Anamorphic Construction For The Winternitz OTS Scheme Family

1st Lucas Mayr
Departamento de Informática e Estatística
Universidade Federal de Santa Catarina
Florianópolis, Brazil
lucas.mayr@posgrad.ufsc.br

2nd João Gabriel Feres
Departamento de Informática e Estatística
Universidade Federal de Santa Catarina
Florianópolis, Brazil
joao.feres@grad.ufsc.br

3rd Bruno Bianchi Pagani
Departamento de Informática e Estatística
Universidade Federal de Santa Catarina
Florianópolis, Brazil
bruno.bianchi.pagani@grad.ufsc.br

4th Ricardo Custódio
Departamento de Informática e Estatística
Universidade Federal de Santa Catarina
Florianópolis, Brazil
ricardo.custodio@ufsc.br

***Abstract*—The Winternitz One-Time Signature (WOTS) scheme is a fundamental primitive in post-quantum cryptography that relies solely on the security of its hash function, has been standardized, and serves as a critical building block for standardized hash-based signatures such as XMSS, LMS, and SPHINCS. However, schemes are traditionally analyzed under the assumption that their private keys are correctly utilized and kept secure. This assumption is broken when facing an adversary with high surveillance capabilities that can order the disclosure of users' private keys. Anamorphic cryptography is a branch of covert communication research that investigates how cryptography operates in such adversarial settings. In this article, we propose the first anamorphic constructions for the WOTS family of signature schemes and show that these constructions are indistinguishable from their traditional counterparts. We formalize these constructions within a game-based framework and prove their security under standard anamorphic assumptions, showing that the resulting schemes satisfy anamorphic indistinguishability. Lastly, we briefly discuss and compare the anamorphic length capabilities and the characteristics of each construction method. Our results show that WOTS and its variants can support an anamorphic covert channel that is resilient in highly adversarial environments.**

***Index Terms*—anamorphic encryption, digital signature, secrecy, covert communication, anamorphic signatures, Winternitz signature scheme**

In secure communications, several fundamental assumptions are taken for granted; they are generally realistic and have supported the development of practical security models. However, analyzing such assumptions through the lens of *heavy surveillance* reveals that some common presumptions are false [1]. Consider the watchful eye of a "big brother-like" entity capable of monitoring and reading all communication and punishing anyone who attempts to communicate in a manner it cannot decipher. The existence of this dictatorial adversary breaks two main fundamental assumptions: *sender-freedom* and *receiver-privacy* [2, §1].

Under typical environments, the sender-freedom assumption states that the sender can choose which messages to encrypt at their own discretion. However, under a dictatorial adversary, the sender may be forced to encrypt messages against their will, such that the meaning of *communicating with someone* is lost, as the message might have been encrypted under duress. Conversely, receiver-privacy assumes that any secrecy is maintained by hiding secrets, such as a private key, from attackers. Still, the existence of our dictatorial adversary implies these secrets must be handed over to them.

Early solutions to this problem base themselves on the *authentication without secrecy* [3] paradigm or by creating straightforward subliminal channels [4]. Consequently, the authentication channel may be used to send covert messages; to send m bits of information with r bits of authentication, $m + r'$ bits must be sent, where $|r'| < |r|$. Inevitably, in order to embed s bits of covert messages into r, the parties lose some degree of authentication capability. Thus, an adversary would be able to find that their probability of tricking the receiver into receiving an "authentic" message is lower than expected, alerting the adversary; that is, an honest r and the covert channel $m + r'$ are not sufficiently indistinguishible.

Anamorphic encryption [2] was introduced as a novel solution to this issue. Their proposal assumes that if users can trick the dictatorial adversary without creating *any* suspicion, they may exchange secret messages among themselves without fear of reprisal. Particularly, anamorphic encryption schemes may generate both traditional and anamorphic keys: one to be used as usual (and possibly shared with the Dictator) and another to be shared with the intended recipients. Thus, when the adversary demands to see the private key, the user discloses only the normal counterpart, thereby avoiding suspicion while maintaining secrecy of the subliminal channel. The required characteristics of an anamorphic encryption scheme are [2, §3]: i) anamorphic key pairs must be indistinguishable from their normal counterparts, and ii) ciphertexts generated by an anamorphic key must be indistinguishable from their usual counterparts.

In light of the threat posed by post-quantum computers and

their impact on the traditional security of signature schemes, novel schemes that are resistant to these threats have not only been proposed but also recently standardized. Among the candidates for post-quantum security, hash-based signature schemes have surfaced as a prime choice due to their minimal security assumptions and well-understood mathematical foundations. Specifically, the Winternitz One-Time Signature (WOTS) and its variants, WOTS+ and WOTS constant sum (WOTS-CS), have been used as standalone schemes and as building blocks in other standardized hash-based schemes, such as XMSS [5], LMS [6], and SPHINCS [7].

While many traditional signature schemes have been found to be anamorphic [8], no construction for the WOTS family itself has been proposed to date. In this work, we exploit the indistinguishability between ciphertexts and random strings [8] to enable the embedding of anamorphic messages while remaining indistinguishable from a standard construction to the anamorphic Dictator. Furthermore, we show that a straightforward adaptation also enables anamorphism in the WOTS+ and constant-sum variants of the Winternitz scheme. Moreover, we show that these anamorphic constructions are indistinguishable from their normal counterparts via game-theoretic proofs grounded in prior anamorphic findings, discuss their differences, and their anamorphic message length.

The paper is structured as follows: i) Section I presents the related work; ii) Section II provides the necessary preliminaries on WOTS and anamorphic definitions; iii) Section III discusses the threat model of the anamorphic Dictator used in this work; iv) Section IV details our proposed anamorphic constructions and their indistinguishability proofs; and v) Section V discusses our findings and future works.

I. Related Work

Anamorphic encryption has similarities to other covert communication protocols and techniques [4], such as key-escrow [9], [10], [11], kleptography [12], [13], steganography[14], [15], deniable encryption [16], and subvertable backdoored encryption [17]. However, these techniques either involve artifacts that our adversary can subvert or do not fully achieve the anamorphic encryption indistinguishability requirements [18], [14], [2], [16].

We present a brief outline of these methods and how they are unfit to meet our requirements: (i) key escrow aims to allow authorities to break secrecy in specific scenarios [18]; (ii) kleptography attacks existing cryptosystems by leaking data to an adversary, which in our case would be the Dictator themselves; (iii) steganography is hard to deploy in widespread scenarios [2] and might be broken by techniques that depend on MAC keys [14] (which the adversary may demand); (iv) deniable encryption is capable of generating *fake* coin tosses to trick adversaries but must be done *before* encrypting the message [16]; and (v) subvertable backdoored encryption is only used as the basis for key exchanges of some anamorphic encryption constructions [2, §5.3] that sacrifice zero-latency for dispensing the need for pre-shared information.

Banfi et al. [19] propose a generic extension that can add anamorphic qualities to public-key encryption (PKE) schemes. Furthermore, they formally define the natural property of PKE used for these anamorphic extensions. They discuss some limitations of the original proposal [2] and show how they can be addressed: decoupling regular and anamorphic key generation and providing a means for a receiver, privy to the anamorphic secrets, to detect the existence of a covert message when deciphering the message with the normal key.

Błaśkiewicz et al. [20] demonstrated how to use anamorphic cryptography to subvert an existing, legitimate, and trustworthy public blockchain to create a parallel covert ledger, hiding covert signatures (which the authors named *darknet signatures*) within the original signature. Furthermore, they present a simple construction of this attack with Schnorr signatures and discuss the possible methods to prevent the creation of covert ledgers.

