\ell} \rfloor}$.
3. Parse $\bar{a} \in \mathbb{Z}_{2^{\ell}}$ as $\mathsf{bin}(\bar{a}) = (\bar{a}_0 || \bar{a}_1) \in \{0,1\}^{\ell}$ with $\bar{a}_0 \in \{0,1\}^{\ell_m}$ and $\bar{a}_1 \in \{0,1\}^{\lambda}$ and return $\perp$ if $\bar{a}_1 \neq H(K, \bar{a}_0, \mu, \tilde{a})$. Otherwise, compute and return $\hat{\mu} = \bar{a}_0 \oplus h(k_m, \tilde{a}) \in \{0,1\}^{\ell_m}$.

Efficiency. The scheme can use the same parameters as ordinary GQ signatures (with a 3072-bit modulus at the 128-bit security level).

Its anamorphic mode offers a nearly optimal bandwidth rate that asymptotically approaches 1 as the security parameter increases. Indeed, in order to resist factorization attacks, the modulus N should have $\nu = \Omega(\lambda^3/\mathrm{polylog}(\lambda))$ bits while anamorphic messages can have up to $\ell_m \leq \nu - 2\lambda - 1$ bits. If we assume that $|e| = \lambda$, the bandwidth rate is thus $\ell_m/(\nu+\lambda) = 1 - \frac{3\lambda+1}{\nu+\lambda} = 1 - O(\frac{\mathrm{polylog}(\lambda)}{\lambda^2})$. We thus use the super-linear growth of $\nu = \log N$ (as a function of the security parameter) to our advantage. Concretely, if we use $\nu = 3072$ and $|e| = 128$ at the 128-bit security level, the fraction of the signature that can carry anamorphic message bits can be as large as 88%.

In Appendix D.2 and D.3, we provide comparisons with anamorphic schemes obtained from RSA-PSS and the swapped-GQ scheme of [48].

Anamorphic Extension. We remark that steps 1-3 of aGen are identical to those of KGen. Following the notion of *anamorphic extension* of Banfi *et al.* [8], this allows introducing an anamorphic channel in an existing key pair $(\mathsf{vk}, \mathsf{sk})$ *after* its deployment and without any change in the verification key vk. Indeed, the signer can initially run KGen to set up a normal key pair and subsequently run steps 4-5 of aGen so as to create a double key when the need arises.

Theorem 4. *The scheme is dictator unforgeable in the* DUF-CASA *sense. For any PPT adversary $\mathcal{A}$ making at most Q_H random oracle queries and Q_D anamorphic decryption queries, we have* $\mathbf{Adv}^{\mathsf{DUF\text{-}CASA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) \leq (Q_H + Q_D + 1)/2^\lambda$.

Proof. The proof considers a sequence of games. In each game, we let W_i denote the event that Game_i outputs 1.

Game_0**:** This is the real DUF-CASA experiment. The challenger handles random oracle queries in the standard way using lazy sampling and faithfully answers signing and anamorphic decryption queries. Each query $\mathcal{O}_{\mathsf{aSig}}(\mu, \hat{\mu})$ or $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ made by the adversary $\mathcal{A}$ triggers random oracle queries of the form $h(k_m, \cdot)$ and $H(K, \cdot, \mu, \cdot)$, where k_m and K are part of the double key dk chosen by the challenger. By definition, $\mathcal{A}$'s advantage is $\Pr[W_0]$.

Game_1**:** This game is like Game_0 with the difference that the experiment halts and outputs 0 if $\mathcal{A}$ makes a direct (i.e., not triggered by a query to $\mathcal{O}_{\mathsf{aSig}}$) random oracle query of the form $H(K, \cdot, \cdot, \cdot)$, where $K \in \{0,1\}^\lambda$ is part of the double key dk chosen by the game at the outset. Both games are identical until the event F_1 that a hash query $H(K, \cdot, \cdot, \cdot)$ is made. Since $K \sim U(\{0,1\}^\lambda)$ is chosen independently of $\mathcal{A}$'s view at the outset of the game, we have $\Pr[F_1] \leq Q_H/2^\lambda$. Thus, $|\Pr[W_1] - \Pr[W_0]| \leq Q_H/2^\lambda$.

Game_2**:** We modify the $\mathcal{O}_{\mathsf{aDec}}$ oracle. At the first $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ query such that $\hat{\mu} = \mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig}) \neq \perp$ and $(\mu, \hat{\mu}, \mathsf{asig}) \notin S$ (i.e., asig was not the result of an anamorphic signing query $\mathcal{O}_{\mathsf{aSig}}(\mu, \hat{\mu})$), the challenger halts and outputs 0 as the result of the experiment. If we call F_2 the latter event, Lemma 6 shows that $\Pr[F_2] \leq Q_D/2^\lambda$, which implies $|\Pr[W_2] - \Pr[W_1]| \leq Q_D/2^\lambda$ since the two games are identical until F_2 occurs.

Game_3**:** This game is identical to Game_2 except that the challenger outputs 0 as the result of the experiment and ignores the adversary's output $(\hat{\mu}^\star, \mathsf{asig}^\star)$. By the same argument as in Lemma 6, we have $|\Pr[W_3] - \Pr[W_2]| \leq 1/2^\lambda$ since we can see $(\mu^\star, \mathsf{asig}^\star)$ as the adversary's $(Q_D + 1)$-th query to $\mathcal{O}_{\mathsf{aDec}}$.

In Game_3, we have $\Pr[W_3] = 0$ since the challenger always outputs 0. The triangle inequality then leads to the stated upper bound on $\mathcal{A}$'s advantage. □

Lemma 6. *We have* $|\Pr[W_2] - \Pr[W_1]| \leq \Pr[F_2] \leq Q_D/2^\lambda$.

Proof. For each query $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig} = (c, z))$, let $a = z^e \cdot y^{-c} \bmod N$ and let

$$\mathsf{bin}(a) = \big(\bar{a}_0 \mid\mid \bar{a}_1 \mid\mid \mathsf{bin}(\tilde{a})\big) \tag{4}$$

with $\bar{a}_0 \in \{0,1\}^{\ell_m}$, $\bar{a}_1 \in \{0,1\}^\lambda$ and where $\tilde{a} = \lfloor a/2^\ell \rfloor \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}$.

Due to the change introduced in Game_1, the output distributions of the experiment in Game_2 can only differ from the one of Game_1 if F_2 occurs before F_1 (i.e., before $\mathcal{A}$ makes a direct random oracle query of the form $H(K, \cdot, \cdot, \cdot)$). Since F_2 implies $\hat{\mu} = \mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig}) \neq \perp$, it implies $\bar{a}_1 = H(K, \bar{a}_0, \mu, \tilde{a})$ in (4). Moreover, we must have $(\mu, \hat{\mu}, \mathsf{asig}) \notin S$ if F_2 happens.

Since aDec returns $\perp$ if $\mathsf{Ver}(\mathsf{avk}, \mu, \mathsf{asig}) = 0$ at step 1, we first note that

$(\mu, \hat{\mu}, a) \neq (\mu', \hat{\mu}', a' \triangleq z'^e \cdot y^{-c'})$ for any $(\mu', \hat{\mu}', (c', z')) \in S$. Indeed, Ver only accepts a unique $z = \big(a \cdot y^c\big)^{1/e} \bmod N$ for a given pair (μ, a) such that $a \cdot y^c \in \mathbb{Z}_N^*$. Since the $\mathcal{O}_{\mathsf{aSig}}$ oracle never outputs a signature for which $a' \cdot y^{H_{\mathsf{FS}}(a', \mu')} \notin \mathbb{Z}_N^*$, we cannot have $(\mu, \hat{\mu}, a) = (\mu', \hat{\mu}', a' = z'^e \cdot y^{-c'})$ with $c' = H_{\mathsf{FS}}(a, \mu) = c$ and $z' \neq z$. Having shown that $(\mu, \hat{\mu}, a) \neq (\mu', \hat{\mu}', z'^e \cdot y^{-c'})$ for each $(\mu', \hat{\mu}', (c', z')) \in S$, we now consider for each tuple $(\mu', \hat{\mu}', (c', z')) \in S$ the corresponding $a' = z'^e \cdot y^{-c'}$ and its decomposition $\mathsf{bin}(a') = \big(\bar{a}'_0 \mid\mid \bar{a}'_1 \mid\mid \mathsf{bin}(\tilde{a}')\big)$.

The condition $(\mu, \hat{\mu}, a) \neq (\mu', \hat{\mu}', a')$ implies either: (i) $(\mu, \hat{\mu}, \tilde{a}) \neq (\mu', \hat{\mu}', \tilde{a}')$; or (ii) $(\mu, \hat{\mu}, \tilde{a}) = (\mu', \hat{\mu}', \tilde{a}')$ and $(\bar{a}_0, \bar{a}_1) \neq (\bar{a}'_0, \bar{a}'_1)$. Case (ii) is not compatible with F_1 since we would have $\bar{a}_0 = \hat{\mu} \oplus h(k_m, \tilde{a}) = \hat{\mu}' \oplus h(k_m, \tilde{a}') = \bar{a}'_0$, so that we could only have $(\bar{a}_0, \mu, \hat{\mu}, \tilde{a}) = (\bar{a}'_0, \mu', \hat{\mu}', \tilde{a}')$ and $\bar{a}_1 \neq \bar{a}'_1$, in which case $\mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig})$ would output $\perp$ since $\bar{a}'_1$ is an incorrect hash value. In case (i), we have either $(\mu, \tilde{a}) \neq (\mu', \tilde{a}')$ or $(\mu, \tilde{a}) = (\mu', \tilde{a}')$ and $\hat{\mu} \neq \hat{\mu}'$, which implies $\bar{a}_0 = \hat{\mu} \oplus h(k_m, \tilde{a}) \neq \hat{\mu}' \oplus h(k_m, \tilde{a}') = \bar{a}'_0$. Either way, case (i) implies that $H(K, \bar{a}_0, \mu, \tilde{a})$ has never been defined when answering a $\mathcal{O}_{\mathsf{aSig}}$-query.

Therefore, since the random oracle value $H(K, \bar{a}_0, \mu, \tilde{a})$ has never been defined when answering direct hash queries or anamorphic signing queries, the probability to have $\bar{a}_1 = H(K, \bar{a}_0, \mu, \tilde{a})$ in (4) is only $1/2^\lambda$. We obtain the stated upper bound on $\Pr[F_2]$ by a union bound over all Q_D queries to $\mathcal{O}_{\mathsf{aDec}}$. □

We now prove recipient unforgeability in the random oracle model under the ϕ-hiding assumption. We note that it is possible to prove security under the standard RSA assumption using the Forking Lemma. However, a tighter security proof can be obtained via the ϕ-hiding assumption by adapting ideas from [2,1].

Theorem 5. *In the random oracle model, the scheme is recipient unforgeable in the* RUF-CAMA *sense under the ϕ-hiding assumption if $N > 2^{\ell_m + 2\lambda + 1}$. For any PPT adversary $\mathcal{A}$ making at most Q_H random oracle queries and Q_S (regular or anamorphic) signing queries, there is a ϕ-hiding distinguisher $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\mathsf{RUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) \leq \mathbf{Adv}^{\phi\text{-hide}}_{\mathcal{B}}(\lambda) + (Q_S + 1)(Q_S + Q_H + 1)/2^{\lambda - 1}.$$

Proof. In each game, we let W_i denote the event that the Game_i experiment outputs 1. For each i, we also denote by $\mathcal{O}^i_{\mathsf{Sig}}$ and $\mathcal{O}^i_{\mathsf{aSig}}$ the normal and anamorphic signing oracles of Game_i.

Game_0**:** This is the RUF-CAMA experiment of Definition 7, where the adversary is given avk and dk at the beginning of the game. The challenger handles random oracle queries using lazy sampling and faithfully answers regular and anamorphic signing queries. Each query $\mathcal{O}^0_{\mathsf{Sig}}(\mu)$ or $\mathcal{O}^0_{\mathsf{aSig}}(\mu, \hat{\mu})$ made by the adversary $\mathcal{A}$ triggers random oracle queries of the form $h(k_m, \cdot)$ and $H(K, \cdot, \mu, \cdot)$, where k_m and K are part of the double key dk given to $\mathcal{A}$ at the outset of the game. By definition, $\mathcal{A}$'s advantage is $\Pr[W_0]$.

Game_1**:** This game is like Game_0 except that the challenger halts and outputs 0 as the result of the experiment in the event F_1 that the anamorphic signing oracle $\mathcal{O}^1_{\mathsf{aSig}}$ chooses $a' \hookleftarrow \mathbb{Z}_N$ such that $\tilde{a} = \lfloor a'/2^\ell \rfloor$ previously appeared in a

random oracle query of the form $h(k_m, \tilde{a})$ or $H(K, \cdot, \cdot, \tilde{a})$ or in an earlier $\mathcal{O}^1_{\mathsf{aSig}}$-query. We have $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1]$ since the two games are identical until F_1 occurs and Lemma 7 shows that $\Pr[F_1] \leq Q_S \cdot (Q_H + Q_S)/2^\lambda$.

Game_2: We modify the anamorphic signing oracle $\mathcal{O}^1_{\mathsf{aSig}}$ of Game_1 as follows.

$\mathcal{O}^2_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, h, H)$,

1. Choose $a \hookleftarrow U(\mathbb{Z}_N)$ and compute
$$\tilde{a} = \lfloor a/2^\ell \rfloor \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}, \qquad \bar{a} = a \bmod 2^\ell,$$
where $\ell = \ell_m + \lambda$. Abort and output 0 if $\tilde{a}$ appeared in an earlier hash or $\mathcal{O}^2_{\mathsf{aSig}}$-query.
2. Let $(\bar{a}_0 || \bar{a}_1) = \mathsf{bin}(\bar{a}) \in \{0,1\}^\ell$, where $\bar{a}_0 \in \{0,1\}^{\ell_m}$, $\bar{a}_1 \in \{0,1\}^\lambda$.
3. Program $h(k_m, \tilde{a}) = \hat{\mu} \oplus \bar{a}_0 \in \{0,1\}^{\ell_m}$ and $H(K, \bar{a}_0, \mu, \tilde{a}) = \bar{a}_1$.
4. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_e$.
5. If $\gcd(a \cdot y^c \bmod N, N) \neq 1$, return $\perp$. Otherwise, compute the e-th root $z = (a \cdot y^c)^{1/e} \bmod N$ using p and q.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

Lemma 9 shows that Game_2 is statistically indistinguishable from Game_1 and that $|\Pr[W_2] - \Pr[W_1]| \leq Q_S/2^{\lambda+1}$.

Game_3: In this game, we modify the anamorphic signing oracle which now chooses $a \hookleftarrow U(\mathbb{Z}_N^*)$ instead of $a \hookleftarrow U(\mathbb{Z}_N)$ at step 1. Also, it does not explicitly check that $\gcd(a \cdot y^c \bmod N, N) = 1$ at step 5 since this always the case (recall that $y \in \mathbb{Z}_N^*$). Now, $\mathcal{O}^3_{\mathsf{aSig}}$ is specified as follows:

$\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, h, H)$,

1. Pick $a \hookleftarrow U(\mathbb{Z}_N^*)$. Compute $\bar{a} = a \bmod 2^\ell$ and $\tilde{a} = \lfloor a/2^\ell \rfloor \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}$. Abort and output 0 if $\tilde{a}$ appeared in an earlier hash or $\mathcal{O}^3_{\mathsf{aSig}}$-query.
2. Let $(\bar{a}_0 || \bar{a}_1) = \mathsf{bin}(\bar{a}) \in \{0,1\}^\ell$, where $\bar{a}_0 \in \{0,1\}^{\ell_m}$, $\bar{a}_1 \in \{0,1\}^\lambda$.
3. Program $h(k_m, \tilde{a}) = \hat{\mu} \oplus \bar{a}_0 \in \{0,1\}^{\ell_m}$ and $H(K, \bar{a}_0, \mu, \tilde{a}) = \bar{a}_1$.
4. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_e$.
5. Compute the e-th root $z = (a \cdot y^c)^{1/e} \bmod N$ using p and q.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

The only change is thus the distribution of a which is now uniform in the group of units $\mathbb{Z}_N^*$ instead of uniform over the full ring $\mathbb{Z}_N$. However, the two distributions are at most $(p + q - 1)/N < 2/\min(p, q) < 2^\lambda$ apart in terms of statistical distance and we have $|\Pr[W_3] - \Pr[W_2]| \leq Q_S/2^\lambda$.

Game_4: This game is like Game_3 except that the challenger aborts and outputs 0 in the event F_4 that $H_{\mathsf{FS}}(a, \mu)$ was already defined (because of an earlier random oracle query to H_{FS}) at step 3 of $\mathcal{O}^4_{\mathsf{aSig}}$ or at step 2 of $\mathcal{O}^4_{\mathsf{Sig}}$. Apart from this modification, the signing oracles $\mathcal{O}^4_{\mathsf{Sig}}$ and $\mathcal{O}^4_{\mathsf{aSig}}$ are identical to $\mathcal{O}^3_{\mathsf{Sig}}$ and $\mathcal{O}^3_{\mathsf{aSig}}$, respectively. Since a is chosen uniformly in an exponentially large space $\mathbb{Z}_N^*$, we have $|\Pr[W_4] - \Pr[W_3]| \leq \Pr[F_4] \leq Q_S \cdot (Q_S + Q_H)/2^\lambda$.

Game_5**:** We modify again the anamorphic signing oracle $\mathcal{O}^4_{\mathsf{aSig}}(\mu, \hat{\mu})$, which is now replaced by the following oracle.

$\mathcal{O}^5_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu}$ using $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$,

1. Choose $z \hookleftarrow U(\mathbb{Z}_N^*)$, $c \hookleftarrow U(\mathbb{Z}_e)$ and compute $a = z^e \cdot y^{-c} \bmod N$. Then, compute
$$\tilde{a} = \lfloor a/2^\ell \rfloor \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}, \qquad \bar{a} = a \bmod 2^\ell,$$
where $\ell = \ell_m + \lambda$. Abort and output 0 if $\tilde{a}$ appeared in an earlier hash or $\mathcal{O}^5_{\mathsf{aSig}}$-query.
2. Let $(\bar{a}_0 || \bar{a}_1) = \mathsf{bin}(\bar{a}) \in \{0,1\}^\ell$, where $\bar{a}_0 \in \{0,1\}^{\ell_m}$, $\bar{a}_1 \in \{0,1\}^\lambda$.
3. Program $h(k_m, \tilde{a}) = \hat{\mu} \oplus \bar{a}_0 \in \{0,1\}^{\ell_m}$ and $H(K, \bar{a}_0, \mu, \tilde{a}) = \bar{a}_1$.
4. If $H_{\mathsf{FS}}(a, \mu)$ was already defined, abort and return 0. Otherwise, program $H_{\mathsf{FS}}(a, \mu) = c$ and output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

The output distribution of $\mathcal{O}^5_{\mathsf{aSig}}$ is exactly the same as that of $\mathcal{O}^4_{\mathsf{aSig}}$. Indeed, $\mathcal{O}^4_{\mathsf{aSig}}$ samples a uniformly in $\mathbb{Z}_N^*$, which implies that $z = (a \cdot y^c)^{1/e} \bmod N$ is itself uniform in $\mathbb{Z}_N^*$ since $\gcd(e, \varphi(N)) = 1$ and the mapping $x \mapsto x^e \bmod N$ is one-to-one over $\mathbb{Z}_N^*$. We have $\Pr[W_5] = \Pr[W_4]$.

Game_6**:** We modify again the normal signing oracle $\mathcal{O}^5_{\mathsf{aSig}}(\mu)$ as follows.

$\mathcal{O}^6_{\mathsf{Sig}}(\mu)$: To sign μ,

1. Choose $z \hookleftarrow U(\mathbb{Z}_N^*)$, $c \hookleftarrow U(\mathbb{Z}_e)$ and compute $a = z^e \cdot y^{-c} \bmod N$.
2. Program $H_{\mathsf{FS}}(a, \mu) = c$. If it is already defined, abort and return 0.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

The output distribution of $\mathcal{O}^6_{\mathsf{Sig}}$ is exactly the same as that of $\mathcal{O}^5_{\mathsf{Sig}}$ by the same argument as in the previous transition. We have $\Pr[W_6] = \Pr[W_5]$. After this change, we note that $\mathcal{O}^6_{\mathsf{aSig}}$ does not use the factorization of N and $\mathcal{O}^6_{\mathsf{Sig}}$ no longer uses the e-th root of the public key $y = x^e \bmod N$.

Game_7**:** We change the key generation phase. Instead of choosing $y = x^e \bmod N$ for $x \hookleftarrow U(\mathbb{Z}_N^*)$, aGen chooses $y \hookleftarrow U(\mathbb{Z}_N^*)$ uniformly (which is possible since x is not used by $\mathcal{O}_{\mathsf{Sig}}$ nor $\mathcal{O}_{\mathsf{aSig}}$ in Game_6). Since $\gcd(e, \varphi(N)) = 1$, the distribution of avk remains as in Game_6 and we have $\Pr[W_7] = \Pr[W_6]$.

Game_8**:** We now modify the distribution of the public key. Instead of choosing (e, N) such that $\gcd(e, \varphi(N)) = 1$, aGen chooses (e, N) such that $e \,|_1\, \varphi(N)$. Under the ϕ-hiding assumption for the set of λ-bit primes $\mathcal{E} = \mathcal{P}_\lambda$, this change goes unnoticed and we have $|\Pr[W_8] - \Pr[W_7]| \leq \mathbf{Adv}^{\phi\text{-hide}}_{\mathcal{B}}(\lambda)$. Indeed, if this modified public key significantly impacts $\mathcal{A}$'s behavior, the challenger can immediately turn $\mathcal{A}$ into a ϕ-hiding distinguisher since the factorization of N is not used at all in Game_7.

In Game_8, Lemma 10 shows that $\Pr[W_8] \leq (Q_H + 1)/2^{\lambda - 1}$, which is negligible since Q_H is polynomial. We then obtain the stated upper bound on the adversary's advantage $\Pr[W_0] = \mathbf{Adv}^{\mathsf{RUF\text{-}CAMA}}_{\mathcal{A}, \Pi, \Sigma}(\lambda)$ by counting probabilities. □

Lemma 7. *In* Game_1*, we have* $\Pr[F_1] \leq Q_S \cdot (Q_H + Q_S)/2^\lambda$.

Proof. At each anamorphic signing query, $\mathcal{O}^1_{\mathsf{aSig}}$ samples $a' \hookleftarrow \mathbb{Z}_N$ and computes $\tilde{a} = \lfloor a'/2^\ell \rfloor$, which means that $\tilde{a}$ is sampled from a distribution with min-entropy at least $\log N - (\ell + 1)$. Indeed, for each $i \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}$, we have

$$\Pr\left[\lfloor a'/2^\ell \rfloor = i \mid a' \hookleftarrow \mathbb{Z}_N\right] = \begin{cases} \frac{2^\ell}{N} & \text{if} \quad i \in [1, \lfloor N/2^\ell \rfloor - 1] \\ \frac{2^\ell}{N} + \frac{N \bmod 2^\ell}{N} & \text{if} \quad i = 0 \end{cases}$$

so that the maximal probability of any $\tilde{a} \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}$ in the support of the distribution is bounded by $p_{\max} \triangleq \frac{2^\ell}{N} + \frac{2^\ell}{N} = \frac{2^{\ell+1}}{N}$, where $\ell = \ell_m + \lambda$. Since $N > 2^{\ell_m + 2\lambda + 1} = 2^{\ell + 1 + \lambda}$, we have $p_{\max} < 2^{-\lambda}$. For each $\mathcal{O}^1_{\mathsf{aSig}}$-query, the probability to choose an $\tilde{a} = \lfloor a'/2^\ell \rfloor$ that appeared in an earlier $\mathcal{O}^1_{\mathsf{aSig}}$-query or in an earlier random oracle input is $\leq (Q_H + Q_S)/2^\lambda$. By a union bound over all signing queries, we obtain $\Pr[F_1] \leq Q_S \cdot (Q_H + Q_S)/2^\lambda$. □

Lemma 8. *Let integers* $\ell, L > 0$ *with* $L > \ell$*. The statistical distance between* $D_1 = \{x \bmod \ell \mid x \hookleftarrow \mathbb{Z}_L\}$ *and* $D_2 = \{x \hookleftarrow \mathbb{Z}_\ell\}$ *is bounded by* $\Delta(D_1, D_2) \leq \ell/L$*. In particular, for* $L \geq \ell \cdot 2^\lambda$*, the statistical distance is at most* $2^{-\lambda}$*.*

Lemma 9. *We have* $|\Pr[W_2] - \Pr[W_1]| \leq Q_S/2^{\lambda+1}$.

