Computing on Authenticated Data: New Privacy Definitions and Constructions

Nuttapong Attrapadung, Benoit Libert, Thomas Peters

▶ **To cite this version:**

Nuttapong Attrapadung, Benoit Libert, Thomas Peters. Computing on Authenticated Data: New Privacy Definitions and Constructions. 2012. hal-00730665v4

HAL Id: hal-00730665

https://inria.hal.science/hal-00730665v4

Preprint submitted on 27 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computing on Authenticated Data: New Privacy Definitions and Constructions

Nuttapong Attrapadung[1] ⋆, Benoît Libert[2] ⋆⋆, and Thomas Peters[2] ⋆⋆⋆

[1] Research Institute for Secure Systems, AIST (Japan)
[2]Université catholique de Louvain, ICTEAM Institute (Belgium)

Abstract. Homomorphic signatures are primitives that allow for public computations on authenticated data. At TCC 2012, Ahn *et al.* defined a framework and security notions for such systems. For a predicate P, their notion of P-homomorphic signature makes it possible, given signatures on a message set M, to publicly derive a signature on any message m' such that $P(M, m') = 1$. Beyond unforgeability, Ahn *et al.* considered a strong notion of privacy – called strong context hiding – requiring that derived signatures be perfectly indistinguishable from signatures newly generated by the signer. In this paper, we first note that the definition of strong context hiding may not imply unlinkability properties that can be expected from homomorphic signatures in certain situations. We then suggest other definitions of privacy and discuss the relations among them. Our strongest definition, called *complete* context hiding security, is shown to imply previous ones. In the case of linearly homomorphic signatures, we only attain a slightly weaker level of privacy which is nevertheless stronger than in previous realizations in the standard model. For subset predicates, we prove that our strongest notion of privacy is satisfiable and describe a completely context hiding system with constant-size public keys. In the standard model, this construction is the first one that allows signing messages of arbitrary length. The scheme builds on techniques that are very different from those of Ahn *et al.*

Keywords. Homomorphic signatures, provable security, privacy, unlinkability, standard model.

1 Introduction

With the advent of fully homomorphic encryption [26], much attention has been paid to the problem of computing on encrypted data (see, e.g., [26, 39]) in the recent years. This also revived the interest of the research community in homomorphic signatures, which allow for computations on authenticated data.

Informally, a signer has a set of messages $\{m_i\}_{i=1}^k$ and generates a corresponding set of signatures $\{\sigma_i\}_{i=1}^k$ with $\sigma_i = \mathsf{Sign}(\mathsf{sk}, m_i)$ for each i. The signed dataset $\{(m_i, \sigma_i)\}_{i=1}^k$ is then archived on a remote server. Later on, the server can publicly compute $(m, \sigma) = \mathsf{Evaluate}(\mathsf{pk}, \{(m_i, \sigma_i)\}_{i=1}^k, f)$ such that $\mathsf{Verify}(\mathsf{pk}, m, \sigma) = 1$, where $m = f(m_1, \ldots, m_k)$ for some function f.

In the last decade, the area was investigated by several lines of research: examples include homomorphic signatures for arithmetic functions [11, 24, 12, 13] but also redactable signatures [36, 16, 17, 15] and various other forms of algebraic signatures [35, 8, 28, 29].

Recently, Ahn *et al.* [4] defined a framework for computing on signed data. For a predicate P, their notion of P-homomorphic signature allows anyone who observes signatures on a message m to publicly derive signatures on messages m' such that $P(m, m') = 1$. This framework is geared towards capturing homomorphic signatures supporting quoting and redacting, arithmetic functions

⋆ This author is supported by KAKENHI (Grant-in-Aid for Young Scientists B) No. 22700020. This work was done while the author visited ENS Paris.

⋆⋆ This author was supported by the Belgian Fund for Scientific Research (F.R.S.-F.N.R.S.) via a "Collaborateur scientifique" fellowship.

⋆⋆⋆ Supported by the Walloon Region Camus Project.

and more. Ahn *et al.* [4] gave thorough definitions for the unforgeability of P-homomorphic signatures. Besides, they introduced a strong notion of privacy, called *strong context hiding*, that captures the infeasibility of linking a derived signature to the signature it was derived from. A scheme is said strongly context hiding when a derived signature is statistically indistinguishable from a freshly generated signature, *even* when the original signature is available.

1.1 Related Work

Homomorphic signatures were first suggested by Desmedt [21] and further considered by Johnson, Molnar, Song and Wagner [34]. Boneh, Freeman, Katz and Waters [11] used them to sign vector spaces in order to prevent pollution attacks in network coding. They adapted the definitions of [34] to the network coding setting and designed a linearly homomorphic scheme in the random oracle model using bilinear maps. Gennaro, Katz, Krawczyk and Rabin subsequently described a homomorphic signature [24] over the integers based on the RSA assumption in the random oracle model. Later on, Boneh and Freeman [12] gave a linearly homomorphic construction over binary fields. They also formalized a notion, called *weak privacy*, which requires derived signatures to hide the original dataset they were derived from.

In the network coding scenario, constructions in the standard model were given by Attrapadung and Libert [5] and Catalano, Fiore and Warinschi [18, 19]. Recently, Freeman [22] defined a framework for constructing linearly homomorphic signatures satisfying enhanced security properties. In the standard model, the framework of [22] notably provides constructions based on the RSA, Diffie-Hellman and Strong Diffie-Hellman assumptions. In the meantime, Boneh and Freeman [13] used lattices to move beyond linear functions and described homomorphic signatures (in the random oracle model) supporting the evaluation of multivariate polynomials over signed data.

Recently, Ahn *et al.* [4] realized strongly context hiding P-homomorphic signatures for quoting and subset predicates: a signed message allows deriving signatures on substrings or arbitrary subsets of that message, respectively. They also showed that linearly homomorphic signatures [11, 12, 18, 22] give P-homomorphic signatures allowing for the computation of weighted averages and Fourier transforms on signed data. The construction of [11] was notably shown strongly context hiding thanks to its uniqueness of signatures property.

1.2 Our Contributions

New Definitions of Privacy. In this paper, we first reconsider the definition of strong context hiding security in [4] and point out a subtlety that arises in the context of randomizable signatures. While the definition of Ahn *et al.* [4] aims at perfect indistinguishability, it only considers honestly generated original signatures. In specific schemes, signatures may satisfy the verification algorithm without being produced by the legitimate signing algorithm. Signatures [32, 5, 25] derived from Waters' dual system encryption technique [41] – which is currently the only known way to prove the standard unforgeability property for certain predicates – are typical examples. For these constructions, the definition of [4] does not guarantee the unlinkability when the original signature is adversarially chosen (e.g., by re-randomizing original signatures). This may be a concern in certain applications. In network coding, suppose that we want to hide the path taken by specific packets. If a curious target node colludes with some intermediate nodes that maliciously re-randomize signatures on the road, they may infer information on the rest of the path downstream.

To address this issue, we suggest other definitions of unlinkability and discuss the relations

among them. We first define a security property, called *adaptive context hiding*, that allows for adversarially-generated original signatures. Since this definition only asks for computational security, it does not imply strong context hiding security [4]: we show examples of schemes that are context hiding according to one definition and fall short of satisfying the other one. In order to unify these definitions, we thus define a notion of *completely context hiding* homomorphic signature, which requires statistical unlinkability and implies *both* strong and adaptive context hiding properties.

New Linearly Homomorphic Signatures. Using the dual system technique [41, 32], we describe a new linearly homomorphic signature and prove it (in the standard model) both strongly context hiding and context hiding on adversarially-chosen signatures with private key exposure. To our knowledge, all previous such schemes fail to simultaneously satisfy both security notions. The scheme of [5] is actually the only strongly context hiding realization in the standard model but, as we shall see, it is provably not adaptively context hiding. Since the new construction is only adaptively context hiding for computationally bounded distinguishers, it does not meet our strongest definition. This shortcoming seems inherent to all signature schemes [5, 25] based on the dual system paradigm. We leave it as an open problem to achieve information-theoretic unlinkability in that sense without resorting to the random oracle model.

If we settle for weak context hiding security[1] (as in most linearly homomorphic signatures [12, 22]), a variant of our scheme provides the shortest linearly homomorphic signature based on a simple assumption in the standard model. At the expense of being context hiding in a weaker sense than [11], the scheme can be proved unforgeable under the standard computational Diffie-Hellman (CDH) assumption. Each signature consists of two group elements and one scalar, which shortens Freeman's CDH-based signatures [22] by about 25%.

Handling Subset Predicates for Messages of Arbitrary Length. Finally, the paper puts forward a new method for dealing with subset predicates. In [34], Johnson *et al.* described set homomorphic signatures supporting both union and subset operations in the random oracle model. Ahn *et al.* [4] showed how to obtain such signatures from a certain class of ciphertext-policy attribute-based encryption (CP-ABE) systems, by applying a Naor-like transformation [10]. With currently available fully secure CP-ABE schemes [31, 37], this technique is limited to support messages of bounded length: the maximal length n_{max} of original messages must be fixed at key generation time and public keys comprise at least $O(n_{max})$ group elements. This limitation could be avoided using a fully secure unbounded [33] CP-ABE scheme. However, no such system is currently available: the only known [33, 30] unbounded ABE constructions to date are selectively secure key-policy ABE schemes.

To fill this gap, we suggest an alternative design principle which yields constant-size public keys and allows signing messages of arbitrary length in the standard model. Our construction departs from the ABE-based approach of [4] and rather uses the randomizability properties of Groth-Sahai proofs [27]. In a nutshell, when original signatures are computed for a set of words $\{m_1, \ldots, m_n\}$, the signer generates a fresh public key pk', which is certified using the long-term secret key of the system, and uses sk' to compute $\sigma_i = \mathsf{Sign}(sk', m_i)$ for each i. This construction is made unlinkable by letting pk' and all signatures $\{\sigma_i\}_{i=1}^n$ appear in committed form, accompanied with non-interactive witness indistinguishable proofs of their validity. The general idea is instantiated

[1] This property relaxes strong context hiding security by only requiring the indistinguishability when the original signatures are not given.

by combining the structure-preserving signature of [1] with Waters signatures [40] – which are both partially randomizable – in such a way that we only need to manipulate linear pairing product equations (in the terminology of [27]). This makes it easy to re-randomize Groth-Sahai proofs when deriving signatures. As a result, the system provably satisfies our strongest definition of unlinkability.

We believe this approach to be of interest in its own right for the design of P-homomorphic signatures. Indeed, if we compare it with the dual system technique [41], it allows us to more easily obtain completely context hiding schemes.

1.3 Organization

We first review previous security definitions for P-homomorphic signatures and introduce new definitions of privacy in Section 2.1. Section 3 discusses the relations among these privacy definitions. In Section 4, we describe a new linearly homomorphic constructions, for which a CDH-based weakly context-hiding variant is described in Appendix F. Section 5 finally presents our completely context hiding system for subset predicates.

2 Background

2.1 Definitions for Homomorphic Signatures

Definition 1 ([4]). *Let $\mathcal{M}$ be a message space and $2^{\mathcal{M}}$ be its powerset. Let $P : 2^{\mathcal{M}} \times \mathcal{M} \to \{0,1\}$ be a predicate. A message m' is said* **derivable** *from $M \subset \mathcal{M}$ if $P(M, m') = 1$. As in [4], $P^i(M)$ is the set of messages derivable from $P^{i-1}(M)$, where $P^0(M) := \{m' \in \mathcal{M} \mid P(M, m') = 1\}$. Finally, $P^*(M) := \cup_{i=0}^{\infty} P^i(M)$ denotes the set of messages derivable from M by iterated derivation.*

Definition 2 ([4]). *A P-homomorphic signature for a predicate $P : 2^{\mathcal{M}} \times \mathcal{M} \to \{0,1\}$ is a triple of algorithms $(\mathsf{Keygen}, \mathsf{SignDerive}, \mathsf{Verify})$ such that:*

$\mathsf{Keygen}(\lambda)$: *takes as input a security parameter $\lambda \in \mathbb{N}$ and outputs a key pair $(\mathsf{sk}, \mathsf{pk})$. As in [4], the private key sk is seen as a signature on the empty tuple $\varepsilon \in \mathcal{M}$.*

$\mathsf{SignDerive}\big(\mathsf{pk}, (\{\sigma_m\}_{m\in M}, M), m'\big)$: *is a possibly randomized algorithm that takes as input a public key pk, a set of messages $M \subset \mathcal{M}$, a corresponding set of signatures $\{\sigma_m\}_{m\in M}$ and a derived message $m' \in \mathcal{M}$. If $P(M, m') = 0$, it returns $\perp$. Otherwise, it outputs a derived signature σ'*

$\mathsf{Verify}(\mathsf{pk}, \sigma, m)$: *is a deterministic algorithm that takes as input a public key pk, a signature σ and a message m. It outputs 0 or 1.*

Note that the empty tuple $\varepsilon \in \mathcal{M}$ satisfies $P(\varepsilon, m) = 1$ for each $m \in \mathcal{M}$. Like [4], we define the algorithm $\mathsf{Sign}(\mathsf{pk}, \mathsf{sk}, m)$ that runs $\mathsf{SignDerive}(\mathsf{pk}, (\mathsf{sk}, \varepsilon), m)$ and returns the resulting output. For any set $M = \{m_1, \ldots, m_k\} \subset \mathcal{M}$, we define $\mathsf{Sign}(\mathsf{sk}, M) := \{\mathsf{Sign}(\mathsf{sk}, m_1), \ldots, \mathsf{Sign}(\mathsf{sk}, m_k)\}$. Also, $\mathsf{Verify}(\mathsf{pk}, M, \{\sigma_m\}_{m\in M}) = 1$ means that $\mathsf{Verify}(\mathsf{pk}, m, \sigma_m) = 1$ for each $m \in M$.

Correctness. It is mandated that, for all pairs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(\lambda)$, for any set $M \subset \mathcal{M}$, any message $m' \in \mathcal{M}$ such that $P(M, m') = 1$, then, we have

- $\mathsf{SignDerive}(\mathsf{pk}, (\mathsf{Sign}(\mathsf{sk}, M), M), m') \neq \perp$.
- $\mathsf{Verify}\big(\mathsf{pk}, m', \mathsf{SignDerive}(\mathsf{pk}, (\mathsf{Sign}(\mathsf{sk}, M), M), m')\big) = 1$.

Definition 3 ([4]). *A P-homomorphic signature* $(\mathsf{Keygen}, \mathsf{SignDerive}, \mathsf{Verify})$ *is* **unforgeable** *if no probabilistic polynomial-time (PPT) adversary has non-negligible advantage in this game:*

1. *The challenger generates* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(\lambda)$ *and gives* pk *to the adversary $\mathcal{A}$. It initializes two initially empty tables T and Q.*
2. *$\mathcal{A}$ adaptively interleaves the following queries.*
   - *Signing queries: $\mathcal{A}$ chooses a message $m \in \mathcal{M}$. The challenger replies by choosing a handle h, runs $\sigma \leftarrow \mathsf{Sign}(sk, m)$ and stores (h, m, σ) in a table T. The handle h is returned to $\mathcal{A}$.*
   - *Derivation queries: $\mathcal{A}$ chooses a vector of handles $\vec{h} = (h_1, \ldots, h_k)$ and a message $m' \in \mathcal{M}$. The challenger retrieves the tuples $\{(h_i, m_i, \sigma_i)\}_{i=1}^{k}$ from T and returns $\perp$ if one of these does not exist. Otherwise, it defines $M := (m_1, \ldots, m_k)$ and $\{\sigma_m\}_{m \in M} = \{\sigma_1, \ldots, \sigma_k\}$. If $P(M, m') = 1$, the challenger runs $\sigma' \leftarrow \mathsf{SignDerive}\big(\mathsf{pk}, (\{\sigma_m\}_{m \in M}, M), m'\big)$, chooses a handle h', stores (h', m', σ') in T and returns h' to $\mathcal{A}$.*
   - *Reveal queries: $\mathcal{A}$ chooses a handle h. If no tuple of the form (h, m', σ') exists in T, the challenger returns $\perp$. Otherwise, it returns σ' to $\mathcal{A}$ and adds (m', σ') to the set Q.*
3. *$\mathcal{A}$ outputs a pair (σ', m') and wins if the following conditions hold.*
   - $\mathsf{Verify}(\mathsf{pk}, m', \sigma') = 1$.
   - *If $M \subset \mathcal{M}$ is the set of messages in Q, then $m' \notin P^*(M)$.*

Definition 4 ([4]). *A homomorphic signature* $(\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ *is* **strongly context hiding** *for the predicate P if, for all key pairs* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(\lambda)$*, for all messages $M \subset \mathcal{M}^*$ and $m' \in \mathcal{M}$ such that $P(M, m') = 1$, the following two distributions are statistically close:*

$$\big\{(\mathsf{sk}, \{\sigma_m\}_{m \in M} \leftarrow \mathsf{Sign}(\mathsf{sk}, M),\ \mathsf{Sign}(sk, m'))\big\}_{sk, M, m'},$$
$$\big\{\big(\mathsf{sk}, \{\sigma_m\}_{m \in M} \leftarrow \mathsf{Sign}(\mathsf{sk}, M),\ \mathsf{SignDerive}\big(\mathsf{pk}, (\{\sigma_m\}_{m \in M}, M), m'\big)\big)\big\}_{sk, M, m'}.$$

In [4] Ahn *et al.* showed that, if a scheme is strongly context hiding, then Definition 3 can be simplified by removing the $\mathsf{SignDerive}$ and Reveal oracles and only providing the adversary with an ordinary signing oracle.

As we will see, specific constructions leave a gap between signatures accepted by the verification algorithm and those generated by the original signing procedure. For these schemes, a stronger definition than Definition 4 may be necessary in some situations.

To illustrate this, we first give an alternative definition which is almost identical to the computational security definition of [4][Appendix A]: the only difference is that, in the challenge phase, one of the signatures is supplied by the adversary instead of being honestly generated by the challenger. This modification is motivated by re-randomizable signatures. It allows for adversaries who attempt to re-randomize one of the signatures obtained from the oracle in order to embed some subliminal information that would help them win the game.

Definition 5. *A P-homomorphic signature* $(\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ *is* **weakly adaptively context hiding** *if no PPT adversary has non-negligible advantage in the following game:*

1. *The challenger runs* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Keygen}(\lambda)$ *and gives* pk *to the adversary.*
2. *The adversary $\mathcal{A}$ adaptively interleaves queries exactly as in Definition 3.*

3. *The adversary $\mathcal{A}$ chooses a message set $M \subset \mathcal{M}$ together with a set of signatures $\{\sigma_m\}_{m\in M}$ as well as another message $m' \in \mathcal{M}$. If $P(M, m') = 0$ or $\mathsf{Verify}(\mathsf{pk}, M, \{\sigma_m\}_{m\in M}) = 0$, return $\perp$. Otherwise, the challenger flips a fair binary coin $\beta \xleftarrow{R} \{0,1\}$. If $\beta = 0$, it computes a derived signature $\sigma^\star = \mathsf{SignDerive}\big(\mathsf{pk}, (\{\sigma_m\}_{m\in M}, M), m'\big)$. If $\beta = 1$, it computes $\sigma^\star = \mathsf{Sign}(sk, m')$. In either case, $\sigma^\star$ is sent as a challenge to $\mathcal{A}$.*
4. *$\mathcal{A}$ is allowed to make another series of queries as in stage 2.*
5. *Eventually, $\mathcal{A}$ outputs a bit $\beta' \in \{0,1\}$ and wins if $\beta' = \beta$. As usual, $\mathcal{A}$'s advantage is defined to be $\mathbf{Adv}(\mathcal{A}) = |\Pr[\beta' = \beta] - 1/2|$.*

The latter definition can be seen as an analogue of a definition of unlinkability given by Prabhakaran and Rosulek [38] for homomorphic encryption: both models account for adversarially-chosen original signatures or ciphertexts.