Catalano et al. [21] show that anamorphism can be extended to homomorphic encryption while preserving its homomorphic properties for both the normal and the covert messages. Furthermore, they present a generic construction of anamorphic encryption based on hybrid and identity-based encryption. Moreover, they decouple the anamorphic key into an encryption component and a decryption component, enabling the creation of multiple covert channels with a single anamorphic key.

The notion of anamorphic signatures [8] was proposed alongside a stronger cryptographic control setting in which encryption either does not exist or is effectively invalidated, or controlled, by the adversary. The authors introduce the concept of anamorphism inside authentication channels. They demonstrate how these signatures may be used to covertly *watermark* digital objects to trace their origins. Furthermore, they also present several examples of anamorphic signature schemes based on known signature schemes [8], expand the proofs of anamorphic signatures to one-time signature schemes via an adapted oracle model, how these schemes can be extended to many-time signature schemes through the Naor-Yung paradigm [22], and finally, how to use these to construct secure anamorphic encryption schemes. They present several constructions based on classical signature schemes and an adaptation for Merkle trees by replacing the uniformly random numbers used in these schemes with the ciphered anamorphic message, which was proven to be indistinguishible [8]. However, the Winternitz OTS scheme was not included in this work, possibly due to the lack of a straightforward substitution.

II. Background

A. Digital signature schemes

Digital signatures are classic examples of how public-key cryptography can be employed. Such algorithms provide three main properties: authentication, which verifies the identity information associated with an entity; integrity, which guarantees that signed data has not been tampered with; and non-repudiation, which ensures that the signer cannot deny having signed a given message. Often, the message being signed is the digest of the original. A formal definition is given below.

Definition 1 ([23])**.** Let $\lambda \in \mathbb{N}$ be a security parameter and a message m such that $m \in \{0,1\}^*$. A *digital signature scheme* is a triple of probabilistic polynomial-time algorithms: i) *key generation* $\textsc{Gen}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk}) \in (\{0,1\}^* \times \{0,1\}^*)$, which outputs private and public keys, bound by a predefined property; ii) the *signature generation* $\textsc{Sig}(\mathsf{sk}, m) \to \sigma \in \{0,1\}^*$; and iii) the *signature verification* $\textsc{Dec}(\mathsf{pk}, m, \sigma) \to \{0,1\}$, all of which must satisfy $\Pr[\textsc{Dec}(\mathsf{pk}, m, \textsc{Sig}(\mathsf{sk}, m)) = 1] = 1$ for any $(\mathsf{sk}, \mathsf{pk})$.

B. Public Key Encryption

In a public-key encryption (PKE) scheme, messages can be encrypted with the public key and decrypted with the corresponding private key, allowing them to be sent through insecure channels where only the intended recipient can decipher the original message. We recall the common syntax used for public-key encryption:

Definition 2 ([19])**.** A *public-key encryption* scheme is a tuple $(\mathsf{KGen_p}, \mathsf{Enc}, \mathsf{Dec})$ where: i) the *key generation* $(\mathsf{KGen_e}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk}) \in (\{0,1\}^* \times \{0,1\}^*)$, which outputs private and public keys, bound by a predefined property; ii) the *encryption algorithm* $\textsc{Enc}(\mathsf{pk}, m) \to c \in \{0,1\}^*$, when necessary, we make the randomness $r \in R$ explicit by writing $\textsc{Enc}(\mathsf{pk}, m; r) \to c$; and iii) the *decryption algorithm* $\textsc{Dec}(\mathsf{sk}, c) \to m$ either decrypts c back to the original message m or errors out.

C. Winternitz one-time signature scheme

Winternitz is a One-Time Signature (OTS) scheme that is based almost entirely on the continuous application of a cryptographic hash function $\mathcal{H}$ to the private key, thereby generating both the signature and the public key. Our definition of the scheme is based on [24]. The notation $X_i^j \coloneqq \{X_i, \cdots, X_j\}$ will sometimes be adopted for brevity when needed.

Consider $\lambda \in \mathbb{N}$ as the security parameter, $n \in \mathbb{N}$ to be the message length, a cryptographic hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^n$, any message $m \in \{0,1\}^n$, and $w \geq 1$ to be the Winternitz parameter, usually a power of two, such that $\log_2 w$ represents the number of bits in each block. Let $t_1 = \lceil \frac{n}{\log_2 w} \rceil$, $t_2 = \lfloor \frac{\log_2(t_1(w-1))}{\log_2 w} \rfloor + 1$ and $t = t_1 + t_2$, which illustrate the total number of blocks for the scheme. We define, for an arbitrary function f, $f^i(x)$ to be $f(f(\dots f(x)\dots))$ applied i times, with $f^0(x) = x$; hereby named a function chain. We present the algorithms that form the signature scheme below:

$\textsc{Gen}(1^\lambda)$. Generate $\lambda \cdot t$ random bits and set the private key sk to be composed by t random words of equal size λ, i.e., $\mathsf{sk} = (x_0, \dots, x_{t-1})$. The public key is obtained by applying the hash function $\mathcal{H}$ to each block of the private key $w-1$ times. That is, $\mathsf{pk} = (\mathcal{H}^{w-1}(x_0), \dots, \mathcal{H}^{w-1}(x_{t-1}))$.

$\textsc{Sig}(\mathsf{sk}, m)$. Slice m into a tuple of t_1 elements of $\log_2 w$ bits $B_1 = (b_0, \dots, b_{t_1-1})$. Calculate the checksum $cs = \sum_{i=0}^{t-1}(w-1-b_{t-i})$ and slice it into an analogous tuple of t_2 elements of $\log_2 w$ bits $B_2 = (b_{t_1}, \dots, b_{t-1})$. Then, we have that $\sigma = \left(\sigma_i = \mathcal{H}^{b_i}(x_i)\right)_{i=0}^{t-1} = (\sigma_0, \dots, \sigma_{t-1})$.

$\textsc{Ver}(\mathsf{pk}, m, \sigma)$. Obtain B_1 and B_2 from m as per the traditional $\textsc{Sig}$ step. Then, the signature σ is valid if and only if $\mathsf{pk} \iff (\mathcal{H}^{w-1-b_0}(\sigma_0), \dots, \mathcal{H}^{w-1-b_{t-1}}(\sigma_{t-1}))$, and invalid otherwise.

D. Anamorphic Lamport Tagging Scheme

The Lamport Tagging Scheme is a one-time signature system defined by a tuple ($\textsc{Gen}$, $\textsc{Sig}$, $\textsc{Dec}$) and a one-way function $f : \{0,1\}^* \to \{0,1\}^n$ [25]. Initially, for a message of length l, we generate two random bit strings of length l each one acting as a private key such that $x = (x_0, x_1) \leftarrow \{1,0\}^\lambda$, where $(x_0, x_1) = ((x_{0,1}, \cdots, x_{0,l}), ((x_{1,1}, \cdots, x_{1,l}))$. Next, we compute each public key section $y_{i,j} = f(x_{i,j})$, in such a way $i \in \{0,1\}$ and $1 \leq j \leq l$. To sign a message $m = (m_1, \cdots, m_l)$, we select the corresponding entries from our private key $x_{i,j}$ based on the value and position of the message bit; for example, if the message bit m_2 is equal to 1, we select $x_{1,2}$, repeating this process for the entire message. Thus, we generate a signature $\sigma = (\sigma_1, \cdots, \sigma_l)$ based on a mix of x_0 and x_1. Finally, to verify this signature, we compare $f(\sigma_j)$ to $y_{m_j,j}$; if they all match, the signature is valid.