Proof. In Game_2, at each anamorphic signing query, $\mathcal{O}^2_{\mathsf{aSig}}$ programs $h(k_m, \cdot)$ and $H(K, \cdot, \cdot, \cdot)$ using the ℓ-bit string $(\bar{a}_0 || \bar{a}_1) = \mathsf{bin}(\bar{a}) \in \{0,1\}^\ell$. In Game_1, $h(k_m, \tilde{a})$ can be written $h(k_m, \tilde{a}) = \hat{\mu} \oplus \bar{a}_0$, for some uniform $\bar{a}_0 \sim U(\{0,1\}^{\ell_m})$, and we have $H(K, \bar{a}_0, \mu, \tilde{a}) = \bar{a}_1 \sim U(\{0,1\}^\lambda)$.

In Game_2, $(\bar{a}_0 || \bar{a}_1) = \mathsf{bin}(a \mod 2^\ell)$ is no longer a truly random string since N is not a power of 2. However, the distribution of $(\bar{a}_0 || \bar{a}_1)$ is in bijection with the distribution $\{\bar{a} = a \bmod 2^\ell \mid a \hookleftarrow U(\mathbb{Z}_N)\}$, which is within statistical distance $1/2^{\lambda+1}$ from $U(\mathbb{Z}_{2^\ell})$ by Lemma 8. Hence, the distribution of $(\bar{a}_0 || \bar{a}_1)$ is within distance $2^{-(\lambda+1)}$ from the uniform distribution $U(\{0,1\}^\ell)$ at each $\mathcal{O}^2_{\mathsf{aSig}}$-query. The stated upper bound follows from a union bound over all signing queries. □

Lemma 10. *In* Game_8*, we have* $\Pr[W_8] \leq (Q_H + 1)/2^{\lambda-1}$.

Proof. In Game_8, e divides $\varphi(N)$ exactly once and y is chosen uniformly in $\mathbb{Z}^*_N$. So, if we call E the event that y happens to be an e-th residue in $\mathbb{Z}^*_N$, we have $\Pr[E] = 1/e < 1/2^{\lambda-1}$ and we can write $\Pr[W_8] \leq \Pr[W_8 | \neg E] + \Pr[E]$.

The term $\Pr[W_8 | \neg E]$ can be bounded using the same argument as in [1]. Namely, for any hash query of the form $H_{\mathsf{FS}}(a, \cdot)$, there exists at most one output $c_a \in \mathbb{Z}_e$ such that $a \cdot y^{c_a} \bmod N$ is an e-th residue, meaning that there exists $z \in \mathbb{Z}^*_N$ that would make (c_a, z) a valid signature. Indeed, if there were at least two distinct $c_a, c'_a \in \mathbb{Z}_e$ such that $a \cdot y^{c_a} = z^e \bmod N$ and $a \cdot y^{c'_a} = z'^e \bmod N$ for some $z, z' \in \mathbb{Z}^*_N$, it would contradict the condition $\neg E$ since we would have $y^{c_a - c'_a} = (z/z')^e \bmod N$ and $\gcd(c_a - c'_a, e) = 1$.

For any fresh hash query $H_{\mathsf{FS}}(a, \cdot)$ (i.e., not triggered by a signing query),

the probability that the random oracle outputs the unique $c_a \in \mathbb{Z}_e$ such that $a \cdot y^{c_a}$ is an e-th residue is only $1/e < 1/2^{\lambda-1}$. By a union bound over all hash queries, we obtain $\Pr[W_8|\neg E] \leq Q_H/2^{\lambda-1}$, which completes the proof. □

Theorem 6. *In the random oracle model, the scheme provides stealthiness. For any PPT adversary $\mathcal{A}$ making at most Q_H random oracle queries and Q_S signing queries, we have* $\mathbf{Adv}^{\mathsf{Stealthiness}}_{\mathcal{D},\Pi,\Sigma}(\lambda) \leq (Q_S^2 + 2Q_S + Q_H)/2^{\lambda-1}$. (The proof is given in Appendix D.1).

In Appendix E, we adapt the anamorphic triplet to the case of 2^t-root signatures. The technique is similar with the difference that we need to embed the anamorphic message in the subgroup of quadratic residues.

References

1. Abdalla, M., Benhamouda, F., Pointcheval, D.: On the tightness of forward-secure signature reductions. In: PKC (2013). https://doi.org/10.1007/978-3-642-36362-7_19
2. Abdalla, M., Fouque, P.A., Lyubashevsky, V., Tibouchi, M.: Tightly-secure signatures from lossy identification schemes. In: Eurocrypt (2012). https://doi.org/10.1007/978-3-642-29011-4_34
3. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: STOC (1996). https://doi.org/10.1145/237814.237838
4. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Math. Cryptol. **9**(3), 169–203 (2015)
5. Aldous, D., Diaconis, P.: Strong uniform times and finite random walks. Adv. in Appl. Math. **8**(1) (1987)
6. Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Crypto (2009). https://doi.org/10.1007/978-3-642-03356-8_3
7. Banerjee, S., Pal, T., Rupp, A., Slamanig, D.: Simple public key anamorphic encryption and signature using multi-message extensions. Cryptology ePrint Archive Report 2025/370 (2025)
8. Banfi, F., Gegier, K., Hirt, M., Maurer, U., Rito, G.: Anamorphic encryption, revisited. In: Eurocrypt (2024). https://doi.org/10.1007/978-3-031-58723-8_1
9. Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., Peikert, C.: More efficient commitments from structured lattice assumptions. In: SCN (2018). https://doi.org/10.1007/978-3-319-98113-0_20
10. Bellare, M., Neven, G.: Multi-signatures in the plain public key model and a general forking lemma. In: ACM-CCS (2006). https://doi.org/10.1145/1180405.1180453
11. Bellare, M., Ristov, T.: A characterization of chameleon hash functions and new, efficient designs. In: Asiacrypt (2008). https://doi.org/10.1007/978-3-540-89255-7_9
12. Bellare, M., Rogaway, P.: The exact security of digital signatures: How to sign with RSA and Rabin. In: Eurocrypt (1996). https://doi.org/10.1007/3-540-68339-9_34

13. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Crypto (2004). https://doi.org/10.1007/978-3-540-28628-8_3
14. Boneh, D., X., B.: Short signatures without random oracles. In: Eurocrypt (2004)
15. Bonte, C., Iliashenko, I., Park, J., Pereira, H.V.L., Smart, N.P.: FINAL: faster FHE instantiated with NTRU and LWE. In: Asiacrypt (2022). https://doi.org/10.1007/978-3-031-22966-4_7
16. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: STOC (2013). https://doi.org/10.1145/2488608.2488680
17. Cachin, C., Micali, S., Stadler, M.: Computationally private information retrieval with polylogarithmic communication. In: Eurocrypt (1999). https://doi.org/10.1007/3-540-48910-X_28
18. Catalano, D., Giunta, E., Migliaro, F.: Anamorphic encryption: New constructions and homomorphic realizations. In: Eurocrypt (2024). https://doi.org/10.1007/978-3-031-58723-8_2
19. Catalano, D., Giunta, E., Migliaro, F.: Limits of black-box anamorphic encryption. In: Crypto (2024). https://doi.org/10.1007/978-3-031-68379-4_11
20. Catalano, D., Giunta, E., Migliaro, F.: Generic anamorphic encryption, revisited: New limitations and constructions. In: Eurocrypt (2025), https://eprint.iacr.org/2024/1119
21. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. of Cryptology (10) (1997). https://doi.org/10.1007/S001459900030
22. Coron, J.S.: On the exact security of Full Domain Hash. In: Crypto (2000). https://doi.org/10.1007/3-540-44598-6_14
23. Coron, J.S.: Optimal security proofs for PSS and other signature schemes. In: Eurocrypt (2002). https://doi.org/10.1007/3-540-46035-7_18
24. Ducas, L., van Woerden, W.: NTRU fatigue: How stretched is overstretched? In: Tibouchi, M., Wang, H. (eds.) Asiacrypt (2021). https://doi.org/10.1007/978-3-030-92068-5_1
25. Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Crypto (1984). https://doi.org/10.1007/3-540-39568-7_2
26. Even, S., Goldreich, O., Micali, S.: On-Line/Off-Line Digital Signature. J. of Cryptology **9**(1) (1996). https://doi.org/10.1007/BF02254791
27. Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G.: Leakage-resilient signatures. In: TCC (2010). https://doi.org/10.1007/978-3-642-11799-2_21
28. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Crypto (1986). https://doi.org/10.1007/3-540-47721-7_12
29. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC (2008). https://doi.org/10.1145/1374376.1374407
30. Goh, E.J., Jarecki, S.: A signature scheme as secure as the Diffie-Hellman problem. In: Eurocrypt (2003). https://doi.org/10.1007/3-540-39200-9_25
31. Goldwasser, S., Micali, S., Rivest, R.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. of Computing (17) (1988). https://doi.org/10.1137/0217017
32. Guillou, L., Joye, M.: GQ2 vs. ECC: A comparative study of two efficient authentication technologies. In: International Conference on Affective Computing and Intelligent Interaction (ICACII). Lecture Notes in Information Technology (2012), https://marcjoye.github.io/papers/GJ12eccgq2.pdf

33. Guillou, L., Quisquater, J.J.: A "Paradoxical" Indentity-Based Signature Scheme Resulting from Zero-Knowledge. In: Crypto (1988). https://doi.org/10.1007/0-387-34799-2_16
34. Håstad, J., Impagliazzo, R., Levin, L., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing (1999). https://doi.org/10.1137/S0097539793244708
35. Housley, R., Schaad, J., Kaliski, B.: Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 4055 (2005). https://doi.org/10.17487/RFC4055, https://www.rfc-editor.org/info/rfc4055
36. ISO/IEC1488-2: Information technology – Security techniques – Digital signatureswith appendix – Part 2: Integer factorization based mechanisms. International Organization for Standardization (2008)
37. Jaeger, J., Stracovsky, R.: Dictators? Friends? Forgers. Breaking and Fixing Unforgeability Definitions for Anamorphic Signature Schemes. In: Asiacrypt (2024). https://doi.org/10.1007/978-981-96-0888-1_4
38. Kakvi, S.: On the security of RSA-PSS in the wild. In: Security Standardisation Research Conference (SSR) (2019). https://doi.org/10.1145/3338500.3360333
39. Kaliski, B.: IEEE standard specifications for public-key cryptography. IEEE Std 1363 (Aug 2000). https://doi.org/10.1109/IEEESTD.2000.92292
40. Katz, J.: Digital Signatures. Springer (2010). https://doi.org/10.1007/978-0-387-27712-7
41. Katz, J., Wang, N.: Efficiency improvement for signature schemes with tight reduction. In: ACM-CCS (2003). https://doi.org/10.1145/948109.948132
42. Kutylowski, M., Persiano, G., Phan, D.H., Yung, M., Zawada, M.: Anamorphic Signatures: Secrecy From a Dictator Who Only Permits Authentication! In: Crypto (2023). https://doi.org/10.1007/978-3-031-38545-2_25
43. Kutylowski, M., Persiano, G., Phan, D.H., Yung, M., Zawada, M.: The self-anticensorship nature of encryption: On the prevalence of anamorphic cryptography. In: PoPETS. No. 4 (2023). https://doi.org/10.56553/POPETS-2023-0104
44. Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: Asiacrypt (2009). https://doi.org/10.1007/978-3-642-10366-7_35
45. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Eurocrypt (2012). https://doi.org/10.1007/978-3-642-29011-4_34
46. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Eurocrypt (2010). https://doi.org/10.1007/978-3-642-13190-5_1
47. Micali, S.: A secure and efficient digital signature algorithm. Technical Report MIT/LCS/TM-501, Massachusetts Institute of Technology (Mar 1994)
48. Micali, S., Reyzin, L.: Improving the exact security of digital signature schemes. Jo. of Cryptology **15**(2) (2002). https://doi.org/10.1007/S00145-001-0005-8
49. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Eurocrypt (2012). https://doi.org/10.1007/978-3-642-29011-4_41
50. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: STOC (1989). https://doi.org/10.1145/73007.73011
51. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Crypto (1992). https://doi.org/10.1007/3-540-48071-4_3
52. Ong, H., Schnorr, C.P.: Fast signature generation with a Fiat-Shamir like scheme. In: Eurocrypt (1990). https://doi.org/10.1007/3-540-46877-3_38

53. Persiano, G., Phan, D.H., Yung, M.: Anamorphic encryption: Private communication against a dictator. In: Eurocrypt (2022). https://doi.org/10.1007/978-3-031-07085-3_2
54. Persiano, G., Phan, D.H., Yung, M.: Public-Key Anamorphism in (CCA-Secure) Public-Key Encryption. In: Crypto (2024). https://doi.org/10.1007/978-3-031-68379-4_13
55. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. J. of Cryptology (2000). https://doi.org/10.1007/S001450010003
56. Prest, T., Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon. Technical report. National Institute of Standards and Technology (2020), http://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions
57. RSA Laboratories: PKCS#1 v2.1: RSA Cryptography Standard. Draft 2 (Jan 2001)
58. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Crypto (1989). https://doi.org/10.1007/0-387-34805-0_22
59. Schnorr, C.P.: Security of 2^t-root identification and signatures. In: Crypto (1996). https://doi.org/10.1007/3-540-68697-5_12
60. Shamir, A., Tauman, Y.: Improved Online/Offline Signature Schemes. In: Crypto (2001). https://doi.org/10.1007/3-540-44647-8_21
61. Shoup, V.: On the security of a practical identification scheme. In: Eurocrypt (1996). https://doi.org/10.1007/3-540-68339-9_30
62. Wang, Y., Chen, R., Huang, X., Yung, M.: Sender-anamorphic encryption reformulated: Achieving robust and generic constructions. In: Asiacrypt (2023). https://doi.org/10.1007/978-981-99-8736-8_5
63. Waters, B.: Efficient identity-based encryption without random oracles. In: Eurocrypt (2005). https://doi.org/10.1007/11426639_7

A Deferred Background

A.1 Pseudorandom Functions

We recall the standard definition of pseudorandom function families.

Definition 8. *Let λ be a security parameter and let $\kappa = \kappa(\lambda)$. A pseudorandom function family is a family of efficiently computable functions $\mathsf{PRF}\colon \mathcal{K}\times\{0,1\}^\kappa \to \mathcal{R}$ with key space $\mathcal{K}(\lambda)$ and range $\mathcal{R}(\lambda)$ such that no PPT distinguisher has non-negligible advantage, as defined hereunder. Let Ω be the set of all functions from $\{0,1\}^\kappa$ to $\mathcal{R}$. The advantage of a PRF distinguisher $\mathcal{D}$ making Q evaluation queries is defined as*

$$\mathbf{Adv}_Q^{\mathcal{D},\mathsf{prf}}(\lambda) := \left|\Pr[\mathcal{D}^{\mathsf{PRF}(K,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{D}^{F(\cdot)}(1^\lambda) = 1]\right| ,$$

where the probability is taken over the random choice of $K \hookleftarrow U(\mathcal{K})$ and $F \hookleftarrow U(\Omega)$ and the coin tosses of $\mathcal{D}$.

A.2 Lattices, Trapdoors and Discrete Gaussians

The SIS assumption [3] is commonly used in the design of lattice-based signatures [29,44,45]. We recall its definition with respect to the ℓ_2-norm next.

Definition 9. *Take integers $q, n, m, \beta = \mathsf{poly}(\lambda)$. The $\mathsf{SIS}_{q,n,m,\beta}$ assumption posits that for any PPT algorithm $\mathcal{A}$, the following holds:*

$$\mathbf{Adv}^{\mathsf{SIS}}_{\mathcal{A}}(\lambda) := \Pr\left[\mathbf{A}\cdot\mathbf{s}=\mathbf{0} \bmod q \ \wedge\ 0<\|\mathbf{s}\|\leq\beta \mid \mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times m}) \ \text{;} \ \mathbf{s}\leftarrow\mathcal{A}(\mathbf{A})\right] \leq \mathsf{negl}(\lambda).$$

The $\mathsf{HNFSIS}_{q,n,m,\beta}$ problem replaces $\mathbf{A}$ by $[\mathbf{I}_n \mid \bar{\mathbf{A}}]$ where $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{n\times m-n})$.

A full rank n-dimensional lattice Λ is a discrete additive subgroup of $\mathbb{R}^n$. A basis of Λ consists of vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ such that $\Lambda = \{\sum_{i=1}^n z_i \boldsymbol{b}_i : z_i \in \mathbb{Z}\}$. We often represent a basis as a matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$ whose columns are the basis vectors. Further, for a basis matrix $\mathbf{B}$, $\|\mathbf{B}\|_{GS}$ denotes the length of the longest vector in the matrix resulting from Gram-Schmidt orthogonalizing the columns of $\mathbf{B}$. For a matrix $\mathbf{R}$, $s_1(\mathbf{R})$ denotes the first singular value of $\mathbf{R}$.

We often take $\mathbf{R} \sim \mathcal{P}^{n_R\times M_R}$ where $\mathcal{P}$ is the distribution over $\{0, \pm 1\}$ that takes the value 0 with probability 1/2 and ± 1 each with probability 1/4. We have the following lemma stated in [49, Section 5.2].

Lemma 11. *If $\mathbf{R} \sim \mathcal{P}^{n_R\times m_R}$, then $s_1(\mathbf{R}) \leq \sqrt{n_R} + \sqrt{m_R}$ with probability $2^{-\Omega(n_R+m_R)}$.*

For a lattice Λ, $D_{\Lambda,\sigma}$ denotes the discrete Gaussian distribution over Λ with parameter σ. Concretely, $D_{\Lambda,\sigma}$ has support on Λ and probability mass function $\rho_{\Lambda,\sigma}(\boldsymbol{x}) = \exp(-\pi\|\boldsymbol{x}\|^2/\sigma^2)/\sum_{\boldsymbol{y}\in\Lambda}\exp(-\pi\|\boldsymbol{y}\|^2/\sigma^2)$. When $\Lambda = \mathbb{Z}$, we simply write D_σ as short-hand for $D_{\mathbb{Z},\sigma}$. We can also sample Gaussians on cosets of lattices $\Lambda + \boldsymbol{c}$, denoted by $D_{\Lambda+\boldsymbol{c},\sigma}$. The probability mass function is then given by $\rho_{\Lambda+\boldsymbol{c},\sigma}(\boldsymbol{x}) = \exp(-\pi\|\boldsymbol{x}\|^2/\sigma^2)/\sum_{\boldsymbol{y}\in\Lambda+\boldsymbol{c}}\exp(-\pi\|\boldsymbol{y}\|^2/\sigma^2)$ over the support $\Lambda + \boldsymbol{c}$. Finally, the discrete Gaussian centered on $\boldsymbol{v}$ over a lattice Λ, denoted $D_{\Lambda,\boldsymbol{v},\sigma}$, has support on Λ and probability mass function given by

$$\rho_{\Lambda,\boldsymbol{v},\sigma}(\boldsymbol{x}) = \exp(-\pi\|\boldsymbol{x}-\boldsymbol{v}\|^2/\sigma^2)/\sum_{\boldsymbol{y}\in\Lambda}\exp(-\pi\|\boldsymbol{y}-\boldsymbol{v}\|^2/\sigma^2).$$

For any $\boldsymbol{v} \in \mathbb{Z}^n$, $D_{\boldsymbol{v},\sigma}$ will be used as short-hand for $D_{\mathbb{Z}^n,\boldsymbol{v},\sigma}$.

For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we define the lattice $\Lambda^\perp(\mathbf{A}) \triangleq \{\boldsymbol{x} \in \mathbb{Z}^m : \mathbf{A}\cdot\boldsymbol{x} = 0 \bmod q\}$. We also define the coset $\Lambda^\perp_{\boldsymbol{u}}(\mathbf{A}) \triangleq \{\boldsymbol{x} \in \mathbb{Z}^m : \mathbf{A}\cdot\boldsymbol{x} = \boldsymbol{u} \bmod q\}$ for any $\boldsymbol{u} \in \mathbb{Z}_q^n$. The lemma below is usually instantiated with the gadget matrix $\mathbf{G} = \mathbf{I}_n \otimes (1, 2, \ldots, 2^{\log(q)-1})$, for which there is a known basis $\mathbf{S}$ of $\Lambda^\perp(\mathbf{G})$ satisfying $\|\mathbf{S}\|_{GS} \leq \sqrt{5}$ [49]. More generally, for $\mathbf{G} = \mathbf{I}_n \otimes (1, b, \ldots, b^{\log(q)-1})$, there is a known basis satisfying $\|\mathbf{S}\|_{GS} \leq \sqrt{b^2+1}$.

Lemma 12 (Lemma 5.3 [49]). *For distribution $\mathcal{D}$, $n(\lambda), \bar{m}(\lambda), q(\lambda)$ and sufficiently large $m = O(n\log(q))$, there exists an efficient randomized algorithm*

$\mathsf{TrapGen}_{\mathcal{D}}$ *that takes as input matrices* $\mathbf{A} \in \mathbb{Z}_q^{n \times \bar{m}}$, $\mathbf{G} \in \mathbb{Z}_q^{n \times (m-\bar{m})}$ *and outputs a matrix* $\mathbf{A_R} = [\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R} + \mathbf{G}]$ *and a trapdoor* $\mathbf{R} \sim \mathcal{D}^{\bar{m} \times (m-\bar{m})}$.

Given $\mathbf{R}$ *and a basis* $\mathbf{S}$ *of* $\Lambda^{\perp}(\mathbf{G})$*, one can compute a basis* $\mathbf{B}$ *of* $\Lambda^{\perp}(\mathbf{A_R})$ *such that* $\|\mathbf{B}\|_{GS} \leq (s_1(\mathbf{R}) + 1) \cdot \|\mathbf{S}\|_{GS}$.

In cases where the entries of the matrix $\mathbf{R} \in \mathbb{Z}^{\bar{m} \times (m-\bar{m})}$ are *not* independent and identically distributed, we may write $\mathsf{TrapGen}_{\mathcal{D}'}$ where $\mathcal{D}'$ has support over $\mathbb{Z}^{\bar{m} \times (m-\bar{m})}$. In this case, $\mathbf{R} \hookleftarrow \mathcal{D}'$.

The following theorem gives a method of sampling exactly from discrete Gaussians. We choose to use this particular result to avoid sampling to within some small statistical distance as in [29,49].

Lemma 13 (Lemma 2.3 [16]). *There exists an efficient algorithm that, given a basis* $\mathbf{B}$ *of an* n*-dimensional lattice* Λ*,* $\boldsymbol{c} \in \mathbb{R}^n$ *and a parameter* $r > \|\mathbf{B}\|_{GS} \cdot \sqrt{\ln(2n+4)/\pi}$ *outputs a sample distributed exactly as* $D_{\Lambda + \boldsymbol{c}, r}$.

We define $\mathsf{SampleD}(\mathbf{A_R}, \mathbf{R}, \boldsymbol{u}, r)$ for a full rank $\mathbf{A_R}$ using the following steps: (i) obtain the short basis $\mathbf{B}$ of $\Lambda^{\perp}(\mathbf{A_R})$ from Lemma 12, (ii) use $\mathbf{B}$ to sample from $D_{\Lambda^{\perp}(\mathbf{A_R}) + \boldsymbol{c}, r}$ using the algorithm implicit in Lemma 13 where $\boldsymbol{c}$ is an arbitrary vector such that $\mathbf{A_R} \cdot \boldsymbol{c} = \boldsymbol{u}$. If r is large enough, $\mathsf{SampleD}$ outputs a sample of $D_{\Lambda_{\boldsymbol{u}}^{\perp}(\mathbf{A_R}), r}$ by the above lemma.

A.3 Statistical Preliminaries

The leftover hash lemma will be useful to prove statistical closeness to uniformity. The lemma is given below, with the trivial modification that the hash function domain and ranges are vectors over $\mathbb{Z}_q$ rather than $\{0,1\}$. The min-entropy of a distribution $\mathcal{W}$ is defined as $\mathbf{H}_{\infty}(\mathcal{W}) = -\log(\max_w' \Pr_{w \hookleftarrow \mathcal{W}}[w = w'])$.