We will see that Definitions 4 and 5 do not imply each other. While incomparable, we believe that they both make sense in practice. For example, when it comes to conceal the path followed by packets in network coding signatures, Definition 5 ensures that each node only learns the last node visited by incoming packets, even if it colludes with another node far upstream.

Towards unifying previous definitions, we now simplify Definition 5 as follows. Instead of providing the adversary $\mathcal{A}$ with a signing oracle, $\mathcal{A}$ is directly given the private key at the beginning.

Definition 6. *A P-homomorphic signature is* **adaptively context hiding** *if no PPT adversary has non-negligible advantage in the following game:*

1. *The challenger runs $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Keygen}(\lambda)$ and hands $(\mathsf{sk}, \mathsf{pk})$ to $\mathcal{A}$.*
2. *The adversary $\mathcal{A}$ chooses a message set $M \subset \mathcal{M}$ together with a set of signatures $\{\sigma_m\}_{m\in M}$ as well as another message $m' \in \mathcal{M}$. If $P(M, m') = 0$ or $\mathsf{Verify}(\mathsf{pk}, M, \{\sigma_m\}_{m\in M}) = 0$, return $\perp$. Otherwise, the challenger flips a fair binary coin $\beta \xleftarrow{R} \{0,1\}$. If $\beta = 0$, it computes a derived signature $\sigma^\star = \mathsf{SignDerive}\big(\mathsf{pk}, (\{\sigma_m\}_{m\in M}, M), m'\big)$. If $\beta = 1$, it computes $\sigma^\star = \mathsf{Sign}(sk, m')$. In either case, $\sigma^\star$ is sent as a challenge to $\mathcal{A}$.*
3. *Eventually, $\mathcal{A}$ outputs a bit $\beta' \in \{0,1\}$ and wins if $\beta' = \beta$. As usual, $\mathcal{A}$'s advantage is defined to be $\mathbf{Adv}(\mathcal{A}) = |\Pr[\beta' = \beta] - 1/2|$.*

While the latter definition seems sufficient for many applications, it still does not imply Definition 4 and we may want signatures to be unlinkable in the statistical sense. The resulting stronger definition implies both Definition 6 and Definition 4 and goes as follows.

Definition 7. *A P-homomorphic signature* $(\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ *is* **completely context hiding** *if, for all pairs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(\lambda)$, all messages $M \subset \mathcal{M}^*$ and $m' \in \mathcal{M}$ such that $P(M, m') = 1$, for all $\{\sigma_m\}_{m\in M}$ such that $\mathsf{Verify}(\mathsf{pk}, M, \{\sigma_m\}_{m\in M}) = 1$, the following distributions are statistically close*

$$\Big\{\big(\mathsf{sk},\ \{\sigma_m\}_{m\in M},\ \mathsf{Sign}(sk, m')\big)\Big\}_{\mathsf{sk},M,m'},$$
$$\Big\{\big(\mathsf{sk},\ \{\sigma_m\}_{m\in M},\ \mathsf{SignDerive}\big(\mathsf{pk}, (\{\sigma_m\}_{m\in M}, M), m'\big)\big)\Big\}_{\mathsf{sk},M,m'}.$$

In all schemes based on the dual system approach [5, 25], the existence of an alternative distribution of acceptable signatures makes it seemingly impossible to satisfy the above definition. In these schemes, the combination of strong (*i.e.*, Definition 4) and adaptive context hiding security thus appears as the best we can hope for. For this reason, we chose to present Definition 6 first instead

of directly working with Definition 7.

Definition 7 assumes honestly generated keys $(\mathsf{sk}, \mathsf{pk})$. It can be strengthened by allowing the adversary to generate a pair $(\mathsf{sk}, \mathsf{pk})$ of its own. In the random oracle model, the construction of [11] is easily seen to satisfy such a stronger definition (if we assume that all public keys live in a cyclic group which is part of common public parameters) because it has unique signatures. In the standard model, we do not know of any scheme that would be secure in that sense.

In the following, we can satisfy Definition 7 with our homomorphic signature for subset predicates. In the case of linearly homomorphic signatures, we are only able to meet Definition 6.

2.2 Complexity Assumptions

We consider groups $(\mathbb{G}, \mathbb{G}_T)$ of composite order $N = p_1p_2p_3$, for which a bilinear map $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T$ is computable. For each $i \in \{1, 2, 3\}$, we denote by $\mathbb{G}_{p_i}$ the subgroup of order p_i. Also, for all distinct i, j, we call $\mathbb{G}_{p_ip_j}$ the subgroup of order p_ip_j. An important property of composite order groups is that pairing two elements of order p_i and p_j, with $i \neq j$, always gives the identity element $1_{\mathbb{G}_T}$.

In these groups, we rely on the following assumptions introduced in [32].

Assumption 1 Given $g \xleftarrow{R} \mathbb{G}_{p_1}, X_3 \xleftarrow{R} \mathbb{G}_{p_3}$, and T, it is infeasible to efficiently decide if $T \in_R \mathbb{G}_{p_1p_2}$ or $T \in_R \mathbb{G}_{p_1}$.

Assumption 2 Let $g, X_1 \xleftarrow{R} \mathbb{G}_{p_1}, X_2, Y_2 \xleftarrow{R} \mathbb{G}_{p_2}, Y_3, Z_3 \xleftarrow{R} \mathbb{G}_{p_3}$. Given a tuple (g, X_1X_2, Z_3, Y_2Y_3) and T, it is hard to decide if $T \in_R \mathbb{G}$ or $T \in_R \mathbb{G}_{p_1p_3}$.

Assumption 3 Let elements $g, w, g^t, X_1 \xleftarrow{R} \mathbb{G}_{p_1}$ with $t \xleftarrow{R} \mathbb{Z}_N, X_2, Y_2, Z_2 \xleftarrow{R} \mathbb{G}_{p_2}, X_3, Y_3, Z_3 \xleftarrow{R} \mathbb{G}_{p_3}$. Given $(g, w, g^t, X_1X_2, X_3, Y_2Y_3)$, and $T \in \mathbb{G}$, decide if $T = w^tZ_3$ or $T = w^tZ_2Z_3$.

Assumption 4 Let $g \xleftarrow{R} \mathbb{G}_{p_1}$, $X_2, Y_2, Z_2 \xleftarrow{R} \mathbb{G}_{p_2}$, $X_3 \xleftarrow{R} \mathbb{G}_{p_3}$ and $a, b, c \xleftarrow{R} \mathbb{Z}_N$. Given elements $(g, g^a, g^b, g^{ab}X_2, X_3, g^cY_2, Z_2)$, it is infeasible to compute $e(g, g)^{abc}$.

We also use bilinear maps $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ over groups of prime order p. In these groups, we rely on the following hardness assumptions.

Definition 8 ([9]). *The* **Decision Linear Problem** *(DLIN) in $\mathbb{G}$, is to distinguish the distributions $(g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ and $(g^a, g^b, g^{ac}, g^{bd}, g^z)$, where $a, b, c, d \xleftarrow{R} \mathbb{Z}_p^*$, $z \xleftarrow{R} \mathbb{Z}_p^*$. The* **Decision Linear Assumption** *is the intractability of DLIN for any PPT distinguisher $\mathcal{D}$.*

Definition 9 ([1]). *In a group $\mathbb{G}$, the* **q-Simultaneous Flexible Pairing Problem** *(q-SFP) is, given $\big(g_z,\ h_z,\ g_r,\ h_r,\ a,\ \tilde{a},\ b,\ \tilde{b} \in \mathbb{G}\big)$ and q tuples $(z_j, r_j, s_j, t_j, u_j, v_j, w_j) \in \mathbb{G}^7$ such that*

$$e(a, \tilde{a}) = e(g_z, z_j) \cdot e(g_r, r_j) \cdot e(s_j, t_j), \qquad e(b, \tilde{b}) = e(h_z, z_j) \cdot e(h_r, u_j) \cdot e(v_j, w_j), \tag{1}$$

to find a new tuple $(z^\star, r^\star, s^\star, t^\star, u^\star, v^\star, w^\star) \in \mathbb{G}^7$ satisfying (1) and such that $z^\star \notin \{1_{\mathbb{G}}, z_1, \ldots, z_q\}$.

2.3 Structure-Preserving Signatures

Privacy-preserving protocols often require to sign elements of bilinear groups as if they were ordinary messages. Abe, Haralambiev and Ohkubo [1, 2] (AHO) described such an efficient structure-preserving signature. The description hereunder assumes public parameters $\mathsf{pp} = \big((\mathbb{G}, \mathbb{G}_T),\ g\big)$ consisting of bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, where $\lambda \in \mathbb{N}$ and a generator $g \in \mathbb{G}$.

Keygen(pp, n)**:** given an upper bound $n \in \mathbb{N}$ on the number of group elements per signed message, choose generators $G_r, H_r \xleftarrow{R} \mathbb{G}$. Pick $\gamma_z, \delta_z \xleftarrow{R} \mathbb{Z}_p$ and $\gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$, for $i = 1$ to n. Then, compute $G_z = G_r^{\gamma_z}$, $H_z = H_r^{\delta_z}$ and $G_i = G_r^{\gamma_i}$, $H_i = H_r^{\delta_i}$ for each $i \in \{1, \ldots, n\}$. Finally, choose $\alpha_a, \alpha_b \xleftarrow{R} \mathbb{Z}_p$ and define $A = e(G_r, g^{\alpha_a})$ and $B = e(H_r, g^{\alpha_b})$. The public key is defined to be

$$pk = \big(G_r,\ H_r,\ G_z,\ H_z,\ \{G_i, H_i\}_{i=1}^n,\ A,\ B\big) \in \mathbb{G}^{2n+4} \times \mathbb{G}_T^2$$

while the private key is $sk = \big(\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n\big)$.

Sign$(sk, (M_1, \ldots, M_n))$**:** to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using sk, choose $\zeta, \rho_a, \rho_b, \omega_a, \omega_b \xleftarrow{R} \mathbb{Z}_p$ and compute $\theta_1 = g^{\zeta}$ as well as

$$\theta_2 = g^{\rho_a - \gamma_z \zeta} \cdot \prod_{i=1}^{n} M_i^{-\gamma_i}, \qquad \theta_3 = G_r^{\omega_a}, \qquad \theta_4 = g^{(\alpha_a - \rho_a)/\omega_a},$$
$$\theta_5 = g^{\rho_b - \delta_z \zeta} \cdot \prod_{i=1}^{n} M_i^{-\delta_i}, \qquad \theta_6 = H_r^{\omega_b}, \qquad \theta_7 = g^{(\alpha_b - \rho_b)/\omega_b},$$

The signature consists of $\sigma = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) \in \mathbb{G}^7$.

Verify$(pk, \sigma, (M_1, \ldots, M_n))$**:** given $\sigma = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7)$, return 1 iff these equalities hold:

$$A = e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(\theta_3, \theta_4) \cdot \prod_{i=1}^{n} e(G_i, M_i),$$
$$B = e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(\theta_6, \theta_7) \cdot \prod_{i=1}^{n} e(H_i, M_i).$$

The scheme was proved [1, 2] existentially unforgeable under chosen-message attacks under the q-SFP assumption, where q is the number of signing queries.

As showed in [1, 2], signature components $\{\theta_i\}_{i=2}^7$ can be publicly randomized to obtain a different signature $\{\theta'_i\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk, \sigma)$ on $(M_1, \ldots, M_n)$. After randomization, we have $\theta'_1 = \theta_1$ while $\{\theta'_i\}_{i=2}^7$ are uniformly distributed among the values $(\theta_2, \ldots, \theta_7)$ such that the equalities $e(G_r, \theta'_2) \cdot e(\theta'_3, \theta'_4) = e(G_r, \theta_2) \cdot e(\theta_3, \theta_4)$ and $e(H_r, \theta'_5) \cdot e(\theta'_6, \theta'_7) = e(H_r, \theta_5) \cdot e(\theta_6, \theta_7)$ hold. This re-randomization is performed by choosing $\varrho_2, \varrho_5, \mu, \nu \xleftarrow{R} \mathbb{Z}_p$ and computing

$$\begin{aligned} \theta'_2 &= \theta_2 \cdot \theta_4^{\varrho_2}, & \theta'_3 &= (\theta_3 \cdot G_r^{-\varrho_2})^{1/\mu}, & \theta'_4 &= \theta_4^{\mu} \\ \theta'_5 &= \theta_5 \cdot \theta_7^{\varrho_5}, & \theta'_6 &= (\theta_6 \cdot H_r^{-\varrho_5})^{1/\nu}, & \theta'_7 &= \theta_7^{\nu}. \end{aligned} \tag{2}$$

As a result, $\{\theta'_i\}_{i \in \{3,4,6,7\}}$ are statistically independent of the message and other signature components. This implies that, in privacy-preserving protocols, re-randomized $\{\theta'_i\}_{i \in \{3,4,6,7\}}$ can be safely given in the clear as long as $(M_1, \ldots, M_n)$ and $\{\theta'_i\}_{i \in \{1,2,5\}}$ are given in committed form.

3 Separation Results

Separating Definitions 4 and 5. Let us consider the following variant[2] of the construction in [5], which relies on the Lewko-Waters signatures [32] and bilinear groups whose order is a product $N = p_1 p_2 p_3$ of three primes. If n denotes the dimension of signed vectors, the public key is

[2] This variant is obtained by applying Freeman's framework [22] to Lewko-Waters signatures [33], which guarantees its unforgeability.

$\mathsf{pk} = \big(g, e(g,g)^\alpha, u, v, \{h_i\}_{i=1}^n, X_3\big)$, where $\alpha \in_R \mathbb{Z}_N$, $g, u, v, h_1, \ldots, h_n \in \mathbb{G}_{p_1}$, $X_3 \in \mathbb{G}_{p_3}$ and the private key consists of $\mathsf{sk} = (g^\alpha, \kappa)$, where κ is the seed of a pseudorandom function. The latter is used to de-randomize the scheme and make sure that all vectors of the same file will be signed using partially identical random coins.

To sign a vector $\vec{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_N^n$ using the file identifier τ, the signer computes a pseudo-random $r = \Psi(\kappa, \tau) \in \mathbb{Z}_N$ which is used to compute

$$(\sigma_1, \sigma_2, \sigma_3) = \Big(g^\alpha \cdot (u^\tau \cdot v)^r \cdot R_3,\ g^r \cdot R_3',\ (\prod_{i=1}^n h_i^{v_i})^r \cdot R_3''\Big),$$

with $R_3, R_3', R_3'' \stackrel{R}{\leftarrow} \mathbb{G}_{p_3}$. The homomorphic property follows from the fact that all vectors of the same dataset are signed using the same $r \in \mathbb{Z}_N$. The homomorphic evaluation algorithm proceeds in the obvious way and combines signatures $\{(\sigma_{i,1}, \sigma_{i,2}, \sigma_{i,3})\}_{i=1}^\ell$ by linearly combining the $\{\sigma_{i,3}\}_{i=1}^\ell$ and re-randomizing the $\mathbb{G}_{p_3}$ components. Note that the underlying exponent r is not re-randomized, so that all $\{(\sigma_{i,1}, \sigma_{i,2})\}_{i=1}^\ell$ share the same $\mathbb{G}_{p_1}$ components.

It is easy to see that the construction is strongly context hiding in the sense of Definition 4. Indeed, the signing algorithm is honestly run in the first distribution of Definition 4. This implies that, for any message set $M = \{(\tau, \vec{v}_1), \ldots, (\tau, \vec{v}_k)\} \subset \mathcal{M}$, the underlying $\log_g(\sigma_2)$ will have the same value no matter if the second signature (σ_1, σ_2) is produced by Sign or SignDerive.

However, the scheme does not satisfy Definition 5. Indeed, in step 2, the adversary can first invoke the signing oracle on k occasions to obtain signatures for some set $M = \{(\tau, \vec{v}_1), \ldots, (\tau, \vec{v}_k)\}$ of its choice. If we denote by $\{\sigma_m\}_{m \in M}$ the resulting signatures, the adversary re-randomizes $\{\sigma_m\}_{m \in M}$ in such a way that each randomized σ_m is of the form $\big(g^\alpha \cdot (u^\tau \cdot v)^{r'} \cdot \tilde{R}_3,\ g^{r'} \cdot \tilde{R}_3',\ (\prod_{i=1}^n h_i^{v_i})^{r'} \cdot \tilde{R}_3''\big)$, for some fresh $r' \in_R \mathbb{Z}_N$. The adversary $\mathcal{A}$ can then choose a random message $m' \in \mathcal{M}$ such that $P(M, m') = 1$ and send $\big((M, \{\sigma_m'\}_{m \in M}), m'\big)$ to the challenger. The latter returns a challenge signature $\sigma^\star = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star)$ on m' and $\mathcal{A}$ can immediately figure out if $\sigma^\star$ is fresh or derived, by testing if $e(\sigma_2^\star, g) = e(\sigma_{m,2}, g)$. With overwhelming probability, the latter equality only holds if $\beta = 0$.

Separating Definitions 5 and 6. The original construction of [5] works exactly like the scheme outlined in the previous paragraph with the difference that it prevents public randomizations of the $\mathbb{G}_{p_1}$ components of signatures $(\sigma_1, \sigma_2, \sigma_3)$. More precisely, the scheme makes use of an additional collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_N$. If the file identifier is τ, a vector $\vec{v} = (v_1, \ldots, v_n)$ is signed by computing $r = \Psi(\kappa, \tau) \in \mathbb{Z}_N$, $\tau' = H(\tau, e(g,g)^r)$ and returning

$$(\sigma_1, \sigma_2, \sigma_3) = \Big(g^\alpha \cdot (u^{\tau'} \cdot v)^r \cdot R_3,\ g^r \cdot R_3',\ (\prod_{i=1}^n h_i^{v_i})^r \cdot R_3''\Big),$$

with $R_3, R_3', R_3'' \stackrel{R}{\leftarrow} \mathbb{G}_{p_3}$. The security proof of [5] implies that, if the adversary is given signatures $\{(\sigma_{i,1}, \sigma_{i,2}, \sigma_{i,3})\}_{i=1}^\ell$ on messages $(\tau, \vec{v}_1), \ldots, (\tau, \vec{v}_\ell)$, the adversary cannot generate a signature $(\sigma_1, \sigma_2, \sigma_3)$ on $(\tau, \vec{y})$ such that $e(\sigma_2, g) \neq e(\sigma_{i,2}, g)$ for each i. Essentially, since $(\sigma_{i,1}, \sigma_{i,2})$ can be seen as a Lewko-Waters signature on the message $H(\tau, e(g,g)^r)$, any valid signature $(\sigma_1, \sigma_2, \sigma_3)$ for which $e(\sigma_{i,2}, g) \neq e(\sigma_2, g)$ implies either an attack against the signature scheme of [32] or a breach in the collision-resistance of H.