The proof of the anamorphism of the Lamport Tagging Scheme in [8] is based on Theorem 1: in the key generation step, being dk the double key, and x_0 and x_1 the random length $l\lambda$ bit strings generated, the first λ bits of each sequence are switched to the anamorphic message encrypted with an encryption scheme in the terms of Theorem 1. For simplicity, we assume that dk denotes the symmetric key generated by this encryption scheme. Then, the process in this step follows the same steps as the original algorithm. Next, in the verification algorithm, the first λ bits of the signature are decrypted using dk and the anamorphic message is recovered. We refer readers to [8] for a more in-depth proof of the LTS anamorphism.

Theorem 1 (§3.1 [8])**.** *Assume that there exists a one-way function, then there exists an IND-CPA secure symmetric encryption scheme that produces pseudorandom ciphertexts, i.e., those generated ciphertexts are indistinguishable from a random distribution.*

III. Threat Model

To analyze the security of our anamorphic constructions, we adopt the standard Dictator model introduced by Persiano et al. [2], with the additional adaptations for authentication channels by Kutylowski et al. [8]. That is, the Dictator has access to any standard private key and has prohibited message encryption, leaving digital signatures as the only means of transmitting covert messages. That is, the Dictator exerts heavy surveillance over all communication, implying: i) Heavy Surveillance: the Dictator can demand and inspect any standard private key and message generated by a user; and ii) Encryption Prohibition: the Dictator can prohibit standard encryption, allowing only digital signatures for communication and authentication purposes.

However, it does not have access to the algorithms being employed, including key generation, signature generation, or verification. Specifically for the purposes of this work, the Dictator does not control any of the following: i) Algorithm Control: While the Dictator knows the public specifications

of the algorithms, they cannot monitor their execution in a private environment; ii) Randomness Generation: The Dictator can not force the use of specific seeds, nor request the seed used in an algorithm; iii) Signature Generation Time: The Dictator does not perform side-channel attacks, such as timing analysis, to infer irregularities in user algorithms and schemes; and iv) Ciphertext Indistinguishability: The Dictator cannot distinguish between ciphertexts and random strings as per Theorem 1, which was proven in [8]. Lastly, following the findings of [8, §2.1], we remark that anamorphic signatures are IND-CPA secure and will show only the anamorphic property of our constructions through the indistinguishability property of Theorem 1.

IV. Anamorphic WOTS

Before diving into the constructions themselves, we model the adversary as the classical Dictator from *Anamorphic Encryption* [2], who has access to any standard private key generated by any entity and can read ciphertexts, generate ciphertexts, and forge signatures. Furthermore, we include the additional encryption limitations proposed by Kutylowski et al. [8], in which entities cannot generate ciphertexts; instead, messages must be sent to the Dictator, which then generates the ciphertexts. This additional restriction requires anamorphic communication to be performed using signature schemes.

We begin by looking at the predecessor of hash-based signatures, the *Lamport Tagging Scheme* (LTS), and generating both a normal game and an anamorphic game for it; showing that it can transmit anamorphic messages; that is, anamorphism for one-time signature schemes, then we show that the same principle can be used for the padded version of WOTS and WOTS+ in two distinct, but equivalent, variations; finally, we extend the arguments from classical WOTS to the constant-sum variant where performance can be made deterministic.

A. WOTS Construction

We recall from Subsection II-C that for WOTS, the part that contains the usual random generation that would be replaced by an anamorphic message is also the secret key sk; however, unlike LTS, WOTS does not reveal any part of its secret key as part of its scheme and, consequently, requires other venues of anamorphic construction. In WOTS-based schemes, the choice of encoding dictates many characteristics such as security, signature size, and overall performance; this property has been exploited by WOTS-based schemes, where they repeatedly randomize a padding or nonce to improve either verification or signature performance by way of increasing or decreasing the total number of hashes needed to reach the public key from the signature or to reach the signature from the private key correspondingly. Furthermore, these processes have already been proven secure [26] and have been used to improve performance in embedded devices [26].

For this anamorphic construction, we have two venues we can manipulate through the addition of a random padding or nonce (henceforth used interchangeably); following previous constructions of anamorphic signatures, we exploit the randomness of this nonce and exchange it for our anamorphic message; we remark that while *suspicion* has not been explicitly defined for the anamorphic model, if a scheme that aims to improve verification speeds ends up substituting a legitimate nonce for an anamorphic message that results in a signature with bad verification performance, it can easily be taken as suspicious, and thus, should be chosen with this in mind.

Following the proof of the Lamport Signature Scheme as defined above, assume that we have two implementations, one that runs the usual WOTS scheme with a randomized nonce as in Figure 1, wherein the signer generates a WOTS key pair $(sk, pk) \leftarrow \mathsf{Gen}(1^\lambda)$; to sign a message the signer samples a uniformly random padding $p \in \{0,1\}^\eta \mid \eta \in \mathbb{N}_0$ and appends it to the message m before calculating its digest to be signed. That is, the signature is computed over $\mathcal{H}(m\|p)$. We remark that the random padding p can be sampled multiple times until a $\mathcal{H}(m\|p)$ with acceptable performance is reached. However, this step is omitted from the game for clarity, as it does not affect any step in the proof of anamorphic properties under our threat model. We assume the acceptance predicate depends only on $\mathcal{H}(m\|p)$, modeled as a random oracle. Under this assumption, conditioning on acceptance is independent of which bits of p carry the ciphertext, so the distribution of accepted p coincides in both games up to the pseudo-randomness of the encrypted anamorphic message c_μ.

Furthermore, we also present our anamorphic construction of the WOTS scheme in Figure 2. In this construction, the signer generates an anamorphic double key $dk \leftarrow \mathsf{KGen}_E(1^\lambda)$ to encrypt the anamorphic message m_μ such that $c_\mu \leftarrow \mathsf{Enc}(dk, m_\mu)$, and $|c_\mu| = \alpha$; then, it uses this ciphertext as part of the padding field used in the traditional scheme to embed the anamorphic message. Note that it must still reserve some bits to allow the algorithm to find suitable performance. That is, we have $p \leftarrow c_\mu\|z$, where $z \leftarrow \{0,1\}^{\eta-\alpha}$ carrying the anamorphic message and our signature σ computed over $\mathcal{H}(m\|c_\mu\|z)$.

Moreover, as the inclusion of a nonce changes the checksum value of the WOTS signature, the embedding of the anamorphic message can also be performed inside the checksum itself instead of the padding, as seen in Figure 3. Note, however, that the most significant bits of the checksum must be used for their intended purpose, leading to a small loss of anamorphic message space. That is, the resulting checksum must be sufficiently small or sufficiently large to avoid raising questions about the scheme (verification-focused instance vs. signature-performance instance).

Let $\mu T_w = (\mu Gen, \mu Sig, \mu Dec)$ be our anamorphic WOTS construction with padding length η. μGen exchanges the random padding for our anamorphic message ciphertext c_μ with length $|c_\mu| < \eta$, then uses the leftover bits to sample z to find passable performance; μSig and μDec follow the steps fo the traditional scheme as usual. A receiver that knows of the embedded anamorphic message is able to recover c_μ from the padding bits and retrieve m_μ by decrypting $\mathsf{Dec}(dk, c_\mu)$.