Lemma 14 ([34]). *Assume a universal family of hash functions* $\{H_x : \mathbb{Z}_q^m \to \mathbb{Z}_q^n\}_{x \in X}$ *(i.e., that* $\forall \boldsymbol{a} \neq \boldsymbol{b} \in \mathbb{Z}_q^m, \Pr_{x \hookleftarrow U(X)}[H_x(\boldsymbol{a}) = H_x(\boldsymbol{b})] = q^{-n}$*). Then for any random variable* $\mathcal{W}$ *with support over* $\mathbb{Z}_q^m$*, the statistical distance between the distributions of*

1. $(H_x(\boldsymbol{w}), x)$ *where* $x \hookleftarrow U(X)$ *and* $\boldsymbol{w} \hookleftarrow \mathcal{W}$*, and*
2. $(\boldsymbol{y}, x)$ *where* $x \hookleftarrow U(X)$ *and* $\boldsymbol{y} \hookleftarrow U(\mathbb{Z}_q^n)$

is at most $\frac{1}{2}\sqrt{2^{-\mathbf{H}_{\infty}(\mathcal{W})} \cdot q^n}$.

The hash family that left-multiplies by matrices of the form $[\mathbf{I}_n \mid \mathbf{A}] \in \mathbb{Z}_q^{n \times m}$ is a universal family of hash functions for prime q. This follows from [9, Lemma 3] by setting the ring to be the field $\mathbb{Z}_q$. As a result, we have the following corollary.

Lemma 15. *Let* q *be a prime and let* $\mathcal{W}$ *be a random variable with support over* $\mathbb{Z}_q^m$*. The distributions*

1. $([\mathbf{I}_n \mid \mathbf{A}], [\mathbf{I}_n \mid \mathbf{A}] \cdot \boldsymbol{w})$ *where* $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times (m-n)})$ *and* $\boldsymbol{w} \hookleftarrow \mathcal{W}$*, and*
2. $([\mathbf{I}_n \mid \mathbf{A}], \boldsymbol{y})$ *where* $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times (m-n)})$ *and* $\boldsymbol{y} \hookleftarrow U(\mathbb{Z}_q^n)$

are within a statistical distance $\frac{1}{2}\sqrt{2^{-\mathbf{H}_\infty(\mathcal{W})} \cdot q^n}$ *of each other*

The following lemma bounds the min-entropy of a discrete Gaussian.

Lemma 16 ([45]). *For* $\sigma > 3/\sqrt{2\pi}$, $\mathbf{H}_\infty(D_\sigma^m) \geq m$

The following lemma will be useful in the security proofs of Section B.

Lemma 17. *Let* $\mathcal{P}'$ *denote the distribution* $\mathcal{P}^{\bar{m}\times(m-\bar{m})}$ *with support restricted to matrices* $\mathbf{R}$ *such that* $s_1(\mathbf{R}) \leq \sqrt{\bar{m}} + \sqrt{m-\bar{m}}$. *Take any* $\mathbf{G} \in \mathbb{Z}_q^{n\times(m-\bar{m})}$ *and* $\sigma > 3/\sqrt{2\pi}$. *The statistical distance between the distributions*

$$\left\{ ([\mathbf{I}_n \mid \mathbf{A}],\ \mathbf{W}) \;\middle|\; \begin{array}{c} \mathbf{A}' \hookleftarrow U(\mathbb{Z}_q^{n\times(\bar{m}-n)}), \\ (\mathbf{A},\mathbf{R}) \leftarrow \mathsf{TrapGen}_{\mathcal{P}'}([\mathbf{I}_n|\mathbf{A}'],\mathbf{G}), \\ \mathbf{W} \hookleftarrow U(\mathbb{Z}_q^{n\times Q}) \end{array} \right\}$$

and

$$\left\{ ([\mathbf{I}_n \mid \mathbf{A}],\ [\mathbf{I}_n \mid \mathbf{A}]\cdot\mathbf{Y}) \mid \mathbf{A} \leftarrow U\left(\mathbb{Z}_q^{n\times(m-n)}\right),\ \mathbf{Y} \hookleftarrow D_\sigma^{m\times Q} \right\}$$

is at most $\frac{m-\bar{m}}{2} \cdot 2^{-(\bar{m}-n\log q)/2} + \frac{Q}{2} \cdot 2^{-(m-n\log q)/2} + 2^{-\Omega(m)}$.

Proof. Take, the first distribution in the lemma statement. Suppose we replace $\mathcal{P}'$ by $\mathcal{P}^{\bar{m}\times(m-\bar{m})}$. Then the statistical distance between the resulting distribution and the one from the lemma statement is at most the statistical distance between $\mathcal{P}'$ and $\mathcal{P}^{\bar{m}\times(m-\bar{m})}$. This follows from a data processing inequality. The aforementioned distance is given by the probability that $s_1(\mathbf{R}) > \sqrt{\bar{m}}+\sqrt{m-\bar{m}}$. This probability is at most $2^{-\Omega(m)}$ by Lemma 11.

Now note that $\mathbf{H}_\infty(\mathcal{P}^{\bar{m}}) = \bar{m}$. Lemma 15 then tells us that, for a uniform $\mathbf{A}' \in \mathbb{Z}_q^{n\times(\bar{m}-n)}$, the output distribution of the first matrix in $\mathsf{TrapGen}_{\mathcal{P}}([\mathbf{I}_n \mid \mathbf{A}'], \mathbf{G})$ is within statistical distance $(m-\bar{m}) \cdot 2^{-(\bar{m}-n\log q)/2-1}$ from $[\mathbf{I}_n \mid \mathbf{A}]$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times(m-n)}$ is uniform.

Lemma 16 tells us that an m-dimensional discrete Gaussian with parameter $\sigma > 3/\sqrt{2\pi}$ has min-entropy at least m. Together with Lemma 15, this means that the two distributions of the lemma statement (but with $\mathbf{A}$ chosen uniformly in both) are with statistical distance $Q \cdot 2^{-(m-n\log q)/2-1}$ from one another. Note the factor of Q that comes from the number of columns in $\mathbf{W}$. Therefore, a triangle inequality on statistical distances implies the result. □

We next recall a standard rejection sampling result.

Lemma 18 ([45]). *Let* h *be a distribution with support* V *whose elements have norm less than* T. *If* $\sigma > \alpha T$ *for any positive* α *and* $M = \exp(12/\alpha + 1/4\alpha^2)$, *the output distributions of the following two procedures are within statistical distance* $2^{-100}/M$, *and the probability that* $\mathcal{A}$ *outputs something in the first procedure is at least* $\frac{1-2^{-100}}{M}$:

Procedure 1:

1. $\boldsymbol{v} \hookleftarrow h$
2. $\boldsymbol{z} \hookleftarrow D_{\boldsymbol{v},\sigma}^m$

3. *Output* $(\boldsymbol{z}, \boldsymbol{v})$ *with probability* $\min\left(\frac{D_\sigma^m(\boldsymbol{z})}{M \cdot D_{\boldsymbol{v},\sigma}^m(\boldsymbol{z})}, 1\right)$

Procedure 2:

1. $\boldsymbol{v} \hookleftarrow h$
2. $\boldsymbol{z} \hookleftarrow D_\sigma^m(\boldsymbol{z})$
3. *Output* $(\boldsymbol{z}, \boldsymbol{v})$ *with probability* $1/M$.

Finally, we recall a useful lemma pertaining to the number of short solutions to an under-determined linear system.

Lemma 19 ([45]). *For any* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *where* $m > 64 + n \log q / \log(2d+1)$*, for randomly chosen* $\boldsymbol{s} \hookleftarrow U(\{-d, \ldots, 0, \ldots, d\}^m)$*, with probability* $1 - 2^{-100}$*, there exists a second vector* $\boldsymbol{s}' \in \{-d, \ldots, 0, \ldots, d\}^m$ *such that* $\mathbf{A} \cdot \boldsymbol{s} = \mathbf{A} \cdot \boldsymbol{s}'$.

B A Construction from Lyubashevsky Signatures

In this section, we describe an anamorphic triplet for Lyubashevsky's Fiat-Shamir-with-abort signatures [44]. The construction below is based on the variant from [45] which relies on the standard SIS assumption in unstructured lattices. It can be adapted to the ring variant of the same scheme too.

In the anamorphic triplet, the aGen algorithm generates the public matrix along with a Micciancio-Peikert trapdoor [49]. Since this trapdoor has no reason to exist at any time in a normal use of the scheme, we include it in the trapdoor key tk used by aSig rather than in the anamorphic signing key ask. In addition, the proof of dictator (strong) unforgeability requires to not reveal the trapdoor to the dictator.

$\mathsf{KGen}(1^\lambda)$: Given security parameter 1^λ,

1. Choose prime modulus $q = \mathsf{poly}(\lambda)$, bounds $d, \kappa = O(1)$, dimensions $k, n = O(\lambda), \bar{m} \geq n \log(q) + 2\lambda, m = \bar{m} + n \log(q)$ such that $n > \lambda$ and $\binom{k}{\kappa} > 2^\lambda$. Further, choose Gaussian parameter $\sigma = \widetilde{O}(\sqrt{m})$ such that $\sigma > (\sqrt{\bar{m}} + \sqrt{m - \bar{m}} + 1) \cdot \sqrt{5 \ln(2m+4)/\pi}$,[5] repetition rate[6] $M = O(1)$ and bound $\eta = O(1)$.
2. Sample $\bar{\mathbf{A}} \hookleftarrow U\left(\mathbb{Z}_q^{n \times (m-n)}\right)$ and set $\mathbf{A} = [\mathbf{I}_n \mid \bar{\mathbf{A}}] \in \mathbb{Z}_q^{n \times m}$.
3. Sample $\mathbf{S} \hookleftarrow U(\{-d, \ldots, 0, \ldots, d\}^{m \times k})$ and set $\mathbf{T} = \mathbf{A} \cdot \mathbf{S}$.
4. Choose a hash function $H_{\mathsf{FS}} : \mathbb{Z}_q^n \times \mathcal{M} \to \{\boldsymbol{v} \in \{-1, 0, 1\}^k : \|\boldsymbol{v}\|_1 \leq \kappa\}$.
5. Define message space $\mathcal{M}$.

Output $\mathsf{vk} = (\mathbf{A}, \mathbf{T})$ and $\mathsf{sk} = (\mathbf{A}, \mathbf{S})$.

$\mathsf{aGen}(1^\lambda)$: Given security parameter 1^λ,

1. Choose prime modulus $q = \mathsf{poly}(\lambda)$, bounds $d, \kappa = O(1)$, dimensions $k, n = O(\lambda), \bar{m} \geq n \log(q) + 2\lambda, m = \bar{m} + n \log(q)$ such that $n > \lambda$ and $\binom{k}{\kappa} > 2^\lambda$. Further, choose Gaussian parameter $\sigma = \widetilde{O}(\sqrt{m})$ such that $\sigma > (\sqrt{\bar{m}} + \sqrt{m - \bar{m}} + 1) \cdot \sqrt{5 \ln(2m+4)/\pi}$, repetition rate $M = O(1)$ and bound $\eta = O(1)$.

[5] This is chosen so that we may perform Gaussian pre-image sampling

[6] Intuitively, M is the parameter from Lemma 18

2. Sample $\mathbf{A}' \hookleftarrow U(\mathbb{Z}_q^{n\times(\bar{m}-n)})$ and set $\mathbf{G} = \mathbf{I}_n \otimes (1, 2, \ldots, 2^{\lfloor \log(q) \rfloor})$. Next, generate $(\mathbf{A}, \mathbf{R}) \hookleftarrow \mathsf{TrapGen}_{\mathcal{P}'}([\mathbf{I}_n \mid \mathbf{A}'], \mathbf{G})$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ is of the form $\mathbf{A} = [\mathbf{I}_n \mid \bar{\mathbf{A}}]$ and the underlying trapdoor $\mathbf{R}$ is sampled according to the distribution $\mathcal{P}'$ over $\{0, \pm 1\}^{\bar{m}\times(m-\bar{m})}$.[7]
3. Sample $\mathbf{S} \hookleftarrow U(\{-d, \ldots, 0, \ldots, d\}^{m\times k})$ and set $\mathbf{T} = \mathbf{A} \cdot \mathbf{S}$.
4. Choose a hash function $H_{\mathsf{FS}} : \mathbb{Z}_q^n \times \mathcal{M} \to \{\boldsymbol{v} \in \{-1, 0, 1\}^k : \|\boldsymbol{v}\|_1 \leq \kappa\}$.
5. Define message space $\mathcal{M}$, set $t = \lambda/\lfloor \log(q) \rfloor, \ell = n - 2t$ and define the anamorphic message space as $\hat{\mathcal{M}} = \mathbb{Z}_q^\ell$.
6. Choose keys $k_m \hookleftarrow U(\mathcal{K}_m)$, $K \hookleftarrow U(\mathcal{K})$ for pseudorandom functions

$$\mathsf{PRF} : \mathcal{K}_m \times \mathbb{Z}_q^t \to \mathbb{Z}_q^\ell,$$
$$\mathsf{PRF}' : \mathcal{K} \times (\mathbb{Z}_q^\ell \times \mathcal{M} \times \mathbb{Z}_q^t) \to \mathbb{Z}_q^t.$$

Output $\mathsf{avk} = (\mathbf{A}, \mathbf{T})$, $\mathsf{ask} = (\mathbf{A}, \mathbf{S})$, $\mathsf{dk} = (\mathsf{avk}, t, k_m, K)$ and $\mathsf{tk} = \mathbf{R}$.

Sig(sk, μ)**:** To sign $\mu \in \mathcal{M}$ with signing key $\mathsf{sk} = \mathbf{S}$,

1. Sample $\boldsymbol{y} \hookleftarrow D_\sigma^m$.
2. Compute $\boldsymbol{c} = H_{\mathsf{FS}}(\mathbf{A} \cdot \boldsymbol{y}, \mu) \in \{-1, 0, 1\}^k$.
3. Output $(\boldsymbol{z} \triangleq \boldsymbol{y} + \mathbf{S} \cdot \boldsymbol{c},\ \boldsymbol{c})$ with probability $\min\left(\frac{D_\sigma^m(\boldsymbol{z})}{M \cdot D_{\mathbf{S}\boldsymbol{c},\sigma}^m(\boldsymbol{z})}\right)$.

aSig$(\mathsf{ask}, \mathsf{tk}, \mathsf{dk}, \mu, \hat{\mu})$**:** Given $\mathsf{ask} = (\mathbf{A}, \mathbf{S})$, $\mathsf{tk} = \mathbf{R}$, $\mathsf{dk} = (\mathsf{avk}, t, k_m, K)$ and anamorphic message $\hat{\mu} \in \mathbb{Z}_q^\ell$,

1. Sample $\boldsymbol{w}_1 \hookleftarrow U(\mathbb{Z}_q^t)$ and compute $\boldsymbol{w}_2 = \hat{\mu} + \mathsf{PRF}_{k_m}(\boldsymbol{w}_1) \in \mathbb{Z}_q^\ell$ and $\boldsymbol{w}_3 = \mathsf{PRF}'_K(\boldsymbol{w}_2, \mu, \boldsymbol{w}_1)$.
2. Define $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3) \in \mathbb{Z}_q^n$ and sample $\boldsymbol{y} \hookleftarrow \mathsf{SampleD}(\mathbf{A}, \mathbf{R}, \boldsymbol{w}, \sigma)$.
3. Compute $\boldsymbol{c} = H_{\mathsf{FS}}(\mathbf{A} \cdot \boldsymbol{y}, \mu)$.
4. Output $(\boldsymbol{z} \triangleq \boldsymbol{y} + \mathbf{S} \cdot \boldsymbol{c},\ \boldsymbol{c})$ with probability $\min\left(\frac{D_\sigma^m(\boldsymbol{z})}{M \cdot D_{\mathbf{S}\boldsymbol{c},\sigma}^m(\boldsymbol{z})}\right)$.

Ver$(\mathsf{vk}, \mu, \mathsf{sig})$**:** Parse sig as $\mathsf{sig} = (\boldsymbol{z}, \boldsymbol{c})$ and return 1 if $c = H(\mathbf{A} \cdot \boldsymbol{z} - \mathbf{T} \cdot \boldsymbol{c}, \mu)$ and $\|\boldsymbol{z}\| \leq \eta\sigma\sqrt{m}$. Otherwise return 0.

aDec$(\mathsf{dk}, \mu, \mathsf{asig})$**:** Given $\mathsf{asig} = (\boldsymbol{z}, \boldsymbol{c})$ and $\mathsf{dk} = (\mathsf{avk}, t, k_m, K)$, do the following:

1. If $\mathsf{Ver}(\mathsf{avk}, \mu, \mathsf{asig})$ returns 0, output $\perp$. Otherwise, parse $\boldsymbol{w} \triangleq \mathbf{A}\boldsymbol{z} - \mathbf{T}\boldsymbol{c}$ as $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3) \in \mathbb{Z}_q^t \times \mathbb{Z}_q^\ell \times \mathbb{Z}_q^t$ and set $\hat{\mu} = \boldsymbol{w}_2 - \mathsf{PRF}_{k_m}(\boldsymbol{w}_1)$.
2. If $\boldsymbol{w}_3 \neq \mathsf{PRF}'_K(\boldsymbol{w}_2, \mu, \boldsymbol{w}_1)$, return $\perp$. Otherwise, return $\hat{\mu}$.

The signature length amounts to $\lambda + m \cdot \log \|\boldsymbol{z}\|_\infty$ bits. By [45, Lemma 4.4], $\|\boldsymbol{z}\|_\infty \leq \sigma\sqrt{2\lambda}$ with overwhelming probability. If we set $\sigma = \widetilde{O}(\sqrt{m})$, we have $\|\boldsymbol{z}\|_\infty \leq \widetilde{O}(\sqrt{\lambda m}) = \widetilde{O}(n\sqrt{\log q})$ and $\log \|\boldsymbol{z}\|_\infty = O(\log n)$. The bandwidth rate is then

$$\frac{n \log q - 2\lambda}{m \cdot \log \|\boldsymbol{z}\|_\infty + \lambda} \approx \frac{n \log q - 2\lambda}{(2n \log q + 2\lambda) \log \|\boldsymbol{z}\|_\infty + \lambda}.$$

which is roughly $\frac{1}{2 \log \|\boldsymbol{z}\|_\infty}$, where $\log \|\boldsymbol{z}\|_\infty = \Theta(\log n)$.

MAKING $\mathbf{A}$ NARROWER. The number of columns in $\mathbf{A}$ is $m = \bar{m} + n \log q$. In the above description, we need to set $m \geq 2\lambda + 2n \log q$, whereas a Lyubashevsky

[7] $\mathcal{P}'$ is the distribution as defined in the statement of Lemma 17

public key would have $m \approx 64 + n \log_{2d+1}(q)$ in the best case [45]. Therefore, there is a need to make $\mathbf{A}$ narrower when using normal parameters.

In the anamorphic scheme, the term $\bar{m}$ must be large enough to ensure that $[\mathbf{I}_n \mid \mathbf{A}'] \cdot \mathbf{R}$ is statistically uniform in aGen and $n \log q$ is the number of columns in the gadget matrix $\mathbf{G}$. One can reduce $\bar{m}$ by increasing the min-entropy of $\mathbf{R}$'s columns. Alternatively, we can set $\bar{m} = 2n$ and $\mathbf{R} \hookleftarrow D_{\sigma'}^{2n \times (m-2n)}$ and rely on an LWE assumption stating the pseudorandomness of $[\mathbf{I}_n \mid \mathbf{A}'] \cdot \mathbf{R} = \mathbf{R}_1 + \mathbf{A}' \cdot \mathbf{R}_2$. Additionally, we may reduce the width of $\mathbf{G}$ by using a gadget matrix with a larger decomposition base: e.g., by setting $\mathbf{G} = \mathbf{I}_n \otimes (1, B, \ldots, B^{\lceil \log_B(q) \rceil - 1})$, the width of $\mathbf{G}$ decreases to $\lceil \log_B(q) \rceil$.

These changes allow us to choose $m = 2n + n \log_B(q)$ at the cost of a larger lower bound on the standard deviation of the $\mathsf{SampleD}$ algorithm. Concretely, Lemma 12, Lemma 13 and [49, Lemma 2.9] tell us that the lower bound is

$$(s_1(\mathbf{R}) + 1) \cdot \sqrt{B^2 + 1} \cdot \sqrt{\ln(2m + 4)/\pi} \leq (\tfrac{\sigma' \cdot (\sqrt{2n} + \sqrt{k})}{\sqrt{2\pi}} + 1) \cdot \sqrt{B^2 + 1} \cdot \sqrt{\ln(2m + 4)/\pi}.$$

For example, with parameter set III from [45, Figure 2], we have $m = 3253$, $n = 512$, $k = 512$, $\sigma = 300926$ and $q \approx 2^{33}$. In order to achieve the correct number of columns in $\mathbf{A}$, we may take $B = 305$. It follows that the largest value of σ' allowing to use $\mathsf{SampleD}$ with parameter $\sigma = 300926$ is $\sigma' \approx 27$. Using the lattice estimator [4],[8] this value of σ' leads to an $\mathsf{HNF\text{-}LWE}$ problem that has around 70 bits of security which is too low for practical use. A similar calculation for parameter set II, suggests a base of $B = 4$, $\sigma' \approx 96$ and around 130 bits of LWE security. Using the bound $\|\boldsymbol{z}\|_\infty \leq 12\sigma$ to estimate signature size as in [45], this parameter set leads to a bandwidth rate of around 9%. Unfortunately, the most efficient parameter sets take $m = 2n$ and are out of reach for MP12 trapdoors. Nonetheless, one can *heuristically* use an unstructured version of the NTRU trapdoors suggested below to instantiate an anamorphic mode.

Ring variant. We now discuss the ring variant of Lyubashevsky's signature [45] over $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$ for power-of-two n. The matrix $\mathbf{A}$ is replaced by a vector $\boldsymbol{a} = [1, a'_1, \ldots, a'_{m/n-1}] \in \mathcal{R}_q^{1 \times m/n}$.

The instantiation leading to the smallest value m/n sets $m/n = 2$ and uses the ring-LWE problem to hide the signing key. In order to achieve such narrow trapdoored matrices, we can replace $\mathsf{TrapGen}$ and $\mathsf{SampleD}$ by the trapdoor generation and pre-image sampling from Falcon [56]. These trapdoors set $m = 2n$ by using an NTRU assumption, and the trapdoor basis allows to sample pre-images with parameter $\sigma_{\mathsf{pre}} \triangleq 1.17 \cdot \sqrt{q} \cdot \sqrt{\ln(2m + 4)/\pi}$ which is much smaller than σ for the most efficient parameter sets IV and V suggested in [45]. Further, for the parameters of Lyubashevsky signatures, the analogue of the vector $\boldsymbol{w}$ can be shown indistinguishable from uniform by a ring LWE assumption [46] with a large security margin. Then, under the assumption that Falcon verification keys

[8] https://github.com/malb/lattice-estimator/commit/6b25d9d34c57b8f6eb0b10799b411ac45de54199

are indistinguishable from uniform, all of the proofs still hold and we have an anamorphic mode for realistic parameters.

It should be mentioned that Falcon uses rejection sampling during its trapdoor generation algorithm, so the assumption that Falcon keys are indistinguishable from uniform is not implied by a standard decisional NTRU assumption. However, one may weaken this assumption by increasing the term $1.17 \cdot \sqrt{q}$ from Falcon in order to satisfy $\sigma_{\mathsf{pre}} \approx \sigma$. In practice though, decisional NTRU parameters are derived by considering known attacks on the search NTRU problem (see e.g. [15]). Again using the lattice estimator (which considers the possibility of attacks in the overstretched regime [24]), parameters IV and V from [45] suggest that the corresponding decisional NTRU problem has around 160 bits of security. In this case, the bandwidth rate reaches 58% and 64% for parameter sets IV and V respectively.

SECURITY. We now prove the dictator unforgeability, stealthiness, and recipient unforgeability of the scheme as described above.