Let us consider an adversary in the sense of Definition 5. Since signatures cannot be publicly randomized, when the adversary enters the challenge phase in step 3, it can only choose a message set $M = \{(\tau, \vec{v}_1), \ldots, (\tau, \vec{v}_\ell)\}$ and signatures $\{(\sigma_{m,1}, \sigma_{m,2}, \sigma_{m,3})\}_{m \in M}$ for which $\{e(\sigma_{m,2}, g)\}_{m \in M}$

has the same value as in signatures obtained from the signing oracle at step 2. Therefore, the only way for $\mathcal{A}$ to have non-negligible advantage in the game of Definition 5 is to choose $(M, \{\sigma_m\}_{m\in M})$ where $\{\sigma_m\}_{m\in M}$ is obtained by introducing a $\mathbb{G}_{p_2}$ component in a signature obtained from the signing oracle. Otherwise, the distribution of the challenge signature $(\sigma_1^\star, \sigma_2^\star, \sigma_3^\star)$ does not depend on $\beta \in \{0,1\}$ in step 3. Using the same arguments as in the proof of Theorem 1, we can prove that Assumption 1 can be broken if $\mathcal{A}$ can output a set $\{\sigma_m\}_{m\in M}$ where one of the signatures contains a $\mathbb{G}_{p_2}$ component. If H is collision-resistant and under the assumptions used in [5], the scheme is thus weakly adaptively context hiding.

Now, we easily observe that the original scheme of [5] is not adaptively context hiding in the sense of Definition 6. Recall that the adversary is given the private key $\mathsf{sk} = (g^\alpha, \kappa)$ at the beginning of the game. In the challenge phase, it can thus choose a message set $M \subset \mathcal{M}$ and signatures $\{\sigma_m\}_{m\in M}$ for which each σ_m is of the form $(\sigma_{m,1}, \sigma_{m,2}, \sigma_{m,3}) = \big(g^\alpha \cdot (u^{\tau'} \cdot v)^{r'} \cdot R_3,\ g^{r'} \cdot R_3',\ (\prod_{i=1}^n h_i^{v_i})^{r'} \cdot R_3''\big)$, with $R_3, R_3', R_3'' \in_R \mathbb{G}_{p_3}$, and for some random $r' \in_R \mathbb{Z}_N \backslash \{\Psi(\kappa, \tau)\}$. When receiving $(M, \{\sigma_m\}_{m\in M})$ and m' such that $P(M, m') = 1$, the challenger runs $\mathsf{SignDerive}$ on $\{\sigma_m\}_{m\in M}$ if $\beta = 0$. If $\beta = 1$, it ignores $\{\sigma_m\}_{m\in M}$ and simply generates a fresh signature on m'. In the letter case, the challenge signature $(\sigma_1^\star, \sigma_2^\star, \sigma_3^\star)$ is such that $\log_g(\sigma_2^\star) = \Psi(\kappa, \tau) \bmod p_1$ and, since the adversary knows κ, it can easily test whether $e(\sigma_2^\star, g) = e(g, g)^{\Psi(\kappa,\tau)}$ and, if so, return $\beta' = 1$.

Later on, we will see an example of scheme that satisfies Definition 6 but fails to be secure as per Definition 4. The two definitions are thus incomparable.

4 An Adaptively Context Hiding Linearly Homomorphic Scheme in the Standard Model

So far, the scheme of [5] is seemingly the only linearly homomorphic signature in the standard model to satisfy Definition 4. This section presents a linearly homomorphic signature satisfying both Definition 4 and the adaptive context hiding property captured by Definition 6.

The scheme works over groups whose order is a product $N = p_1p_2p_3$ of three primes. Like [5], it builds on Lewko-Waters signatures, where public keys contain $\big(g, e(g,g)^\alpha, u, v\big)$, with $g, u, v \in \mathbb{G}_{p_1}$ and $\alpha \in \mathbb{Z}_N$, and a signature on m consists of $(g^\alpha \cdot (u^m \cdot v)^r \cdot R_3, g^r \cdot R_3')$, for some $R_3, R_3' \in \mathbb{G}_{p_3}$. A difference with [5] is that $e(g,g)^\alpha$ is replaced by g^α in the public key and signatures are obtained by aggregating a Lewko-Waters signature on the file identifier τ and a signed vector hash $(\prod_{i=1}^n g_i^{v_i})^\alpha$ of the vector $\vec{v} = (v_1, \ldots, v_n)$, where $(g_1, \ldots, g_n) \in \mathbb{G}_{p_1}^n$ is part of the public key. We note that $(\prod_{i=1}^n g_i^{v_i})^\alpha$ is not a secure homomorphic signature in general: it can actually be seen as a one-time linearly homomorphic signature where only one message set $M = \{(\tau, \vec{v}_1), \ldots, (\tau, \vec{v}_k)\}$ can be signed. Nevertheless, we will show that aggregating the two components actually provides unforgeability. Moreover, beyond providing a stronger flavor of privacy than [5], it also shortens signatures by 33%.

For simplicity, the scheme is described in terms of composite order groups. It is very plausible that Lewko's techniques [30] apply to translate the scheme in the prime order setting.

4.1 Construction

$\mathsf{Keygen}(\lambda, n)$: given $\lambda \in \mathbb{N}$ and an integer $n \in \mathsf{poly}(\lambda)$, choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of order $N = p_1p_2p_3$, where $p_i > 2^\lambda$ for each $i \in \{1,2,3\}$. Choose $\alpha \xleftarrow{R} \mathbb{Z}_N$, $g, u, v \xleftarrow{R} \mathbb{G}_{p_1}$, $X_{p_3} \xleftarrow{R} \mathbb{G}_{p_3}$, $g_i \xleftarrow{R} \mathbb{G}_{p_1}$ for $i = 1$ to n. Then, select an identifier space $\mathcal{T}$. The private key is $\mathsf{sk} := \alpha$ while the

public key is

$$\mathsf{pk} := \Big((\mathbb{G}, \mathbb{G}_T),\ N,\ g,\ g^{\alpha},\ u,\ v,\ \{g_i\}_{i=1,\ldots,n},\ X_{p_3}\Big).$$

Sign$(\mathsf{sk}, \tau, \vec{v})$**:** on input of a vector $\vec{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_N^n$, a file identifier $\tau \in \mathcal{T}$ and the private key $\mathsf{sk} = \alpha \in \mathbb{Z}_N$, return $\perp$ if[3] $\vec{v} = \vec{0}$. Otherwise, conduct the following steps. First, choose $r \xleftarrow{R} \mathbb{Z}_N$ and $R_3, R_3' \xleftarrow{R} \mathbb{G}_{p_3}$. Then, compute a signature $\sigma = (\sigma_1, \sigma_2)$ as

$$\sigma_1 = (g_1^{v_1} \cdots g_n^{v_n})^{\alpha} \cdot (u^{\tau} \cdot v)^{r} \cdot R_3, \qquad \sigma_2 = g^{r} \cdot R_3',$$

SignDerive$(\mathsf{pk}, \tau, \{(\beta_i, \sigma_i)\}_{i=1}^{\ell})$**:** given pk, a file identifier τ and ℓ tuples (β_i, σ_i), parse σ_i as $\sigma_i = (\sigma_{i,1}, \sigma_{i,2})$ for $i = 1$ to ℓ. Then, choose $\tilde{r} \xleftarrow{R} \mathbb{Z}_N$, $\tilde{R}_3, \tilde{R}_3' \xleftarrow{R} \mathbb{Z}_N$ and compute

$$\sigma_1 = \prod_{i=1}^{\ell} \sigma_{i,1}^{\beta_i} \cdot (u^{\tau} \cdot v)^{\tilde{r}} \cdot \tilde{R}_3 \qquad \sigma_2 = \prod_{i=1}^{\ell} \sigma_{i,2}^{\beta_i} \cdot g^{\tilde{r}} \cdot \tilde{R}_3'$$

and output (σ_1, σ_2).

Verify$(\mathsf{pk}, \tau, \vec{y}, \sigma)$**:** given a public key pk, a signature $\sigma = (\sigma_1, \sigma_2)$ and a message $(\tau, \vec{y})$, where $\tau \in \mathbb{Z}_N$ and $\vec{y} = (y_1, \ldots, y_n) \in (\mathbb{Z}_N)^n$, return 0 if $\vec{y} = \vec{0}$. Otherwise, return 1 if and only if $e(\sigma_1, g) = e(g_1^{y_1} \cdots g_n^{y_n}, g^{\alpha}) \cdot e(u^{\tau} \cdot v, \sigma_2)$.

Verifying the correctness of the scheme is straightforward since pairing an element of $\mathbb{G}_{p_1}$ with an element of $\mathbb{G}_{p_3}$ always gives the identity element in $\mathbb{G}_T$.

4.2 Security

Theorem 1. *The scheme is adaptively context hiding if Assumption 1 holds.* (The proof is given in Appendix D).

As already mentioned, computational adaptive context hiding security does not imply statistical strong context hiding security (cf. Definition 4) in general. Let us consider a simple modification of the scheme. The public key includes $e(g, g)^{\varphi}$, for some $\varphi \in_R \mathbb{Z}_N$ which is *not* part of sk. Original signatures are augmented with $\sigma_3 = e(g, g)^{\varphi \cdot r}$, which is ignored by the verification algorithm. Also, SignDerive replaces σ_3 by a random element of $\mathbb{G}_T$. Although this artificial scheme can be proved adaptively context hiding under Assumptions 1 and 4, it does not meet the requirements of Definition 4.

Yet, it is immediate that the system of Section 4.1 is also secure in the sense of Definition 4. As a consequence, we can use a simplified security definition (Definition 13 in Appendix B) when it comes to prove the unforgeability of the scheme.

Theorem 2. *The scheme is unforgeable assuming that Assumptions 1, 2, 3 and 4 hold.* (The proof is given in Appendix E).

In Appendix F, we show that the same scheme can be safely instantiated in prime order groups if we settle for the weaker privacy definition used in [12, 13, 22]. The unforgeability of this modified scheme can be proved under the standard Diffie-Hellman assumption. To date, this construction turns out to be the shortest linearly homomorphic signature based on a simple assumption.

[3] In the construction, we disallow signatures on the all-zeroes vector $\vec{0}$. This is not a restriction since, in all applications of linearly homomorphic signatures, a unit vector $(0, \ldots, 1, \ldots, 0)$ of appropriate length is appended to signed vectors.

5 A Construction with Short Keys for Subset Predicates

In this section, we use the malleability properties of Groth-Sahai proofs (already exploited in, e.g., [7, 23, 20]) to construct a homomorphic signature for subset predicates. The main advantage over the approach of [4] is that we obtain constant-size[4] public keys in the standard model. In the standard model, the CP-ABE approach of [4] is currently limited to provide linear-size public keys in the maximal length of signed messages.

This limitation could be avoided using a ciphertext-policy adaption of the unbounded key-policy ABE system of [33]. However, the ABE construction of [33] is only known to be selectively secure and, for the time being, no fully secure unbounded CP-ABE system is available. Conceivably, such a scheme can be obtained by extending the techniques of [33]. Still, the resulting system would probably encounter the same difficulties as in Section 4 when it comes to obtain complete context hiding security. In contrast, our scheme is proved completely context hiding and fully (as opposed to selective-message) secure. It also allows for messages of unbounded (but polynomial) length.

In homomorphic signatures for subset predicates, the message space $\mathcal{M}$ can be defined as the set of tuples $\mathcal{M} := \Sigma^*$, where Σ is a set of words. The predicate P is defined in such a way that, for any polynomials $\{n_i\}_i$ and n', we have

$$P\Big(\{m_1, \ldots, m_n\}, \{m'_1, \ldots, m'_{n'}\}\Big) = 1 \quad \Longleftrightarrow \quad (n' \leq n) \wedge (m'_j \in \{m_1, \ldots, m_n\} \text{ for } j = 1 \text{ to } n').$$

The intuition of the scheme begins with the following naive construction, based on any digital signature, that only works when privacy is not a concern. The public key of the scheme is a standard digital signature key pair (sk, pk). When a message $\mathsf{Msg} = \{m_1, \ldots, m_n\}$ must be signed, the signer generates a fresh public key (sk', pk'), certifies pk' by computing $\sigma_{pk'} \leftarrow \mathsf{Sign}(sk, pk')$ and returning $(pk', \sigma_{pk'}, \{\sigma_i = \mathsf{Sign}(sk', m_i)\}_{i=1}^n)$. This simple construction immediately allows signature derivations for subset predicates. Moreover, since each signed set of words Msg involves a different public key pk', there is no way to generate a signature on a message $\mathsf{Msg}^\star$ that mixes words from two distinct signed messages Msg_1, Msg_2. However, the latter construction is trivially not context hiding. To achieve the latter property, instead of leaving pk' and $\{\sigma_i\}_{i=1}^{n'}$ appear in the clear within signatures, we let them appear in committed form and appeal to non-interactive witness indistinguishable (NIWI) arguments of knowledge of these signatures and keys. Then, the randomizability properties of Groth-Sahai proofs (which are recalled in Appendix C) come in handy to obtain the desired privacy properties.

To realize the above idea, we work with Waters signatures [40] and the structure-preserving signature of Abe *et al.* [1, 2] because they make it possible to work with *linear* pairing product equations (*i.e.*, equation (7) in Appendix C). As observed in [23], these equations have proofs that only depend on the randomness of Groth-Sahai commitments and not on the committed witnesses or on the right-hand-side member of the equation. In the $\mathsf{SignDerive}$ algorithm, this allows updating some of the witnesses in such a way that the old proof remains valid.

In the following notations, we define a coordinate-wise pairing $E : \mathbb{G} \times \mathbb{G}^3 \to \mathbb{G}_T^3$ such that, for any element $h \in \mathbb{G}$ and any vector $\vec{g} = (g_1, g_2, g_3)$, we have $E\big(h, \vec{g}\big) = \big(e(h, g_1), e(h, g_2), e(h, g_3)\big)$. In the following, when $X \in \mathbb{G}$ (resp. $Y \in \mathbb{G}_T$), the notation $\iota_{\mathbb{G}}(X)$ (resp. $\iota_{\mathbb{G}_T}(Y)$) will be used to denote the vector $(1_{\mathbb{G}}, 1_{\mathbb{G}}, X) \in \mathbb{G}^3$ (resp. the vector $(1_{\mathbb{G}_T}, 1_{\mathbb{G}_T}, Y) \in \mathbb{G}_T^3$).

[4] By "constant", we mean that it only depends on the security parameter and not on the length of messages to be signed.

Keygen(λ)**:** given a security parameter $\lambda \in \mathbb{N}$, choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Then, do the following.

1. Generate a Groth-Sahai CRS $\mathbf{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3)$ for the perfect witness indistinguishability setting. Namely, choose $\vec{f}_1 = (f_1, 1, g)$, $\vec{f}_2 = (1, f_2, g)$, and $\vec{f}_3 = \vec{f}_1^{\,\xi_1} \cdot \vec{f}_2^{\,\xi_2} \cdot (1,1,g)^{-1}$, with $f_1, f_2 \xleftarrow{R} \mathbb{G}$, $\xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p$.
2. Generate a key pair $(sk_{\mathsf{AHO}}, pk_{\mathsf{AHO}})$ for the AHO signature in order to sign messages consisting of a single group element. This key pair are

$$pk_{\mathsf{AHO}} = \Big(G_r,\ H_r,\ G_z = G_r^{\gamma_z},\ H_z = H_r^{\delta_z},\ G_1 = G_r^{\gamma_1}, H_1 = H_r^{\delta_1},\ A,\ B\Big)$$

   and $sk_{\mathsf{AHO}} = \big(\alpha_a, \alpha_b, \gamma_z, \delta_z, \gamma_1, \delta_1\big)$.
3. Generate parameters for the Waters signature. Namely, choose group elements $h \xleftarrow{R} \mathbb{G}$, and $(u_0, u_1, \ldots, u_L) \xleftarrow{R} \mathbb{G}^{L+1}$. These are used to implement a hash function $H_{\mathbb{G}} : \{0,1\}^L \to \mathbb{G}$ such that, for any string $m = m[1] \ldots m[L] \in \{0,1\}^L$, $H_{\mathbb{G}}(m) = u_0 \cdot \prod_{i=1}^{L} u_i^{m[i]}$.

The public key is defined to be $\mathsf{pk} := \Big((\mathbb{G}, \mathbb{G}_T), g,\ \mathbf{f},\ pk_{\mathsf{AHO}},\ h,\ \{u_i\}_{i=0}^{L}\Big)$ and the private key is $\mathsf{sk} = sk_{\mathsf{AHO}}$. The public key defines $\Sigma = \{0,1\}^L$.

Sign$(\mathsf{sk}, \mathsf{Msg})$**:** on input of a message $\mathsf{Msg} = \{m_i\}_{i=1}^{n}$, where $m_i \in \{0,1\}^L$ for each i, and the private key $\mathsf{sk} = sk_{\mathsf{AHO}}$, do the following.

1. Choose a new public key $X = g^x$ for Waters signatures, with $x \xleftarrow{R} \mathbb{Z}_p$. Generate a Groth-Sahai commitment $\vec{C}_X = \iota_{\mathbb{G}}(X) \cdot \vec{f}_1^{\,r_X} \cdot \vec{f}_2^{\,s_X} \cdot \vec{f}_3^{\,t_X}$, with $r_X, s_X, t_X \xleftarrow{R} \mathbb{Z}_p$.
2. Generate an AHO signature $(\theta_1, \ldots, \theta_7) \in \mathbb{G}^7$ on the group element $X \in \mathbb{G}$. Then, for each $j \in \{1,2,5\}$, generate Groth-Sahai commitments $\vec{C}_{\theta_j} = \iota_{\mathbb{G}}(\theta_j) \cdot \vec{f}_1^{\,r_{\theta_j}} \cdot \vec{f}_2^{\,s_{\theta_j}} \cdot \vec{f}_3^{\,t_{\theta_j}}$. Finally, generate NIWI proofs $\vec{\pi}_{\mathsf{AHO},1}, \vec{\pi}_{\mathsf{AHO},2} \in \mathbb{G}^3$ that committed variables $(X, \theta_1, \theta_2, \theta_5)$ satisfy

$$\begin{aligned} A \cdot e(\theta_3, \theta_4)^{-1} &= e(G_z, \theta_1) \cdot e(G_r, \theta_2) \cdot e(G_1, X) \\ B \cdot e(\theta_6, \theta_7)^{-1} &= e(H_z, \theta_1) \cdot e(H_r, \theta_5) \cdot e(H_1, X) \end{aligned} \tag{3}$$

   These proofs are obtained as

$$\begin{aligned} \vec{\pi}_{\mathsf{AHO},1} &= \big(G_z^{-r_{\theta_1}} G_r^{-r_{\theta_2}} G_1^{-r_X},\ G_z^{-s_{\theta_1}} G_r^{-s_{\theta_2}} G_1^{-s_X},\ G_z^{-t_{\theta_1}} G_r^{-t_{\theta_2}} G_1^{-t_X}\big) \\ \vec{\pi}_{\mathsf{AHO},2} &= \big(H_z^{-r_{\theta_1}} H_r^{-r_{\theta_5}} H_1^{-r_X},\ H_z^{-s_{\theta_1}} H_r^{-s_{\theta_5}} H_1^{-s_X},\ H_z^{-t_{\theta_1}} H_r^{-t_{\theta_5}} H_1^{-t_X}\big) \end{aligned}$$

3. For each $i \in \{1, \ldots, n\}$, generate a Waters signature $(\sigma_{i,1}, \sigma_{i,2})$ on the word $m_i \in \{0,1\}^L$ by computing $(\sigma_{i,1}, \sigma_{i,2}) = \big(h^x \cdot H_{\mathbb{G}}(m_i)^{\chi_i},\ g^{\chi_i}\big)$ for a randomly chosen $\chi_i \xleftarrow{R} \mathbb{Z}_p$. Then, generate a Groth-Sahai commitment $\vec{C}_{\sigma_{i,1}} = \iota_{\mathbb{G}}(\sigma_{i,1}) \cdot \vec{f}_1^{\,r_{i,1}} \cdot \vec{f}_2^{\,s_{i,1}} \cdot \vec{f}_3^{\,t_{i,1}}$, with $r_{i,1}, s_{i,1}, t_{i,1} \xleftarrow{R} \mathbb{Z}_p$, and a NIWI proof $\pi_{W,i}$ that $(X, \sigma_{i,1})$ satisfy

$$e(H_{\mathbb{G}}(m_i), \sigma_{i,2}) = e(X, h)^{-1} \cdot e(\sigma_{i,1}, g). \tag{4}$$

   This proof is obtained as $\pi_{W,i} = \big(h^{r_X} \cdot g^{-r_{i,1}},\ h^{s_X} \cdot g^{-s_{i,1}},\ h^{t_X} \cdot g^{-t_{i,1}}\big)$.
4. Return the signature

$$\sigma = \Big(\vec{C}_X, \{\vec{C}_{\theta_j}\}_{j\in\{1,2,5\}}, \{\theta_j\}_{j\in\{3,4,6,7\}}, \vec{\pi}_{\mathsf{AHO},1}, \vec{\pi}_{\mathsf{AHO},2}, \{\vec{C}_{\sigma_{i,1}}, \sigma_{i,2}, \vec{\pi}_{W,i}\}_{i=1}^{n}\Big). \tag{5}$$

Note that proofs $\vec{\pi}_{\mathsf{AHO},1}, \vec{\pi}_{\mathsf{AHO},2}$ and $\{\vec{\pi}_{W,i}\}_i$ only depend on the randomness used in commitments and not on the committed values or on the left-hand-side members of pairing-product equations (3) and (4).