WOTS NORMAL GAME - rG_{WOTS}

GEN(1^λ):
1. $sk \xleftarrow{\$} \{0,1\}^\lambda$
2. $pk \leftarrow \left(\mathcal{H}^{w-1}(sk_j)\right)_{j=0}^{t-1}$
3. return (sk, pk)

SIG(sk, m):
1. $p \xleftarrow{\$} \{0,1\}^\eta$
2. $B_1 \leftarrow$ base-w$(\mathcal{H}(m\|p))$
3. $cs \leftarrow \sum_{j=0}^{t_1-1}(w-1-b_j)$
4. $B_2 \leftarrow$ base-w(cs)
5. $B \leftarrow B_1\|B_2$
6. $\sigma \leftarrow \left(\mathcal{H}^{b_j}(sk_j)\right)_{j=0}^{t-1}$
7. return σ

VER(pk, m, σ):
1. $r \leftarrow \left(\mathcal{H}^{\,w-1-b_j}(\sigma_j) \stackrel{?}{=} pk_j\right)_{j=0}^{t-1}$
2. return r

Figure 1. Normal game for the WOTS Scheme

WOTS ANAMORPHIC GAME - $\mu\mathsf{G}_{T_w}$

μ **SIG**(sk, dk, m, m_μ):
1. $z \xleftarrow{\$} \{0,1\}^{\eta-\alpha}$
2. $p \leftarrow$ **ENC**$(dk, m_\mu)\|z$
3. $B_1 \leftarrow$ base-w$(\mathcal{H}(m\|p))$
4. $cs \leftarrow \sum_{j=0}^{t_1-1}(w-1-b_j)$
5. $B_2 \leftarrow$ base-w(cs)
6. $B \leftarrow B_1\|B_2$
7. $\sigma \leftarrow \left(\mathcal{H}^{b_j}(sk_j)\right)_{j=0}^{t-1}$
9. return σ

μ **DEC**(dk, p):
1. $c_\mu \leftarrow (p_0, \cdots, p_{\alpha-1})$
2. $m_\mu \leftarrow$ **DEC**(dk, c_μ)
3. return m_μ

Figure 2. Anamorphic game for the Padding variant of the WOTS Scheme.

WOTS ANAMORPHIC GAME (CHECKSUM) - $\mu\mathsf{G}_{T_{wc}}$

μ **SIG**(sk, m, m_μ):
1. $p \xleftarrow{\$} \{0,1\}^\eta$
2. $B_1 \leftarrow$ base-w$(\mathcal{H}(m\|p))$
3. $c_\mu \leftarrow$ **ENC**(dk, m_μ)
4. $B_2 \leftarrow$ base-w(c_μ)
5. $B \leftarrow B_1\|B_2$
6. $\sigma \leftarrow \left(\mathcal{H}^{b_j}(sk_j)\right)_{j=0}^{t-1}$
7. return σ

μ **DEC**(dk, B_2):
1. $m_\mu \leftarrow$ **DEC**(dk, B_2)
2. return m_μ

Figure 3. Anamorphic game for the checksum variant of the WOTS Scheme.

Theorem 2 (μWOTS)**.** *Assuming the existence of a one-way function, there exists an anamorphic construction for the WOTS scheme.*

Proof. Let $\mathcal{E} = (\mathsf{KGen}_E, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric encryption scheme where Enc produces pseudorandom ciphertexts as per Theorem 1. Assume that Enc encrypts n-bit messages into α-bit cyphertexts, where $\alpha < \eta$. Construct the anamorphic triplet $\mu T_w = (\mu Gen, \mu Sig, \mu Dec)$ such that:

1) $\mu Gen(1^\lambda)$: generate the double key $dk \leftarrow \mathsf{KGen}_E(1^\lambda)$. Generate the WOTS key pair (sk, pk) as usual.
2) $\mu Sig(sk, dk, m, m_\mu)$: let $c_\mu \leftarrow \mathsf{Enc}(dk, m_\mu)$ with length $|c_\mu| = \alpha$ and $z \xleftarrow{\$} \{0,1\}^{\eta-\alpha} \mid \eta \in \mathbb{N}$. Sample z until a passable performance threshold is reached. Set the WOTS padding p to $p \leftarrow c_\mu\|z$ and calculate $B_1 \leftarrow$ base-w$\big(\mathcal{H}(m||p)\big)$; next, obtain cs, B_2, and B as usual and generate σ following the traditional steps of the WOTS scheme.
3) $\mu Dec(dk, p)$: retrieve c_μ from $c_\mu \leftarrow p_0^{\alpha-1}$, and decrypt c_μ back to m_μ using the double key; i.e., $m_\mu \leftarrow \mathsf{Dec}(dk, c_\mu)$.

Now, assume for contradiction that WOTS with T_w is not an anamorphic construction. Then, there exists a probabilistic polynomial-time (PPT) Dictator $\mathcal{D}$ with access to the WOTS key pair (sk, pk) of the signer and a signing oracle $\mathcal{O}_d$ that runs either the normal WOTS game rG_{WOTS} or the anamorphic WOTS game $\mu\mathsf{G}_{T_w}$. The Dictator may query signatures from chosen plaintexts m_d and must output a bit indicating whether it believes it is interacting with rG_{WOTS} (bit 0) or $\mu\mathsf{G}_{T_w}$ (bit 1). We note that due to the nature of OTS schemes, a new WOTS key pair (sk, pk) is generated for each query. Consequently, the advantage of the Dictator in guessing correctly is given by:

$$\mathsf{Adv}_{\mathcal{D}} = |\Pr[\mathsf{rG}_{WOTS}(\lambda) = 1] - \Pr[\mu\mathsf{G}_{T_w}(\lambda) = 1]| \quad (1)$$

If the normal game rG_{WOTS} is being played, p is a random sample, and $\mathcal{O}_d$ is simulating the real game to D. Otherwise, if $\mathcal{O}_d$ is playing $\mu\mathsf{G}_{T_w}$, then p carries the anamorphic message ciphertext c_μ, and $p = \big(\mathsf{Enc}(dk, m_\mu)\|z\big)$; i.e., $\mathcal{O}_d$ simulates the anamorphic game for D.

Assume, for contradiction, that there exists a PPT adversary $\mathcal{D}$ with substantial advantage when guessing between those two games, such that $\mathsf{Adv}_{\mathcal{D}} \geq \epsilon(\lambda)$ for some negligible function ϵ. Then, let $\mathcal{A}$ be a PPT adversary that interacts with the challenger stated in Theorem 1 and receives access to an oracle $\mathcal{O}_A$ that returns either ciphertexts $c_\mu \leftarrow \mathsf{Enc}(dk, m_\mu)$ of length α or uniformly random strings of the same length. $\mathcal{A}$ wishes to gain an advantage $\mathsf{Adv}_{\mathcal{A}}$ when distinguishing between these two responses from the oracle.

For this purpose, $\mathcal{A}$ instantiates a Dictator $\mathcal{D}$ and acts as its oracle, receiving plaintext messages to sign using WOTS. For this purpose, whenever it receives a plaintext to sign from $\mathcal{D}$, it generates a WOTS key pair (sk, pk) and queries its own oracle $\mathcal{O}_{\mathcal{A}}$ to generate a padding for the WOTS scheme. Then,

it responds to the query from $\mathcal{D}$ by generating σ utilizing the padding received by $\mathcal{O}_{\mathcal{A}}$ and sends it back to $\mathcal{D}$, listening to its guess. Lastly, it forwards the response bit from $\mathcal{D}$ as its own response. Consequently, if the Dictator responds indicating that the signature was made using the anamorphic WOTS game, $\mathcal{A}$ responds signaling that $\mathcal{O}_{\mathcal{A}}$ generated a ciphertext; and if the Dictator responds indicating that the signature was made using the normal WOTS game, $\mathcal{A}$ responds signaling that $\mathcal{O}_{\mathcal{A}}$ generated a random string.