On difficulty in the proof of dictator unforgeability is that a dictator may try to rerandomize a signature $(\boldsymbol{z}, \boldsymbol{c})$ to $(\boldsymbol{z}', \boldsymbol{c})$ by finding a short vector $\boldsymbol{z}'$ such that $\mathbf{A}\boldsymbol{z}' = \mathbf{A}\boldsymbol{z}$. The proof rules out this case by relying on an HNFSIS assumption on $\mathbf{A}$. However, this requires to first move to a game where the trapdoor of $\mathbf{A}$ is not used and the proof steps through a sequence of hybrids to this end.

Theorem 7. *Suppose $\sigma > \alpha\kappa d\sqrt{m}$ and $M = \exp(12/\alpha + 1/4\alpha^2)$ for some constant α. The scheme provides dictator unforgeability if PRF and PRF' are instantiated by random oracles and the $\mathsf{HNFSIS}_{q,n,m,2\eta\sigma\sqrt{m}}$ problem is hard. In particular, for any PPT adversary $\mathcal{A}$ making at most Q_H random oracle queries and Q_S signing queries, there is a PPT HNFSIS algorithm $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\mathsf{DUF\text{-}CASA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) \leq \mathbf{Adv}^{\mathsf{HNFSIS}_{q,n,m,2\eta\sigma\sqrt{m}}}_{\mathcal{B}}(\lambda) + \frac{Q_H + Q_S^2 + (m - \bar{m} + Q_S)/2 + (Q_D + 1)(Q_S + 2)}{2^{\lambda}} + \frac{1}{2^{\Omega(m)}}.$$

Proof. The proof considers a sequence of games with W_i denoting the event that the experiment Game_i outputs 1.

Game_0**:** This is the $\mathsf{DUF\text{-}CASA}$ experiment with $\mathsf{PRF}(k_m, \cdot)$ instantiated with $H_1(k_m, \cdot)$ and $\mathsf{PRF}'(K, \cdot, \cdot, \cdot)$ instantiated with $H_2(K, \cdot, \cdot, \cdot)$ where the random oracle responses for H_1 and H_2 are sampled lazily. For convenience, we consider $\mathcal{A}$'s output $(\mu^\star, \mathsf{asig}^\star)$ as an anamorphic decryption query and declare $\mathcal{A}$ successful at the first $\mathcal{O}_{\mathsf{aDec}}$-query for which aDec outputs $\hat{\mu} \neq \perp$ and asig is non-trivial. Specifically, W_0 becomes the event that $\mathcal{A}$ queries $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ on (μ, asig) such that $\hat{\mu} = \mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig}) \neq \perp$ and $(\mu, \hat{\mu}, \mathsf{asig}) \notin S$. Clearly, $\mathbf{Adv}^{\mathsf{DUF\text{-}CASA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) \leq \Pr[W_0]$.

Game_1**:** Define F_1 to be the event that $\mathcal{A}$ directly queries either H_1 or H_2 on an input of the form $H_1(k_m, \cdot)$ or $H_2(K, \cdot, \cdot, \cdot)$. Then Game_1 is the same as Game_0 apart from that when the event F_1 occurs, Game_1 aborts and outputs 0. Let Q_{H_1} and Q_{H_2} denote the number of random oracle queries

to H_1 and H_2 respectively. As $\mathcal{A}$'s view is independent of (k_m, K), we have $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1] \leq Q_{H_1}/2^{-\lambda} + Q_{H_2}/2^{-\lambda} \leq Q_H/2^{-\lambda}$.

Game_2**:** This is game is like Game_1 except that the $\mathcal{O}_{\mathsf{aSig}}$ oracle of Game_1 is replaced by $\mathcal{O}^2_{\mathsf{aSig}}$ which is defined as follows:

$\mathcal{O}^2_{\mathsf{aSig}}(\mu, \hat{\mu})$:

1. Sample $\boldsymbol{w} \hookleftarrow U(\mathbb{Z}_q^n)$ and $\boldsymbol{y} \hookleftarrow \mathsf{SampleD}(\mathbf{A}, \mathbf{R}, \boldsymbol{w}, \sigma)$.
2. Parse $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3) \in \mathbb{Z}_q^t \times \mathbb{Z}_q^\ell \times \mathbb{Z}_q^t$ and program
$$H_1(k_m, \boldsymbol{w}_1) = \boldsymbol{w}_2 - \hat{\mu}, \qquad H_2(K, \boldsymbol{w}_2, \mu, \boldsymbol{w}_1) = \boldsymbol{w}_3,$$
aborting and outputting 0 for the game if these inputs have already been programmed.
3. Compute $\boldsymbol{c} = H_{\mathsf{FS}}(\boldsymbol{w}, \mu)$.
4. Output $(\boldsymbol{z} \triangleq \boldsymbol{y} + \mathbf{S}\boldsymbol{c}, \boldsymbol{c})$ with probability $\min\left(\frac{D^m_\sigma(\boldsymbol{z})}{M \cdot D^m_{\mathbf{S}\boldsymbol{c},\sigma}(\boldsymbol{z})}, 1\right)$.

Assuming no programming error occurs, H_1 and H_2 are programmed with uniformly chosen evaluations, so Game_2 and Game_1 have an identical distribution. Game_2 and Game_1 only differ in the event F_2 that two calls to $\mathcal{O}^2_{\mathsf{aSig}}$ attempt to program the same point. If two such calls exist, they necessarily have the same $(\boldsymbol{w}_1, \mu, \hat{\mu})$ since $\boldsymbol{w}_2$ is determined by this tuple. By a birthday bound on $\boldsymbol{w}_1$, we obtain that $|\Pr[W_2] - \Pr[W_1]| \leq \Pr[F_2] \leq Q_S^2/2^\lambda$ since $\boldsymbol{w}_1$ is sampled uniformly from a space of size at least 2^λ.

Game_3**:** This game updates Game_2 in two ways. First, $\mathsf{aGen}(1^\lambda)$ is adapted to sample $\mathbf{A} = [\mathbf{I}_n \mid \bar{\mathbf{A}}]$ for *uniformly* chosen $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{n \times (m-n)})$ so that the trapdoor $\mathbf{R}$ no longer exists. We call this modified generation algorithm aGen'. The $\mathcal{O}^2_{\mathsf{aSig}}$ oracle is also replaced by the following $\mathcal{O}^3_{\mathsf{aSig}}$ oracle:

$\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$:

1. Sample $\boldsymbol{y} \hookleftarrow D^m_\sigma$ and set $\boldsymbol{w} = \mathbf{A} \cdot \boldsymbol{y}$.
2. Parse $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3) \in \mathbb{Z}_q^t \times \mathbb{Z}_q^\ell \times \mathbb{Z}_q^t$. Program $H_1(k_m, \boldsymbol{w}_1) = \boldsymbol{w}_2 - \hat{\mu}$, and $H_2(K, \boldsymbol{w}_2, \mu, \boldsymbol{w}_1) = \boldsymbol{w}_3$. Abort and output 0 for the game if these hash values have already been defined.
3. Compute $\boldsymbol{c} = H_{\mathsf{FS}}(\boldsymbol{w}, \mu)$.
4. Output $(\boldsymbol{z} \triangleq \boldsymbol{y} + \mathbf{S}\boldsymbol{c}, \boldsymbol{c})$ with probability $\min\left(\frac{D^m_\sigma(\boldsymbol{z})}{M \cdot D^m_{\mathbf{S}\boldsymbol{c},\sigma}(\boldsymbol{z})}, 1\right)$.

The only difference between Game_3 and Game_2 is the joint distribution of $(\mathbf{A}, \{(\boldsymbol{w}^i, \boldsymbol{y}^i)\})$ where $(\boldsymbol{w}^i, \boldsymbol{y}^i)$ are the pairs used in the signing queries. In Game_2, $\boldsymbol{w}^i$ is uniformly distributed and given a vector $\boldsymbol{w}^i \in \mathbb{Z}_q^n$, its pre-image $\boldsymbol{y}^i$ is distributed exactly as $D_{\Lambda + \boldsymbol{w}^i, \sigma}$ where $\Lambda = \{\boldsymbol{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \boldsymbol{x} = 0 \bmod q\}$. In Game_3, the tuple $(\boldsymbol{w}^i, \boldsymbol{y}^i)$ is obtained by sampling $\boldsymbol{y}^i \hookleftarrow D^m_\sigma$ and setting $\boldsymbol{w}^i = \mathbf{A} \cdot \boldsymbol{y}^i$. One can equivalently sample $\boldsymbol{y}'^i \hookleftarrow D^m_\sigma$, set $\boldsymbol{w}^i = \mathbf{A} \cdot \boldsymbol{y}'^i$ and resample $\boldsymbol{y}^i \hookleftarrow D_{\Lambda + \boldsymbol{w}^i, \sigma}$ without changing the distribution in Game_3.
Define $\mathbf{W} = [\boldsymbol{w}^1 \mid \cdots \mid \boldsymbol{w}^{Q_S}] \in \mathbb{Z}_q^{n \times Q_S}$ and $\mathbf{Y} = [\boldsymbol{y}^1 \mid \cdots \mid \boldsymbol{y}^{Q_S}] \in \mathbb{Z}_q^{m \times Q_S}$. Then, *conditioned on* some fixed value $(\mathbf{A}, \mathbf{W})$, $\mathbf{Y}$ follows precisely the same distribution in Game_2 and Game_3. It follows that the statistical distance between the distributions of $(\mathbf{A}, \mathbf{W}, \mathbf{Y})$ in Game_2 and Game_3 is equal to the

statistical distance between the distributions of $(\mathbf{A}, \mathbf{W})$ in Game_2 and Game_3. Lemma 17 implies that this distance is at most $\frac{m-\bar{m}}{2} \cdot 2^{-(\bar{m}-n\log q)/2} + \frac{Q_S}{2} \cdot 2^{-(m-n\log q)/2} + 2^{-\Omega(m)}$, implying that

$$|\Pr[W_3] - \Pr[W_2]| \leq (m - \bar{m} + Q_S) \cdot 2^{-\lambda-1} + 2^{-\Omega(m)}.$$

Game_4**:** Let F_4 be the event that $\mathcal{O}_{\mathsf{aDec}}$ is queried on $(\mu, (\boldsymbol{z}', \boldsymbol{c}'))$ such that: (i) $\mathsf{aDec}(\mathsf{dk}, \mu, (\boldsymbol{z}', \boldsymbol{c}')) = \hat{\mu} \neq \perp$; and (ii) There exists $(\mu, \hat{\mu}, (\boldsymbol{z}, \boldsymbol{c})) \in S$ where $\boldsymbol{c} = \boldsymbol{c}'$, $\mathbf{A} \cdot \boldsymbol{z}' - \mathbf{T} \cdot \boldsymbol{c}' = \mathbf{A} \cdot \boldsymbol{z} - \mathbf{T} \cdot \boldsymbol{c}$ and $\boldsymbol{z}' \neq \boldsymbol{z}$. If F_4 happens, Game_4 halts and outputs 0. Apart from this change, Game_4 is identical to Game_3. Clearly, $|\Pr[W_4] - \Pr[W_3]| \leq \Pr[F_4]$. We can build a PPT algorithm $\mathcal{B}$ that solves an $\mathsf{HNFSIS}_{q,n,m,2\eta\sigma\sqrt{m}}$ problem using $\mathcal{A}$ as a subroutine and with advantage $\Pr[F_4]$. Namely, $\mathcal{B}$ takes its HNFSIS instance $\mathbf{A}$ and uses it in step 2 of aGen' (introduced in Game_3). It then faithfully plays the challenger in Game_3. If F_4 occurs, $\mathcal{B}$ outputs $\boldsymbol{z} - \boldsymbol{z}'$. If $\mathcal{A}$ terminates without F_4 occurring, $\mathcal{B}$ outputs $\perp$. If F_4 occurs, we have $\mathbf{A} \cdot \boldsymbol{z} - \mathbf{T} \cdot \boldsymbol{c} = \mathbf{A} \cdot \boldsymbol{z} - \mathbf{T} \cdot \boldsymbol{c}'$ with $\boldsymbol{c} = \boldsymbol{c}'$ implying that $\mathbf{A} \cdot (\boldsymbol{z}' - \boldsymbol{z}) = 0 \bmod q$ and $\|\boldsymbol{z} - \boldsymbol{z}'\| \leq 2\eta\sigma\sqrt{m}$ as both signatures verify. Since $\mathcal{B}$ solves its HNFSIS problem with probability $\Pr[F_4]$, we have

$$|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{HNFSIS}_{q,n,m,2\eta\sigma\sqrt{m}}}(\lambda).$$

Game_5**:** We modify the $\mathcal{O}_{\mathsf{aDec}}$ oracle. At the first query $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ such that $\hat{\mu} = \mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig}) \neq \perp$ and no tuple $(\mu, \hat{\mu}, \mathsf{asig})$ exists in S, the challenger halts and outputs 0 as the result of the experiment. We call the latter event F_5. If F_4 occurs, then both Game_5 and Game_4 output 0. However, if F_5 occurs, but F_4 does not, the two games differ in the adversary's view. It follows that $|\Pr[W_5] - \Pr[W_4]| \leq \Pr[F_5 \wedge (\neg F_4)]$. Together with Lemma 20, this implies that $\Pr[W_5] - \Pr[W_4]| \leq (Q_D + 1) \cdot (Q_S + 2)/2^{-\lambda}$.

In Game_5, we observe that $\Pr[W_5] = 0$ since the output of the experiment is always 0, regardless of $\mathcal{A}$'s queries. This completes the proof. □

Lemma 20. $\Pr[F_5 \wedge (\neg F_4)] \leq (Q_D + 1)(Q_S + 2)/2^{-\lambda}$.

Proof. Let $(\mu, (\boldsymbol{z}', \boldsymbol{c}'))$ be a message-signature pair triggering $F_5 \wedge (\neg F_4)$ and define $\hat{\mu} = \mathsf{aDec}(\mathsf{dk}, \mu, (\boldsymbol{z}', \boldsymbol{c}')) \neq \perp$. There are three possible cases when F_4 does not occur: (i) There is no triple of the form $(\mu, \cdot, \cdot) \in S$; (ii) There is a triple $(\mu, \cdot, (\boldsymbol{c}, \boldsymbol{z})) \in S$ such that $\boldsymbol{c}' \neq \boldsymbol{c}$; (iii) There is a triple $(\mu, \cdot, (\boldsymbol{c}, \boldsymbol{z})) \in S$ such that $\boldsymbol{c}' = \boldsymbol{c}$ but $\boldsymbol{w}' \triangleq \mathbf{A} \cdot \boldsymbol{z}' - \mathbf{T} \cdot \boldsymbol{c}' \neq \mathbf{A} \cdot \boldsymbol{z} - \mathbf{T} \cdot \boldsymbol{c} \triangleq \boldsymbol{w}$ and $\boldsymbol{z}' \neq \boldsymbol{z}$ (note that we cannot have $\boldsymbol{w}' = \boldsymbol{w}$ in case (iii) since it would imply $(\mu, \hat{\mu}, (\boldsymbol{c}', \boldsymbol{z}')) \in S$, which would contradict $\neg F_4$).

We assume that $\mathcal{A}$ never queries H_2 to obtain PRF' values as this causes both Game_4 and Game_5 to output 0 (this condition was introduced in Game_1). Then, in case (i), we note that no signing query ever programmed an input of the form $(K, \cdot, \mu, \cdot)$ for H_2. Then, the probability that case (i) occurs is at most $\leq (Q_D + 1)/2^{-\lambda}$ assuming at most $Q_D + 1$ decryption queries (where the $+1$ term arises from checking the adversary's final signature output).

For case (ii), if $\boldsymbol{c} \neq \boldsymbol{c}'$ yet both signatures verify, we must have that $\boldsymbol{w} \triangleq \mathbf{A}\cdot\boldsymbol{z} - \mathbf{T}\cdot\boldsymbol{c} \neq \mathbf{A}\cdot\boldsymbol{z}' - \mathbf{T}\cdot\boldsymbol{c}' \triangleq \boldsymbol{w}'$ as $\boldsymbol{c} = H_{\mathsf{FS}}(\boldsymbol{w}, \mu)$ and $\boldsymbol{c}' = H_{\mathsf{FS}}(\boldsymbol{w}', \mu)$. However, since $\boldsymbol{w}$ is fully determined by $(\boldsymbol{w}_1, \boldsymbol{w}_2, \mu)$, it must be the case that $(\boldsymbol{w}_1', \boldsymbol{w}_2') \neq (\boldsymbol{w}_1, \boldsymbol{w}_2)$ which means that $H_2(K, \boldsymbol{w}_2, \mu, \boldsymbol{w}_1)$ was not set prior to the aDec query. Therefore, case (ii) also occurs with probability at most $(Q_D + 1)/2^{-\lambda}$.

As for case (iii), we have $\boldsymbol{c} = \boldsymbol{c}'$ corresponding to two distinct H_{FS} random oracle evaluations – one from the decryption oracle query and one from a signing oracle query. Over all $Q_D + 1$ decryption queries, this final case can only happen with probability at most $Q_S \cdot (Q_D + 1)/2^{-\lambda}$ since $\boldsymbol{c}$ is sampled from a set of size at least 2^{λ}. By a union bound, $\Pr[F_5 \wedge (\neg F_4)] \leq (Q_D + 1)(Q_S + 2)/2^{-\lambda}$. □

Theorem 8. *The scheme provides stealthiness if* PRF *and* PRF' *are both pseudorandom function families. For any PPT adversary* $\mathcal{A}$ *making at most* Q_S *signing queries, there exist distinguishers* $\mathcal{B}_0$ *and* $\mathcal{B}_1$ *against the pseudorandomness of* PRF' *and* PRF*, respectively, for which we have*

$$\mathbf{Adv}^{\mathsf{Stealthiness}}_{\mathcal{D},\Pi,\Sigma}(\lambda) \leq \mathbf{Adv}^{\mathcal{B}_0,\mathsf{PRF}'}_{Q_S}(\lambda) + \mathbf{Adv}^{\mathcal{B}_1,\mathsf{PRF}}_{Q_S}(\lambda) + \tfrac{Q_S^2}{2^{\lambda-1}} + \tfrac{1}{2^{\Omega(m)}} + \tfrac{m-\bar{m}+Q_S}{2^{\lambda+1}}.$$

Proof. The proof uses a sequence of games that begins with $\mathsf{AnamorphicG}_{\Sigma}(\lambda, \mathcal{D})$ and ends with $\mathsf{RealG}_{\Pi}(\lambda, \mathcal{D})$. For each i, we let W_i denote the event that adversary $\mathcal{A}$ outputs 1 in Game_i. Further, $\mathcal{O}^i_{\mathsf{aSig}}$ denotes the anamorphic signing oracle of Game_i.

Game_0**:** This is the $\mathsf{AnamorphicG}_{\Sigma}(\lambda, \mathcal{D})$ experiment. At each $\mathcal{O}^0_{\mathsf{aSig}}(\mathsf{ask}, \mu, \hat{\mu})$ query, the challenger faithfully computes $\sigma \leftarrow \mathsf{aSig}(\mathsf{ask}, \mathsf{dk}, \mu, \hat{\mu})$ and returns the result to $\mathcal{A}$. By definition, W_0 is the event that $\mathcal{A}$ outputs 1.

Game_1**:** This game is like Game_0 with the difference that the challenger halts and sets $\mathcal{A}$'s output as 0 in the event F_1 that $\mathcal{O}^1_{\mathsf{aSig}}$ chooses a vector $\boldsymbol{w}_1 \hookleftarrow U(\mathbb{Z}_q^t)$ that collides with the $\boldsymbol{w}_1$ chosen in a previous signing query. By a standard birthday argument, $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1] \leq Q_S^2/q^t \leq Q_S^2/2^{\lambda}$.

Game_2**:** We modify the anamorphic signing oracle $\mathcal{O}^1_{\mathsf{aSig}}(\mu, \hat{\mu})$ of Game_1 so that instead of computing $\boldsymbol{w}_3 = \mathsf{PRF}'_K(\boldsymbol{w}_2, \mu, \boldsymbol{w}_1)$, it computes $\boldsymbol{w}_3 = h$ for a freshly sampled $h \hookleftarrow U(\mathbb{Z}_q^t)$ for each query. If F_1 occurs, then Game_1 and Game_2 have the same output distribution. Otherwise, Game_2 simply replaces the distinct PRF' values in Game_1 with truly uniform values. A straightforward reduction builds a distinguisher $\mathcal{B}_0$ against PRF' that makes Q_S evaluation queries and for which $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}^{\mathcal{B}_0,\mathsf{PRF}'}_{Q_S}(\lambda)$.

Game_3**:** We replace the anamorphic signing oracle $\mathcal{O}^2_{\mathsf{aSig}}(\mu, \hat{\mu})$ of Game_2 by an oracle $\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$ that computes $\boldsymbol{w}_2 = \hat{\mu} + h_{\boldsymbol{w}_1}$ for a freshly sampled $h_{\boldsymbol{w}_1} \hookleftarrow U(\mathbb{Z}_q^{\ell})$ at each query. As before, if F_1 occurs, Game_2 and Game_3 have the same output distribution. When F_1 does not occur, we can rely on the pseudorandomness of PRF by the same argument as in the previous game transition. Concretely, we can build a distinguisher $\mathcal{B}_1$ against PRF such that $|\Pr[W_3] - \Pr[W_2]| \leq \mathbf{Adv}^{\mathcal{B}_1,\mathsf{PRF}}_{Q_S}(\lambda)$.

Game_4**:** We replace $\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$ by an oracle $\mathcal{O}^4_{\mathsf{aSig}}(\mu, \hat{\mu})$ that samples a fresh random $\boldsymbol{w}_2 \hookleftarrow U(\mathbb{Z}_q^\ell)$ at each query. As before, if F_1 occurs, then the output distributions of Game_3 and Game_4 is the same. Therefore we only need to analyze the transition when F_1 does not occur. In this case, the values of $\boldsymbol{w}_2$ in Game_3 are uniformly distributed and independent of the query $(\mu, \hat{\mu})$ as $\hat{\mu}$ is blinded by a fresh uniform sample at each query. The transition from Game_3 to Game_4 is thus only conceptual and $\Pr[W_4] = \Pr[W_3]$.

Game_5**:** We modify $\mathcal{O}^4_{\mathsf{aSig}}$ so that, at step 1, $\mathcal{O}^5_{\mathsf{aSig}}$ does not abort anymore in the event that $\boldsymbol{w}_1$ accidentally coincides with the $\boldsymbol{w}_1$ of an earlier signing query. By the same argument as in the transition from Game_1 to Game_2, we have $|\Pr[W_5] - \Pr[W_4]| \leq Q_S^2/2^\lambda$.