SignDerive$(\mathsf{pk}, \mathsf{Msg}, \mathsf{Msg}', \sigma)$**:** given pk, $\mathsf{Msg} = \{m_i\}_{i=1}^n$ and $\mathsf{Msg}' = \{m'_i\}_{i=1}^{n'}$, return $\perp$ if there exists $i \in \{1, \ldots, n'\}$ such that $m'_i \notin \{m_i\}_{i=1}^n$. Otherwise, parse σ as in (5). For each $i \in \{1, \ldots, n'\}$, let $\rho(i) \in \{1, \ldots, n\}$ be the index such that $m'_i = m_{\rho(i)}$. Then, for each $i \in \{1, \ldots, n'\}$, do the following.

1. Re-randomize the commitment $\vec{C}_X$ and the proofs $\vec{\pi}_{\mathsf{AHO},1}, \vec{\pi}_{\mathsf{AHO},2}$, $\{\vec{\pi}_{W,i}\}_i$ accordingly. Let $\vec{C}'_X$, $\vec{\pi}'_{\mathsf{AHO},1}, \vec{\pi}'_{\mathsf{AHO},2}$, and $\{\vec{\pi}'_{W,i}\}_i$ be the randomized commitment and proofs. Note that, in all of these commitments and proofs (r_X, s_X, t_X) have been updated consistently.
2. Re-randomize $\{\vec{C}_{\theta_j}\}_{j\in\{2,5\}}$ and $\{\theta_j\}_{j\in\{3,4,6,7\}}$ by choosing $\varrho_2, \varrho_5, \mu, \nu$ and computing

$$\vec{C}'_{\theta_2} = \vec{C}_{\theta_2} \cdot \iota_{\mathbb{G}}(\theta_4)^{\varrho_2} \qquad \theta'_3 = \big(\theta_3 \cdot G_r^{-\varrho_2}\big)^{1/\mu} \qquad \theta'_4 = \theta_4^{\mu},$$
$$\vec{C}'_{\theta_5} = \vec{C}_{\theta_5} \cdot \iota_{\mathbb{G}}(\theta_7)^{\varrho_5} \qquad \theta'_6 = \big(\theta_6 \cdot H_r^{-\varrho_5}\big)^{1/\nu} \qquad \theta'_7 = \theta_7^{\nu}.$$

   We note that, although the committed values inside $\vec{C}'_{\theta_2}, \vec{C}'_{\theta_5}$ have changed. The proofs $\pi'_{\mathsf{AHO},1}, \pi'_{\mathsf{AHO},2}$ are still valid for the new committed values. Then, compute $\{\vec{C}''_{\theta_j}\}_{j\in\{1,2,5\}}$ by re-randomizing the commitments $\vec{C}_{\theta_1}, \{\vec{C}'_{\theta_j}\}_{j\in\{2,5\}}$ and re-randomize the proofs $\pi'_{\mathsf{AHO},1}, \pi'_{\mathsf{AHO},2}$ again. Let $\vec{\pi}''_{\mathsf{AHO},1}, \vec{\pi}''_{\mathsf{AHO},2}$ be the re-randomized proofs.
3. For each $i \in \{1, \ldots, n'\}$, choose $\chi'_i \xleftarrow{R} \mathbb{Z}_p$ and compute

$$\vec{C}'_{\sigma_{\rho(i),1}} = \vec{C}_{\sigma_{\rho(i),1}} \cdot \iota_{\mathbb{G}}\big(H_{\mathbb{G}}(m_{\rho(i)})^{\chi'_i}\big), \qquad \sigma'_{\rho(i),2} = \sigma_{\rho(i),2} \cdot g^{\chi'_i}.$$

   Even though the committed value inside $\vec{C}'_{\sigma_{\rho(i),1}}$ has changed, $\vec{\pi}'_{W,\rho(i)}$ remains a valid proof that the updated committed value $\sigma'_{\rho(i),1}$ satisfies $e(X,h)\cdot e(H_{\mathbb{G}}(m_{\rho(i)}), \sigma'_{\rho(i),2}) = e(\sigma'_{\rho(i),1}, g)$. The commitment $\vec{C}'_{\sigma_{\rho(i),1}}$ is then re-randomized and the proof $\vec{\pi}'_{W,\rho(i)}$ is re-randomized accordingly. Let $\vec{C}''_{\sigma_{\rho(i),1}}$ and $\vec{\pi}''_{W,\rho(i)}$ denote the new commitment and proof.
4. Return the signature

$$\sigma' = \Big(\vec{C}'_X, \{\vec{C}''_{\theta_j}\}_{j\in\{1,2,5\}}, \{\theta'_j\}_{j\in\{3,4,6,7\}}, \vec{\pi}''_{\mathsf{AHO},1}, \vec{\pi}''_{\mathsf{AHO},2}, \{\vec{C}''_{\sigma_{\rho(i),1}}, \sigma'_{\rho(i),2}, \vec{\pi}''_{W,\rho(i)}\}_{i=1}^{n'}\Big). \quad (6)$$

Verify$(\mathsf{pk}, \mathsf{Msg}, \sigma)$**:** given pk, σ and $\mathsf{Msg} = \{m_i\}_{i=1}^n$, parse σ as per (5).

1. Return 0 if $\vec{\pi}_{\mathsf{AHO},1} = (\pi_1, \pi_2, \pi_3)$ and $\vec{\pi}_{\mathsf{AHO},2} = (\pi_4, \pi_5, \pi_6)$ do not satisfy.

$$\iota_{\mathbb{G}_T}(A) \cdot E\big(\theta_3, \iota_{\mathbb{G}}(\theta_4)\big)^{-1} = E(G_z, \vec{C}_{\theta_1}) \cdot E(G_r, \vec{C}_{\theta_2}) \cdot E(G_1, \vec{C}_X) \cdot \prod_{j=1}^{3} E(\pi_j, \vec{f}_j)$$

$$\iota_{\mathbb{G}_T}(B) \cdot E\big(\theta_6, \iota_{\mathbb{G}}(\theta_7)\big)^{-1} = E(H_z, \vec{C}_{\theta_1}) \cdot E(H_r, \vec{C}_{\theta_5}) \cdot E(H_1, \vec{C}_X) \cdot \prod_{j=1}^{3} E(\pi_{j+3}, \vec{f}_j).$$

2. Return 1 if, for each i, $\vec{\pi}_{W,i} = (\pi_{W,i,1}, \pi_{W,i,2}, \pi_{W,i,3})$ satisfies

$$E\big(h, \vec{C}_X\big) \cdot E\big(H_{\mathbb{G}}(m_i), (1, 1, \sigma_{i,2})\big) = E(g, \vec{C}_{\sigma_{i,1}}) \cdot \prod_{j=1}^{3} E(\pi_{W,i,j}, \vec{f}_j).$$

   Otherwise, return 0.

From an efficiency point of view, the scheme is asymptotically optimal with constant-size public (and private) keys and signatures consisting of $7n + 22$ group elements for messages of length n.

Theorem 3. *The scheme is completely context hiding.*

Proof. The security of the scheme in the sense of Definition 7 follows from the fact that, when the CRS $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ is generated as in step 1 of the key generation algorithm, commitments $\vec{C}_X$ are always perfectly hiding and proofs are perfectly witness indistinguishable.

Moreover, the components $\{\theta_j\}_{j\in\{3,4,6,7\}}$ of AHO signatures are perfectly re-randomized by the signature derivation algorithm and the same is true for $\{\sigma_{i,2}\}_{i=1}^{n}$. As for Groth-Sahai proofs $\pi_{\mathsf{AHO},1}, \pi_{\mathsf{AHO},2}$ and $\{\pi_{W,i}\}_{i=1}^{n}$, they are also perfectly re-randomized. It comes that the output of $\mathsf{SignDerive}$ is always distributed like a fresh signature. □

Theorem 4. *The scheme is unforgeable if the DLIN and q-SFP assumptions hold in* $\mathbb{G}$.

Proof. Having shown that the scheme is completely context hiding (and thus secure in the sense of Definition 4), we can prove unforgeability using a simplified definition where the adversary only interacts with a signing oracle. The proof uses a sequence of three games where, for each $i \in \{0, 1, 2\}$, we denote by S_i the event that the adversary $\mathcal{A}$ produces a successful forgery at the end of Game_i.

Game_0: This game is the real game. We denote by S_0 the event that the adversary $\mathcal{A}$ manages to output a successful forgery. Obviously, $\mathcal{A}$'s advantage is $\Pr[S_0]$.

Game_1: We modify the generation of the public key and choose $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$ as a Groth-Sahai CRS for the perfect soundness setting, where even an unbounded adversary cannot prove false statements. More precisely, the challenger $\mathcal{B}$ sets up $\vec{f_1} = (f_1, 1, g)$, $\vec{f_2} = (1, f_2, g)$ and $\vec{f_3} = \vec{f_1}^{\xi_1} \cdot \vec{f_2}^{\xi_2}$, with $f_1 = g^{\phi_1}$ and $f_2 = g^{\phi_2}$, for randomly chosen $\phi_1, \phi_2, \xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p$. It is easy to build a DLIN distinguisher that bridges between Game_0 and Game_1. This implies that, under the DLIN assumption, this modification does not significantly affect $\mathcal{A}$'s behavior. We can thus write $|\Pr[S_1] - \Pr[S_0]| \leq \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B})$. Moreover, from Game_1 onwards, all Groth-Sahai commitments are perfectly binding and their content can be extracted using (ϕ_1, ϕ_2) by performing a BBS decryption [9].

Game_2: In this game, $\mathcal{B}$ makes use of the values $(\phi_1, \phi_2) = (\log_g(f_1), \log_g(f_2))$ that were defined in Game_1. When $\mathcal{A}$ outputs a forgery $\sigma^\star$, the challenger $\mathcal{B}$ uses (ϕ_1, ϕ_2) to extract $X^\star$ from the Groth-Sahai commitment $\vec{C}_X^\star$ contained in $\sigma^\star$ (recall that, due to the modification introduced in Game_1, $\vec{C}_X^\star$ is now an extractable commitment). We raise a failure event, called F_2, and let the challenger $\mathcal{B}$ abort if the extracted $X^\star$ was never used in any signing query. Clearly, any occurrence of F_2 immediately contradicts the unforgeability of the AHO signature and thus the q-SFP assumption. We can thus write $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2] \leq \mathbf{Adv}^{q\text{-SFP}}(\mathcal{B})$.

In Game_2, we can easily prove that, conditionally on $\neg F_2$, event S_2 only occurs with negligible probability if the Diffie-Hellman assumption holds. Let $(\sigma^\star, \mathsf{Msg}^\star = \{m_1^\star, \ldots, m_{n^\star}^\star\})$ denote $\mathcal{A}$'s forgery. If F_2 does not occur, we know that $X^\star$, which is extracted from the commitment $\vec{C}_X^\star$ contained in $\sigma^\star$, *did* appear in some signing query. Let $j \in \{1, \ldots, q\}$ denote the index of that query $\mathsf{Msg}_j = \{m_{j,1}, \ldots, m_{j,n_j}\}$. If $\sigma^\star$ is a successful forgery, we know that $\mathsf{Msg}^\star \not\subseteq \mathsf{Msg}_j$. In other words, there exists $\ell \in \{1, \ldots, n^\star\}$ such that $m_\ell^\star \notin \mathsf{Msg}_j$. This allows constructing a forger $\mathcal{B}'$ breaking the unforgeability of Waters signatures with probability $\Pr[S_2|\neg F_2]/q$.

Specifically, the forger $\mathcal{B}'$ of Game$_3$ receives as input a public key $(X^\dagger, h, (u_0, u_1, \ldots, u_L))$ for Waters signatures. To generate the public key pk of the system, it generates an AHO key pair $(sk_{\mathsf{AHO}}, pk_{\mathsf{AHO}})$ of its own at the beginning of the game. It also sets up $\mathbf{f} = (\vec{f}_1, \vec{f}_2, \vec{f}_3)$ as an extractable Groth-Sahai CRS. It finally chooses a random index $j^\star \xleftarrow{R} \{1, \ldots, q\}$ as a guess that the extracted $X^\star \in \mathbb{G}$ will coincide with the value X appearing in the $j^\star$-th signing query. At each query $\mathsf{Msg}_j = \{m_{j,1}, \ldots, m_{j,n_j}\}$ such that $j \neq j^\star$, $\mathcal{B}'$ answers the query faithfully, by setting up $X = g^x$ for a randomly chosen $x \xleftarrow{R} \mathbb{Z}_p$ which is used to generate Waters signatures as in the real signing algorithm. If $j = j^\star$, $\mathcal{B}'$ computes $\vec{C}_X$ using its challenge Waters public key $X^\dagger$ at step 1 of the signing algorithm. At step 3 of the signing algorithm, $\mathcal{B}'$ obtains $\{(\sigma_{i,1}, \sigma_{i,2})\}_{i=1}^{n_j}$ by successively invoking its signing oracle for messages $m_{j^\star,1}, \ldots, m_{j^\star,n_{j^\star}}$. At the end of the game, with probability $1/q$, the element $X^\star$ extracted from $\vec{C}^\star_X$ in $\mathcal{A}$'s forgery happens to be $X^\star = X^\dagger$. By hypothesis, $\mathsf{Msg}^\star = \{m^\star_1, \ldots, m^\star_{n^\star}\}$ contains an element $m^\star_\ell$ that $\mathcal{B}'$ did not query to its challenger at the $j^\star$-th signing query. From $\sigma^\star$, $\mathcal{B}'$ can thus extract a valid signature $(\sigma^\star_{\ell,1}, \sigma^\star_{\ell,2})$ on $m^\star_\ell$ and break the security of Waters signatures. The result of [40] implies that $\Pr[S_2|\neg F_2] \leq 8 \cdot q^2 \cdot (L+1) \cdot \mathbf{Adv}^{\mathrm{CDH}}(\mathcal{B}')$.

When collecting probabilities, we find

$$\Pr[S_0] \leq \mathbf{Adv}^{\mathrm{DLIN}}(\mathcal{B}) + 2 \cdot \mathbf{Adv}^{q\text{-SFP}}(\mathcal{B}) + 8 \cdot q^2 \cdot (L+1) \cdot \mathbf{Adv}^{\mathrm{CDH}}(B'').$$

Since the DLIN and q-SFP assumptions both imply the hardness of CDH in the same group, the statement of the theorem follows. □

We note that, if one of the structure-preserving signatures of Abe *et al.* [3] is used in place of the AHO signature, it is easy to build a scheme relying on the DLIN assumption only.

References

1. M. Abe, K. Haralambiev, M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. Cryptology ePrint Archive: Report 2010/133, 2010.
2. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In *Crypto'10*, *LNCS* 6223, pp. 209–236, 2010.
3. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, M. Ohkubo. Constant-Size Structure-Preserving Signatures: Generic Constructions and Simple Assumptions. In *Asiacrypt'12*, *LNCS* series, to appear, 2012. Cryptology ePrint Archive: Report 2012/285.
4. J.-H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, a. shelat, B. Waters. Computing on Authenticated Data. In *TCC 2012*, *LNCS* 7194, pp. 1–20, 2012.
5. N. Attrapadung, B. Libert. Homomorphic Network Coding Signatures in the Standard Model. In *PKC'11*, *LNCS* 6571, pp. 17–34, 2011.
6. P. Barreto, M. Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. In *SAC'05*, *LNCS* 3897, pp. 319–331, 2005.
7. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, H. Shacham. Randomizable Proofs and Delegatable Anonymous Credentials. In *Crypto'09*, *LNCS* 5677, pp. 108–125, 2009.
8. M. Bellare, G. Neven. Transitive Signatures Based on Factoring and RSA. In *Asiacrypt'02*, *LNCS* 2501, 397–414, 2002.
9. D. Boneh, X. Boyen, H. Shacham. Short Group Signatures. In *Crypto'04*, *LNCS* 3152, pp. 41–55. Springer, 2004.
10. D. Boneh, M. Franklin. Identity-Based Encryption from the Weil Pairing. In *SIAM* Journal of Computing 32(3), pp. 586–615, 2003, earlier version in *Crypto'01*, *LNCS* 2139, pp. 213–229, 2001.
11. D. Boneh, D. Freeman, J. Katz, B. Waters. Signing a Linear Subspace: Signature Schemes for Network Coding. In *PKC'09*, *LNCS* 5443, pp. 68–87, 2009.
12. D. Boneh, D. Freeman. Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-Based Signatures. In *PKC'11*, *LNCS* 6571, pp. 1–16, 2011.