That is, if the oracle $\mathcal{O}_{\mathcal{A}}$ returns uniform random strings, then p is uniformly random, and the simulation is equivalent to rG_{WOTS}; otherwise, if the oracle $\mathcal{O}_{\mathcal{A}}$ returns a ciphertext, then $c_\mu \subseteq p$ and the simulation is equivalent to the anamorphic game $\mu\mathsf{G}_{T_w}$.

Consequently, any non-negligible advantage of Dictator $\mathcal{D}$ in distinguishing between the anamorphic and the real game translates directly into a non-negligible advantage for adversary $\mathcal{A}$ in distinguishing ciphertexts from uniform distributions, contradicting Theorem 1's indistinguishability of ciphertexts. Therefore, no PPT adversary can distinguish between the normal game rG_{WOTS} and the anamorphic game $\mu\mathsf{G}_{T_w}$ with non-negligible advantage, and the construction satisfies anamorphic indistinguishability.

□

We remark that the checksum variation follows the same arguments for its anamorphic indistinguishability proof. Intuitively, instead of sampling $p \leftarrow c_\mu \| z$ until it finds acceptable performance, it samples $p \leftarrow \{0,1\}^\eta$ until it finds a B_2 with passable performance that contains $c_\mu \subseteq B_2$. We also note that this rejection-sampling method may require more samples to find a suitable B_2, but the considerably small length of B_2 should yield feasible performance. As in the anamorphic WOTS padding proof above, the adversary could exploit the checksum field to gain an advantage in breaking the indistinguishability property of Theorem 1, since both preserve a seemingly uniform random distribution under their anamorphic encodings.

B. WOTS+ Construction

In contrast to the previous construction, the WOTS+ variant introduces a random number sampled from a uniform distribution, similar to the Lamport construction, such that we can exploit this characteristic in a manner similar to some previously introduced techniques reliant on random numbers [8]. Unlike the traditional WOTS, WOTS+, shown in Figure 4, introduces two new parameters: r, which is a set of $w-1$ random strings of size λ used to improve the entropy of the private key, and k, a function key of a key space $\mathcal{K}_\lambda$ of a family of cryptographic hash functions $\mathcal{H}_k : \{0,1\}^\lambda \rightarrow \{0,1\}^\lambda$. Using them, the chaining function is defined as $\mathfrak{C}_k^i(x,k) = \mathcal{H}_k\big(\mathfrak{C}_k^{i-1}(x,r) \oplus r_i\big)$ for $i \geq 1$, with base case $\mathfrak{C}_k^0(x,r) = x$. Furthermore, the public key is evaluated as $pk = (pk_0, , \cdots, pk_{t-1}) = \big((r,k), \mathfrak{C}_k^{w-1}(sk_0, r), \cdots, \mathfrak{C}_k^{w-1}(sk_{t-1}, r)\big)$. Both signing and verification steps operate similarly, differing only in the hashing function $\mathfrak{C}_k$ and the uniformly random parameter r.

Consequently, we exploit the existence of this uniformly random sample in the form of the variable r in the same manner as the constructions made by [8]. Thus, our anamorphic construction for the WOTS+ variant modifies the key generation step GEN to include the covert message m_μ as a parameter embedded in the randomness of r, i.e., $\mu\textsc{Gen}(1^\lambda, dk, m_\mu)$.

During the generation process, we substitute the would-be legitimate $w-1$ strings of parameter r with strings containing our encrypted anamorphic message such that $r \leftarrow \textsc{Enc}(dk, m_\mu)$. Since the verification step also receives r as part of the public key, we can recover our anamorphic message m_μ such that $m_\mu \leftarrow \textsc{Dec}(dk, r) = \textsc{Dec}(dk, c_\mu)$. This process is seen in Figure 5.

Informally, we recall that we reach our anamorphism through the substitution of the uniformly random parameter r with our encrypted message $\textsc{Enc}(dk, m_\mu)$. Thus, to distinguish between $\textsc{Enc}(dk, m_\mu)$ and r is to imply that the Dictator can differentiate between an encryption function and a random uniform distribution. However, as we discussed in the LTS games and through the results of [8] in Theorem 1, we know them to be indistinguishable; thus, this oracle can not exist, and the Dictator is unable to differentiate between a normal and an anamorphic game with probability higher than the allowed negligible function $\epsilon(\lambda)$, i.e., our anamorphic construction of WOTS+ is indistinguishable from the traditional scheme. In a more formal manner, we present the following Theorem:

Theorem 3 (μWOTS+)**.** *Assuming the existence of a one-way function, there exists an anamorphic construction for the WOTS+ scheme.*

Proof. Let $\mathcal{E} = (\mathsf{KGen}_E, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric encryption scheme where Enc produces pseudorandom ciphertexts indistinguishable from a random distribution as per Theorem 1. Assume, for simplicity, that Enc encrypts n-bit messages into $\lambda(w-1)$-bit cyphertexts. Construct the anamorphic triplet $\mu T_{w+} = (\mu Gen, \mu Sig, \mu Dec)$ such that:

1) $\mu Gen(1^\lambda, dk, m_\mu)$: generate the double key dk where $dk \leftarrow \mathsf{KGen}_E(1^\lambda)$. Calculate $r \leftarrow \mathsf{Enc}(dk, m_\mu)$ and use it to generate the WOTS+ key pair (sk, pk).
2) $\mu Sig(m, sk, r)$: follows the same steps of the normal game without alterations.
3) $\mu Dec(dk, r)$: retrieve r from the public key and recover the anamorphic message $m_\mu \leftarrow \mathsf{Dec}(dk, r)$.

Let rG_{WOTS+} be the normal game running a standard instance of WOTS+ and $\mu\mathsf{G}_{T_{w+}}$ an anamorphic construction of WOTS+. The advantage a Dictator would have in discerning between the different games is given by:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D}} = |\ &\Pr[\mathsf{rG}_{WOTS+}(\lambda) = 1] \\ &- \Pr[\mu\mathsf{G}_{T_w+}(\lambda) = 1]\ |\end{aligned} \tag{2}$$

Then, there exists a PPT Dictator $\mathcal{D}$ and an oracle $\mathcal{O}_d$ that runs either the normal Gen steps from rG_{WOTS+} or the altered steps of $\mu\mathsf{Gen}$ from $\mu\mathsf{G}_{T_{w+}}$. The Dictator may query key pairs and must output a bit indicating whether it believes it is interacting with rG_{WOTS+} (bit 0) or $\mu\mathsf{G}_{T_{w+}}$ (bit 1). For contradiction, assume that $\mu\mathsf{G}_{T_{w+}}$ is not an anamorphic construction, then, for some negligible function $\epsilon(\lambda)$, the

WOTS+ NORMAL GAME - rG_{WOTS+}

GEN(1^λ):
1. $sk \xleftarrow{\$} \{0,1\}^{t\lambda}$
2. $r \xleftarrow{\$} \{0,1\}^{(w-1)\lambda}$
3. $k \xleftarrow{\$} \mathcal{K}_\lambda$
4. $pk \leftarrow (r,k) \| \left(\mathfrak{C}_k^{w-1}(x_j, r)\right)_{j=0}^{t-1}$
5. return (sk, pk)

SIG(m, sk, r):
1. $B_1 \leftarrow$ base-w(m)
2. $C \leftarrow \sum_{j=0}^{t_1-1}(w-1-b_j)$
3. $B_2 \leftarrow$ base-w(C)
4. $B \leftarrow B_1 \| B_2$
5. $\sigma = (\mathfrak{C}_k^{b_j}(sk_j, r))_{j=0}^{t-1}$
6. return σ