Game_6**:** Now, the challenger changes the key generation algorithm from $(\mathsf{ask}, \mathsf{avk}, \mathsf{dk}, \mathsf{tk}) \hookleftarrow \mathsf{aGen}(1^\lambda)$ to $(\mathsf{vk}, \mathsf{sk}) \hookleftarrow \mathsf{KGen}(1^\lambda)$. Note that tk no longer exists, so this game additionally uses $\mathcal{O}^6_{\mathsf{aSig}}$ that first samples $\boldsymbol{y} \hookleftarrow D_\sigma^m$ and sets $\boldsymbol{w}$ afterwards to $\boldsymbol{w} = \mathbf{A}\boldsymbol{y}$. In particular, $\mathcal{O}^6_{\mathsf{aSig}}$ does not use the trapdoor to sample any pre-image. Let $(\boldsymbol{w}^i, \boldsymbol{y}^i)$ denote the values used in the i-th oracle call and define $\mathbf{W} = [\boldsymbol{w}^1 \mid \ldots \mid \boldsymbol{w}^{Q_S}] \in \mathbb{Z}_q^{n \times Q_S}$, $\mathbf{Y} = [\boldsymbol{y}^1 \mid \ldots \mid \boldsymbol{y}^{Q_S}] \in \mathbb{Z}_q^{m \times Q_S}$. Game_5 and Game_6 differ only in the distribution of $(\mathbf{A}, \mathbf{W}, \mathbf{Y})$. However, in both games, $\mathbf{Y}$ is identically distributed given $(\mathbf{A}, \mathbf{W})$ (it is just a Gaussian matrix conditioned on $\mathbf{AY} = \mathbf{W}$). This means the statistical distance between $(\mathbf{A}, \mathbf{W}, \mathbf{Y})$ in Game_5 and Game_6 is equal to the statistical distance between $(\mathbf{A}, \mathbf{W})$. In Game_5, $\mathbf{W}$ is uniformly distributed. Therefore, Lemma 17 directly bounds the statistical distance in question and we obtain

$$\begin{aligned} |\Pr[W_6] - \Pr[W_5]| &\leq \tfrac{m-\bar{m}}{2} \cdot 2^{-(\bar{m} - n\log q)/2} + \tfrac{Q_S}{2} \cdot 2^{-(m - n\log q)/2} + 2^{-\Omega(m)} \\ &\leq (m - \bar{m} + Q_S) \cdot 2^{-\lambda - 1} + 2^{-\Omega(m)}. \end{aligned}$$

Note that $\mathcal{O}^6_{\mathsf{aSig}}$ is exactly the $\mathcal{O}_{\mathsf{Sig}}$ oracle. Moreover, Game_6 gives normal key pairs to $\mathcal{A}$. It follows that Game_6 is exactly the $\mathsf{RealG}_\Pi(\lambda, \mathcal{D})$ experiment. Applying the triangle inequality completes the proof. □

Theorem 9. *Suppose $\sigma > \alpha \cdot \kappa d\sqrt{m}$ and $M = \exp(12/\alpha + 1/4\alpha^2)$ for some constant α. The scheme provides recipient unforgeability if PRF, PRF' and H_{FS} are all modeled as random oracles and the $\mathsf{HNFSIS}_{q,n,m,2\eta\sigma\sqrt{m}}$ problem is hard. In particular, for any PPT adversary $\mathcal{A}$ making at most Q_H random oracle queries and Q_S signing queries define $\epsilon \triangleq \mathbf{Adv}^{\mathsf{RUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}(\lambda)$. Then there exists a PPT algorithm $\mathcal{B}$ solving the $\mathsf{HNFSIS}_{q,n,m,\beta}$ problem for $\beta = 2(\eta\sigma + d\kappa)\sqrt{m}$ with probability at least*

$$\epsilon' \cdot \left(\tfrac{\epsilon'}{Q_S + Q_H + 1} + \tfrac{1}{2^\lambda} \right) \cdot \tfrac{1 - 2^{-100}}{2}$$

where

$$\epsilon' \geq \epsilon - \left(\tfrac{3Q_S \cdot (Q_S + Q_H) + (m - \bar{m} + Q_S)/2}{2^\lambda} + \tfrac{1}{2^{\Omega(m)}} + \tfrac{2Q_S}{M \cdot 2^{100}} \right).$$

Proof. The proof considers a sequence of games with W_i denoting the event that Game_i returns 1. For each i, $\mathcal{O}^i_{\mathsf{Sig}}/\mathcal{O}^i_{\mathsf{aSig}}$ will denote the normal/anamorphic signing oracle in Game_i.

Game_0**:** This is $\mathcal{G}^{\mathsf{RUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}(\lambda)$ where $\mathsf{PRF}_k(x)$ is implemented as $H_1(k,x)$ and $\mathsf{PRF}'_k(x)$ is implemented as $H_2(k,x)$, where H_1 and H_2 are random oracles. These random oracles are simulated via lazy sampling.

Game_1**:** This is Game_0 except for a change to the normal signing oracle $\mathcal{O}_{\mathsf{Sig}}$. Instead of computing $\boldsymbol{c} = H_{\mathsf{FS}}(\mathbf{A}\cdot\boldsymbol{y},\mu)$, $\mathcal{O}^1_{\mathsf{Sig}}(\mu)$ samples a random $\boldsymbol{c}$ uniformly from $\{\{-1,0,1\}^k : \|\boldsymbol{v}\|_1 \le \kappa\}$ and programs $H_{\mathsf{FS}}(\mathbf{A}\cdot\boldsymbol{y},\mu)$ to equal $\boldsymbol{c}$. If the programming fails due to an attempt to program a point that is already defined, then Game_1 aborts and outputs 0. Let F_1 denote the event that, during an $\mathcal{O}_{\mathsf{Sig}}$-query, the value of $\boldsymbol{y}$ leads to $\boldsymbol{w} = \mathbf{A}\cdot\boldsymbol{y}$ where either: (i) $\boldsymbol{w}$ was encountered in a previous $\mathcal{O}_{\mathsf{Sig}}/\mathcal{O}_{\mathsf{aSig}}$-query; or (ii) $\boldsymbol{w}$ is a prefix of a previous direct query made to H_{FS} by $\mathcal{A}$. The distribution of Game_1 is then identical to Game_0 unless F_1 occurs. Note that, for any $\mathbf{A} = [\mathbf{I}_n \mid \bar{\mathbf{A}}]$ and $\boldsymbol{t} \in \mathbb{Z}_q^n$,

$$\begin{aligned}\Pr[\mathbf{A}\cdot\boldsymbol{y} = \boldsymbol{t} \mid \boldsymbol{y} \hookleftarrow D^m_\sigma] &= \Pr[\boldsymbol{y}_1 = \boldsymbol{t} - \bar{\mathbf{A}}\boldsymbol{y}_0 \mid \boldsymbol{y} \hookleftarrow D^m_\sigma] \\ &\le \max_{\boldsymbol{t}'\in\mathbb{Z}_q^n} \Pr[\boldsymbol{y}_1 = \boldsymbol{t}' \mid \boldsymbol{y}_1 \hookleftarrow D^n_\sigma] \le 2^{-n} \le 2^{-\lambda}\end{aligned}$$

where the second to last inequality follows from Lemma 16. This allows us to conclude that $|\Pr[W_1] - \Pr[W_0]| \le \Pr[F_1] \le Q_S\cdot(Q_S+Q_H)/2^{-\lambda}$.

Game_2**:** This is Game_1 with $\mathcal{O}^1_{\mathsf{Sig}}$ replaced by $\mathcal{O}^2_{\mathsf{Sig}}(\mu)$ described as follows:

1. Sample $\boldsymbol{c} \hookleftarrow \{\boldsymbol{v}\in\{-1,0,1\}^k : \|\boldsymbol{v}\| \le \kappa\}$ and $\boldsymbol{z} \hookleftarrow D^m_\sigma$.
2. With probability $1/M$: Output $(\boldsymbol{z},\boldsymbol{c})$ and program $H_{\mathsf{FS}}(\mathbf{A}\boldsymbol{z} - \mathbf{T}\boldsymbol{c},\mu)$ (if the programming fails, abort and set the game output to 0).

The difference between Game_2 and Game_1 can be seen through the joint output distribution of $(\boldsymbol{c},\boldsymbol{z})$. The statistical distance between $(\mathbf{S}\cdot\boldsymbol{c},\boldsymbol{z})$ in the two games is equal to the statistical distance of $(\boldsymbol{c},\boldsymbol{z})$. This is because given a value $\mathbf{S}\cdot\boldsymbol{c}$, $\boldsymbol{c}$ is just a uniformly chosen pre-image. Lemma 18 then shows that $|\Pr[W_2] - \Pr[W_1]| \le Q_S\cdot 2^{-100}/M$.

Game_3**:** This is Game_2 with the $\mathcal{O}^2_{\mathsf{aSig}}$ oracle replaced by the following oracle.

$\mathcal{O}^3_{\mathsf{aSig}}(\mu,\hat{\mu})$:

1. Sample $\boldsymbol{w} \hookleftarrow U(\mathbb{Z}_q^n)$. Parse $\boldsymbol{w} = (\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{w}_3) \in \mathbb{Z}_q^t \times \mathbb{Z}_q^\ell \times \mathbb{Z}_q^t$ and program
$$H_1(k_m,\boldsymbol{w}_1) = \boldsymbol{w}_2 - \hat{\mu}, \qquad H_2(K,\boldsymbol{w}_2,\mu,\boldsymbol{w}_1) = \boldsymbol{w}_3.$$
If either of these points were already defined, abort and output 0.
2. Sample $\boldsymbol{y} \hookleftarrow \mathsf{SampleD}(\mathbf{A},\mathbf{R},\boldsymbol{w},\sigma)$.
3. Compute $\boldsymbol{c} = H_{\mathsf{FS}}(\boldsymbol{w},\mu)$.
4. Output $(\boldsymbol{z} \triangleq \boldsymbol{y} + \mathbf{S}\boldsymbol{c},\boldsymbol{c})$ with probability $\min\left(\frac{D^m_\sigma(\boldsymbol{z})}{M\cdot D^m_{\mathbf{S}\boldsymbol{c},\sigma}(\boldsymbol{z})},1\right)$.

Let F_3 denote the event that, during an $\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$ query, the sampled $\boldsymbol{w}_1$ was either: (i) identical to the $\boldsymbol{w}_1$ sampled in a previous $\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$ query; or (ii) part of a previous direct H_1-query of the form $(k_m, \boldsymbol{w}_1)$ or a previous direct H_2-query of the form $(K, \boldsymbol{w}_2, \mu, \boldsymbol{w}_1)$. If F_3 does not happen, then the programming in step 1 never fails and Game_3 and Game_2 are identical from $\mathcal{A}$'s view. We obtain $|\Pr[W_3] - \Pr[W_2]| \leq \Pr[F_3] \leq Q_S \cdot (Q_S + Q_H)/2^{-\lambda}$ since $\boldsymbol{w}_1$ is sampled uniformly from a space of size greater than 2^λ.

Game_4: This game makes two changes. First, instead of running step 2 of aGen, Game_4 sets $\mathbf{A} = [\mathbf{I}_n \mid \bar{\mathbf{A}}]$ where $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{n \times (m-n)})$. As a result, the challenger sets $\mathsf{tk} = \perp$ as no trapdoor $\mathbf{R}$ is sampled. We call this modified aGen algorithm aGen'. The $\mathcal{O}^3_{\mathsf{aSig}}$ oracle is then changed into the following oracle.

$\mathcal{O}^4_{\mathsf{aSig}}(\mu, \hat{\mu})$:

1. Sample $\boldsymbol{y} \hookleftarrow D^m_\sigma$ and set $\boldsymbol{w} = \mathbf{A} \cdot \boldsymbol{y}$.
2. Compute $\boldsymbol{c} = H_{\mathsf{FS}}(\boldsymbol{w}, \mu)$.
3. Parse $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3) \in \mathbb{Z}_q^t \times \mathbb{Z}_q^\ell \times \mathbb{Z}_q^t$. Program $H_1(k_m, \boldsymbol{w}_1) = \boldsymbol{w}_2 - \hat{\mu}$, and $H_2(K, \boldsymbol{w}_2, \mu, \boldsymbol{w}_1) = \boldsymbol{w}_3$. Abort and output 0 if either of these hash values were previously defined.
4. Output $(\boldsymbol{z} \triangleq \boldsymbol{y} + \mathbf{S}\boldsymbol{c}, \boldsymbol{c})$ with probability $\min\left(\frac{D^m_\sigma(\boldsymbol{z})}{M \cdot D^m_{\mathbf{S}\boldsymbol{c},\sigma}(\boldsymbol{z})}, 1\right)$.

We now show that these changes are statistically indistinguishable. Define $\mathbf{W} = [\boldsymbol{w}^1 \mid \cdots \mid \boldsymbol{w}^{Q_S}] \in \mathbb{Z}_q^{n \times Q_S}$ and $\mathbf{Y} = [\boldsymbol{y}^1 \mid \cdots \mid \boldsymbol{y}^{Q_S}] \in \mathbb{Z}_q^{n \times Q_S}$ where $(\boldsymbol{y}^i, \boldsymbol{w}^i)$ is the pair $(\boldsymbol{y}, \boldsymbol{w})$ used when answering the i-th anamorphic signing oracle call. The statistical distance between $(\mathbf{A}, \mathbf{W}, \mathbf{Y})$ bounds the probability of distinguishing the two games. Further, given $(\mathbf{A}, \mathbf{W})$, $\mathbf{Y}$ is distributed exactly as a discrete Gaussian conditioned on $\mathbf{A} \cdot \mathbf{Y} = \mathbf{W}$ in both games. Therefore, the statistical distance between $(\mathbf{A}, \mathbf{W})$ in Game_3 and Game_4 is identical to the statistical distance between $(\mathbf{A}, \mathbf{W}, \mathbf{Y})$ in the two games. By Lemma 17, it follows that

$$|\Pr[W_4] - \Pr[W_3]| \leq (m - \bar{m} + Q_S) \cdot 2^{-\lambda - 1} + 2^{-\Omega(m)}.$$

Game_5: This game changes $\mathcal{O}^4_{\mathsf{aSig}}$ into to the following $\mathcal{O}^5_{\mathsf{aSig}}$ oracle.

$\mathcal{O}^5_{\mathsf{aSig}}(\mu, \hat{\mu})$:

1. Sample $\boldsymbol{y} \hookleftarrow D^m_\sigma$ and compute $\boldsymbol{w} = \mathbf{A} \cdot \boldsymbol{y}$.
2. Sample $\boldsymbol{c} \hookleftarrow U(\{\boldsymbol{v} \in \{-1, 0, 1\}^k : \|\boldsymbol{v}\|_1 \leq \kappa\})$.
3. Parse $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3) \in \mathbb{Z}_q^t \times \mathbb{Z}_q^\ell \times \mathbb{Z}_q^t$ and program

$$H_1(k_m, \boldsymbol{w}_1) = \boldsymbol{w}_2 - \hat{\mu}, \qquad H_2(K, \boldsymbol{w}_2, \mu, \boldsymbol{w}_1) = \boldsymbol{w}_3,$$

   aborting and outputting 0 if these hashes are already defined.
4. Output $(\boldsymbol{z} \triangleq \boldsymbol{y} + \mathbf{S}\boldsymbol{c}, \boldsymbol{c})$ with probability $\min\left(\frac{D^m_\sigma(\boldsymbol{z})}{M \cdot D^m_{\mathbf{S}\boldsymbol{c},\sigma}(\boldsymbol{z})}, 1\right)$ and program $H_{\mathsf{FS}}(\mathbf{A} \cdot \boldsymbol{z} - \mathbf{T} \cdot \boldsymbol{c}, \mu) = \boldsymbol{c}$, aborting and returning 0 in the game if this point is already defined.

Let F_5 denote the event that, during a $\mathcal{O}^4_{\mathsf{aSig}}(\mu, \hat{\mu})$-query, the sampled $\boldsymbol{w}$ was either: (i) equal to the $\boldsymbol{w}$ sampled in a previous $\mathcal{O}^4_{\mathsf{aSig}}(\mu, \hat{\mu})$ query; or (ii) equal to $\mathbf{A} \cdot \boldsymbol{z} - \mathbf{T} \cdot \boldsymbol{c}$ where $(\boldsymbol{z}, \boldsymbol{c})$ was the response to a previous $\mathcal{O}^4_{\mathsf{Sig}}(\mu)$ query; or (iii) part of a previous direct H_{FS}-query of the form $(\boldsymbol{w}, \mu)$ made by $\mathcal{A}$. If F_5 does not occur, then both Game_5 and Game_4 have the same output distribution. By a birthday bound and uniformity of $\boldsymbol{w}$, we have $|\Pr[W_5] - \Pr[W_4]| \leq \Pr[F_5] \leq Q_S \cdot (Q_S + Q_H)/q^m \leq Q_S \cdot (Q_S + Q_H)/2^{-\lambda}$.

Game_6**:** This game updates the $\mathcal{O}^5_{\mathsf{aSig}}$ oracle to $\mathcal{O}^6_{\mathsf{aSig}}$. On input $(\mu, \hat{\mu})$, $\mathcal{O}'_{\mathsf{aSig}}$ performs the following steps:

$\mathcal{O}^6_{\mathsf{aSig}}(\mu, \hat{\mu})$:

1. Sample $\boldsymbol{z} \hookleftarrow D^m_\sigma$, $\boldsymbol{c} \hookleftarrow U(\{\boldsymbol{v} \in \{-1, 0, 1\}^k : \|v\|_1 \leq \kappa\})$.
2. Set $\boldsymbol{w} = \mathbf{A}\boldsymbol{z} - \mathbf{T}\boldsymbol{c}$
3. Parse $\boldsymbol{w}$ as $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3) \in \mathbb{Z}^t_q \times \mathbb{Z}^\ell_q \times \mathbb{Z}^t_q$. Program

$$H_1(k_m, \boldsymbol{w}_1) = \boldsymbol{w}_2 - \hat{\mu}, \qquad H_2(K, \boldsymbol{w}_2, \mu, \boldsymbol{w}_1) = \boldsymbol{w}_3, \qquad H_{\mathsf{FS}}(\boldsymbol{w}, \mu) = \boldsymbol{c},$$

   and abort if these hashes were previously defined.
4. Output $(\boldsymbol{z}, \boldsymbol{c})$ with probability $1/M$ and program $H_{\mathsf{FS}}(\mathbf{A}\boldsymbol{z} - \mathbf{T}\boldsymbol{c}) = \boldsymbol{c}$, aborting and outputting 0 if this point is already defined.

Similarly to the transition between Game_1 and Game_2, Lemma 18 implies that $|\Pr[W_6] - \Pr[W_5]| \leq Q_S \cdot 2^{-100}/M$.

Game_6 uses a public key matrix of the form $\mathbf{A} = [\mathbf{I}_n \mid \bar{\mathbf{A}}]$, where $\bar{\mathbf{A}}$ is distributed as $U(\mathbb{Z}_q^{n \times (m-n)})$. Further, neither of the signing oracles use the signing key $\mathbf{S}$. Using these facts, Lemma 21 shows that we may use the Forking Lemma to translate forgeries in Game_6 into HNFSIS solutions. □

Lemma 21. *There exists a PPT algorithm $\mathcal{B}$ that solves the $\mathsf{HNFSIS}_{q,n,m,\beta}$ problem for $\beta = 2(\eta\sigma + d\kappa)\sqrt{m}$ with probability at least*

$$\Pr[W_6] \cdot \left(\frac{\Pr[W_6]}{Q_S + Q_H + 1} + \frac{1}{2^\lambda} \right) \cdot \frac{1 - 2^{-100}}{2}.$$

Proof. Algorithm $\mathcal{B}$ uses the Forking Lemma using an algorithm $\mathcal{A}'$ described as follows. The input of $\mathcal{A}'$ consists of $(x, (h_1, \ldots, h_Q); \rho)$, where $x = \mathbf{A} = [\mathbf{I}_n \mid \bar{\mathbf{A}}]$ is an $\mathsf{HNFSIS}_{q,n,m,\beta}$ instance, $(h_1, \ldots, h_Q)$ are used to answer random oracle queries in order, and ρ denotes the random coins used by $\mathcal{A}'$.

We set $Q = Q_S + Q_H + 1$ in order to have sufficiently many h_i's to answer $\mathcal{A}$'s signing queries/direct random oracle queries plus a potential query to verify $\mathcal{A}$'s purported forgery. More precisely, $(h_i)_{i \in [Q]}$ are used to answer H_{FS}-queries only and ρ is used to answer H_1 and H_2-queries. Algorithm $\mathcal{A}'$ begins by running aGen' (see Game_4), using its input x as the "$\mathbf{A}$"-matrix. After this, $\mathcal{A}'$ runs Game_6 with adversary $\mathcal{A}$ (this is possible as the signing oracle does not require a trapdoor for $\mathbf{A}$). $\mathcal{A}'$ outputs (J, σ) where $(J, \mathsf{sig}) = (\perp, \perp)$ if Game_6 outputs 0. Otherwise, $\mathcal{A}$ outputs a new valid message-signature pair $(\mu^\star, (\boldsymbol{z}^\star, \boldsymbol{c}^\star)) \notin S$. In this case, J is set to be the index of the random oracle query with input

$(\mathbf{A}\cdot\boldsymbol{z}^\star - \mathbf{T}\cdot\boldsymbol{c}^\star, \mu^\star)$, where $\boldsymbol{c}^\star = h_J$ and $\mathsf{sig} = (\mu^\star, \boldsymbol{z}^\star, \boldsymbol{c}^\star)$. The HNFSIS algorithm $\mathcal{B}$ first runs $F_{\mathcal{A}'}$. If it obtains an output $(1, \mathsf{sig}, \mathsf{sig}')$, we write $\mathsf{sig} = (\mu^\star, \boldsymbol{z}^\star, \boldsymbol{c}^\star)$, $\mathsf{sig}' = (\mu', \boldsymbol{z}', \boldsymbol{c})$. In this case, $\mathcal{B}$ outputs $\boldsymbol{z}^\star - \boldsymbol{z}' - \mathbf{S}\cdot(\boldsymbol{c}^\star - \boldsymbol{c}')$, where $\mathbf{S}$ is taken from the secret key sampled by $\mathcal{A}'$ via its random tape ρ. If it obtains an output $(0, \perp, \perp)$ from $F_{\mathcal{A}'}$, $\mathcal{B}$ aborts.

In Lemma 1, we have $acc = \Pr[W_6]$. Then, the probability that $\mathcal{B}$ does not abort and obtains a candidate HNFSIS solution is $\mathsf{frk} \geq acc\cdot(\frac{acc}{Q_S+Q_H} + \frac{1}{2^\lambda})$. What remains is to show that $\mathcal{B}$'s output is a valid $\mathsf{HNFSIS}_{q,n,m,\beta}$ solution. Firstly, as ρ and $(h_1, \ldots, h_{J-1})$ are fixed when $\mathcal{A}'$ is run twice, the input to the J-th random oracle query is the same on both runs. It follows that $\mathbf{A}\cdot\boldsymbol{z}^\star - \mathbf{A}\cdot\mathbf{S}\cdot\boldsymbol{c}^\star = \mathbf{A}\cdot\boldsymbol{z}' - \mathbf{A}\cdot\mathbf{S}\cdot\boldsymbol{c}'$ and thus

$$\mathbf{A}\cdot(\boldsymbol{z}^\star - \boldsymbol{z}' - \mathbf{S}\cdot(\boldsymbol{c}^\star - \boldsymbol{c}')) = \mathbf{0} \mod q.$$

It can be checked that $\|\boldsymbol{z}^\star - \boldsymbol{z}' - \mathbf{S}\cdot(\boldsymbol{c}^\star - \boldsymbol{c}')\| \leq (2\eta\sigma + 2d\kappa)\sqrt{m}$. To conclude the proof, we need to analyze the probability that $\boldsymbol{z}^\star - \boldsymbol{z}' - \mathbf{S}\cdot(\boldsymbol{c}^\star - \boldsymbol{c}') \neq \mathbf{0}$.

We know that $\boldsymbol{c}^* \neq \boldsymbol{c}'$ due to the definition of $F_{\mathcal{A}'}$. Let i be an index where $c_i^\star \neq c_i'$. By Lemma 19, with probability at least $1 - 2^{-100}$, there is at least one other matrix $\mathbf{S}'$ such that $\mathbf{S}$ and $\mathbf{S}'$ differ only in the i-th column and $\mathbf{A}\cdot\mathbf{S} = \mathbf{A}\cdot\mathbf{S}'$. Since $\mathcal{A}$'s view is independent of $\mathbf{S}$ and depends only on $\mathbf{T} = \mathbf{A}\cdot\mathbf{S}$, $\mathcal{A}$ cannot tell if $\mathbf{S}'$ or $\mathbf{S}$ was used as a secret key. Therefore, the probability that $\boldsymbol{z}^\star - \boldsymbol{z}' = \mathbf{S}\cdot(\boldsymbol{c}^\star - \boldsymbol{c}')$ is at most $(1 - 2^{-100})/2$, and $\mathcal{B}$ obtains a solution with probability at least $\mathsf{frk}\cdot(1 - 2^{-100})/2$. □

C Deferred Material for Anamorphic Okamoto-Schnorr Signatures

C.1 Anamorphic Triplet in Original Okamoto-Schnorr Signatures

This section gives a simplified anamorphic triplet for Okamoto-Schnorr signature with two generators. The construction is a special case of the one in Section 3.1 for $n = 1$. The anamorphic mode is simpler and only uses two pseudorandom functions. Its security analysis follows from that of the general case.

KGen(1^λ)**:** Given a security parameter 1^λ,

1. Choose a cyclic group $\mathbb{G}$ of prime order q such that $2^{\ell_q} < q < 2^{\ell_q} + 2^\lambda$ with $\ell_q > 2\lambda$. Choose generators $g, h \hookleftarrow U(\mathbb{G})$.
2. Choose $x_1, x_2 \hookleftarrow U(\mathbb{Z}_q)$ and compute $X = g^{x_1}\cdot h^{x_2}$.
3. Choose a hash function $H : \{0,1\}^* \to \mathbb{Z}_q$.