13. D. Boneh, D. Freeman. Homomorphic Signatures for Polynomial Functions. In *Eurocrypt'11*, *LNCS* 6632, pp. 149–168, 2011.
14. D. Boneh, E. Shen, B. Waters. Strongly Unforgeable Signatures Based on Computational Diffie-Hellman. In *PKC'06*, *LNCS* 3958, pp. 229–240, 2009.
15. C. Brzuska, H. Busch, O. Dagdelen, M. Fischlin, M. Franz, S. Katzenbeisser, M. Manulis, C. Onete, A. Peter, B. Poettering, D. Schröder Redactable Signatures for Tree-Structured Data: Definitions and Constructions. In *ACNS'10*, *LNCS* 6123, pp. 87–104, 2010.
16. C. Brzuska, M. Fischlin, T. Freudenreich, A. Lehmann, M. Page, J. Schelbert, D. Schröder, F. Volk. Security of Sanitizable Signatures Revisited. In *PKC'09*, *LNCS* 3376, pp. 317–336, 2009.
17. C. Brzuska, M. Fischlin, A. Lehmann, D. Schröder. Unlinkability of Sanitizable Signatures. In *PKC'10*, *LNCS* 6056, pp. 444-461, 2010.
18. D. Catalano, D. Fiore, B. Warinschi. Adaptive Pseudo-free Groups and Applications. In *Eurocrypt'11*, *LNCS* 6632, pp. 207–223, 2011.
19. D. Catalano, D. Fiore, B. Warinschi. Efficient Network Coding Signatures in the Standard Model. In *PKC'12*, *LNCS* 7293, pp. 680–696, 2012.
20. M. Chase, M. Kohlweiss, A. Lysyanskaya, S. Meiklejohn. Malleable Proof Systems and Applications. In *Eurocrypt'12*, *LNCS* 7237, pp. 281-300, 2012.
21. Y. Desmedt. Computer security by redefining what a computer is. In *New Security Paradigms Workshop (NSPW) 1993*, pp. 160–166, 1993.
22. D. Freeman. Improved security for linearly homomorphic signatures: A generic framework. In *PKC'12*, *LNCS* 7293, pp. 697–714, 2012.
23. G. Fuchsbauer. Commuting Signatures and Verifiable Encryption. In *Eurocrypt'11*, *LNCS* 6632, pp. 224-245, 2011.
24. R. Gennaro, J. Katz, H. Krawczyk, T. Rabin. Secure Network Coding over the Integers. In *PKC'10*, *LNCS* 6056, pp. 142–160, 2010.
25. M. Gerbush, A. Lewko, A. O'Neill, B. Waters. Dual Form Signatures: An Approach for Proving Security from Static Assumptions. In *Asiacrypt'12*, *LNCS* series, to appear, 2012. Cryptology ePrint Archive: Report 2012/261.
26. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC'09*, pp. 169–178, 2009.
27. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
28. A. Hevia, D. Micciancio. The Provable Security of Graph-Based One-Time Signatures and Extensions to Algebraic Signature Schemes. In *Asiacrypt'02*, LNCS 2501, pp. 379-396, 2002.
29. E. Kiltz, A. Mityagin, S. Panjwani, B. Raghavan. Append-Only Signatures. In *ICALP'05*, LNCS 3580, pp. 434–445, 2005.
30. A. Lewko. Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting. In *Eurocrypt'12*, *LNCS* 7237, pp. 318–335, 2012.
31. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *Eurocrypt 2010*, *LNCS* 6110, pp. 62–91, 2010.
32. A. Lewko, B. Waters. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC 2010*, LNCS 5978, pp. 455-479, Springer, 2010.
33. A. Lewko, B. Waters. Unbounded HIBE and Attribute-Based Encryption. In *Eurocrypt'11*, *LNCS* 6632, pp. 149–168, 2011.
34. R. Johnson, D. Molnar, D. Song, D. Wagner. Homomorphic Signature Schemes. In *CT-RSA'02*, LNCS 2271, pp. 244–262, 2002.
35. S. Micali, R. Rivest. Transitive Signature Schemes. In *CT-RSA'02*, *LNCS* 2271, pp. 236-243, 2002.
36. K. Miyazaki, G. Hanaoka, H. Imai. Digitally signed document sanitizing scheme based on bilinear maps. In *AsiaCCS'06*, pp. 343-354, 2006.
37. T. Okamoto, K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *Crypto'10*, *LNCS* 6223, pp. 191–208, 2010.
38. M. Prabhakaran, M. Rosulek. Homomorphic Encryption with CCA Security. In *ICALP 2008*, *LNCS* 5126, pp. 667-678, 2008.
39. M. van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Eurocrypt'10*, *LNCS* 6110, pp. 22–43, 2010.
40. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Eurocrypt'05*, *LNCS* 3494, pp. 114–127, 2005.
41. B. Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Crypto'09*, *LNCS* series, 2009.

A Definition of Weak Privacy

In the definitional framework of Ahn *et al.* [4], weak privacy can be defined by generalizing the definition given by Boneh and Freeman [12] in the context of linearly homomorphic signatures. Intuitively, weak privacy does not hide the fact that a signature derivation took place but only hides the original message set. This is captured by Definition 10, which is almost identical to Definition 6: the only difference is that, in the challenge phase, the adversary only chooses $M_0, M_1 \subset \mathcal{M}$, $m' \in \mathcal{M}$ and is not allowed to see the original signatures.

Definition 10. *A P-homomorphic signature* $(\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ *is* **weakly context hiding** *if no PPT adversary has non-negligible advantage in the following game:*

1. *The challenger runs* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Keygen}(\lambda)$ *and gives* $(\mathsf{sk}, \mathsf{pk})$ *to the adversary.*
2. *The adversary* $\mathcal{A}$ *chooses two message sets* $M_0, M_1 \subset \mathcal{M}$ *and a message* $m' \in \mathcal{M}$ *such that* $P(M_0, m') = P(M_1, m') = 1$*. The challenger flips a fair coin* $\beta \xleftarrow{R} \{0,1\}$ *and computes* $\{\sigma_m\}_{m \in M_\beta} \leftarrow \mathsf{Sign}(sk, M_\beta)$ *as well as* $\sigma^\star \leftarrow \mathsf{SignDerive}\big(\mathsf{pk}, (\{\sigma_m\}_{m \in M_\beta}, M_\beta), m'\big)$*. In either case,* $\sigma^\star$ *is sent as a challenge to* $\mathcal{A}$*.*
3. *The adversary* $\mathcal{A}$ *outputs a bit* $\beta' \in \{0,1\}$ *and wins if* $\beta' = \beta$*. As always,* $\mathcal{A}$*'s advantage is measured as the distance* $\mathbf{Adv}(\mathcal{A}) = |\Pr[\beta' = \beta] - 1/2|$*.*

B Definitions for Linearly Homomorphic Signatures

In linearly homomorphic signatures, the message space $\mathcal{M}$ consists of pairs $\mathcal{M} := \mathcal{T} \times \mathbb{Z}_p^n$ for some positive integers p, n and where $\mathcal{T}$ is a tag space. The predicate P is defined in such a way that $P(\varepsilon, m) = 1$ for each $m \in \mathcal{M}$ and

$$P\Big(\{(\tau_1, \vec{v}_1), \ldots, (\tau_k, \vec{v}_k)\}, (\tau, \vec{v})\Big) = 1 \quad \Longleftrightarrow \quad \big(\tau = \tau_1 = \cdots = \tau_k\big) \wedge \big(\vec{v} \in \mathrm{span}(\vec{v}_1, \ldots, \vec{v}_k)\big)$$

In the context of linearly homomorphic signatures, the definitions can be specialized as in [11].

Definition 11. *A linearly homomorphic signature scheme is a tuple of efficient algorithms* $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$

$\mathsf{Keygen}(\lambda, n)$: *is a probabilistic algorithm that takes as input a security parameter* $\lambda \in \mathbb{N}$ *and an integer* $n \in \mathsf{poly}(\lambda)$ *denoting the length of vectors to be signed. It outputs a key pair* $(\mathsf{pk}, \mathsf{sk})$ *and the description of a tag (i.e., a file identifier) space* $\mathcal{T}$*.*

$\mathsf{Sign}(\mathsf{sk}, \tau, \vec{v})$: *is a possibly randomized algorithm that takes in a private key* sk*, a file identifier* $\tau \in \mathcal{T}$ *and a vector* $\vec{v}$*. It outputs a signature* σ*.*

$\mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\beta_i, \sigma_i)\}_{i=1}^{\ell})$: *is a (possibly randomized) algorithm that takes as input a public key* pk*, a file identifier* τ *and* ℓ *tuples* (β_i, σ_i)*, each one of which consists of a weight* β_i *and a signature* σ_i*. Intuitively, the output is a signature* σ *on the vector* $\vec{y} = \sum_{i=1}^{\ell} \beta_i \vec{v}_i$*, where* σ_i *is a signature on* $\vec{v}_i$*.*

$\mathsf{Verify}(\mathsf{pk}, \tau, \vec{y}, \sigma)$: *is a deterministic algorithm that takes as input a public key* pk*, a file identifier* $\tau \in \mathcal{T}$*, a signature* σ *and a vector* $\vec{y}$*. It outputs* 0 *or* 1*.*

Correctness is formulated by mandating that, for all security parameters $\lambda \in \mathbb{N}$, all integers $n \in \mathsf{poly}(\lambda)$ and all triples $(\mathsf{pk}, \mathsf{sk}, \mathcal{T}) \leftarrow \mathsf{Keygen}(\lambda, n)$, the following holds.

1. For all $\tau \in \mathcal{T}$ and all n-vectors $\vec{y}$, if $\sigma = \mathsf{Sign}(\mathsf{sk}, \tau, \vec{y})$, then we have $\mathsf{Verify}(\mathsf{pk}, \tau, \vec{y}, \sigma) = 1$.
2. For all $\tau \in \mathcal{T}$, any $\ell > 0$ and any set of triples $\{(\beta_i, \sigma_i, \vec{v}_i)\}_{i=1}^{\ell}$, if $\mathsf{Verify}(\mathsf{pk}, \tau, \vec{v}_i, \sigma_i) = 1$ for each $i \in \{1, \ldots, \ell\}$, then it holds that $\mathsf{Verify}\big(\mathsf{pk}, \tau, \sum_{i=1}^{\ell} \beta_i \vec{v}_i, \mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\beta_i, \sigma_i)\}_{i=1}^{\ell})\big) = 1$.

SECURITY. In this context, Definition 3 can be rephrased as follows. We note that the definition captures the stronger definition used by Freeman [22]. In addition, file identifiers can be chosen by the adversary and are not assumed to be uniformly distributed. As a result, any easy-to-remember string can serve as a file identifier.

Definition 12. *A linearly homomorphic signature scheme* $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ *is secure if no probabilistic polynomial time (PPT) adversary has non-negligible advantage (as a function of the security parameter* $\lambda \in \mathbb{N}$*) in the following game:*

1. *The adversary* $\mathcal{A}$ *chooses an integer* $n \in \mathbb{N}$ *and sends it to the challenger who runs* $\mathsf{Keygen}(\lambda, n)$ *and obtains* $(\mathsf{pk}, \mathsf{sk})$ *before sending* pk *to* $\mathcal{A}$*.*
2. *On polynomially-many occasions,* $\mathcal{A}$ *can interleave the following kinds of queries.*
   - *Signing queries:* $\mathcal{A}$ *chooses a tag* $\tau \in \mathcal{T}$ *and a vector* $\vec{v}$*. The challenger chooses a handle* h *and computes* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \tau, \vec{v})$*. It stores* $(h, (\tau, \vec{v}, \sigma))$ *in a table* T *and returns* h *to* $\mathcal{A}$*.*
   - *Derivation queries:* $\mathcal{A}$ *chooses a vector of handles* $\vec{h} = (h_1, \ldots, h_k)$ *and a message* $(\tau, \vec{y}) \in \mathcal{M}$*. The challenger retrieves the tuples* $\{(h_i, (\tau, \vec{v}_i), \sigma_i)\}_{i=1}^{k}$ *from* T *and returns* $\perp$ *if one of these does not exist or if there exists* $i \in \{1, \ldots, k\}$ *such that* $\tau_i \neq \tau$*. Otherwise, it defines* $M := \big((\tau, \vec{v}_1), \ldots, (\tau, \vec{v}_k)\big)$*. If* $\vec{y} \notin \mathrm{span}(\vec{v}_1, \ldots, \vec{v}_k)$*, the challenger returns* $\perp$*. Otherwise, it runs* $\sigma' \leftarrow \mathsf{SignDerive}\big(\mathsf{pk}, (\{\sigma_i\}_{i=1}^{k}, M), \vec{y}\big)$*, chooses a handle* h'*, stores* $(h', (\tau, \vec{y}), \sigma')$ *in* T *and returns* h' *to* $\mathcal{A}$*.*
   - *Reveal queries:* $\mathcal{A}$ *chooses a handle* h*. If no tuple of the form* $(h, (\tau, \vec{v}), \sigma')$ *exists in* T*, the challenger returns* $\perp$*. Otherwise, it returns* σ' *to* $\mathcal{A}$ *and adds* $((\tau, \vec{v}), \sigma')$ *to the set* Q*.*
3. $\mathcal{A}$ *outputs an identifier* $\tau^\star$*, a signature* $\sigma^\star$ *and a vector* $\vec{y} \in \mathbb{Z}_N^n$*. The adversary* $\mathcal{A}$ *is deemed successful if* $\mathsf{Verify}(\mathsf{pk}, \tau^\star, \vec{y^\star}, \sigma^\star) = 1$ *and either of the following holds:*
   - *(Class I):* $\tau^\star \neq \tau_i$ *for any entry* $(\vec{\tau}_i, .)$ *in* Q *and* $\vec{y^\star} \neq \vec{0}$*.*
   - *(Class II):* $\tau^\star = \tau_i$ *for* $k_i > 0$ *entries* $(\tau_i, .)$ *in* Q *and* $\vec{y^\star} \notin V_i$*, where* V_i *denotes the subspace spanned by all vectors* $\vec{v}_1, \ldots, \vec{v}_{k_i}$ *for which an entry of the form* $(\tau^\star, \vec{v}_j)$*, with* $j \in \{1, \ldots, k_i\}$*, appears in* Q*.*

$\mathcal{A}$'s advantage is its probability of success taken over all coin tosses.

It is assumed that, for each subspace V_i, the adversary only queries linearly independent vectors $\vec{v}_1, \ldots, \vec{v}_{k_i}$. This is not a limitation since, in practical applications such as network coding [11], the signer typically augments the signed vectors with a unit vector so as to make sure they will be independent.

In [4], Ahn *et al.* proved that, when the scheme is strongly context hiding in the sense of Definition 4, the definition of unforgeability can be simplified by only providing the adversary with a signing oracle. In this case, the definition can be restated as follows.

Definition 13. *A linearly homomorphic signature scheme* $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ *is secure if no probabilistic polynomial time (PPT) adversary has non-negligible advantage (as a function of the security parameter* $\lambda \in \mathbb{N}$*) in the following game:*

1. *The adversary $\mathcal{A}$ chooses an integer $n \in \mathbb{N}$ and sends it to the challenger who runs* $\mathsf{Keygen}(\lambda, n)$ *and obtains* $(\mathsf{pk}, \mathsf{sk})$ *before sending* pk *to* $\mathcal{A}$.
2. *On a polynomial number of occasions, $\mathcal{A}$ chooses a tag $\tau \in \mathcal{T}$ and a vector $\vec{v}$. The challenger returns $\sigma = \mathsf{Sign}(\mathsf{sk}, \tau, \vec{v})$ to $\mathcal{A}$.*
3. *$\mathcal{A}$ outputs an identifier $\tau^\star$, a signature $\sigma^\star$ and a vector $\vec{y} \in \mathbb{Z}_N^n$. The adversary $\mathcal{A}$ is deemed successful if* $\mathsf{Verify}(\mathsf{pk}, \tau^\star, \vec{y}^\star, \sigma^\star) = 1$ *and either of the following holds:*
   - *(Class I): $\tau^\star \neq \tau_i$ for any i and $\vec{y}^\star \neq \vec{0}$.*
   - *(Class II): $\tau^\star = \tau_i$ for some $i \in \{1, \ldots, q\}$ and $\vec{y}^\star \notin V_i$, where V_i denotes the subspace spanned by all vectors $\vec{v}_1, \ldots, \vec{v}_{k_i}$ that have been queried for τ_i.*

C Groth-Sahai Proof Systems

The Groth-Sahai (GS) techniques [27] can be based on the DLIN assumption, where they use prime order groups and a common reference string containing three vectors $\vec{f_1}, \vec{f_2}, \vec{f_3} \in \mathbb{G}^3$, where $\vec{f_1} = (f_1, 1, g)$, $\vec{f_2} = (1, f_2, g)$ for some $f_1, f_2 \in \mathbb{G}$. To commit to a group element $X \in \mathbb{G}$, one chooses $r, s, t \xleftarrow{R} \mathbb{Z}_p^*$ and computes $\vec{C} = (1, 1, X) \cdot \vec{f_1}^{\,r} \cdot \vec{f_2}^{\,s} \cdot \vec{f_3}^{\,t}$. In the perfect soundness setting, we have $\vec{f_3} = \vec{f_1}^{\,\xi_1} \cdot \vec{f_2}^{\,\xi_2}$ where $\xi_1, \xi_2 \in \mathbb{Z}_p^*$. Commitments $\vec{C} = (f_1^{r+\xi_1 t}, f_2^{s+\xi_2 t}, X \cdot g^{r+s+t(\xi_1+\xi_2)})$ are then extractable (and distributed as Boneh-Boyen-Shacham (BBS) ciphertexts [9]) using $\beta_1 = \log_g(f_1)$, $\beta_2 = \log_g(f_2)$. In the witness indistinguishability (WI) setting, vectors $\vec{f_1}, \vec{f_2}, \vec{f_3}$ are linearly independent so that $\vec{C}$ is a perfectly hiding commitment. Under the DLIN assumption, the two kinds of CRS are computationally indistinguishable.

To prove that committed variables satisfy a set of relations, the prover computes one commitment per variable and one proof element for each relation. Efficient non-interactive witness indistinguishable (NIWI) proofs are available for pairing-product equations, which are relations of the type

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T, \tag{7}$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{ij} \in \mathbb{Z}_p$, for $i, j \in \{1, \ldots, n\}$.

In equations of the form (7), proofs for quadratic equations – where some factors are pairings of two variables – consist of 9 group elements whereas linear equations (*i.e.*, where $a_{ij} = 0$ for all i, j in equation (7)) only consist of 3 group elements each.

D Proof of Theorem 1

The proof considers two games.

Game$_{real}$: is the real game. It begins with the challenger generating a key pair $(\mathsf{pk}, \mathsf{sk})$ and giving $\mathsf{sk} = \alpha \in \mathbb{Z}_N$ to the adversary. In the game of Definition 6, the adversary supplies a message-set $M = \{m_1, \ldots, m_k\}$, with $k \leq n$, the corresponding signature set $\{\sigma_m\}_{m \in M}$ and a message m' such that $P(M, m') = 1$. The challenger flips a coin $\beta \in \{0, 1\}$ and computes a challenge signature $(\sigma_1^\star, \sigma_2^\star)$ as a derived (resp. fresh) signature on m' if $\beta = 0$ (resp. $\beta = 1$). The game ends with the adversary $\mathcal{A}$ outputting a bit $\beta' \in \{0, 1\}$ and the challenger outputs 1 if $\beta' = \beta$. We call S_{real} the latter event.

Game$_1$: is like Game$_{real}$ with the difference that the challenger halts and outputs 1 in the event, which we call E_1 hereafter, that $\{\sigma_m\}_{m\in M}$ contains a signature (σ_1, σ_2) where either σ_1 or σ_2 has a non-trivial component of order p_2. We call S_1 the event that the challenger eventually outputs 1 at the end of Game$_1$. Note that the challenger makes use of the factorization of N to detect $\mathbb{G}_{p_2}$ components in Game$_1$. Still, we claim that there exists a distinguisher $\mathcal{D}$ against Assumption 1 for which we have $|\Pr[S_{real}] - \Pr[S_1]| \leq \mathbf{Adv}^1(\mathcal{D})/2 + 1/2^\lambda$, as explained below.