VER(pk, m, σ):
1. $B_1 \leftarrow$ base-w(m)
2. $C \leftarrow \sum_{j=0}^{t_1-1}(w-1-b_j)$
3. $B_2 \leftarrow$ base-w(C)
4. $B \leftarrow B_1 \| B_2$
5. $R \xleftarrow{} pk \overset{?}{=} (r,k) \| (\mathfrak{C}_k^{w-1-b_j}(\sigma_j, r_{j+1}^{w-1}))_{j=0}^{t-1}$
6. return R

Figure 4. Normal game for the WOTS+ Scheme

WOTS+ ANAMORPHIC GAME - $\mu\mathsf{G}_{T_{w+}}$

μ **GEN**$(1^\lambda, dk, m_\mu)$:
1. $sk \xleftarrow{\$} \{0,1\}^{t\lambda}$
2. $r \leftarrow$ **ENC**(dk, m_μ)
3. $k \xleftarrow{\$} \mathcal{K}_\lambda$
4. $pk \xleftarrow{} (r,k) \| \left(\mathfrak{C}_k^{w-1}(x_j, r)\right)_{j=0}^{t-1}$
5. return (sk, pk)

μ **DEC**(dk, r):
1. $m_\mu \leftarrow$ **DEC**(dk, r)
2. return m_μ

Figure 5. Anamorphic game for the WOTS+ Scheme.

advantage $\mathsf{Adv}_{\mathcal{D}}$ of the Dictator must be substantial such that $\mathsf{Adv}_{\mathcal{D}} \geq \epsilon(\lambda)$.

Now, let $\mathcal{A}$ be a PPT adversary that challenges the pseudo-randomness characteristic of Theorem 1 through an oracle $\mathcal{O}_{\mathcal{A}}$. For each query made by $\mathcal{A}$, $\mathcal{O}_{\mathcal{A}}$ returns either a ciphertext or a random string, and the adversary must guess which type of string it has been given, responding with a bit 0 for the random string and a bit 1 for the ciphertext.

For this purpose, $\mathcal{A}$ constructs a Dictator $\mathcal{D}$ and behaves as its oracle. Then, $\mathcal{A}$ queries its oracle for a string and uses it to construct a WOTS+ private key, sends it to $\mathcal{D}$, and forwards its response as its own.

That is, if the private key built by $\mathcal{A}$ used a ciphertext, it is equivalent to the anamorphic game Gen; otherwise, if it used a random string, it is equivalent to the normal game Gen. Observe that r is the only part of the key pair that is generated differently in both games. Consequently, any non-negligible advantage of Dictator $\mathcal{D}$ translates directly into a non-negligible advantage for adversary $\mathcal{A}$ in distinguishing ciphertexts from uniform distributions, contradicting Theorem 1. Therefore, no PPT adversary can distinguish between the normal game rG_{WOTS+} and the anamorphic game $\mu\mathsf{G}_{T_{w+}}$ with non-negligible advantage, and the construction satisfies anamorphic indistinguishability.

□

C. WOTS Constant-sum Construction

This variant of WOTS+ uses an alternative encoding to base-w, so the running time performance of the procedures is independent of the raw bits of the signed message. That is, in the classical WOTS+ scheme, the number of hashes needed to sign a message for each Winternitz block always depends solely on the base-w encoding of the original message; this constant-sum variant (WOTS-CS), on the other hand, utilizes techniques that manipulate the final hash such that it results in a fixed number of remaining hash operations. WOTS-CS is particularly beneficial for systems with constrained computational resources, such as low-power IoT devices, cryptographic hardware modules, or systems with stringent bandwidth requirements.

In this scheme, WOTS-CS exchanges the Winternitz parameter w for a corresponding variable n that serves a similar purpose of numbering the *hash-chain* length in the WOTS scheme. WOTS-CS also introduces the constant-sum parameter $s \in \mathbb{N}$, analogous to the original checksum purpose of WOTS, indicating the total number of remaining hashes for a particular signature, and $t \in \mathbb{N}$, which represents the number of hash chains. Additionally, WOTS-CS introduces a mapping function $\Phi(s,n)$ such that $\sum_{i=0}^{t-1} \Phi\big(\mathcal{H}(m)\big)_i = s$, enforcing the same constraints of the checksum value. The key generation process is analogous to WOTS+, while the signing and verification algorithms also include the mapping function Φ, as shown in Figure 6.

We remark that WOTS-CS inherits the same anamorphic venue from WOTS+, where the anamorphic ciphertext is used in place of parameter r. However, the unique properties of the mapping function Φ allow a second, promising construction based on the selection of the constant-sum parameter s. Unlike the WOTS construction, which relies on rejection sampling, WOTS-CS is able to deterministically find a suitable Φ that maps to the chosen constant-sum parameter s. For the former construction, the same indistinguishability proof used for WOTS+ applies.

Figure 7 shows the anamorphic steps based on the normal game for the WOTS-CS Scheme. We observe that this construction relieon the substitution of the r parameter for the encrypted anamorphic message, which follows the same steps

WOTS-CS NORMAL GAME - $\mathsf{rG}_{WOTS-CS}$

GEN$(1^\lambda, s, n)$:
1. $sk \xleftarrow{\$} \{0,1\}^{t\lambda}$
2. $r \xleftarrow{\$} \{0,1\}^{(w-1)\lambda}$
3. $k \xleftarrow{\$} \mathcal{K}_\lambda$
4. $pk \leftarrow (r,k) \| \left(\mathfrak{C}_k^{w-1}(x_j, r)\right)_{j=0}^{t-1}$
5. return (sk, pk)

SIG(m, sk, k, s, n):
1. $(b_0, ..., b_{t-1}) \leftarrow \Phi_{s,n}(m)$
2. $\sigma \leftarrow \left(\Phi_{s,n}^{n-b_j}(sk_j, k)\right)_{j=0}^{t-1}$
3. return σ

VER(pk, m, σ):
1. $(b_0, ..., b_{t-1}) \leftarrow \Phi_{s,n}(m)$
2. $r \leftarrow pk \stackrel{?}{=} \left(\mathfrak{C}^{b_j}(\sigma_j, (k_{n-b_{t-1}}, n))\right)_{j=0}^{t-1}$
3. return r

Figure 6. Normal game for the WOTS-CS Scheme

of the anamorphic game for WOTS+. This property allows the direct transference of the method used to enable anamorphism in WOTS+ to WOTS-CS.

WOTS-CS ANAMORPHIC GAME $\mu\mathsf{G}_{T_{cs}}$

μ **GEN**$(1^\lambda, dk, m_\mu)$:
1. $sk \xleftarrow{\$} \{0,1\}^{t\lambda}$
2. $r \leftarrow$ **ENC**(dk, m_μ)
3. $k \xleftarrow{\$} \mathcal{K}_\lambda$
4. $pk \leftarrow (r,k) \| \left(\mathfrak{C}_k^{w-1}(x_j, r)\right)_{j=0}^{t-1}$
5. return (sk, pk)

μ**DEC**(dk, r):
1. $m_\mu \leftarrow$ DEC(dk, r)
2. return m_μ

Figure 7. Anamorphic game for the WOTS-CS scheme.

Furthermore, we note that when the user selects the value of the sum parameter in WOTS-CS, they may choose a sum parameter that contains the encrypted anamorphic message c_μ as part of the parameter itself; i.e., $c_\mu \subseteq s$. This parameter is then decrypted during the verification stage of the scheme, thereby recovering the covert message. We note that the anamorphic length performance of this scheme depends solely on the choice of s and, therefore, is higher for schemes that prioritize signature speed than for those that prioritize the verification step. However, we remark that this method is not based on the substitution of random strings for a ciphertext, as shown to be indistinguishable by Theorem 1, and therefore might raise suspicion depending on the chosen sum parameter value.