Output $\mathsf{vk} = \big(\mathbb{G}, g, h, X, H\big)$, $\mathsf{sk} = (x_1, x_2)$.

aGen(1^λ)**:** Given a security parameter 1^λ,

1. Choose a cyclic group $\mathbb{G}$ of prime order q such that $2^{\ell_q} < q < 2^{\ell_q} + 2^\lambda$ with $\ell_q > 2\lambda$. Choose $g \hookleftarrow U(\mathbb{G})$ and set $h = g^\omega$ where $\omega \hookleftarrow U(\mathbb{Z}_q)$.
2. Choose $x_1, x_2 \hookleftarrow U(\mathbb{Z}_q)$ and compute $X = g^{x_1}\cdot h^{x_2}$.
3. Choose a hash function $H : \{0,1\}^* \to \mathbb{Z}_q$.

4. Define an anamorphic message length $\ell_m \leq \lambda$ and define the anamorphic message space as $\hat{\mathcal{M}} = \{0,1\}^{\ell_m}$.
5. Choose keys $k_m \hookleftarrow U(\mathcal{K}_m)$, $K \hookleftarrow U(\mathcal{K})$ for pseudorandom functions

$$\mathsf{PRF} : \mathcal{K}_m \times \mathbb{G} \to \{0,1\}^{\ell_m},$$
$$\mathsf{PRF}' : \mathcal{K} \times \big(\{0,1\}^{\ell_m} \times \mathcal{M} \times \mathbb{G}\big) \to \{0,1\}^{t},$$

where $\mathcal{M}$ is the space of regular messages and $t = \ell_q - \ell_m$.

Output $\mathsf{avk} = \big(\mathbb{G}, g, h, X, H\big)$, $\mathsf{ask} = (x_1, x_2)$, $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m)$ and $\mathsf{tk} = \omega$.

Sig(sk, μ)**:** To sign $\mu \in \mathcal{M}$ using $\mathsf{sk} = (x_1, x_2)$,

1. Choose $r_1, r_2 \hookleftarrow U(\mathbb{Z}_q)$ and compute $R = g^{r_1} \cdot h^{r_2}$.
2. Compute $c = H(R, \mu) \in \mathbb{Z}_q$.
3. Compute $s_1 = r_1 + c \cdot x_1 \bmod q$ and $s_2 = r_2 + c \cdot x_2 \bmod q$.

Output $\mathsf{sig} = (c, s_1, s_2) \in \mathbb{Z}_q^3$.

aSig$(\mathsf{ask}, \mathsf{tk}, \mathsf{dk}, \mu, \hat{\mu})$**:** To sign $\mu \in \mathcal{M}$ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x_1, x_2)$, $\mathsf{tk} = \omega$, and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m)$, conduct the following steps.

1. Choose $r \hookleftarrow U(\mathbb{Z}_q)$ and compute $R = g^r$.
2. Compute $a = \hat{\mu} \oplus \mathsf{PRF}_{k_m}(R) \in \{0,1\}^{\ell_m}$ and then

$$s_2 = \mathsf{bin}^{-1}\big(a\big) + 2^{\ell_m} \cdot \mathsf{bin}^{-1}\big(\mathsf{PRF}'_K(a, \mu, R)\big) \mod q$$

3. Compute $c = H(R, \mu) \in \mathbb{Z}_q$.
4. Compute $s_1 = r + c \cdot (x_1 + \omega x_2) - \omega \cdot s_2 \bmod q$.

Output $\mathsf{sig} = (c, s_1, s_2) \in \mathbb{Z}_q^3$.

Ver$\big(\mathsf{vk}, \mu, \mathsf{sig}\big)$**:** Parse sig as $\mathsf{sig} = (c, s_1, s_2) \in \mathbb{Z}_q^3$. If $c = H\big(g^{s_1} \cdot h^{s_2} \cdot X^{-c}, \mu\big)$, return 1. Otherwise, return 0.

aDec$\big(\mathsf{dk}, \mu, \mathsf{asig}\big)$**:** Given μ, $\mathsf{sig} = (c, s_1, s_2) \in \mathbb{Z}_q^3$ and $\mathsf{dk} = (\mathsf{avk}, k_m, k, \ell_m)$,

1. Return $\perp$ if $\mathsf{Ver}(\mathsf{avk}, \mu, \mathsf{asig}) = 0$. Let $R = g^{s_1} \cdot h^{s_2} \cdot X^{-c}$.
2. Let $a_0 = s_2 \bmod 2^{\ell_m}$ and $a_1 = \lfloor s_2 / 2^{\ell_m} \rfloor \in \mathbb{Z}_{\lfloor q/2^{\ell_m} \rfloor}$.
3. Parse $a_0 \in \mathbb{Z}_{2^{\ell_m}}$ as $a = \mathsf{bin}(a_0) \in \{0,1\}^{\ell_m}$ and return $\perp$ if $\mathsf{bin}(a_1) \neq \mathsf{PRF}'_K(a, \mu, R)$. Otherwise, return $\hat{\mu} = a \oplus \mathsf{PRF}_{k_m}(R) \in \{0,1\}^{\ell_m}$.

C.2 Proof of Lemma 5

Proof. We first define an adversary $\mathcal{A}'$ that fits neatly into the forking algorithm $F_{\mathcal{A}'}$ (Figure 1) by adapting $\mathcal{A}$. In particular, $\mathcal{A}'$ inputs $((\mathsf{avk}, \mathsf{dk}), (h_1, \ldots, h_Q), \rho)$ where $Q = Q_S + Q_H + 1$. The pair $(\mathsf{avk}, \mathsf{dk})$ stands for $\mathcal{A}$'s input whereas ρ denotes the random coins of $\mathcal{A}$ as well as random coins used in $\mathcal{O}'_{\mathsf{Sig}}$ and $\mathcal{O}'_{\mathsf{aSig}}$ that are not random oracle evaluations. In particular, ρ contains the tuples $(s_1, \ldots, s_{n+1})$ used to answer $\mathcal{O}'_{\mathsf{Sig}}$-queries and the tuples $(s'_1, r_2, \ldots, r_{n+1})$ that are used to answer $\mathcal{O}'_{\mathsf{aSig}}$-queries. Elements $(h_1, \ldots, h_Q)$ are used to answer fresh (i.e., non-repeated) random oracle queries that may come directly from $\mathcal{A}$, from a signing

oracle query, or the verification of $\mathcal{A}$'s forgery. Algorithm $\mathcal{A}'$ then runs Game_2 as played by $\mathcal{A}$ until it outputs a candidate forgery $\sigma = (\mu^\star, \mathsf{sig}^\star)$. If Game_2 outputs 1, $\mathcal{A}'$ parses $\mathsf{sig}^\star = (c, s_1, \ldots, s_{n+1})$, setting J to be the index of the random oracle query with input $(X^{-c} \cdot \prod_{i=1}^{n+1} g_i^{s_i}, \mu)$ and $c = h_J$. Otherwise, $\mathcal{A}'$ sets $J = 0$ and $\sigma = \perp$. The final output of $\mathcal{A}'$ is defined to be (J, σ).

We now describe the main reduction $\mathcal{B}$ that will solve the discrete logarithm problem using $\mathcal{A}'$ and the forking algorithm. Algorithm $\mathcal{B}$ begins by using its discrete logarithm instance X_1 to run a modified version of aGen. In particular, it faithfully runs step 2 of aGen by computing $g_i = g^{\omega_i}$ for each $i \in [2, n+1]$ using randomly chosen $\omega_2, \ldots, \omega_{n+1} \hookleftarrow U(\mathbb{Z}_q)$. However, at step 3 of aGen, $\mathcal{B}$ samples $x_2, \ldots, x_{n+1} \hookleftarrow U(\mathbb{Z}_q)$ and sets $X = X_1 \cdot \prod_{i=2}^{n+1} g_i^{x_i}$. All other steps of aGen remain unchanged but, since $\mathcal{B}$ does not know $x_1 = \log_g(X_1)$, it only stores $\mathsf{ask} = (x_2, \ldots, x_{n+1})$. Then, $\mathcal{B}$ runs $F_{\mathcal{A}'}((\mathsf{avk}, \mathsf{dk}))$ as defined in Figure 1 and aborts if $F_{\mathcal{A}'}$ yields an output of the form $(0, \star, \star)$ ($F_{\mathcal{A}'}$ can run $\mathcal{A}'$ as a subroutine because the oracles of Game_2 do not require knowledge of $x_1 = \log_g(X_1)$). If $\mathcal{B}$ does not abort after running $F_{\mathcal{A}'}$, it parses $F_{\mathcal{A}'}$'s output as

$$(1, \sigma, \widetilde{\sigma}) = (1, (\mu, (c, s_1, \ldots, s_{n+1})), (\tilde{\mu}, \tilde{c}, (\tilde{s}_1, \ldots, \tilde{s}_{n+1}))) \tag{5}$$

and outputs $x_1 = \frac{s_1 - \tilde{s}_1 + \sum_{i=2}^{n+1} \omega_i \cdot (s_i - \tilde{s}_i)}{c - \tilde{c}} - \sum_{i=2}^{n+1} \omega_i \cdot x_i$.

We claim that $\mathcal{B}$ correctly outputs the discrete logarithm of X_1 if it does not abort. Note that the random coins of $\mathcal{A}'$ are fixed across both runs in $F_{\mathcal{A}'}$. These random coins are used to define $\mathcal{A}$'s random coins and random values used in $\mathcal{O}'_{\mathsf{Sig}}/\mathcal{O}'_{\mathsf{aSig}}$ that do not correspond to random oracle evaluations. Additionally, the first $J-1$ programmed random oracle points $h_1, \ldots, h_{J-1}$ are fixed across both runs of $\mathcal{A}'$ in $F_{\mathcal{A}'}$. Now, as Game_2 is an adaptation of the *weak* unforgeability game $\mathcal{G}^{\mathsf{wRUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}$, the J-th random oracle query contains a message μ that was not used an input to any $\mathcal{O}'_{\mathsf{Sig}}$ or $\mathcal{O}'_{\mathsf{aSig}}$-query. In other words, the J-th (and also J'-th) random oracle query does not correspond to a signing oracle query and was performed directly by $\mathcal{A}$. Further, by virtue of the fact that ρ and $h_1, \ldots, h_{J-1}$ is fixed across both runs of $\mathcal{A}'$, the input to the J-th random oracle query is identical across both runs of $\mathcal{A}'$. This implies that when $F_{\mathcal{A}'}$ outputs $(1, \sigma, \widetilde{\sigma})$ as parsed in Equation (5), $\mu = \tilde{\mu}$ and we have the "collision" $X^{-c} \cdot \prod_{i=1}^{n+1} g_i^{s_i} = X^{-\tilde{c}} \cdot \prod_{i=1}^{n+1} g_i^{\tilde{s}_i}$ as both candidate signatures verify correctly. We also have $c \neq \tilde{c}$ as $h_J \neq h_{J'}$ when $F_{\mathcal{A}'}$ outputs $(1, \sigma, \widetilde{\sigma})$. By re-arranging terms in the collision, we obtain

$$g^{\frac{s_1 - \tilde{s}_1}{c - \tilde{c}}} \cdot g^{\cdot \frac{\sum_{i=2}^{n+1} \omega_i \cdot (s_i - \tilde{s}_i)}{c - \tilde{c}}} \cdot g^{-\sum_{i=2}^{n+1} \omega_i \cdot x_i} = X_1$$

which shows that $\frac{s_1 - \tilde{s}_1 + \sum_{i=2}^{n+1} \omega_i \cdot (s_i - \tilde{s}_i)}{c - \tilde{c}} - \sum_{i=2}^{n+1} \omega_i x_i$ is indeed $\log_g(X_1)$.

To conclude the proof, we lower bound the probability that $F_{\mathcal{A}'}$ outputs $(1, \sigma, \widetilde{\sigma})$ using the Forking Lemma. We let acc denote the accepting probability of $\mathcal{A}'$ as defined in the statement of Lemma 1. Since $acc = \Pr[W_2]$, it follows that $\mathcal{B}$ solves its problem instance with probability $\geq \Pr[W_2] \cdot \left(\frac{\Pr[W_2]}{Q_S + Q_H + 1} - 1/q \right)$. $\square$

D Deferred Material for Anamorphic GQ Signatures

D.1 Proof of Theorem 6

Proof. The proof proceeds again with a sequence of games. In the first game, the challenger is running anamorphic experiment $\mathsf{AnamorphicG}_{\Sigma}(\lambda, \mathcal{D})$ with the adversary. In the last game, it will be running the real experiment $\mathsf{RealG}_{\Pi}(\lambda, \mathcal{D})$. We assume that the adversary can query the random oracles h and H in all games although the real experiment $\mathsf{RealG}_{\Pi}(\lambda, \mathcal{D})$ only involves one random oracle H_{FS}. The reason is that, in practice, h, H and H_{FS} would be obtained from a single random oracle using standard techniques to enforce domain separation.

In each game, we call W_i denote the event that the adversary $\mathcal{A}$ outputs 1 in Game_i. We also denote by $\mathcal{O}^i_{\mathsf{aSig}}$ the anamorphic signing oracle of Game_i.

Game_0**:** This game is identical to the $\mathsf{AnamorphicG}_{\Sigma}(\lambda, \mathcal{D})$. The challenger initially runs $((\mathsf{avk}, \mathsf{ask}), \mathsf{dk}) \leftarrow \mathsf{aGen}(1^\lambda)$. It hands $(\mathsf{avk}, \mathsf{ask} = (x, p, q))$ to the adversary $\mathcal{A}$ and keeps dk to itself. At each query $\mathcal{O}^0_{\mathsf{aSig}}(\mathsf{ask}, \mu, \hat{\mu})$, the challenger runs the real anamorphic signing algorithm and returns the result $\sigma \leftarrow \mathsf{aSig}(\mathsf{ask}, \mathsf{dk}, \mu, \hat{\mu})$ to $\mathcal{A}$. We note that each query $\mathcal{O}^0_{\mathsf{aSig}}(\mu, \hat{\mu})$ made by $\mathcal{A}$ triggers random oracle queries of the form $H(k_m, \cdot)$ and $H(K, \cdot, \mu, \cdot)$, where k_m and K are part of dk. We call W_0 the event that $\mathcal{A}$ outputs 1.

Game_1**:** This game is identical Game_0 except that the experiment halts and replaces $\mathcal{A}$'s output by a random bit if $\mathcal{A}$ makes a direct random oracle query (i.e., which is not triggered by $\mathcal{O}^1_{\mathsf{aSig}}$-query) of the form $h(k_m, \cdot)$, $H(K, \cdot, \cdot, \cdot)$, where $k_m, K \in \{0,1\}^\lambda$ are part of the double key dk chosen by the experiment at the outset of the game. Both games are identical until the event F_1 that a hash query of the form $h(k_m, \cdot)$ or $H(K, \cdot, \cdot, \cdot)$ is made. Since $k_m, K \sim U(\{0,1\}^\lambda)$ are initially chosen uniformly and independently of $\mathcal{A}$'s view, we have $\Pr[F_1] \leq Q_H/2^{\lambda-1}$ and thus $|\Pr[W_1] - \Pr[W_0]| \leq Q_H/2^{\lambda-1}$ if Q_H is the number of hash queries.

Game_2**:** This game is like Game_1 with the difference that the experiment halts and replaces $\mathcal{A}$'s output by a random bit in the event F_2, that the anamorphic signing oracle $\mathcal{O}^2_{\mathsf{aSig}}$ chooses $a' \hookleftarrow \mathbb{Z}_N$ such that $\tilde{a} = \lfloor a'/2^\ell \rfloor$ previously appeared in a signing query. We have $|\Pr[W_2] - \Pr[W_1]| \leq \Pr[F_2]$ and the same argument as in the proof of Lemma 7 shows that $\Pr[F_2] \leq Q_S^2/2^\lambda$.

Game_3**:** We modify the anamorphic signing oracle $\mathcal{O}^2_{\mathsf{aSig}}$ of Game_2 as follows.

$\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, h, H)$, let $\ell = \ell_m + \lambda$.

1. Choose $a \hookleftarrow U(\mathbb{Z}_N)$ and compute

$$\tilde{a} = \lfloor a/2^\ell \rfloor \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}, \qquad \bar{a} = a \bmod 2^\ell.$$

   If $\tilde{a}$ appeared in an earlier $\mathcal{O}^3_{\mathsf{aSig}}$-query, abort and output 0.
2. Let $(\bar{a}_0 || \bar{a}_1) = \mathsf{bin}(\bar{a}) \in \{0,1\}^\ell$, where $\bar{a}_0 \in \{0,1\}^{\ell_m}$, $\bar{a}_1 \in \{0,1\}^\lambda$.

3. Program $h(k_m, \tilde{a}) = \hat{\mu} \oplus \bar{a}_0 \in \{0,1\}^{\ell_m}$ and $H(K, \bar{a}_0, \mu, \tilde{a}) = \bar{a}_1$.
4. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_e$.
5. If $\gcd(a \cdot y^c \bmod N, N) \neq 1$, return $\perp$. Otherwise, compute the e-th root $z = (a \cdot y^c)^{1/e} \bmod N$ using p and q.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

By the same argument as in the proof of Lemma 9, Game_3 is statistically indistinguishable from Game_2 and that $|\Pr[W_3] - \Pr[W_2]| \leq Q_S/2^{\lambda+1}$.

Game_4**:** We modify the anamorphic signing oracle $\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$ of Game_3 so that it no longer programs the random oracle $h(k_m, \cdot)$ and $H(K, \cdot, \cdot, \cdot)$ at step 3. In detail, $\mathcal{O}^4_{\mathsf{aSig}}$ proceeds as follows:

$\mathcal{O}^4_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, h, H)$,

1. Choose $a \hookleftarrow U(\mathbb{Z}_N)$. If $\tilde{a} = \lfloor a/2^\ell \rfloor$ appeared in an earlier $\mathcal{O}^4_{\mathsf{aSig}}$-query, abort and output 0.
2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_e$.
3. If $\gcd(a \cdot y^c \bmod N, N) \neq 1$, return $\perp$. Otherwise, compute the e-th root $z = (a \cdot y^c)^{1/e} \bmod N$ using p and q.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

This change has no impact on $\mathcal{A}$'s view since, due to the change introduced in Game_1, the challenger halts and replaces $\mathcal{A}$'s output by a random bit as soon as $\mathcal{A}$ makes a hash query of the form $h(k_m, \cdot)$ or $H(K, \cdot, \cdot, \cdot)$. We have $\Pr[W_4] = \Pr[W_3]$ and we note that $\mathcal{O}^4_{\mathsf{aSig}}$ just ignores its second input $\hat{\mu}$.

Game_5**:** We modify $\mathcal{O}^4_{\mathsf{aSig}}$ in such a way that, at step 1, $\mathcal{O}^5_{\mathsf{aSig}}$ does not abort anymore in the event that $\tilde{a} = \lfloor a/2^\ell \rfloor$ collides with the $\tilde{a}$ of an earlier signing query. By the same argument as in the transition from Game_1 to Game_2, we have $|\Pr[W_5] - \Pr[W_4]| \leq Q_S^2/2^\lambda$.

Game_6**:** We change $\mathcal{O}^5_{\mathsf{aSig}}$ into $\mathcal{O}^6_{\mathsf{aSig}}$, which chooses $a \hookleftarrow U(\mathbb{Z}_N^*)$ instead of $a \hookleftarrow U(\mathbb{Z}_N)$ at step 1. Also, it does not check that $\gcd(a \cdot y^c \bmod N, N) = 1$ at step 1 since this always holds. More precisely, $\mathcal{O}^6_{\mathsf{aSig}}$ is specified as follows:

$\mathcal{O}^6_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, h, H)$,

1. Pick $a \hookleftarrow U(\mathbb{Z}_N^*)$.
2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_e$.
3. Compute the e-th root $z = (a \cdot y^c)^{1/e} \bmod N$ using p and q.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

The distribution of a is now uniform over $\mathbb{Z}_N^*$ instead of being uniform over $\mathbb{Z}_N$. Since the two distributions are within statistical distance $(p+q-1)/N < 2/\min(p,q) < 2^\lambda$, we have $|\Pr[W_6] - \Pr[W_5]| \leq Q_S/2^\lambda$.

Game_7**:** We now change $\mathcal{O}^6_{\mathsf{aSig}}$ into $\mathcal{O}^7_{\mathsf{aSig}}$, which is as follows

$\mathcal{O}^6_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, h, H)$,

1. Pick $r \hookleftarrow U(\mathbb{Z}_N^*)$ and compute $u = r^e \bmod N$.
2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_e$.
3. Compute $z = r \cdot x^c \bmod N$, where $x = y^{1/e} \bmod N$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

We note that the output distribution of $\mathcal{O}^7_{\mathsf{aSig}}$ is exactly the same as that of $\mathcal{O}^6_{\mathsf{aSig}}$ since $\gcd(e, \varphi(N)) = 1$ and the mapping $x \mapsto x^e \bmod N$ is one-to-one over $\mathbb{Z}_N^*$. We have $\Pr[W_7] = \Pr[W_6]$.

In Game_7, the anamorphic signing oracle has been replaced by a regular signing oracle that always returns normal signatures on μ. Since the distribution of $(\mathsf{vk}, \mathsf{sk})$ is also identical to that of $(\mathsf{avk}, \mathsf{ask})$, this concludes the proof. □

D.2 Comparison with RSA-PSS

Jaeger and Stracovsky [37] showed that recipient unforgeability can be obtained from signatures that are secure against chosen-message and randomness attacks (SUF-CRA) and where the signer's random coins publicly appear in the signature. As shown in [37, Section 5], RSA-PSS (just like Rabin-PSS) [12] satisfies both conditions. It thus provides RUF-CAMA security when the random salt of PSS is replaced by a symmetric encryption with pseudorandom ciphertexts. Moreover, by using an *authenticated* encryption scheme with pseudorandom ciphertexts, it is also possible to obtain dictator unforgeability.

However, the length of the pseudorandom authenticated encryption ciphertext is limited by the length of the random salt r in PSS. In principle, the random salt can contain up to $|N| - 2\lambda - 1$ bits if the two underlying random oracles have output length λ. In this case, if the symmetric AE scheme uses a λ-bit nonce and a λ-bit authentication tag (obtained from a PRF or a MAC), it can support anamorphic plaintexts as long as $\nu - 4\lambda = |N| - 4\lambda$ bits, which yields a bandwidth rate $1 - \frac{4\lambda}{\nu}$ that is also close to 1,[9] although it remains slightly below our rate $1 - \frac{3\lambda+1}{\nu+\lambda}$.

However, in practice, the random salt of PSS is much shorter (typically, 32 bytes[10] at the 128-bit security level). A result of Coron [23, Theorem 3] actually says that $|r| \geq 250$ suffices for $\lambda = 128$ and highlights the lack of motivation for using a longer salt.[11] Under the ϕ-hiding assumption, it is even possible to tightly prove security without a random salt at all [38].

Since the length of the random salt is a public parameter taken as input by the verification algorithm, the use of an unusually long salt might raise a dictator's suspicion (and somehow hint that the scheme is being used in anamorphic mode) as it would clearly depart from the PSS standard. This issue does not arise in our anamorphic triplet for GQ signatures where there is no public salt.

[9] It allows 0.8333 for $\lambda = 128$ and $\nu = 3072$, where we nearly obtain 0.88.

[10] The default salt length in public implementations if often 20 bytes [35].

[11] [23, Section 3.3] suggests that has PSS has a tight security proof when $|r| > \log Q_S$ and using a longer salt does not further improve its concrete security.

D.3 Comparison with the Swapping Technique

Micali and Reyzin [48] considered a method allowing to obtain tighter security reductions in certain signature schemes obtained from Σ-protocols. In particular, it can be applied to schemes like GQ, 2^t-root or Micali signatures where the factorization can be used as a trapdoor to compute the response without knowing the signer's internal state after its first Σ-protocol message.

In the case of GQ signatures, the swapping technique consists in first having the signer choose a random challenge $c \hookleftarrow U(\mathbb{Z}_e)$, which is hashed along with μ to obtain a first Σ-protocol message $a = H(c, \mu) \in \mathbb{Z}_N^*$. The factorization can then be used to compute $z \in \mathbb{Z}_N^*$ such that $a = H(c, \mu) = z^e \cdot y^c \bmod N$. A signature is a pair $(c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$ satisfying $a = H(c, \mu) = z^e \cdot y^c \bmod N$.