We note that Game$_{real}$ and Game$_1$ are identical until E_1 occurs. Assuming that $\Pr[E_1]$ is non-negligible, we build an algorithm $\mathcal{B}$ that receives as input ($g \in \mathbb{G}_{p_1}$, $X_{p_3} \in \mathbb{G}_{p_3}$) and finds an element $\eta^\star \in \mathbb{G}_{p_2p_3}$ with a non-trivial $\mathbb{G}_{p_2}$ component. At the outset of the game, $\mathcal{B}$ generates the public key by computing g^α, $u = g^{a_u}$, $v = g^{a_v}$ and $g_i = g^{a_i}$, for $i = 1$ to n, using randomly chosen $\alpha, a_u, a_v, a_1, \ldots, a_n \xleftarrow{R} \mathbb{Z}_N$. Note that, while $\mathsf{sk} = \alpha \in \mathbb{Z}_N^*$ is given to the adversary, no information about $a_u \bmod p_2$, $a_v \bmod p_2$ and $\{a_i \bmod p_2\}_{i=1}^n$ is revealed by the public key. Let us assume that a signature (σ_1, σ_2) on a message $\big(\tau, \vec{v} = (v_1, \ldots, v_n)\big)$ has a non-trivial $\mathbb{G}_{p_2}$ component in either σ_1 or σ_2. If we define $\eta = \sigma_1/g^{\alpha\cdot\langle\vec{a},\vec{v}\rangle} \cdot \sigma_2^{a_u\cdot\tau+a_v}$, we observe that η is an element of $\mathbb{G}_{p_2p_3}$ with a non-trivial $\mathbb{G}_{p_2}$ component with overwhelming probability: indeed, η has no $\mathbb{G}_{p_2}$ component if and only if $\log_{g_{p_2}}(\sigma_1) = (a_u\cdot\tau + a_v)\cdot\log_{g_{p_2}}(\sigma_2) \bmod p_2$, which only occurs with probability $1/p_2 < 1/2^\lambda$ since $a_u \bmod p_2$ and $a_v \bmod p_2$ are uniform in $\mathbb{Z}_{p_2}$ and independent of $\mathcal{A}$'s view. This means that, if $\Pr[E_1]$ is non-negligible, $\mathcal{B}$ can find an element of $\eta^\star \in \mathbb{G}_{p_2p_3}$ with a non-trivial $\mathbb{G}_{p_2}$ component: $\mathcal{B}$ simply computes η_m for each message-signature pair $\big(m = (\tau, \vec{v}), (\sigma_1, \sigma_2)\big)$ produced by $\mathcal{A}$ and computes $\eta^\star$ as a random linear combination of the resulting $\{\eta_m\}_{m\in M}$. Note that an algorithm finding such an element $\eta^\star \in \mathbb{G}_{p_2p_3}$ with probability ε yields a distinguisher with advantage $\varepsilon/2$ against Assumption 1. As a consequence, if $\mathbf{Adv}^1(\mathcal{D})$ denotes the maximal advantage of a PPT distinguisher $\mathcal{D}$ against Assumption 1, we have $\Pr[E_1] \leq \mathbf{Adv}^1(\mathcal{D})/2 + 1/2^\lambda$. Given that $|\Pr[S_{real}] - \Pr[S_1]| \leq \Pr[E_1]$, we find $|\Pr[S_{real}] - \Pr[S_1]| \leq \mathbf{Adv}^1(\mathcal{D})/2 + 1/2^\lambda$.

In Game$_1$, we have $\Pr[S_1|\neg E_1] = 1/2$, so that $\mathcal{B}$ outputs 1 with probability 1/2: indeed, if event E_1 does not occur, the distribution of the challenge signature $(\sigma_1^\star, \sigma_2^\star)$ will be perfectly independent of the challenger's bit $\beta \in \{0, 1\}$. We can thus write $\Pr[S_1] \leq 1/2 + \mathbf{Adv}^1(\mathcal{D})/2 + 1/2^\lambda$ and finally find $|\Pr[S_{real}] - 1/2| \leq \mathbf{Adv}^1(\mathcal{D}) + 1/2^{\lambda-1}$. □

E Proof of Theorem 2

Since the scheme is strongly context hiding (in the sense of Definition 4), we can use the simpler Definition 13 (in Appendix B), where the adversary is only granted access to a signing oracle.

Lemma 1. *For any Class I forger $\mathcal{A}$, there is a forger $\mathcal{B}$ that breaks the security of Lewko-Waters signatures with the same advantage. Consequently, the scheme is secure against Class I forgeries under Assumptions 1, 2 and 4.*

Proof. We consider a variant of Lewko-Waters signatures where the public key consists of group elements $(g, g^\alpha, h, u, v) \in \mathbb{G}_{p_1}$ and $X_{p_3} \in \mathbb{G}_{p_3}$, where $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_3}$ are the subgroups of orders p_1 and p_3, respectively, in a group $\mathbb{G}$ of order $N = p_1p_2p_3$. The private key $\alpha \in \mathbb{Z}_N$ allows signing messages $m \in \mathbb{Z}_N$ as

$$(\sigma_1, \sigma_2) = \big(h^\alpha \cdot (u^m \cdot v)^r \cdot R_3,\ g^r \cdot R_3'\big),$$

where $r \xleftarrow{R} \mathbb{Z}_p$ and $R_3, R_3' \xleftarrow{R} \mathbb{G}_{p_3}$.

If an adversary $\mathcal{A}$ can produce a Class I forgery with non-negligible probability, we construct

a forger $\mathcal{B}$ with the same advantage for Lewko-Waters signatures. Algorithm $\mathcal{B}$ receives as input a Lewko-Waters public key $(g, g^\alpha, h, u, v, X_{p_3})$. It defines $(g_1, \ldots, g_n)$ by setting $g_i = h^{\gamma_i} \cdot g^{\delta_i}$, for $i = 1$ to n, using random $\gamma_1, \ldots, \gamma_n \xleftarrow{R} \mathbb{Z}_N$ and $\delta_1, \ldots, \delta_n \xleftarrow{R} \mathbb{Z}_N$. The adversary $\mathcal{A}$ is given $\mathsf{pk} = \big((\mathbb{G}, \mathbb{G}_T), g, g^\alpha, \{g_i\}_{i=1}^n, u, v, X_{p_3}\big)$.

For each signing query $(\tau, \vec{v} = (v_1, \ldots, v_n))$ made by $\mathcal{A}$, $\mathcal{B}$ obtains a Lewko-Waters signature $(\tilde{\sigma_1}, \tilde{\sigma_2}) = (h^\alpha \cdot H_{\mathbb{G}}(\tau)^r, g^r)$ on the message τ from its own signing oracle. Then, $\mathcal{B}$ transforms this signature into a linearly homomorphic signature. To do this, $\mathcal{B}$ picks $r' \xleftarrow{R} \mathbb{Z}_p$, computes

$$(\sigma_1, \sigma_2) = \Big((g^\alpha)^{\sum_{i=1}^n \delta_i \cdot v_i} \cdot \tilde{\sigma_1}^{\sum_{i=1}^n \gamma_i \cdot v_i} \cdot (u^\tau \cdot v)^{r'}, \tilde{\sigma_2}^{\sum_{i=1}^n \gamma_i \cdot v_i} \cdot g^{r'}\Big).$$

It is straightforward that (σ_1, σ_2) forms a valid signature on $(\tau, \vec{v})$ and it is returned to $\mathcal{A}$.

Eventually, $\mathcal{A}$ outputs a valid signature $(\sigma_1^\star, \sigma_2^\star)$ on some message $(\tau^\star, \vec{v}^\star = (v_1^\star, \ldots, v_n^\star))$. Since $\mathcal{A}$ is a Class I forger, we $\tau^\star$ did not appear in any signing query. For this reason, $\mathcal{B}$ can forge a Lewko-Waters signature by computing

$$(\sigma_1, \sigma_2) = \Big(\big((g^\alpha)^{-\sum_{i=1}^n \delta_i v_i^\star} \cdot \sigma_1^\star\big)^{1/(\sum_{i=1}^n \gamma_i v_i^\star)}, \sigma_2^{\star 1/(\sum_{i=1}^n \gamma_i v_i^\star)}\Big),$$

which is possible whenever $\sum_{i=1}^n \gamma_i \cdot v_i^\star \neq 0 \bmod N$. However, since $\vec{v}^\star \neq \vec{0}$, we can only have $\sum_{i=1}^n \gamma_i \cdot v_i^\star = 0$ with negligible chance since $\{\gamma_i\}_{i=1}^n$ are independent of $\mathcal{A}$'s view. □

Lemma 2. *The scheme is secure against class II forgeries under Assumptions 1, 2, 3 and 4.*

Proof. The proof proceeds as in [5, 25] and follows the dual system methodology [41, 32]. It is easy to observe that any identifier-vector-signature triple $(\tau, \vec{y}, \sigma)$ satisfying the verification equation $e(\sigma_1, g) = e(\prod_{i=1}^n h_i^{y_i}, g^\alpha) \cdot e(u^\tau \cdot v, \sigma_2)$ must be of the form:

$$\sigma_1 = (\prod_{i=1}^n g_i^{y_i})^\alpha \cdot (u^\tau \cdot v)^r \cdot g_{p_2}^{w_1} \cdot R_3, \qquad \sigma_2 = g^r \cdot g_{p_2}^{w_2} \cdot R_3', \tag{8}$$

where $g_{p_2} \in \mathbb{G}_{p_2}$, for some $r \in \mathbb{Z}_{p_1}$, $w_1, w_2 \in \mathbb{Z}_{p_2}$ and $R_3, R_3' \in \mathbb{G}_{p_3}$.

We will distinguish two types of signatures as follows.

- Type A: $(w_1, w_2) = (0, 0) \bmod p_2$.
- Type B: $(w_1, w_2) \neq (0, 0) \bmod p_2$.

We can also distinguish two kinds of Type B signatures: Type B.1 signatures are such that $w_2 \neq 0$ whereas Type B.2 signatures have $w_2 = 0$. We will call Type A forgery (resp. Type B forgery) a fake signature of Type A (resp. Type B) which is produced by the forger.

The proof considers a sequence of $q + 3$ games. It starts with the real attack game Game_{real} followed by $\text{Game}_1, \text{Game}_2, \text{Game}_3, \text{Game}_{4.1}, \ldots, \text{Game}_{4.q}$, where q is the number of *distinct* tags $\tau_1, \ldots, \tau_q$ appearing in signing queries.

Game_{real}: is the real game. The game ends with the adversary outputting an alleged forgery. If the adversary $\mathcal{A}$ is indeed successful, the challenger outputs 1. We call S_{real} the latter event.

Game_1: GUESSING THE INDEX OF $\tau^\star$. This game is identical to Game_{real} with the difference that, at the outset of the game, the challenger chooses $j^\star \xleftarrow{R} \{1, \ldots, q\}$ at random and thereby guesses that $\mathcal{A}$'s forgery will involve the $j^\star$-th distinct tag appearing in signing queries. If the guess

eventually turns out to be wrong, the challenger halts and outputs a random bit. Otherwise, (*i.e.*, if $\tau^\star$ was indeed the $j^\star$-th distinct queried tag), the challenger outputs 1 if $\mathcal{A}$'s forgery is a valid one. We denote by S_1 the event of the challenger outputting 1 in Game_1. Since the choice of $j^\star$ is independent of $\mathcal{A}$'s view, we clearly have $\Pr[S_1] = \Pr[S_{real}]/q$.

Game_2: Eliminating zero inner products in the exponent. This game is like Game_1 but the challenger halts and reports failure if the adversary's forgery involves a vector $\vec{y}^\star$ such that $\prod_{i=1}^n g_i^{y_i^\star} = 1_{\mathbb{G}}$. It is easy to see that, under the discrete logarithm assumption in $\mathbb{G}_{p_1}$, Game_2 does not significantly depart from Game_1. Note that we are not introducing any extra assumption since Assumption 3 implies the hardness of computing discrete logarithms in $\mathbb{G}_{p_1}$.

Game_3: Restriction modulo p_2. It is as Game_2 but the game will further abort if the following event, called E_3, occurs. If $V_{j^\star}$ denotes the subspace $\text{span}(\vec{v}_{j^\star,1}, \ldots, \vec{v}_{j^\star,k})$ of vectors that are signed under the tag $\tau_{j^\star}$, we define E_3 as the event that $\mathcal{A}$ outputs a class (ii) forgery (*i.e.,* $\tau^\star = \tau_{j^\star}$ and $\vec{y}^\star \notin V_{j^\star}$) but for which[5] $\vec{y}^\star \bmod p_2 \in V_j \bmod p_2$. Lemma 3 shows that, under Assumption 1 and Assumption 2, the difference between Game_2 and Game_3 is negligible. Then, Lemma 4 shows that, if $\mathcal{A}$ can output a Type B forgery in Game_3, Assumption 1 is false.

$\text{Game}_{4.(j_1,j_2)}$ ($1 \leq j_1 \leq q$, $1 \leq j_2 \leq n-1$): Hybrid types. It is as Game_0 but signing queries are handled in a hybrid way. Each query $(\tau_{l_1}, \vec{v}_{l_2})$ is indexed by a pair (l_1, l_2), where l_1 identifies the l_1-th distinct tag τ_{l_1} chosen by $\mathcal{A}$ during the game and l_2 denotes the l_2-th distinct vector involving τ_{l_1}. Each query (l_1, l_2) is answered as follows.

Case $l_1 < j_1$ or $(l_1 = j_1) \cap (l_2 \leq j_2)$: the challenger returns a Type B signature if $l_1 \neq j^\star$ and a Type A signature if $l_1 = j^\star$.

Case $l_1 > j_1$ or $(l_1 = j_1) \cap (l_2 > j_2)$: the challenger always returns a Type A signature.

Lemma 5 demonstrates that, in all the games of this sub-sequence, the adversary has about the same probability of outputting a Type A signature, unless there is a distinguisher against Assumption 2. For convenience, we define $\text{Game}_{4.(0,n-1)}$ as being identical to Game_3.

Game_5: Switching signatures for $\tau_{j^\star}$ to Type B. Is like $\text{Game}_{4.(q,n-1)}$ with the difference that, at the signing query involving the $j^\star$-th distinct tag $\tau_{j^\star}$, the challenger now returns a Type B.2 signature instead of a Type A signature. Lemma 6 shows that, under Assumption 3, this modification does not noticeably increase $\mathcal{A}$'s probability to output a Type B signature in the forgery stage. At this point, we only have to worry about Type A forgeries

In Game_5, Lemma 7 shows that, if $\mathcal{A}$ is able to produce a Type A forgery with non-negligible probability, it is possible to break Assumption 4. □

Lemma 3. *Any significant difference between the adversary's behaviors in Game_2 and Game_3 contradicts either Assumption 1 or Assumption 2.*

Proof. Game_2 and Game_3 proceed identically unless the adversary $\mathcal{A}$ outputs a forgery involving a message $(\tau^\star, \vec{y}^\star)$ such that $\tau^\star = \tau_{j^\star} \bmod N$ but $\det(M_{V_{j^\star}}) = 0 \bmod p_2$ and $\det(M_{V_{j^\star}}) \neq 0 \bmod N$, where $M_{V_{j^\star}} \in \mathbb{Z}_N^{n \times n}$ is the matrix

$$M_{V_{j^\star}} = \left(\begin{array}{cccc} & | & & | \\ R_{n \times (n-m_{j^\star}-1)} & \vec{v}_{j^\star,1}^\top & \cdots & \vec{v}_{j^\star,m_{j^\star}}^\top \\ & | & & | \end{array} \begin{array}{c} | \\ \vec{y}^{\star\top} \\ | \end{array} \right),$$

[5] We denote by $V \bmod p_2$ the subspace V reduced in $\mathbb{Z}_{p_2}^n$. More precisely, for any subspace $V = \text{span}(\vec{v}_1, \ldots, \vec{v}_m) \subset \mathbb{Z}_N^n$, the notation $V \bmod p_2$ denotes $\text{span}(\vec{v}_1 \bmod p_2, \ldots, \vec{v}_m \bmod p_2) \subset \mathbb{Z}_{p_2}^n$.

with $m_{j^\star} = \dim(V_{j^\star}) < n$, such that $R_{n\times(n-m_{j^\star}-1)}$ is a $n \times (n - m_{j^\star} - 1)$ matrix whose columns are orthogonal to $\mathrm{span}(\vec{v}_{j^\star,1}, \ldots, \vec{v}_{j^\star,m_{j^\star}}, \vec{y}^\star)$ (such a matrix can be obtained via the Gram-Schmidt process). The matrix has the desired properties since $\vec{y}^\star \bmod p_2 \in V_{j^\star} \bmod p_2$ although $\vec{y}^\star \notin V_{j^\star}$. The simulator $\mathcal{B}$ can extract a non-trivial factor of N by computing $\gcd(\det(M_{V_{j^\star}}), N)$. As shown in [32][Lemma 5], this allows breaking either Assumption 1 or Assumption 2 depending on which factor is extracted. □

Lemma 4. *Under Assumption 1, no PPT adversary can output a Type B forgery in Game$_3$.*

Proof. We show that, if the adversary outputs a Type B forgery in Game$_3$, there is a distinguisher $\mathcal{B}$ that, given (g, X_3, T), decides if $T \in_R \mathbb{G}_{p_1}$ or $T \in_R \mathbb{G}_{p_1p_2}$. This algorithm proceeds as in the proof of Theorem 1, by finding an element of $\mathbb{G}_{p_2p_3}$ with a non-trivial $\mathbb{G}_{p_2}$ component.

The distinguisher $\mathcal{B}$ sets up the public key pk by choosing $\alpha, a_u, a_v, a_1, \ldots, a_n \stackrel{R}{\leftarrow} \mathbb{Z}_N$ and computing g^α, $u = g^{a_u}$, $v = g^{a_v}$ and $g_i = g^{a_i}$ for $i = 1$ to n. It also sets $X_{p_3} = X_3$. We define $\vec{a} = (a_1, \ldots, a_n)$. Algorithm $\mathcal{B}$ can answer all signing queries by faithfully running the signing algorithm since it knows the private key $\mathsf{sk} = \alpha \in \mathbb{Z}_N$.

At the end, $\mathcal{A}$ outputs a file identifier $\tau^\star$, a Type-B signature $(\sigma_1^\star, \sigma_2^\star)$ and a vector $\vec{y}^\star$. The algorithm $\mathcal{B}$ then computes

$$\eta^\star = \sigma_1^\star / g^{\alpha \cdot \langle \vec{a}, \vec{y}^\star \rangle} \cdot \sigma_2^{\star\, a_u \cdot \tau + a_v}. \tag{9}$$

The $\mathbb{G}_{p_1}$ components of $(\sigma_1^\star, \sigma_2^\star)$ necessarily cancel out in the right-hand-side member of (9). For this reason, if either $\sigma_1^\star$ or $\sigma_2^\star$ has a non-trivial $\mathbb{G}_{p_2}$ component, so has $\eta^\star \in \mathbb{G}_{p_2p_3}$ with all but negligible probability. Indeed, as long as $a_u, a_v \bmod p_2$ are information theoretically hidden to the adversary (which is the case since they are uncorrelated to $a_u, a_v \bmod p_1$), we can only have $\log_{g_{p_2}}(\sigma_1^\star) = (a_u \cdot \tau^\star + a_v) \cdot \log_{g_{p_2}}(\sigma_2^\star)$ by pure chance, with probability $1/p_2 < 1/2^\lambda$. there must be an element of $\mathbb{G}_{p_2p_3}$ with non-trivial $\mathbb{G}_{p_2}$ component among η_1, η_2. Therefore, the distinguisher $\mathcal{B}$ can conclude that $T \in \mathbb{G}_{p_1p_2}$ if and only if either $e(T, \eta^\star) \neq 1_{\mathbb{G}_T}$ or $e(T, \eta^\star) \neq 1_{\mathbb{G}_T}$. □

Lemma 5. *For each $j_1 \in \{1, \ldots, q\}$ and $j_2 \in \{1, \ldots, n-1\}$, the adversary outputs a Type A forgery with negligibly different probabilities in Game$_{4.(j_1,j_2)}$ and Game$_{4.(j_1,j_2-1)}$ if Assumption 2 holds. Under the same assumption, he outputs a Type A forgery with about the same probability in Game$_{4.(j_1-1,n-1)}$ and Game$_{4.(j_1,1)}$.*

Proof. Let us assume that a forger $\mathcal{A}$ has significantly better probability of outputting a Type B forgery in the aforementioned games. We describe a distinguisher $\mathcal{B}$ that breaks Assumption 2 with non-negligible advantage.