D. Tree-Based Signature Anamorphism Inheritance

In this subsection, we show that tree-like signature schemes that admit hash-based OTS schemes inherit the leaf signatures anamorphism when employing the anamorphic construction of WOTS and its variants. For brevity, we present a proof for binary trees and later extend it to n-ary trees.

We show by contradiction that a tree-based signature scheme that employs Winternitz-based signatures at its leaves inherits their anamorphic properties from the chosen leaf signature scheme. That is, we show that if this tree did not inherit the anamorphic properties of its leaf signature scheme, an adversary would be able to gain a significant advantage when breaking the properties of indistinguishability proven by Theorem 1.

Theorem 4. *Assuming the existence of a one-way function, let γ be a hash-based binary tree signature scheme whose leaves are instantiated with a Winternitz-based signature scheme that admits an anamorphic construction. Consequently, γ is anamorphic.*

Proof. Let γ be a binary tree of height h, where the root corresponds to $h = 0$, and each leaf corresponds to a Winternitz-based signature scheme. Each signature must also reveal a path to the root, called the authentication path. We remark that this path is deterministic and independent of the message encoding and the maximum number of leaves; hence, the number of signatures equals 2^h. Furthermore, the anamorphic triplet is inherited from the leaf corresponding to the chosen anamorphic construction; e.g., μT_w when choosing WOTS.

We construct two games: the normal game rG_γ, where every leaf signature uses a standard construction, and the anamorphic game $\mu\mathsf{G}_\gamma$, where one of the leaves contains an anamorphic construction of the chosen hash-based signature scheme. The root and the authentication path of the tree are computed as usual for either game.

Assume, for contradiction, that a tree γ with an anamorphic leaf does not inherit the anamorphic property of indistinguishability from its leaf signature scheme. Consequently, there must exist a PPT Dictator $\mathcal{D}_\gamma$ with advantage $\mathsf{Adv}_{\mathcal{D}_\gamma} > \epsilon(\lambda)$ for a negligible function $\epsilon(\lambda)$, access to every signer key pair (sk, pk) for γ, and an oracle $\mathcal{O}_\gamma$ that simulates either the honest tree through rG_γ or an anamorphic game $\mu\mathsf{G}_{T_\gamma}$ for the tree given a set of up to 2^h messages. That is, the Dictator may query up to 2^h signatures from chosen plaintexts and must output a bit indicating whether it believes it is engaging with rG_γ (bit 0) or $\mu\mathsf{G}_\gamma$ (bit 1). The advantage of the Dictator guessing correctly is given by:

$$\mathsf{Adv}_{\mathcal{D}} = |\Pr[\mathsf{rG}_\gamma(\lambda) = 1] - \Pr[\mu\mathsf{G}_{T_\gamma}(\lambda) = 1]| \quad (3)$$

Let $\mathcal{A}_\gamma$ be a PPT adversary that attempts to challenge the indistinguishability presented in Theorem 1 and has access to an oracle $\mathcal{O}_\mathcal{A}$ that returns either ciphertexts of length α or uniformly random strings, also of length α. The objective of $\mathcal{A}$ is to gain a non-negligible advantage when distinguishing

between these responses. To that end, $\mathcal{A}_\gamma$ instantiates a Dictator $\mathcal{D}_\gamma$ and acts as its oracle.

To reach this goal, when queried by $\mathcal{D}_\gamma$, the adversary $\mathcal{A}_\gamma$ selects one of our anamorphic hash-based constructions, generates the appropriate keys, queries $\mathcal{O}_\mathcal{A}$ to generate an α-length random string, and uses it to sign a message using one of the leaves of γ. Then, it uses a classic random number to generate every other signature in the tree. Then, it forwards this tree to $\mathcal{D}_\gamma$ and repeats its output for its own query; i.e., if the Dictator indicates an anamorphic game, the adversary indicates that the random string it received from $\mathcal{O}_\mathcal{A}$ was a ciphertext. If the Dictator indicates a normal game, the adversary declares that it was given a real random string. However, we know through Theorem 1 that these strings are indistinguishable; therefore, there can not exist a Dictator capable of differentiating between a normal and an anamorphic tree-based signature scheme that uses a WOTS-like signature scheme in a leaf.

□

Corollary 1. *Assuming the existence of a one-way function. Let γ be a binary hash-based tree signature scheme whose leaves are instantiated with a Winternitz-based signature scheme that admits an anamorphic construction, λ the signature scheme security parameter, and $\mathcal{D}_b$ a PPT adversary with $\mathsf{Poly}(\lambda) \geq 2^h$. Then, a tree full of anamorphic constructions is also indistinguishable from a normal tree construction.*

Proof. Let $\mathsf{Adv}_\mathcal{D}$ be the advantage of the PPT Dictator $\mathcal{D}_b$ in detecting an anamorphic construction in the tree, and Adv_L the advantage of detecting an anamorphic construction in a given leaf. We know from the previous Theorems that $\mathsf{Adv}_\mathsf{L} \leq \epsilon(\lambda)$; then the advantage the Dictator could have in detecting an anamorphic construction in a γ tree is given by $\mathsf{Adv}_\mathcal{D} \leq 2^h \mathsf{Adv}_\mathsf{L} \leq 2^h\, \epsilon(\lambda) \leq \mathsf{Poly}(\lambda)\, \epsilon(\lambda) \leq \epsilon'(\lambda)$. That is, the sum of all the leaf advantages is also negligible for an adversary of strength $2^h \leq \mathsf{Poly}(\lambda)$. Furthermore, this property can be extended to non-binary and unbalanced trees as long as the maximum number of leaves $\mathcal{L}$ is outside the capabilities of the adversary such that $\mathsf{Poly}_A(\lambda) \geq \mathcal{L}$.

□

Finally, it is straightforward to see that existing Merkle tree signature schemes, such as XMSS and SPHINCS, also inherit the properties demonstrated in this section, since the hash-based signatures on the leaves are independent of the authentication path. Likewise, as noted in [8], we can increase the anamorphic message length by combining WOTS+ anamorphism and Merkle tree anamorphism, as they employ independent mechanisms for their respective anamorphic constructions.

E. Anamorphic Length Efficiency

While comparing anamorphic length efficiency across algorithms is not a primary focus of anamorphic cryptography research, it remains valuable to measure and discuss these differences. For WOTS, the anamorphic length efficiency varies significantly across constructions, as each exploits distinct sources of randomness within its signature schemes. We denote the maximum anamorphic message length (in bits) as ω and explore each variant below. For the padded WOTS construction, the anamorphic message length is directly proportional to the size of the padding field added to the message and can be made as small or as long as desired through direct substitution. Conversely, the checksum variant is more constrained as it is limited by the checksum field, such that ω_{wots} can be calculated via $\omega_{wots} = t_2 log_2(w)$; that is, the number of bits in the Winternitz blocks reserved for the checksum.

Conversely, the anamorphic construction for WOTS+ relies on the r parameter rather than the checksum field. Thus, ω_{wots+} is equal to the total space initially reserved for the legitimate r. Namely, our anamorphic message length $\omega_{wots+} = \lambda(w-1)$. Comparing the performances of WOTS and WOTS+ as seen in Table I and looking at the ratio of $\frac{\omega_{wots+}}{\omega_{wots}} = \frac{\lambda(w-1)}{t_2 log_2(w)}$ for some common parameters, we observe that the length constraints for WOTS are much stricter than those of WOTS+ due to the limited space of the checksum field. Nonetheless, it is imperative that we not completely dismiss WOTS, as the value of anamorphic constructions lies in their potential to be constructed, not in their intrinsic performance, since the Dictator might allow only a select set of algorithms. Furthermore, the WOTS-CS variant can mimic the performance of WOTS+ when exploiting the substitution for the r parameter, exhibiting the same message length $\omega_{sum} = \lambda(w-1)$. Finally, the checksum (or padding) anamorphism in WOTS can be used in tandem with that of WOTS+, leading to a marginal increase in anamorphic length efficiency.