Their scheme can be proven secure under chosen message and randomness attacks by applying Coron's technique [22] for FDH signatures. In short, each random oracle query $H(c, \mu)$ is answered by programming $H(c, \mu) = z^e \cdot y^c \bmod N$ for a random $z \hookleftarrow U(\mathbb{Z}_N^*)$ with probability $Q_S/(Q_S + 1)$. With probability $1/(Q_S + 1)$, the reduction programs $H(c, \mu) = z'^e \cdot y^{c'} \bmod N$ with $z' \hookleftarrow U(\mathbb{Z}_N^*)$, $c' \hookleftarrow U(\mathbb{Z}_e)$. With non-negligible probability, the adversary only makes signing queries (μ, c) such that $H(c, \mu) = z^e \cdot y^c \bmod N$, which allows the reduction to return a signature for the randomness c. Then, if the adversary forges on a message $\mu^\star$ such that the reduction programmed $H(c^\star, \mu^\star) = z'^e \cdot y^{c'} \bmod N$ using a random $c' \hookleftarrow U(\mathbb{Z}_e)$, we have $c^\star \neq c'$ with probability $1 - 1/e$. Then, the reduction obtains a collision $z^{\star e} \cdot y^{c^\star} = z'^e \cdot y^{c'} \bmod N$, which allows computing $y^{1/e} \bmod N$. The analysis of [22] shows that, if the adversary has advantage ϵ, the reduction succeeds with probability $\geq \epsilon \cdot (1 - 1/e)/(\exp(1) \cdot (Q_S + 1))$.

While the swapped-GQ signature can be plugged into the RRep transformation to obtain a recipient unforgeable anamorphic signatures, the length of $c \in \mathbb{Z}_e$ is not larger than the salt of PSS since e is typically a λ-bit prime.

E Anamorphic Signatures from 2^t-Root Signatures

In this section, we give an anamorphic signature construction based on the 2^t-root signature scheme initially proposed by Ong and Schnorr [52] and further investigated in [47,61,59]. The standard 2^t-root signature scheme differs from GQ signatures in that the prime public exponent e is replaced by a power of 2.

We consider the variant described by Shoup [61], which relies on the hardness of factoring Blum integers. While the scheme has a security proof for general RSA moduli [59], its anamorphic triplet relies on the specific properties of Blum integers when the anamorphic message is encoded in the signature.

In the following, we denote by $\mathbb{QR}_N = \{x^2 \bmod N \mid x \in \mathbb{Z}_N^*\}$ the subgroup of quadratic residues in $\mathbb{Z}_N^*$. Recall that a Blum integer is an RSA modulus $N = pq$ where $p = q = 3 \bmod 4$ and that, for such a modulus, modular squaring is a permutation over $\mathbb{QR}_N$. For any $y \in \mathbb{QR}_N$, we denote by $y^{1/2} \bmod N$ the unique square root of y which is itself in $\mathbb{QR}_N$.

We also recall that, for a prime p and an element $a \in \mathbb{Z}_p^*$, the Legendre

symbol of a is defined as $(a \mid p) \triangleq a^{(p-1)/2} \bmod p$ and satisfies $(a \mid p) = 1$ if a is a square in $\mathbb{Z}_p^*$ and $(a \mid p) = -1$ otherwise.

When $N = pq$ is an RSA modulus, the Jacobi symbol of $a \in \mathbb{Z}_N^*$ is defined as $J_N(a) = (a \mid p) \cdot (a \mid q)$ and is efficiently computable from a and N. We denote by $\mathbb{J}_N \triangleq \{a \in \mathbb{Z}_N^* \mid J_N(a) = 1\}$ the multiplicative subgroup of elements with Jacobi symbol 1 in $\mathbb{Z}_N^*$. For Blum integers, we always have $-1 \in \mathbb{J}_N \setminus \mathbb{QR}_N$.

E.1 Description

The scheme uses the same parameters as in the scheme described in, e.g., [61].

As in the anamorphic GQ signature of Section 3.2, the anamorphic channel can be set up as an anamorphic extension of an already deployed key pair produced by the normal key generation procedure KGen. Indeed, the first three steps of KGen and aGen are identical, which allows creating double keys on-the-fly without any modification in the existing public key vk.

$\mathsf{KGen}(1^\lambda)$: Given a security parameter 1^λ,

1. Choose an RSA modulus $N = pq$ such that $2^{\nu(\lambda)-1} < N < 2^{\nu(\lambda)}$, where p, q are primes such that $p = q = 3 \bmod 4$, for some polynomial $\nu : \mathbb{N} \to \mathbb{N}$. Choose an integer $t \geq \lambda$.
2. Choose $x \hookleftarrow U(\mathbb{Z}_N^*)$ and compute $y = x^{2^t} \bmod N$.
3. Choose a hash function $H_{\mathsf{FS}} : \{0,1\}^* \to \mathbb{Z}_{2^t}$.

Output $\mathsf{vk} = \big(N, t, y, H_{\mathsf{FS}}\big)$, $\mathsf{sk} = (x, p, q)$.

$\mathsf{aGen}(1^\lambda)$: Given a security parameter 1^λ,

1. Choose an RSA modulus $N = pq$ such that $2^{\nu(\lambda)-1} < N < 2^{\nu(\lambda)}$, where p, q are primes such that $p = q = 3 \bmod 4$, for some polynomial $\nu : \mathbb{N} \to \mathbb{N}$. Choose an integer $t \geq \lambda$.
2. Choose $x \hookleftarrow U(\mathbb{Z}_N^*)$ and compute $y = x^{2^t} \bmod N$.
3. Choose a hash function $H_{\mathsf{FS}} : \{0,1\}^* \to \mathbb{Z}_{2^t}$ modeled as a random oracle.
4. Define an anamorphic message length $\ell_m < \nu - 2\lambda - 1$ and define the anamorphic message space as $\widehat{\mathcal{M}} = \{0,1\}^{\ell_m}$.
5. Choose random keys $k_m, K \hookleftarrow U(\{0,1\}^\lambda)$ for hash functions

$$h : \{0,1\}^\lambda \times \mathbb{Z}_{\lfloor N/2^\ell \rfloor} \to \{0,1\}^{\ell_m},$$
$$H : \{0,1\}^\lambda \times \{0,1\}^{\ell_m} \times \mathcal{M} \times \mathbb{Z}_{\lfloor N/2^\ell \rfloor} \to \{0,1\}^\lambda.$$

6. Choose a random $u \in \mathbb{Z}_N^* \setminus \mathbb{J}_N$ (i.e., such that $J_N(u) = -1$).

Output $\mathsf{avk} = \big(N, t, y, H_{\mathsf{FS}}\big)$, $\mathsf{ask} = (x, p, q)$, $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$ and $\mathsf{tk} = \emptyset$.

$\mathsf{Sig}(\mathsf{sk}, \mu)$: To sign $\mu \in \mathcal{M}$,

1. Choose $r \hookleftarrow U(\mathbb{Z}_N^*)$ and compute $a = r^{2^t} \bmod N$.
2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_{2^t}$.
3. Compute $z = r \cdot x^c \bmod N$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_e \times \mathbb{Z}_N^*$.

aSig$(\mathsf{ask}, \mathsf{tk}, \mathsf{dk}, \mu, \hat{\mu})$**:** To sign $\mu \in \mathcal{M}$ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$, let $\ell = \ell_m + \lambda$.

1. Encrypt $\hat{\mu}$ as an element of $\mathbb{QR}_N$. Namely,
   a. Choose $a' \hookleftarrow U(\mathbb{Z}_N)$ and compute $\tilde{a} = \lfloor a'/2^{\ell} \rfloor \in \mathbb{Z}_{\lfloor N/2^{\ell} \rfloor}$.
   b. Compute $\bar{a}_0 = \hat{\mu} \oplus h(k_m, \tilde{a}) \in \{0,1\}^{\ell_m}$ and then
   $$\begin{aligned} a &= \mathsf{bin}^{-1}\big(\bar{a}_0 \,||\, H(K, \bar{a}_0, \mu, \tilde{a}) \,||\, \mathsf{bin}(\tilde{a})\big) \\ &= \mathsf{bin}^{-1}(\bar{a}_0) + 2^{\ell_m} \cdot \mathsf{bin}^{-1}(H(K, \bar{a}_0, \mu, \tilde{a})) + 2^{\ell} \cdot \tilde{a} \end{aligned}$$
   If $\gcd(a, N) \neq 1$, return $\perp$.
   c. If $J_N(a) = -1$, set $a := u \cdot a \bmod N$.
   d. If $a \notin \mathbb{QR}_N$, set $a := -a \bmod N$.
2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_{2^t}$.
3. Using p and q, compute $\bar{z} = \big(a \cdot y^c\big)^{1/2^{t-1}} \bmod N$. Then, compute z as a random square root of $\bar{z}$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$.

Ver$\big(\mathsf{vk}, \mu, \mathsf{sig}\big)$**:** Parse sig as $(c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$. If $c = H_{\mathsf{FS}}(z^{2^t} \cdot y^{-c} \bmod N, \mu)$, return 1. Otherwise, return 0.

aDec$\big(\mathsf{dk}, \mu, \mathsf{asig}\big)$**:** Given μ, $(c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$, let $\ell = \ell_m + \lambda$. Then, do the following.

1. Return $\perp$ if $\mathsf{Ver}(\mathsf{avk}, \mu, \mathsf{asig}) = 0$. Otherwise, let $a = z^{2^t} \cdot y^{-c} \bmod N$.
2. For each pair $(i, j) \in \{0,1\}^2$, compute $a_{i,j} = a \cdot (-1)^i \cdot u^{-j} \bmod N$. Then, do the following.
   a. Let $\bar{a}_{i,j} = a_{i,j} \bmod 2^{\ell}$ and $\tilde{a}_{i,j} = \lfloor a_{i,j}/2^{\ell} \rfloor \in \mathbb{Z}_{\lfloor N/2^{\ell} \rfloor}$.
   b. Parse $\bar{a}_{i,j} \in \mathbb{Z}_{2^{\ell}}$ as a string $\mathsf{bin}(\bar{a}_{i,j}) = (\bar{a}_{i,j,0} || \bar{a}_{i,j,1}) \in \{0,1\}^{\ell}$ where $\bar{a}_{i,j,0} \in \{0,1\}^{\ell_m}$ and $\bar{a}_{i,j,1} \in \{0,1\}^{\lambda}$. If
   $$\bar{a}_{i,j,1} = H(K, \bar{a}_{i,j,0}, \mu, \tilde{a}_{i,j}), \tag{6}$$
   halt and return
   $$\hat{\mu} = \bar{a}_{i,j,0} \oplus h(k_m, \tilde{a}_{i,j}) \in \{0,1\}^{\ell_m}. \tag{7}$$
3. Return $\perp$.

Correctness. The anamorphic decryption algorithm can output an incorrect result with negligible probability when the test (6) is accidentally satisfied for two distinct pairs $(i, j) \in \{0,1\}^2$. If $H(K, \cdot, \cdot, \cdot)$ behaves as a random function, it occurs with probability smaller than $2^{-\lambda-2}$.

A Variant Based on Micali Signatures. It is possible to describe a similar anamorphic triplet for Micali's signature [47]. The main difference with [61] is

that [47] uses Williams integers[12] (which are Blum integers for which $p = 3 \bmod 8$ and $q = 7 \bmod 8$) and its secret key x is a 2^{t+1}-th root of $4^{-1} \bmod N$ while the challenge space is $\mathbb{Z}_{2^t}$. The first message of the prover is then $a = r^{2^{t+1}} \bmod N$ and the verifier checks that $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_{2^t}$, where $a = z^{2^{t+1}} \cdot 4^c \bmod N$.

A Variant Based on GQ2 Signatures. GQ2 signatures are a variant of 2^t-root signatures that appeared in the ISO/IEC 14888-2 standard [36]. As described in [32], the GQ2 is almost identical to 2^t-root signatures. The difference is that the public key contains a quadratic residue $G \in \mathbb{QR}_N$ instead of $y = x^{2^t} \bmod N$, while the secret key is a 2^{t+1}-th root of G. When GQ2 is instantiated with Blum integers, it can be endowed with an anamorphic triplet in the same way as in the above construction for 2^t-root signatures.

E.2 Security

We first prove recipient unforgeability since it is the main part of the security analysis that differs from GQ signatures.

Theorem 10. *In the random oracle model, the scheme is recipient weakly unforgeable assuming that factoring Blum integers is infeasible. Let a PPT adversary $\mathcal{A}$ making at most Q_H queries to H and Q_S signing oracle queries in $\mathcal{G}^{\mathsf{wRUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}$. If $\epsilon \triangleq \Pr[\mathcal{G}^{\mathsf{wRUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) = 1] - 2Q_S \cdot (Q_S + Q_H + 1)/2^{\lambda-1}$ and $N > 2^{\ell_m+2\lambda+1}$, then there exists a PPT algorithm $\mathcal{B}$ that can factor N with probability at least*

$$\epsilon \cdot \left(\frac{\epsilon}{Q_S + Q_H + 1} - 2^{-\lambda} \right).$$

Proof. The proof uses a sequence of games where, for each index i, we denote by W_i the event that the wRUF-CAMA experiment outputs 1 at the end of the Game_i. We also denote by $\mathcal{O}^i_{\mathsf{Sig}}$ and $\mathcal{O}^i_{\mathsf{aSig}}$ the normal and anamorphic signing oracles of Game_i.

Game_0**:** This is the wRUF-CAMA experiment, where the adversary $\mathcal{A}$ receives avk and dk at the onset of the game. The challenger $\mathcal{B}$ handles random oracle queries in the standard way and faithfully answers regular and anamorphic signing queries. Each query $\mathcal{O}^0_{\mathsf{Sig}}(\mu)$ or $\mathcal{O}^0_{\mathsf{aSig}}(\mu, \hat{\mu})$ made by $\mathcal{A}$ triggers random oracle queries of the form $H(k_m, \cdot)$ and $H(K, \cdot, \mu, \cdot)$, where k_m and K are part of the double key dk, which is initially given to $\mathcal{A}$. By definition, $\mathcal{A}$'s advantage is $\Pr[W_0]$.

Game_1**:** This game is identical to Game_0 with the difference that the challenger replaces the output of the experiment by 0 in the event F_1 that, at step 1.a, $\mathcal{O}^1_{\mathsf{aSig}}$ chooses $a' \sim U(\mathbb{Z}_N)$ such that $\tilde{a} = \lfloor a'/2^\ell \rfloor$ previously appeared in a random oracle query of the form $h(k_m, \tilde{a})$ or $H(K, \cdot, \cdot, \tilde{a})$ or in a $\mathcal{O}^1_{\mathsf{aSig}}$-query. We have $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1]$. By applying exactly the same argument as in the proof of Lemma 7, we obtain $\Pr[F_1] \leq Q_S \cdot (Q_H + Q_S)/2^\lambda$.

[12] It allows setting $u = 2$ in the double key dk since $J_N(2) = -1$ for Williams integers.

Game_2**:** We modify the anamorphic signing oracle $\mathcal{O}^1_{\mathsf{aSig}}(\mu, \hat{\mu})$ of Game_1 in the following way.

$\mathcal{O}^2_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$,

1. Encrypt $\hat{\mu}$ as follows.
   a. Choose $a \hookleftarrow U(\mathbb{Z}_N)$. If $\gcd(a, N) \neq 1$, return $\perp$. Otherwise, compute
   $$\tilde{a} = \lfloor a/2^\ell \rfloor \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}, \qquad \bar{a} = a \bmod 2^\ell,$$
   where $\ell = \ell_m + \lambda$. If $\tilde{a}$ appeared in an earlier hash or $\mathcal{O}^2_{\mathsf{aSig}}$-query, abort and output 0.
   b. Let $(\bar{a}_0 || \bar{a}_1) = \mathsf{bin}(\bar{a}) \in \{0,1\}^\ell$, where $\bar{a}_0 \in \{0,1\}^{\ell_m}$, $\bar{a}_1 \in \{0,1\}^\lambda$.
   c. Program $h(k_m, \tilde{a}) = \hat{\mu} \oplus \bar{a}_0 \in \{0,1\}^{\ell_m}$ and $H(K, \bar{a}_0, \mu, \tilde{a}) = \bar{a}_1$.
   d. If $J_N(a) = -1$, set $a := u \cdot a \bmod N$.
   e. If $a \notin \mathbb{QR}_N$, set $a := -a \bmod N$.
2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_{2^t}$.
3. Using p and q, compute $\bar{z} = \left(a \cdot y^c\right)^{1/2^{t-1}} \bmod N$. Then, compute z as a random square root of $\bar{z}$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}^*_N$.

Lemma 9 implies that Game_2 is statistically indistinguishable from Game_1 and that $|\Pr[W_2] - \Pr[W_1]| \leq Q_S/2^{\lambda+1}$.

Game_3**:** We modify the anamorphic signing oracle that now chooses $a \hookleftarrow U(\mathbb{Z}^*_N)$ instead of $a \hookleftarrow U(\mathbb{Z}_N)$ at step 1.a. Now, $\mathcal{O}^3_{\mathsf{aSig}}$ is specified as follows:

$\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$,

1. Encrypt $\hat{\mu}$ as follows.
   a. Choose $a \hookleftarrow U(\mathbb{Z}^*_N)$. Compute
   $$\tilde{a} = \lfloor a/2^\ell \rfloor \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}, \qquad \bar{a} = a \bmod 2^\ell,$$
   where $\ell = \ell_m + \lambda$. Abort and output 0 if $\tilde{a}$ appeared in an earlier hash or $\mathcal{O}^3_{\mathsf{aSig}}$-query.
   b. Let $(\bar{a}_0 || \bar{a}_1) = \mathsf{bin}(\bar{a}) \in \{0,1\}^\ell$, where $\bar{a}_0 \in \{0,1\}^{\ell_m}$, $\bar{a}_1 \in \{0,1\}^\lambda$.
   c. Program $h(k_m, \tilde{a}) = \hat{\mu} \oplus \bar{a}_0 \in \{0,1\}^{\ell_m}$ and $H(K, \bar{a}_0, \mu, \tilde{a}) = \bar{a}_1$.
   d. If $J_N(a) = -1$, set $a := u \cdot a \bmod N$.
   e. If $a \notin \mathbb{QR}_N$, set $a := -a \bmod N$.
2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_{2^t}$.
3. Using p and q, compute $\bar{z} = \left(a \cdot y^c\right)^{1/2^{t-1}} \bmod N$. Then, compute z as a random square root of $\bar{z}$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$.

The only change is the distribution of a which is now $U(\mathbb{Z}_N^*)$ instead of $U(\mathbb{Z}_N)$. However, the two distributions are at most $(p+q-1)/N < 2^\lambda$ apart in terms of statistical distance and we have $|\Pr[W_3] - \Pr[W_2]| \leq Q_S/2^\lambda$.

Game_4**:** This game is like Game_3 except that the challenger aborts and outputs 0 in the event F_4 that $H_{\mathsf{FS}}(a, \mu)$ was already defined during a direct query to H_{FS} made by $\mathcal{A}$, or at step 3 of $\mathcal{O}^4_{\mathsf{aSig}}$, or at step 2 of $\mathcal{O}^4_{\mathsf{Sig}}$. Apart from this modification, the signing oracles $\mathcal{O}^4_{\mathsf{Sig}}$ and $\mathcal{O}^4_{\mathsf{aSig}}$ are identical to $\mathcal{O}^3_{\mathsf{Sig}}$ and $\mathcal{O}^3_{\mathsf{aSig}}$, respectively. Since a is chosen uniformly in an exponentially large space $\mathbb{Z}_N^*$, we have $|\Pr[W_4] - \Pr[W_3]| \leq \Pr[F_4] \leq Q_S \cdot (Q_S + Q_H)/2^\lambda$.

Game_5**:** We modify again the anamorphic signing oracle $\mathcal{O}^4_{\mathsf{aSig}}(\mu, \hat{\mu})$, which is now replaced by the following oracle.

$\mathcal{O}^5_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu}$ using $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$,

1. Encrypt $\hat{\mu}$ as follows.
   a. Pick $z \hookleftarrow U(\mathbb{Z}_N^*)$, $c \hookleftarrow U(\mathbb{Z}_{2^t})$. Compute $a_z = z^{2^t} \cdot y^{-c} \bmod N$. Next, choose random bits $(i, j) \hookleftarrow U(\{0,1\} \times \{0,1\})$ and compute $a = a_z \cdot (-1)^i \cdot u^{-j} \bmod N$.
   b. Compute
   $$\tilde{a} = \lfloor a/2^\ell \rfloor \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}, \qquad \bar{a} = a \bmod 2^\ell,$$
   where $\ell = \ell_m + \lambda$. Abort and output 0 if $\tilde{a}$ appeared in an earlier hash or $\mathcal{O}^5_{\mathsf{aSig}}$-query.
   c. Let the string $(\bar{a}_0 || \bar{a}_1) = \mathsf{bin}(\bar{a}) \in \{0,1\}^\ell$, where $\bar{a}_0 \in \{0,1\}^{\ell_m}$ and $\bar{a}_1 \in \{0,1\}^\lambda$.
   d. Program $h(k_m, \tilde{a}) = \hat{\mu} \oplus \bar{a}_0 \in \{0,1\}^{\ell_m}$ and $H(K, \bar{a}_0, \mu, \tilde{a}) = \bar{a}_1$.
2. If $H_{\mathsf{FS}}(a_z, \mu)$ was already defined, abort and return 0. Otherwise, program $H_{\mathsf{FS}}(a_z, \mu) = c$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$.

We claim that the output distribution of $\mathcal{O}^5_{\mathsf{aSig}}$ is exactly the same as that of $\mathcal{O}^4_{\mathsf{aSig}}$. Indeed, $\mathcal{O}^4_{\mathsf{aSig}}$ chooses a uniformly in $\mathbb{QR}_N$ at step 1.e, which implies that $\bar{z} = (a \cdot y^c)^{1/2^{t-1}} \bmod N$ is itself uniform in $\mathbb{QR}_N$ since the mapping $x \mapsto x^{1/2^{t-1}} \bmod N$ is one-to-one over $\mathbb{QR}_N$. Accordingly, since $a_z \sim U(\mathbb{QR}_N)$ at step 1.a of $\mathcal{O}^5_{\mathsf{aSig}}$, the distribution of a is uniform over $\mathbb{Z}_N^*$ at the end of step 1.a, exactly as in step 1.a of $\mathcal{O}^4_{\mathsf{aSig}}$. We thus have $\Pr[W_5] = \Pr[W_4]$.

Game_6**:** We modify the normal signing oracle $\mathcal{O}^5_{\mathsf{aSig}}(\mu)$ as follows.

$\mathcal{O}^6_{\mathsf{Sig}}(\mu)$: To sign μ,

1. Choose $z \hookleftarrow U(\mathbb{Z}_N^*)$, $c \hookleftarrow U(\mathbb{Z}_{2^t})$ and compute $a = z^{2^t} \cdot y^{-c} \bmod N$.
2. If $H_{\mathsf{FS}}(a, \mu)$ was already defined, abort and return 0. Otherwise, program $H_{\mathsf{FS}}(a, \mu) = c$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$.

The output distribution of $\mathcal{O}^6_{\mathsf{Sig}}$ is the same as that of $\mathcal{O}^5_{\mathsf{Sig}}$ by the same argument as in the previous transition. We have $\Pr[W_6] = \Pr[W_5]$.

In Game_6, we note that $\mathcal{O}^6_{\mathsf{aSig}}$ does not use the factorization of N and $\mathcal{O}^6_{\mathsf{Sig}}$ does not use the 2^t-th root of the public key y.

Game_7**:** We modify the key generation. Instead of choosing $y = x^{2^t} \bmod N$ for a random $x \hookleftarrow \mathbb{Z}_N^*$, we compute $y = x^2 \bmod N$ with $x \hookleftarrow \mathbb{Z}_N^*$. Since squaring is a permutation over $\mathbb{QR}_N$, this change does not affect the distribution of the public key y. We have $\Pr[W_7] = \Pr[W_6]$.

In Game_7, Lemma 22 shows that $\Pr[W_7]$ is negligible if factoring is hard. □

The proof of Lemma 22 relies on the Forking Lemma and an argument that was already used in the security proof of the standard 2^t-root signatures. This argument (see, e.g., [40, Theorem 8.7]) says that, although the underlying Σ-protocol does not have special soundness in the usual sense, it is possible to compute a square root of y from two suitably related forgeries (i.e., two Σ-protocol transcripts with the same first message of the prover).