Algorithm $\mathcal{B}$ takes as input $(g, X_1X_2, Z_3, Y_2Y_3, T)$ and uses the adversary $\mathcal{A}$ to decide if $T \in_R \mathbb{G}_{p_1p_3}$ or $T \in_R \mathbb{G}$. Let $\tau_1, \ldots, \tau_q$ denote all the adversarially-chosen tags involved in signing queries throughout the game. Recall that $\mathcal{A}$ must obtain Type B signatures at queries involving τ_i such that $i \in \{1, \ldots, j-1\} \backslash \{j^\star\}$ and Type A signatures for tags τ_i such that $i \in \{j^\star\} \cup \{j+1, \ldots, q\}$. As for tag τ_j, the distribution of the signatures will depend on T if $j \neq j^\star$. In the latter case, we will simulate the interaction so that the resulting signature will be of Type A (hence emulating Game$_{4.j}$) if $T \in_R \mathbb{G}_{p_1}$ and of Type B (as in Game$_{4.(j+1)}$) if $T \in_R \mathbb{G}$. It comes that $\mathcal{A}$ will eventually produce a Type B forgery with about the same probability in either case if Assumption 2 holds. We then show that the distinguisher $\mathcal{B}$ can indeed distinguish whether $\mathcal{A}$'s forgery will be of Type A or Type B with overwhelming probability.

To this end, $\mathcal{B}$ prepares the public key pk by choosing $\alpha, a_u, a_v, a_1, \ldots, a_n \stackrel{R}{\leftarrow} \mathbb{Z}_N$, and setting $u = g^{a_u}$, $v = g^{a_v}$, $g_i = g^{a_i}$ for $i = 1$ to n. The public key $\mathsf{pk} = \big(g, g^\alpha, u, v, g_1, \ldots, g_n, Z_3\big)$ is given to $\mathcal{A}$. Each signing query is indexed by a pair (l_1, l_2) where $l_1 \in \{1, \ldots, q\}$ is the index of the involved tag τ_{l_1} and $l_2 \in \{1, \ldots, n-1\}$ denotes the index of the l_2-th queried vector for τ_{l_1}. The way to answer signing queries depends on their index (l_1, l_2).

[**Case** $l_1 < j_1$ or $(l_1 = j_1) \cap (l_2 < j_2)$:] At the l_2-th signing query $(\tau_{l_1}, \vec{v}_{l_2})$ involving the l_1-th distinct tag τ_{l_1} chosen by $\mathcal{A}$, $\mathcal{B}$ creates a Type A signature (using the private key $\alpha \in \mathbb{Z}_N$) if $l_1 = j^\star$. Otherwise (*i.e.*, if $l_1 \neq j^\star$), it chooses a fresh random exponent $r \stackrel{R}{\leftarrow} \mathbb{Z}_N$. It then chooses $z_1, z_2 \stackrel{R}{\leftarrow} \mathbb{Z}_N$ and computes a Type-B signature (σ_1, σ_2) on $\big(\tau_{l_1}, \vec{v}_{l_2} = (v_1, \ldots, v_n)\big)$ as

$$\sigma_1 = (\prod_{i=1}^{n} g_i^{v_i})^\alpha \cdot (u^\tau \cdot v)^r \cdot (Y_2 Y_3)^{z_1}, \qquad \sigma_2 = g^r \cdot (Y_2 Y_3)^{z_2}$$

[**Case** $l_1 > j_1$ or $(l_1 = j_1) \cap (l_2 > j_2)$:] In this case, $\mathcal{A}$ simply computes a Type A signature using the private key $\mathsf{sk} = \alpha \in \mathbb{Z}_N$ as specified by the signing algorithm.

[**Case** $(l_1 = j_1) \cap (l_2 = j_2)$:] To answer the j_2-th signing query $(\tau_{j_1}, \vec{v}_{j_2} = (v_1, \ldots, v_n))$ involving the j_1-th distinct tag, $\mathcal{B}$ proceeds as follows. If $j_1 = j^\star$, it simply returns a Type A signature using $\mathsf{sk} = \alpha$. Otherwise, it chooses $z_1 \stackrel{R}{\leftarrow} \mathbb{Z}_N$, $Z_3', Z_3'' \stackrel{R}{\leftarrow} \mathbb{G}_{p_3}$ and computes

$$\sigma_1 = (\prod_{i=1}^{n} g_i^{v_i})^\alpha \cdot T^{a_u \tau_{j_1} + a_v} \cdot Z_3', \qquad \sigma_2 = T \cdot Z_3''. \tag{10}$$

If $T \in \mathbb{G}_{p_1 p_2}$, (σ_1, σ_2) is easily seen to form a Type A signature. If $T \in \mathbb{G}$, (σ_1, σ_2) is a Type B signature with $(w_1, w_2) = \big(\log_{g_{p_2}}(T) \cdot (a_u \cdot \tau_{l_1} + a_v), \log_{g_{p_2}}(T)\big)$.

At the end of the game, $\mathcal{A}$ outputs a forgery $\sigma^\star = (\sigma_1^\star, \sigma_2^\star)$ and a vector $\vec{y}^\star = (y_1^\star, \ldots, y_n^\star)$ and a file identifier $\tau^\star$. At this point, $\mathcal{B}$ halts and outputs a random bits if $\tau_{j^\star} \neq \tau^\star$ since it incorrectly guessed $j^\star \in_R \{1, \ldots, q\}$. Otherwise (*i.e.*, if $\tau^\star = \tau_{j^\star}$), we know that, if the (j_1, j_2)-th signing query was answered using a Type B signature, $\tau^\star$ is necessarily different from the tag τ_{j_1} appearing in that signing query. Moreover, $(\sigma_1^\star, \sigma_2^\star)$ satisfies

$$e(\sigma_1^\star, g) = e(\prod_{i=1}^{n} h_i^{y_i^\star}, g^\alpha) \cdot e(u^{\tau^\star} \cdot v, \sigma_2^\star). \tag{11}$$

At this stage, $\mathcal{B}$ halts and checks whether the forgery is of Type A or B. If the forgery is of Type A, it returns 0 (meaning that $T \in_R \mathbb{G}_{p_1 p_3}$). If the forgery is believed to be of Type B, $\mathcal{B}$ rather bets on $T \in_R \mathbb{G}_{p_1 p_2 p_3}$ and outputs 1.

When it comes to figure out which kind of forgery $\mathcal{A}$ comes up with, $\mathcal{B}$ uses its input value $X_1 X_2$ as follows. It computes

$$\eta = \frac{\sigma_1^\star}{(\prod_{i=1}^{n} g_i^{y_i^\star})^\alpha \cdot \sigma_2^{\star\, a_u \tau^\star + a_v}}. \tag{12}$$

We can easily check from equation (11) that the $\mathbb{G}_{p_1}$ component of each term is canceled out in the right-hand-side member of (12). If $e(X_1 X_2, \eta) = 1$, then algorithm $\mathcal{B}$ decides that $\sigma^\star$ is of Type A.

Otherwise, it is considered as a Type B signature. To see why this test works with overwhelming probability, we note that, if $(\sigma_1^\star, \sigma_2^\star)$ is a Type B signature, we have

$$e(X_1X_2, \eta) = e(X_2, g_{p_2})^{w_1^\star - w_2^\star(a_u\tau^\star + a_v)}.$$

If $\sigma^\star$ is of Type B, it can only be interpreted as a Type A signature if and only if

$$w_1^\star - w_2^\star \cdot (a_u\tau^\star + a_v) = 0 \bmod p_2. \tag{13}$$

However, this occurs with negligible probability since the value $a_u \cdot \tau^\star + a_v \bmod p_2$ is information theoretically hidden to $\mathcal{A}$. Indeed, if $\tau^\star = \tau_{j^\star}$, we know that either: (1) $\tau^\star \neq \tau_{j_1}$; (2) $\tau^\star = \tau_{j_1}$ but the (j_1, j_2)-th query was answered by returning a Type A signature so that $\mathcal{A}$ has not seen $a_u \cdot \tau_{j_1} + a_b \bmod p_2$. In either case, no information about $a_u \cdot \tau^\star + a_v \bmod p_2$ is leaked to $\mathcal{A}$ and relation (10) can only be satisfied with negligible probability. □

Lemma 6. *If Assumption 3 holds, the probability that the adversary outputs a Type B forgery is not noticeably higher in Game$_5$ than in Game$_{4.(q,n-1)}$.*

Proof. Towards a contradiction, we assume that a forger $\mathcal{A}$ has significantly better probability of outputting a Type B forgery in Game$_5$ than in Game$_{4.(q,n-1)}$. We describe a distinguisher $\mathcal{B}$ that breaks Assumption 3 with non-negligible advantage.

Algorithm $\mathcal{B}$ takes as input $(g, w, g^t, X_1X_2, X_3, Y_2Y_3, T)$ and decides if $T = w^t Z_3$ or $T = w^t Z_2 Z_3$ for some $Z_2 \in_R \mathbb{G}_{p_2}$ and $Z_3 \in_R \mathbb{G}_{p_3}$. Let $\tau_1, \ldots, \tau_q$ denote the tags appearing in signing queries throughout the game. Recall that $\mathcal{A}$ must obtain Type B signatures at queries involving τ_i such that $i \in \{1, \ldots, j-1\} \backslash \{j^\star\}$. As for signatures pertaining to $\tau_{j^\star}$, their distribution will depend on whether T has a $\mathbb{G}_{p_2}$ component or not. If yes, the signature will be a Type B. Otherwise, it will be a Type A signature. If Assumption 3 holds, $\mathcal{A}$ will thus produce a Type B forgery with nearly the same probability in either case. We show that the distinguisher $\mathcal{B}$ can indeed distinguish whether $\mathcal{A}$'s forgery will be of Type A or Type B with overwhelming probability.

Algorithm $\mathcal{B}$ prepares the public key pk by choosing $\alpha, a_u, a_v, a_1, \ldots, a_n \xleftarrow{R} \mathbb{Z}_N$, and setting $g^\alpha = g^t$ (which implicitly defines $\alpha = t$), $u = g^{a_u}$, $v = g^{a_v}$, $g_i = w^{a_i}$ for $i = 1$ to n. The public key $\mathsf{pk} = \big(g, g^\alpha, u, v, \{g_i\}_{i=1}^n, X_3\big)$ is given to $\mathcal{A}$. Note that the private key $\mathsf{sk} = t$ is not available in this game. However, $\mathcal{B}$ will be able to answer signing queries as follows. Recall that each signing query is indexed by a pair (l_1, l_2) where $l_1 \in \{1, \ldots, q\}$ is the index of the involved tag τ_{l_1} and $l_2 \in \{1, \ldots, n-1\}$ denotes the index of the l_2-th queried vector for τ_{l_1}.

[**Case** $l_1 \neq j^\star$:] If $l_1 \neq j^\star$, $\mathcal{B}$ handles the signing query $\big(\tau_{l_1}, \vec{v}_{l_2} = (v_{l_2,1}, \ldots, v_{l_2,n})\big)$ by choosing $r \xleftarrow{R} \mathbb{Z}_N$, $z_1, z_2 \xleftarrow{R} \mathbb{Z}_N$ and computes a Type-B signature (σ_1, σ_2) on $\big(\tau_{l_1}, \vec{v}_{l_2} = (v_1, \ldots, v_n)\big)$ as

$$\sigma_1 = T^{\sum_{i=1}^n a_i \cdot v_{l_2,i}} \cdot (u^\tau \cdot v)^r \cdot (Y_2Y_3)^{z_1}, \qquad \sigma_2 = g^r \cdot (Y_2Y_3)^{z_2}.$$

Observe that (σ_1, σ_2) always forms a Type B signature regardless of whether T has a $\mathbb{G}_{p_2}$ component. Moreover, inner products $\langle \vec{a}, \vec{v}_{l_2} \rangle \bmod p_2$ are perfectly hidden by randomizers $(Y_2Y_3)^{z_1}$ and $(Y_2Y_3)^{z_1}$, should T have a non-trivial $\mathbb{G}_{p_2}$ component.

[**Case** $l_1 = j^\star$:] To answer signing queries $(\tau_{j^\star}, \vec{v}_{l_2} = (v_{l_2,1}, \ldots, v_{l_2,n}))$ involving the $j^\star$-th tag $\tau_{j^\star}$, $\mathcal{B}$ chooses $z_1 \xleftarrow{R} \mathbb{Z}_N$, $Z_3', Z_3'' \xleftarrow{R} \mathbb{G}_{p_3}$ and computes

$$\sigma_1 = T^{\sum_{i=1}^n a_i \cdot v_{l_2,i}} \cdot (u^{\tau_{j^\star}} \cdot v)^r \cdot Z_3', \qquad \sigma_2 = g^r \cdot Z_3''. \tag{14}$$

If $T = w^t Z_3$, (σ_1, σ_2) forms a Type A signature. If $T = w^t Z_2 Z_3$ for some non-trivial Z_2, (σ_1, σ_2) is a Type B signature with $(w_1, w_2) = \big(\log_{g_{p_2}}(Z_2) \cdot \sum_{i=1}^{n} a_i \cdot v_{l_2,i}, 0\big)$.

When $\mathcal{A}$ terminates, he outputs a class (ii) forgery $\sigma^\star = (\sigma_1^\star, \sigma_2^\star)$, a vector $\vec{y}^\star = (y_1^\star, \ldots, y_n^\star)$ and a file identifier $\tau^\star$. At this point, the distinguisher $\mathcal{B}$ halts and outputs a random bits if $\tau_{j^\star} \neq \tau^\star$ since it was unfortunate when guessing $j^\star \in_R \{1, \ldots, q\}$. Otherwise (*i.e.*, if $\tau^\star = \tau_{j^\star}$), by hypothesis, we have $\vec{y}^\star \notin \mathrm{span}(\vec{v}_1, \ldots, \vec{v}_{n-1})$, where $\vec{v}_1, \ldots, \vec{v}_{n-1}$ are the vectors queried for $\tau^\star = \tau_{j^\star}$.

In order to decide if $T = w^t Z_3$ or $T = w^t Z_2 Z_3$, $\mathcal{B}$ uses its input value $X_1 X_2$ to determine whether $\sigma^\star$ is a Type A or Type B forgery. To this end, it computes the value $\eta = \sigma_1^\star / \sigma_2^{\star a_u \tau^\star + a_v}$, which is of the form

$$\eta = w^{t \cdot \langle \vec{a}, \vec{y}^\star \rangle} \cdot g_{p_2}^{\tilde{\theta}} \cdot \tilde{R}_3, \tag{15}$$

for some $\tilde{\theta} \in \mathbb{Z}_{p_2}$ and $\tilde{R}_3 \in \mathbb{G}_{p_3}$. Then, $\mathcal{B}$ checks if the equality

$$e(X_1 X_2, \eta) = e(X_1 X_2, T)^{\langle \vec{a}, \vec{y}^\star \rangle} \tag{16}$$

holds. If it does, $\mathcal{B}$ decides that $\sigma^\star$ must be a Type A forgery and outputs 0 to indicate that $T = w^t Z_3$. If equation (16) does not hold, $\mathcal{B}$ rather bets that $\sigma^\star$ is a Type B forgery and outputs 1 (meaning that $T = w^t Z_2 Z_3$).

To see why $\mathcal{B}$ is a distinguisher with non-negligible advantage, let us first assume that $T = w^t Z_3$. In this case, the distinguisher $\mathcal{B}$ is playing $\mathrm{Game}_{4.(q,n-1)}$ with $\mathcal{A}$, so that previous lemmas imply that $\sigma^\star$ must be a Type A forgery unless Assumption 1 or Assumption 2 is false. If $\sigma^\star$ is of Type A, we have $\tilde{\theta} = 0$, so that the equality (16) is indeed satisfied and $\mathcal{B}$ outputs 0.

Now, let us assume that $T = w^t \cdot g_{p_2}^{\theta} \cdot R_3$ for some non-zero $\theta \in \mathbb{Z}_{p_2}^*$. In this case, $\mathcal{B}$ is playing Game_5 with $\mathcal{A}$. By hypothesis, the probability for $\sigma^\star$ to be a Type B forgery is now non-negligible, so that $\tilde{\theta} \neq 0$. Given that

$$e(X_1 X_2, T^{\langle \vec{a}, \vec{y}^\star \rangle}) = e(X_1, w^t) \cdot e(X_2, g_{p_2})^{\theta \cdot \langle \vec{a}, \vec{y}^\star \rangle},$$

relation (16) can only hold if $\tilde{\theta} = \theta \cdot \langle \vec{a}, \vec{y}^\star \rangle \bmod p_2$. However, by hypothesis, we also know that $\vec{y}^\star \bmod p_2 \notin \mathrm{span}(\vec{v}_1 \bmod p_2, \ldots, \vec{v}_{n-1} \bmod p_2)$. This means implies that the value $\langle \vec{a}, \vec{y}^\star \rangle \bmod p_2$ is independent of the information that $\mathcal{A}$ could possibly infer from signing queries via (14). It is thus completely unpredictable by $\mathcal{A}$ and the equality (16) is only satisfied with negligible probability. If $T = w^t Z_2 Z_3$, we thus conclude that $\mathcal{B}$ returns 1 with overwhelming probability. □

Lemma 7. *Any PPT algorithm $\mathcal{A}$ outputting a Type A forgery in Game_5 allows breaking Assumption 4.*

Proof. We specify an algorithm $\mathcal{B}$ that takes as input $\big(g, g^a, g^b, g^{ab} X_2, X_3, g^c Y_2, Z_2\big)$ and aims at computing $T = e(g,g)^{abc}$ using $\mathcal{A}$. To do so, $\mathcal{B}$ generates $\mathsf{pk} = \big(g, g^\alpha, u, v, \{g_i\}_{i=1,\ldots,n}, X_{p_3}\big)$ by choosing $a_u, a_v, a_1, \ldots, a_n \xleftarrow{R} \mathbb{Z}_N$ and setting $X_{p_3} = X_3$, $g^\alpha = g^a$ as well as $u = g^{a_u}$, $v = g^{a_v}$ and $g_i = (g^b)^{a_i}$ for $i = 1$ to n. We define $\vec{a} = (a_1, \ldots, a_n)$. Throughout the game, we also denote by $\tau_1, \ldots, \tau_q$ the tags appearing in $\mathcal{A}$'s signing queries. As in Game_5, $\mathcal{B}$ picks $j^\star \xleftarrow{R} \{1, \ldots, q\}$ as a guess for the index of $\tau^\star$. These signing queries $(\tau_{l_1}, \vec{v}_{l_2})$ are answered as follows.