Table I
COMPARISON OF MAXIMUM ANAMORPHIC MESSAGE LENGTH IN BITS FOR THE WOTS AND WOTS+ ANAMORPHIC CONSTRUCTIONS FOR λ VALUES OF 256 AND 512, AND WINTERNITZ PARAMETERS w OF 16 AND 32.

	$\lambda = 256$		$\lambda = 512$	
	$w = 16$	$w = 32$	$w = 16$	$w = 32$
ω_{wots}	12	15	12	15
ω_{wots+}	3840	7936	7680	15 872
ω_{sum}	3840	7936	7680	15 872
$\omega_{(wots,wots+)}$	3852	7951	7692	15 887
$\omega_{(wots,sum)}$	3852	7951	7692	15 887

V. CONCLUSION

In this work, we analyzed the Lamport Tagging Scheme anamorphic construction and identified that the anamorphic method used can not be directly applied to the Winternitz One-Time Signature family of signature schemes. Next, we presented the first anamorphic construction for the Winternitz One-Time Signature scheme and its variants, demonstrating that the inherent randomness of the schemes can be safely leveraged to embed covert messages in an indistinguishable manner. Subsequently, we built on our findings about WOTS anamorphism and developed two other anamorphic variants of WOTS: WOTS+ and WOTS-CS, paving the way for anamorphic hash-based signature schemes and enabling covert

communication under the heavy surveillance of a dictatorial adversary across multiple distinct methods. Furthermore, we discuss how each variant may employ different anamorphic tactics and show that these anamorphic venues can be used in tandem to increase the maximum anamorphic message length.

Future Work. While the method of swapping a random number from a pseudorandom number distribution for an anamorphic message is a straightforward way to enable anamorphism in digital signatures, it still depends on assumptions that might not hold when facing a harsher adversary, one that controls the algorithms for key and signature generation such that they can only be run "Dictator-side", preventing them from modifying their algorithms and thus, breaking most (if not all!) existing anamorphic constructions. Consequently, the search for other, more exoteric ways of anamorphic construction should be sought, veering away from the usual key generation and signature adaptations. In this direction, we point interested researchers to the Φ function introduced in WOTS-CS as a possible contender for an anamorphic construction that relies solely on the verification step of the signature process, as it might be possible to produce two distinct Φ mapping functions, one legitimate Φ used in the signature step, and one anamorphic Φ_μ that is shared between the participants that can reveal the anamorphic message.

REFERENCES

[1] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptology: Proceedings of Crypto 83*. Springer, 1984, pp. 51–67.

[2] G. Persiano, D. H. Phan, and M. Yung, "Anamorphic encryption: private communication against a dictator," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, ser. Lecture Notes in Computer Science, vol. 13276. Springer, May 2022, pp. 34–63.

[3] G. J. Simmons, "Message authentication without secrecy," in *Secure Communications and Asymmetric Cryptosystems*. Routledge, 2019, pp. 105–139.

[4] X. Chen, J. An, Z. Xiong, C. Xing, N. Zhao, F. R. Yu, and A. Nallanathan, "Covert communications: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2023.

[5] J. Buchmann, E. Dahmen, and A. Hülsing, "Xmss-a practical forward secure signature scheme based on minimal security assumptions," in *International Workshop on Post-Quantum Cryptography*. Springer, 2011, pp. 117–129.

[6] D. McGrew, M. Curcio, and S. Fluhrer, "Leighton-Micali Hash-Based Signatures," RFC 8554, Apr. 2019. [Online]. Available: https://www.rfc-editor.org/info/rfc8554

[7] J.-P. Aumasson, D. J. Bernstein, W. Beullens, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, A. Hülsing, P. Kampanakis, S. Kölbl *et al.*, "Sphincs," *Gaithersburg, MD, USA*, 2019.

[8] M. Kutyłowski, G. Persiano, D. H. Phan, M. Yung, and M. Zawada, "Anamorphic signatures: Secrecy from a dictator who only permits authentication!" in *Annual International Cryptology Conference*. Springer, 2023, pp. 759–790.

[9] D. E. Denning and D. K. Branstad, "A taxonomy for key escrow encryption systems," *Communications of the ACM*, vol. 39, no. 3, pp. 34–40, 1996.

[10] S. Micali, "Fair public-key cryptosystems," in *Annual International Cryptology Conference*. Springer, 1992, pp. 113–138.

[11] M. Green, G. Kaptchuk, and G. Van Laer, "Abuse resistant law enforcement access systems," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2021, pp. 553–583.

[12] A. Young and M. Yung, "Kleptography: Using cryptography against cryptography," in *Advances in Cryptology—EUROCRYPT'97: International Conference on the Theory and Application of Cryptographic Techniques Konstanz, Germany, May 11–15, 1997 Proceedings 16*. Springer, 1997, pp. 62–74.

[13] ——, "The prevalence of kleptographic attacks on discrete-log based cryptosystems," in *Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17*. Springer, 1997, pp. 264–276.

[14] R. L. Rivest *et al.*, "Chaffing and winnowing: Confidentiality without encryption," *CryptoBytes (RSA laboratories)*, vol. 4, no. 1, pp. 12–17, 1998.

[15] R. J. Anderson and F. A. Petitcolas, "On the limits of steganography," *IEEE Journal on selected areas in communications*, vol. 16, no. 4, pp. 474–481, 1998.

[16] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable encryption," in *Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17*. Springer, 1997, pp. 90–104.

[17] L. Von Ahn and N. J. Hopper, "Public-key steganography," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2004, pp. 323–341.

[18] A. M. Froomkin, "Metaphor is the key: cryptography, the clipper chip, and the constitution," *U. Pa. L. Rev.*, vol. 143, p. 709, 1994.

[19] F. Banfi, K. Gegier, M. Hirt, U. Maurer, and G. Rito, "Anamorphic encryption, revisited," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, ser. Lecture Notes in Computer Science, vol. 14652. Springer, May 2024, pp. 3–32.

[20] P. Błaśkiewicz, M. Kutyłowski, and A. Lauks-Dutka, "Darknet signatures," in *2023 International Conference on Data Security and Privacy Protection (DSPP)*. IEEE, Oct. 2023, pp. 232–237.

[21] D. Catalano, E. Giunta, and F. Migliaro, "Anamorphic encryption: New constructions and homomorphic realizations," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, ser. Lecture Notes in Computer Science, vol. 14652. Springer, May 2024, pp. 33–62.

[22] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," in *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, 1989, pp. 33–43.

[23] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*, 1st ed. Cambridge University Press, 2004.

[24] C. Dods, N. P. Smart, and M. Stam, "Hash Based Digital Signature Schemes," in *Cryptography and Coding 2005: Cryptography and Coding*, ser. Lecture Notes in Computer Science, vol. 3796, Dec. 2005, pp. 96–115.

[25] L. Lamport, "Constructing digital signatures from a one way function," 1979.

[26] J. W. Bos, A. Hülsing, J. Renes, and C. van Vredendaal, "Rapidly verifiable xmss signatures," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 137–168, 2021.