Lemma 22. *Let* $\Pr[W_7]$ *denote the probability of* Game_7 *outputting 1 for some PPT adversary* $\mathcal{A}$*. Further, suppose that* $\mathcal{A}$ *makes* Q_H *queries to* H *and* Q_S *queries to its signing oracles. Then, there exists a PPT algorithm* $\mathcal{B}$ *that can factor Blum integers with probability at least*

$$\Pr[W_7] \cdot \left(\frac{\Pr[W_7]}{Q_S + Q_H + 1} - 2^{-\lambda} \right).$$

Proof. We first define an algorithm $\mathcal{A}'$ that fits into the forking algorithm $F_{\mathcal{A}'}$. Algorithm $\mathcal{A}'$ serves as a wrapper for $\mathcal{A}$ and takes in $((\mathsf{avk}, \mathsf{dk}), (h_1, \ldots, h_Q), \rho)$, where $Q = Q_S + Q_H + 1$. The variable $(\mathsf{avk}, \mathsf{dk})$ denotes $\mathcal{A}$'s input whereas ρ contains the random coins of $\mathcal{A}$ as well as random coins used in $\mathcal{O}_{\mathsf{Sig}}$ and $\mathcal{O}_{\mathsf{aSig}}$ that are not random oracle evaluations. In particular, ρ contains the pairs (z, c) used to answer $\mathcal{O}^7_{\mathsf{Sig}}$ queries and (z, c, i, j) used to answer $\mathcal{O}^7_{\mathsf{aSig}}$ queries. Finally, $(h_1, \ldots, h_Q)$ are used to answer fresh random oracle queries that come directly from $\mathcal{A}$, via a signing query or via verifying $\mathcal{A}$'s purported forgery. Algorithm $\mathcal{A}'$ then runs Game_7 as played by $\mathcal{A}$ until it outputs a candidate forgery $\sigma = (\mu, \mathsf{sig})$. If Game_7 outputs 1, then $\mathcal{A}'$ parses $\mathsf{sig} = (c, z)$, setting J to be the index of the random oracle with input $(z^{2^t} \cdot y^{-c} \bmod N, \mu^\star)$ with $c = h_J$. Otherwise if Game_7 outputs 0, $\mathcal{A}'$ sets $J = 0$ and $\sigma = \perp$. The final output of $\mathcal{A}'$ is (J, σ).

We now describe the factoring algorithm $\mathcal{B}$ that will use $\mathcal{A}'$ as a subroutine. Algorithm $\mathcal{B}$ takes as input a Blum integer $N = pq$ which it uses to run a modified version of aGen. Namely, at step 2 of aGen, $\mathcal{B}$ samples $x \hookleftarrow U(\mathbb{Z}_N^*)$ and sets $y = x^2 \bmod N$ as the challenger does in Game_7. Steps 3-6 of aGen remain unchanged except that $\mathcal{B}$ samples $u \hookleftarrow \mathbb{Z}_N^* \setminus \mathbb{J}_N$ using rejection sampling, without

knowing the factorization of N.[13] Since $\mathcal{B}$ does not know the factorization of N, it only stores $\mathsf{ask} = (x, *, *)$, where $*$ is a placeholder for an unknown prime, and includes $(\mathsf{avk}, k_m, K, \ell_m, u)$ in dk. Then, $\mathcal{B}$ runs $F_{\mathcal{A}'}((\mathsf{avk}, \mathsf{dk}))$ as defined in Figure 1, aborting if $F_{\mathcal{A}'}$ yields an output of the form $(0, \star, \star)$. Importantly, $\mathcal{B}$ can run $\mathcal{A}'$ as a subroutine because the signing oracles in Game_7 do not require knowledge of $y^{1/2^t} \bmod N$ nor the factorization of N. If $\mathcal{B}$ does not abort after running $F_{\mathcal{A}'}$, it parses the output of $F_{\mathcal{A}'}$ as

$$(1, \sigma, \widetilde{\sigma}) = \big(1, (\mu, (c, z)), (\tilde{\mu}, (\tilde{c}, \tilde{z}))\big) \tag{8}$$

We now argue that, when $\mathcal{B}$ does not abort, it can factor N with probability $1/2$ from the two related forgeries (8). The random coins of $\mathcal{A}'$ are fixed across both runs in $F_{\mathcal{A}'}$. These random coins are used to define $\mathcal{A}$'s random coins and random values used in $\mathcal{O}^7_{\mathsf{Sig}}/\mathcal{O}^7_{\mathsf{aSig}}$ that do not correspond to random oracle evaluations. Moreover, the first $J-1$ programmed random oracle outputs $h_1, \ldots, h_{J-1}$ are fixed across both runs of $\mathcal{A}'$ in $F_{\mathcal{A}'}$. Then, as Game_7 is an adaptation of the *wea unforgeability* $\mathcal{G}^{\mathsf{wRUF\text{-}CAMA}}_{\mathcal{A},\Pi,\Sigma}$ game, the J-th random oracle query contains a message μ that is not the message of any $\mathcal{O}^7_{\mathsf{Sig}}$ or $\mathcal{O}^7_{\mathsf{aSig}}$-query, which means that the J-th (and also J'-th) random oracle query was not programmed by the signing oracle query and was the response to a direct hash query by $\mathcal{A}$. Further, since ρ and $h_1, \ldots, h_{J-1}$ are fixed in both runs of $\mathcal{A}'$, the input to the J-th random oracle query is identical across both runs of $\mathcal{A}'$. This implies that, when $F_{\mathcal{A}'}$ outputs $(1, \sigma, \widetilde{\sigma})$ as parsed in Equation (8), we have $\mu = \tilde{\mu}$ as well as

$$a = z^{2^t} \cdot y^{-c} \bmod N = \tilde{z}^{2^t} \cdot y^{-\tilde{c}} \bmod N \tag{9}$$

since both signatures verify. Moreover, we have $c \neq \tilde{c}$ as $h_J \neq h_{J'}$ when $F_{\mathcal{A}'}$ outputs $(1, \sigma, \widetilde{\sigma})$. Then, let $\Delta z = z \cdot \tilde{z}^{-1} \bmod N$ and $\Delta c = c - \tilde{c} \in [0, 2^t - 1]$ (we assume w.l.o.g. that $c > \tilde{c}$ when interpreted as integers). Let the largest integer $d \in [0, t-1]$ such that $2^d | \Delta c$ and write $\Delta c = 2^d \cdot c_o$ for an odd $c_o \in \mathbb{Z}$.

Then, (9) can be written

$$(\Delta z)^{2^t} = y^{2^d \cdot c_o} \bmod N$$

Since squaring is one-to-one over $\mathbb{QR}_N$ (and recalling that $y = x^2 \bmod N$ is a square), this implies

$$(\Delta z)^{2^{t-d}} = y^{c_o} \bmod N \tag{10}$$

Then, since c_o is odd, $\mathcal{B}$ can find $\alpha, \beta \in \mathbb{Z}$ such that $\alpha \cdot c_o + \beta \cdot 2^{t-d} = 1$ and (10) then implies

$$\big((\Delta z)^{\alpha} \cdot y^{\beta}\big)^{2^{t-d}} = y \bmod N$$

meaning that $\mathcal{B}$ obtains a square root $x_0 \triangleq \big((\Delta z)^{\alpha} \cdot y^{\beta}\big)^{2^{t-d-1}} \bmod N$ of y. At this point, we have $x_0 \neq \pm x \bmod N$ (which allows $\mathcal{B}$ to factor of N by computing

[13] The probability that $\mathcal{B}$ does not find $u \in \mathbb{Z}^*_N$ such that $J_N(u) = -1$ after λ attempts is at most $2^{-\lambda}$.

$\gcd(x_0 - x, N)$) with probability $1/2$ since the square root x of y chosen by $\mathcal{B}$ at the outset of the reduction is completely independent of the adversary's view.

To conclude the proof, we lower bound the probability that $F_{\mathcal{A}'}$ outputs $(1, \sigma, \widetilde{\sigma})$ using the forking lemma. We call acc the accepting probability of $\mathcal{A}'$ as defined in the statement of Lemma 1. By definition, $acc = \Pr[W_7]$ as $\mathcal{A}'$ outputs $J \neq 0$ if and only if W_7 occurs. Lemma 1 then implies that $F_{\mathcal{A}'}$ outputs $(1, \sigma, \widetilde{\sigma})$ with probability $\geq acc \cdot \left(\frac{acc}{Q_S + Q_H} - 2^{-\lambda}\right)$. It follows that $\mathcal{B}$ can factor N with probability at least

$$\Pr[W_7] \cdot \left(\frac{\Pr[W_7]}{Q_S + Q_H + 1} - 2^{-\lambda}\right).$$

□

Theorem 11. *The scheme is dictator weakly unforgeable in the* wDUF-CASA *sense. For any PPT adversary $\mathcal{A}$ making at most Q_H random oracle queries and Q_D anamorphic decryption queries, we have*

$$\mathbf{Adv}^{\mathsf{wDUF\text{-}CASA}}_{\mathcal{A},\Pi,\Sigma}(\lambda) \leq \frac{Q_H + 4 \cdot (Q_D + 1)}{2^{\lambda}}.$$

Proof. The proof uses a sequence of games. Again, we call W_i the event that the wDUF-CASA experiment outputs 1 at the end of the Game_i.

Game_0**:** This is the real wDUF-CASA experiment. The challenger handles random oracle queries using lazy sampling and honestly answers signing and anamorphic decryption queries. Each query $\mathcal{O}_{\mathsf{aSig}}(\mu, \hat{\mu})$ or $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ made by $\mathcal{A}$ leads the challenger to make random oracle queries of the form $H(k_m, \cdot)$ and $H(K, \cdot, \mu, \cdot)$, where k_m and K are part of the double key dk chosen by the challenger. By definition, $\mathcal{A}$'s advantage is $\Pr[W_0]$.

Game_1**:** This game is like Game_0 except that the experiment halts and outputs 0 if $\mathcal{A}$ makes a direct (i.e., not triggered by $\mathcal{O}_{\mathsf{aSig}}$-query) random oracle query of the form $H(K, \cdot, \cdot, \cdot)$, where $K \in \{0,1\}^{\lambda}$ is part of the double key dk. Both games proceed identically until the event F_1 that a hash query $H(K, \cdot, \cdot, \cdot)$ occurs. Since $K \sim U(\{0,1\}^{\lambda})$ is chosen independently of $\mathcal{A}$'s view, we have $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1] \leq Q_H/2^{\lambda}$.

Game_2**:** We modify the $\mathcal{O}_{\mathsf{aDec}}$ oracle. At the first query $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ such that $\hat{\mu} = \mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig}) \neq \perp$ no tuple of the form $(\mu, \hat{\mu}, \cdot)$ exists in S, the challenger aborts the experiment and outputs 0. If we call F_2 this event, Lemma 23 shows that $\Pr[F_2] \leq Q_D/2^{\lambda}$, which implies $|\Pr[W_2] - \Pr[W_1]| \leq 4 \cdot Q_D/2^{\lambda}$ since the two games are identical until F_2 occurs.

Game_3**:** This game is like Game_2 except that the challenger outputs 0 as the result of the experiment and ignores the adversary's output $(\hat{\mu}^{\star}, \mathsf{asig}^{\star})$. By the same argument as in Lemma 23, we have $|\Pr[W_3] - \Pr[W_2]| \leq 4/2^{\lambda}$ since $(\mu^{\star}, \mathsf{asig}^{\star})$ can be seen as the adversary's $(Q_D + 1)$-th query to $\mathcal{O}_{\mathsf{aDec}}$.

In Game_3, we have $\Pr[W_3] = 0$ since the challenger always outputs 0. □

Lemma 23. *We have* $|\Pr[W_2] - \Pr[W_1]| \leq \Pr[F_2] \leq 4 \cdot Q_D/2^{\lambda}$.

Proof. For each query $\mathcal{O}_{\mathsf{aDec}}(\mu, \mathsf{asig})$ where $\mathsf{asig} = (c, z)$, let $a = z^{2^t} \cdot y^{-c} \bmod N$ and $\hat{\mu} = \mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig})$. For each $(i, j) \in \{0,1\}^2$, let $a_{i,j} = a \cdot (-1)^i \cdot u^{-j} \bmod N$, $\bar{a}_{i,j} = a_{i,j} \bmod 2^\ell$ and $\tilde{a}_{i,j} = \lfloor a_{i,j}/2^\ell \rfloor \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}$. Let also

$$\mathsf{bin}(\bar{a}_{i,j}) = \left(\bar{a}_{i,j,0} \mid\mid \bar{a}_{i,j,1}\right) \in \{0,1\}^\ell. \tag{11}$$

Due to the change introduced in Game_1, the output distributions of the experiment in Game_2 can only differ from the one of Game_1 if F_2 occurs before F_1 (i.e., before $\mathcal{A}$ makes a direct random oracle query of the form $H(K, \cdot, \cdot, \cdot)$). Since F_2 implies $\mathsf{aDec}(\mathsf{dk}, \mu, \mathsf{asig}) \neq \perp$, it happens if and only if there exist $(i, j) \in \{0,1\}^2$ such that $\bar{a}_{i,j,1} = H(K, \bar{a}_{i,j,0}, \mu, \tilde{a}_{i,j})$ in (11).

However, since no tuple $(\mu, \hat{\mu}, \cdot)$ exists in S and no H-query of the form $H(K, \cdot, \cdot, \cdot)$ was made earlier, the hash value $H(K, \bar{a}_{i,j,0}, \mu, \tilde{a}_{i,j})$ cannot have been defined at all since $\hat{\mu}$ is uniquely determined by $\bar{a}_{i,j,0}$ and $\tilde{a}_{i,j}$ in (7). In an $\mathcal{O}_{\mathsf{aDec}}$-query, the probability to have $\bar{a}_{i,j,1} = H(K, \bar{a}_{i,j,0}, \mu, \tilde{a}_{i,j})$ is thus $1/2^\lambda$ for each $(i, j) \in \{0,1\}^2$. We obtain the stated upper bound on $\Pr[F_2]$ by a union bound over all pairs $(i, j) \in \{0,1\}^2$ and all anamorphic decryption queries. □

Theorem 12. *In the random oracle model, the scheme provides stealthiness. For any PPT adversary $\mathcal{A}$ making at most Q_H random oracle queries and Q_S signing queries, we have* $\mathbf{Adv}^{\mathsf{Stealthiness}}_{\mathcal{D},\Pi,\Sigma}(\lambda) \leq (Q_S^2 + 2Q_S + Q_H)/2^{\lambda-1}$.

Proof. The proof is similar to the proof of Theorem 6 and we only outline the changes.

Game_0**:** This game is identical to the $\mathsf{AnamorphicG}_\Sigma(\lambda, \mathcal{D})$. We call W_0 the event that $\mathcal{A}$ outputs 1.

Game_1**:** This game is identical Game_0 except that the experiment aborts and outputs a random bit if $\mathcal{A}$ makes a direct random oracle query of the form $h(k_m, \cdot)$, $H(K, \cdot, \cdot, \cdot)$, where $k_m, K \in \{0,1\}^\lambda$ are part of dk. Both games are identical until the event F_1 that a hash query $h(k_m, \cdot)$ or $H(K, \cdot, \cdot, \cdot)$ occurs. Since $k_m, K \sim U(\{0,1\}^\lambda)$ are chosen uniformly and independently of $\mathcal{A}$'s view, we have $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1] \leq Q_H/2^{\lambda-1}$.

Game_2**:** This game is like Game_1 except that the challenger halts and outputs 0 in the event that $\mathcal{O}^2_{\mathsf{aSig}}$ chooses $a' \leftarrow\!\!\!\!\!\leftarrow \mathbb{Z}_N$ such that $\tilde{a} = \lfloor a'/2^\ell \rfloor$ appeared in an earlier anamorphic signing query. We have $|\Pr[W_2] - \Pr[W_1]| \leq Q_S^2/2^\lambda$ by the same argument as in the proof of Lemma 7.

Game_3**:** We modify the anamorphic signing oracle $\mathcal{O}^2_{\mathsf{aSig}}(\mu, \hat{\mu})$ of Game_2 in the following way.

$\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$,

1. Encrypt $\hat{\mu}$ as follows.

a. Choose $a \hookleftarrow U(\mathbb{Z}_N)$. If $\gcd(a, N) \neq 1$, return $\perp$. Otherwise, compute

$$\tilde{a} = \lfloor a/2^\ell \rfloor \in \mathbb{Z}_{\lfloor N/2^\ell \rfloor}, \qquad \bar{a} = a \bmod 2^\ell,$$

where $\ell = \ell_m + \lambda$. If $\tilde{a}$ appeared in an earlier $\mathcal{O}^3_{\mathsf{aSig}}$-query, abort and output 0.

b. Let $(\bar{a}_0 || \bar{a}_1) = \mathsf{bin}(\bar{a}) \in \{0,1\}^\ell$, where $\bar{a}_0 \in \{0,1\}^{\ell_m}$, $\bar{a}_1 \in \{0,1\}^\lambda$.

c. Program $h(k_m, \tilde{a}) = \hat{\mu} \oplus \bar{a}_0 \in \{0,1\}^{\ell_m}$ and $H(K, \bar{a}_0, \mu, \tilde{a}) = \bar{a}_1$.

d. If $J_N(a) = -1$, set $a := u \cdot a \bmod N$.

e. If $a \notin \mathbb{QR}_N$, set $a := -a \bmod N$.

2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_{2^t}$.

3. Using p and q, compute $\bar{z} = \left(a \cdot y^c\right)^{1/2^{t-1}} \bmod N$. Then, compute z as a random square root of $\bar{z}$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}^*_N$.

By the same argument as in the proof of Lemma 9, Game_3 is statistically indistinguishable from Game_2 and that $|\Pr[W_3] - \Pr[W_2]| \leq Q_S/2^{\lambda+1}$.

Game_4: We modify the anamorphic signing oracle $\mathcal{O}^3_{\mathsf{aSig}}(\mu, \hat{\mu})$ of Game_3 in such a way that it does no longer program $h(k_m, \cdot)$ and $H(K, \cdot, \cdot, \cdot)$ at step 1. Namely, $\mathcal{O}^4_{\mathsf{aSig}}$ proceeds as follows:

$\mathcal{O}^4_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$,

1. Generate a as follows.

a. Choose $a \hookleftarrow U(\mathbb{Z}_N)$. If $\gcd(a, N) \neq 1$, return $\perp$.

b. If $J_N(a) = -1$, set $a := u \cdot a \bmod N$.

c. If $a \notin \mathbb{QR}_N$, set $a := -a \bmod N$.

2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_{2^t}$.

3. Using p and q, compute $\bar{z} = \left(a \cdot y^c\right)^{1/2^{t-1}} \bmod N$. Then, compute z as a random square root of $\bar{z}$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}^*_N$.

This change does not impact $\mathcal{A}$'s view since, due to the change introduced in Game_1, the challenger halts and outputs a random bit as soon as $\mathcal{A}$ makes a query of the form $h(k_m, \cdot)$ or $H(K, \cdot, \cdot, \cdot)$. We have $\Pr[W_4] = \Pr[W_3]$ and we note that $\mathcal{O}^4_{\mathsf{aSig}}$ now ignores its second input $\hat{\mu}$ at each query.

Game_5: We modify $\mathcal{O}^4_{\mathsf{aSig}}$ so that, at step 1.a, $\mathcal{O}^5_{\mathsf{aSig}}$ does not abort anymore in the event that $\tilde{a} = \lfloor a/2^\ell \rfloor$ collides with the $\tilde{a}$ of an earlier anamorphic signing query. By the same argument as in the transition from Game_1 to Game_2, we have $|\Pr[W_5] - \Pr[W_4]| \leq Q_S^2/2^\lambda$.

Game_6: We change $\mathcal{O}^5_{\mathsf{aSig}}$ into $\mathcal{O}^6_{\mathsf{aSig}}$, which now samples $a \hookleftarrow U(\mathbb{Z}^*_N)$ instead of $a \hookleftarrow U(\mathbb{Z}_N)$ at step 1.a. More precisely, $\mathcal{O}^6_{\mathsf{aSig}}$ is specified as follows:

$\mathcal{O}^6_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$,

1. Generate a as follows.
   a. Choose $a \hookleftarrow U(\mathbb{Z}_N^*)$.
   b. If $J_N(a) = -1$, set $a := u \cdot a \bmod N$.
   c. If $a \notin \mathbb{QR}_N$, set $a := -a \bmod N$.
2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_{2^t}$.
3. Using p and q, compute $\bar{z} = \left(a \cdot y^c\right)^{1/2^{t-1}} \bmod N$. Then, compute z as a random square root of $\bar{z}$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$.

The only change is thus the distribution of a which is now uniform over $\mathbb{Z}_N^*$ instead of being uniform over the full ring $\mathbb{Z}_N$. Since the two distributions are within statistical distance $(p + q - 1)/N < 2/\min(p, q) < 2^\lambda$, we have $|\Pr[W_6] - \Pr[W_5]| \leq Q_S/2^\lambda$.

Game_7**:** We now change $\mathcal{O}^6_{\mathsf{aSig}}$ into $\mathcal{O}^7_{\mathsf{aSig}}$, which is as follows

$\mathcal{O}^6_{\mathsf{aSig}}(\mu, \hat{\mu})$: To sign μ and encrypt $\hat{\mu} \in \{0,1\}^{\ell_m}$ using $\mathsf{ask} = (x, p, q)$ and $\mathsf{dk} = (\mathsf{avk}, k_m, K, \ell_m, u, h, H)$,

1. Pick $r \hookleftarrow U(\mathbb{Z}_N^*)$ and compute $u = r^{2^t} \bmod N$.
2. Compute $c = H_{\mathsf{FS}}(a, \mu) \in \mathbb{Z}_{2^t}$.
3. Compute $z = r \cdot x^c \bmod N$.

Output $\mathsf{sig} = (c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$.

The output distribution of $\mathcal{O}^7_{\mathsf{aSig}}$ is exactly the same as that of $\mathcal{O}^6_{\mathsf{aSig}}$ since the mapping $x \mapsto x^{2^t} \bmod N$ is one-to-one on $\mathbb{QR}_N$. We have $\Pr[W_7] = \Pr[W_6]$.

In Game_7, the anamorphic signing oracle has been replaced by a regular signing oracle that always returns normal signatures on μ. Since the distribution of $(\mathsf{vk}, \mathsf{sk})$ is also identical to that of $(\mathsf{avk}, \mathsf{ask})$, this concludes the proof. □

Achieving Strong Unforgeability. The scheme does not provide strong unforgeability against a dictator nor against a dishonest recipient. Indeed, the recipient can always modify a given signature $(c, z) \in \mathbb{Z}_{2^t} \times \mathbb{Z}_N^*$ and replace z by $-z$. Also, since the dictator knows the factorization of N, it can replace z by $z' = z \cdot \zeta$, where $\zeta \neq 1$ is a square root of 1.

However, the 2^t-root signature scheme can be made strongly unforgeable using a simple modification. The idea is to sample both the secret key x and the signing randomness r in the subgroup $\mathbb{J}_N$ instead of the larger group $\mathbb{Z}_N^*$ (which does not affect the distribution of $a = r^{2^t} \bmod N$ and $y = x^{2^t} \bmod N$). This ensures that $z \in \mathbb{J}_N$ at step 3 of the signing algorithm. Moreover, instead of simply outputting $z = r \cdot x^c \bmod N$, the signer can first check that $z \in \{0, \ldots, (N-1)/2\}$ and replace it by $-z$ if this is not the case. The verifier can then additionally check that $z \in \mathbb{J}_N \cap \{0, \ldots, (N-1)/2\}$. This prevents an adversary from replacing z by $z' = z \cdot \zeta$, where ζ is either -1 or a non-trivial square root of 1 (since

$J_N(\zeta) = -1$ in the latter case).

It is possible to prove that this simple method of making the ordinary signature scheme strongly unforgeable also provides dictator and recipient strong unforgeability in the anamorphic setting. We note that a similar modification was suggested in a different context [11, Section 4.5] for the Fiat-Shamir Σ-protocol [28]. A difference is that [11] does not force the Jacobi symbol of the response to be 1. Here, it is necessary to obtain strong unforgeability against a dictator knowing the factorization of N.