[**Case** $l_1 \neq j^\star$:] To generate a signature for $\big(\tau_{l_1}, \vec{v}_{l_2} = (v_{l_2,1}, \ldots, v_{l_2,n})\big)$, $\mathcal{B}$ picks $r \xleftarrow{R} \mathbb{Z}_N$, $z_1, z_2 \xleftarrow{R} \mathbb{Z}_N$ and computes a Type-B.1 signature (σ_1, σ_2) as

$$\sigma_1 = (g^{ab} X_2)^{\sum_{i=1}^{n} a_i \cdot v_{l_2,i}} \cdot (u^\tau \cdot v)^r \cdot (Z_2 \cdot X_3)^{z_1}, \qquad \sigma_2 = g^r \cdot (Z_2 \cdot X_3)^{z_2}.$$

[**Case** $l_1 = j^\star$:] To answer signing queries of the form $\big(\tau_{j^\star}, \vec{v}_{l_2} = (v_{l_2,1}, \ldots, v_{l_2,n})\big)$, $\mathcal{B}$ picks $r \xleftarrow{R} \mathbb{Z}_N$, $z_1, z_2 \xleftarrow{R} \mathbb{Z}_N$ and computes a Type-B.2 signature (σ_1, σ_2) as

$$\sigma_1 = (g^{ab} X_2)^{\sum_{i=1}^n a_i \cdot v_{l_2,i}} \cdot (u^\tau \cdot v)^r \cdot (Z_2 \cdot X_3)^{z_1}, \qquad \sigma_2 = g^r \cdot X_3^{z_2}.$$

Observe that signatures are distributed exactly as in Game_5.

At the end of the game, $\mathcal{A}$ outputs a tag $\tau^\star$, a signature $\sigma^\star = (\sigma_1^\star, \sigma_2^\star)$ of Type-A and a vector $\vec{y}^\star$. At this point, $\mathcal{B}$ reports failure if $\tau^\star \neq \tau_{j^\star}$. Otherwise, algorithm $\mathcal{B}$ can compute

$$\eta = \Big(\frac{\sigma_1^\star}{\sigma_2^{\star a_u \tau^\star + a_v}}\Big)^{1/\langle \vec{a}, \vec{y}^\star \rangle}$$

since $\langle \vec{a}, \vec{y}^\star \rangle \neq 0$ unless the failure event introduced in Game_1 occurs. Given that $\sigma^\star$ is a Type A forgery, it must be of the form $(\sigma_1^\star, \sigma_2^\star) = \big((g^{ab})^{\langle \vec{a}, \vec{y}^\star \rangle} \cdot (u^{\tau^\star} \cdot v)^r \cdot R_3,\ g^r \cdot R_3'\big)$, so that $\eta = g^{ab} \cdot R_3''$, for some $R_3, R_3', R_3'' \in \mathbb{G}_{p3}$. This allows computing

$$T = e\big(g^c Y_2, \eta\big) = e(g,g)^{abc}. \tag{17}$$

To conclude, in Game_5, $\mathcal{A}$'s advantage is thus negligible if Assumption 4 holds. □

F Shorter Weakly Context Hiding Linearly Homomorphic Signatures from the Diffie-Hellman Assumption

This section describes a variant of the scheme in Section 4 that works over groups of prime order p. This variant is inspired by Freeman's construction [22] in that it appeals to a randomized vector hash system. Namely, in the first component of each signature, the first term is a vector hash of the form $(v^s \cdot \prod_{i=1}^n g_i^{v_i})^\alpha$, with $s \xleftarrow{R} \mathbb{Z}_p$ and where $\alpha \in \mathbb{Z}_p$ is the private key.

As a consequence of introducing a randomizer $s \in \mathbb{Z}_p$, the full context hiding property is lost and the scheme can only be proved weakly context hiding (in the sense of Definition 10 in Appendix A). On the other hand, the security of the resulting scheme can be proved under the standard Diffie-Hellman assumption. This gives us the shortest known Diffie-Hellman-based linearly homomorphic signatures in the standard model. In comparison with Freeman's CDH-based signatures [22], signatures are shortened by 25%.

As another advantage over [22], the homomorphic property can be obtained without using a pseudorandom function to sign all vectors of the same file using the same randomness. In the CDH-based construction of [22], a consequence of using a PRF is that individual signatures can be re-randomized in such a way that they still satisfy the verification algorithm but they cannot be combined any longer. In applications to network coding, it may be annoying since malicious nodes can somehow pollute the network by re-randomizing signatures and destroying their compatibility.

In [5], this issue was addressed by preventing signatures from being re-randomized in the appropriate subgroup. In prime-order groups, the same technique requires to apply a chameleon hash function (in the same way as in the Boneh-Shen-Waters signatures [14]), which results in signatures of 3 group elements and 2 elements of $\mathbb{Z}_p$. In comparison, we only need 2 elements of $\mathbb{G}$ and one element of $\mathbb{Z}_p$. We thus obtain the shortest signatures relying on a simple assumption.

Keygen(λ, n)**:** given a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathsf{poly}(\lambda)$, choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Choose $\alpha \xleftarrow{R} \mathbb{Z}_p$, $g, v \xleftarrow{R} \mathbb{G}$ and $u_0, u_1, \ldots, u_L \xleftarrow{R} \mathbb{G}$, for some

$L \in \mathsf{poly}(\lambda)$. These elements $(u_0, \ldots, u_L) \in \mathbb{G}^{L+1}$ will be used to implement a programmable hash function $H_{\mathbb{G}} : \{0,1\}^L \to \mathbb{G}$ such that any n-bit string $m = m[1] \ldots m[L] \in \{0,1\}^L$ has a hash value $H_{\mathbb{G}}(m) = u_0 \cdot \prod_{i=1}^{L} u_i^{m[i]}$. Pick $g_i \xleftarrow{R} \mathbb{G}$ for $i = 1$ to n. Finally, define the identifier space $\mathcal{T} := \{0,1\}^L$. The private key is $\mathsf{sk} := \alpha$ and the public key consists of

$$\mathsf{pk} := \Big((\mathbb{G}, \mathbb{G}_T),\ g,\ g^{\alpha},\ v,\ \{g_i\}_{i=1}^{n},\ \{u_i\}_{i=0}^{L} \Big).$$

Sign$(\mathsf{sk}, \tau, \vec{v})$**:** given a vector $\vec{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_p^n$, a file identifier $\tau \in \{0,1\}^L$ and the private key $\mathsf{sk} = \alpha \in \mathbb{Z}_p$, return $\perp$ if $\vec{v} = \vec{0}$. Otherwise, choose $r, s \xleftarrow{R} \mathbb{Z}_p$. Then, compute a signature $\sigma = (\sigma_1, \sigma_2, s) \in \mathbb{G}^2 \times \mathbb{Z}_p$ as

$$\sigma_1 = (g_1^{v_1} \cdots g_n^{v_n} \cdot v^s)^{\alpha} \cdot H_{\mathbb{G}}(\tau)^r, \qquad \sigma_2 = g^r.$$

SignDerive$(\mathsf{pk}, \tau, \{(\beta_i, \sigma_i)\}_{i=1}^{\ell})$**:** given pk, a file identifier τ and ℓ tuples (β_i, σ_i), parse each signature σ_i as $\sigma_i = (\sigma_{i,1}, \sigma_{i,2}, s_i)$ for $i = 1$ to ℓ. Then, choose $\tilde{r} \xleftarrow{R} \mathbb{Z}_p$ and compute

$$\sigma_1 = \prod_{i=1}^{\ell} \sigma_{i,1}^{\beta_i} \cdot H_{\mathbb{G}}(\tau)^{\tilde{r}} \qquad \sigma_2 = \prod_{i=1}^{\ell} \sigma_{i,2}^{\beta_i} \cdot g^{\tilde{r}} \qquad s = \sum_{i=1}^{\ell} \beta_i \cdot s_i$$

and output (σ_1, σ_2, s).

Verify$(\mathsf{pk}, \tau, \vec{y}, \sigma)$**:** given pk, a signature $\sigma = (\sigma_1, \sigma_2, s)$ and a message $(\tau, \vec{y})$, where $\tau \in \{0,1\}^L$ and $\vec{y} = (y_1, \ldots, y_n) \in (\mathbb{Z}_p)^n$, return 0 if $\vec{y} = \vec{0}$. Otherwise, return 1 if

$$e(\sigma_1, g) = e(g_1^{y_1} \cdots g_n^{y_n} \cdot v^s, g^{\alpha}) \cdot e(H_{\mathbb{G}}(\tau), \sigma_2). \tag{18}$$

and 0 otherwise.

It is easy to re-write the scheme in terms of Type-2 asymmetric pairings $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, for which an efficiently computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is available. Using Barreto-Naehrig curves [6], signature components σ_1 and σ_2 can both live in $\mathbb{G}_1$, so that each vector can be signed using 480 bits.

The security of the scheme relies on the standard computational Diffie-Hellman assumption, as established by the following theorem.

Theorem 5. *The above scheme is unforgeable if the CDH assumption holds in* $\mathbb{G}$.

The statement of the theorem is a consequence of Lemmas 8 and 9, which separately consider Class I and Class II forgeries. Here, since the scheme is only weakly context hiding, we have to consider the unforgeability game of Definition 3.

Lemma 8. *For any Class I forger* $\mathcal{A}$*, there is a forger* $\mathcal{B}$ *that breaks the security of Waters signatures with the same advantage. Consequently, for any Class I forger* $\mathcal{A}$*, there exists an algorithm* $\mathcal{B}$ *solving the CDH problem such that* $\mathbf{Adv}(\mathcal{A}) \leq 8 \cdot q \cdot (L+1) \cdot \big(\mathbf{Adv}^{\mathrm{CDH}}(\mathcal{B}) + \frac{1}{p}\big)$*, where* q *is the number of distinct tags appearing in signing queries.*

Proof. Recall that Waters signatures have a public key comprising a vector $(u_0, \ldots, u_L) \in \mathbb{G}^{L+1}$ and group elements $(g, g^\alpha, h) \in \mathbb{G}^3$. The private key $\alpha \in \mathbb{Z}_p$ is used to sign L-bit messages $m \in \{0,1\}^L$ as $(\sigma_1, \sigma_2) = (h^\alpha \cdot H_{\mathbb{G}}(m)^r, g^r)$, where $r \xleftarrow{R} \mathbb{Z}_p$ and $H_{\mathbb{G}}(m) = u_0 \cdot \prod_{i=1}^{L} u_i^{m[i]}$.

We show that, if an adversary $\mathcal{A}$ can produce a Class I forgery against the linearly homomorphic signature with non-negligible probability, there exists a forger $\mathcal{B}$ with the same advantage for Waters signatures. Algorithm $\mathcal{B}$ receives as input a Waters public key $(g, g^\alpha, h, \{u_i\}_{i=0}^{L})$. It defines $(g_1, \ldots, g_n, v)$ by setting $v = h^{\gamma_v}$ and $g_i = h^{\gamma_i} \cdot g^{\delta_i}$, for $i = 1$ to n, using random $\gamma_v, \gamma_1, \ldots, \gamma_n \xleftarrow{R} \mathbb{Z}_p$ and $\delta_1, \ldots, \delta_n \xleftarrow{R} \mathbb{Z}_p$. The adversary $\mathcal{A}$ is given $\mathsf{pk} = \big((\mathbb{G}, \mathbb{G}_T), g, g^\alpha, v, \{g_i\}_{i=1}^{n}, \{u_i\}_{i=0}^{L}\big)$.

For each signing query $(\tau, \vec{v} = (v_1, \ldots, v_n))$ made by $\mathcal{A}$, $\mathcal{B}$ queries its own signing oracle, requesting a Waters signature on the message $\tau = \tau[1] \ldots \tau[1] \in \{0,1\}^L$. The oracle returns a Waters signature $(\tilde{\sigma_1}, \tilde{\sigma_2}) = (h^\alpha \cdot H_{\mathbb{G}}(\tau)^r, g^r)$, which $\mathcal{B}$ transforms into a linearly homomorphic signature. To do this, $\mathcal{B}$ picks $s, r' \xleftarrow{R} \mathbb{Z}_p$, computes

$$(\sigma_1, \sigma_2) = \Big((g^\alpha)^{\sum_{i=1}^{n} \delta_i v_i} \cdot \tilde{\sigma_1}^{\gamma_v \cdot s + \sum_{i=1}^{n} \gamma_i \cdot v_i} \cdot H_{\mathbb{G}}(\tau)^{r'}, \tilde{\sigma_2}^{\gamma_v \cdot s + \sum_{i=1}^{n} \gamma_i \cdot v_i} \cdot g^{r'}\Big).$$

It is straightforward that $\sigma = (\sigma_1, \sigma_2, s)$ forms a valid signature on $(\tau, \vec{v})$. However, σ is not directly returned to $\mathcal{A}$. Instead, $\mathcal{B}$ chooses a handle and stores $(h, (\tau, \vec{v}), \sigma)$ in the table T.

As for SignDerive and Reveal queries, $\mathcal{B}$ always answers them in the same way as the real oracles, by inspecting the table T and updating it when necessary.

Eventually, $\mathcal{A}$ outputs a valid signature $(\sigma_1^\star, \sigma_2^\star, s^\star)$ on some message $(\tau^\star, \vec{v^\star} = (v_1^\star, \ldots, v_n^\star))$. Since $\mathcal{A}$ is a Class I forger, we know that $\tau^\star$ did not appear in any signing query. For this reason, $\mathcal{B}$ can forge a Waters signature by computing

$$(\sigma_1, \sigma_2) = \Big(\big(\sigma_1^\star \cdot (g^\alpha)^{-\sum_{i=1}^{n} \gamma_i v_i^\star}\big)^{1/(\gamma_v \cdot s^\star + \sum_{i=1}^{n} \gamma_i v_i^\star)}, {\sigma_2^\star}^{1/(\gamma_v \cdot s^\star + \sum_{i=1}^{n} \gamma_i v_i^\star)}\Big),$$

as long as $\gamma_v \cdot s^\star + \sum_{i=1}^{n} \gamma_i v_i^\star \neq 0$. However, this holds with overwhelming probability $1 - 1/p$ since $\{\gamma_i\}_{i=1}^{n}$ are completely independent of $\mathcal{A}$'s view. □

In the proof of Lemma 9, we will use the assumption that, for each identifier, the adversary only queries the signing oracle on independent vectors. We also assume that, for any given pair $(\tau, \vec{v})$, the signing oracle is invoked at most once. This restriction can be removed by making sure (*e.g.*, by computing s as a pseudorandom function of $(\tau, \vec{v})$) that the same exponent s is re-used if multiple queries involve the same pair $(\tau, \vec{v})$.

Lemma 9. *The scheme is secure against Class II forgeries if the CDH assumption holds in $\mathbb{G}$. More precisely, for any Class II forger $\mathcal{A}$, there is an algorithm $\mathcal{B}$ solving the CDH problem such that $\mathbf{Adv}(\mathcal{A}) \leq 8 \cdot q^2 \cdot (L+1) \cdot \big(\mathbf{Adv}^{\mathrm{CDH}}(\mathcal{B}) + \frac{1}{p}\big)$, where q is the number of distinct tags appearing in signing queries.*

Proof. Assuming that a Class II forger has a non-negligible advantage ε, we build an algorithm $\mathcal{B}$ for solving a CDH instance (g, g^a, g^b) with probability about $\varepsilon/(8q^2(L+1))$.

The reduction $\mathcal{B}$ chooses $(u_0, u_1, \ldots, u_L) \in \mathbb{G}^{L+1}$ in the same way as in the proof of Waters' signatures [40], in such a way that for any $\tau \in \{0,1\}^L$, $H_{\mathbb{G}}(\tau) = u_0 \cdot \prod_{i=1}^{L} u_i^{\tau[i]}$ can be written $H_{\mathbb{G}}(\tau) = (g^a)^{J(\tau)} \cdot g^{K(\tau)}$ for some integer-valued functions $J, K : \{0,1\}^L \to \mathbb{Z}_p$. These functions are perfectly invisible from the adversary's view. They are defined in such a way that $J(.)$ has relatively small absolute values and, for any distinct $\tau, \tau_1, \ldots, \tau_q$, it holds that $J(\tau) = 0 \bmod p$ and

$J(\tau_i) \neq 0 \bmod p$ for each $i \in \{1, \ldots, q\}$ with non-negligible probability $\delta = 1/(8 \cdot q \cdot (L+1))$.

Other components of the public key pk are defined as $g^\alpha = g^a$, $v = g^b$ and $g_i = (g^b)^{\gamma_i} \cdot g^{\rho_i}$ with $\gamma_i, \rho_i \xleftarrow{R} \mathbb{Z}_p$ for $i = 1$ to n.

By hypothesis, a Class II forger produces a forgery $(\tau^\star, \vec{y}^\star, \sigma^\star)$ for a file identifier $\tau^\star$ that previously appeared in a signing query but for which $\vec{y}^\star \notin \mathrm{span}(\vec{v}_1, \ldots, \vec{v}_{n-1})$, where $\vec{v}_1, \ldots, \vec{v}_{n-1}$ are the vectors queried for $\tau^\star$ (we assume w.l.o.g. that $\mathcal{A}$ makes exactly $n-1$ signing queries for $\tau^\star$ since, otherwise, $\mathcal{B}$ can simulate signing queries for itself). In the notations hereafter, we denote by $\tau_1, \ldots, \tau_q$ the distinct adversarially-chosen file identifiers successively appearing in $\mathcal{A}$'s queries during the game. At the outset of the game, $\mathcal{B}$ picks a random index $j^\star \xleftarrow{R} \{1, \ldots, q\}$, hoping that $\tau^\star$ will happen to be the $j^\star$-th distinct file identifier chosen by $\mathcal{A}$. Since $j^\star$ is drawn independently of $\mathcal{A}$'s view, $\mathcal{B}$'s guess will be successful (so that $\tau_{j^\star} = \tau^\star$) with probability $1/q$. During the game, $\mathcal{A}$'s queries are answered as follows.

Signing queries: at each signing query $\big(\tau_{l_1}, \vec{v} = (v_1, \ldots, v_n)\big)$ involving the l_1-th distinct tag τ_{l_1}, the reduction $\mathcal{B}$ considers the following cases.

- $l_1 \neq j^\star$: in this case, $\mathcal{B}$ evaluates the function $J(\tau_{l_1})$ and aborts if $J(\tau_{l_1}) = 0 \bmod p$. Otherwise (*i.e.*, if $J(\tau_{l_1}) \neq 0 \bmod p$), $\mathcal{B}$ picks $r, s \xleftarrow{R} \mathbb{Z}_p$ and computes

$$\sigma_1 = H_{\mathbb{G}}(\tau_{l_1})^r \cdot (g^b)^{-\frac{K(\tau_{l_1})}{J(\tau_{l_1})} \cdot (\langle \vec{\gamma}, \vec{v} \rangle + s)} \cdot (g^a)^{\langle \vec{\rho}, \vec{v} \rangle}, \qquad \sigma_2 = g^r \cdot (g^b)^{-\frac{\langle \vec{\gamma}, \vec{v} \rangle + s}{J(\tau_{l_1})}}.$$

  The triple (σ_1, σ_2, s) forms a valid signature on $\vec{v}$ since, if we define $\tilde{r} = r - \frac{b \cdot (\langle \vec{\gamma}, \vec{v} \rangle + s)}{J(\tau_{l_1})}$, we have

$$(\sigma_1, \sigma_2) = \Big((\prod_{i=1}^{n} g_i^{v_i} \cdot v^s)^a \cdot H_{\mathbb{G}}(\tau_{l_1})^{\tilde{r}},\ g^{\tilde{r}} \Big).$$

  The signature $\sigma = (\sigma_1, \sigma_2, s)$ is not directly returned to $\mathcal{A}$ but assigned to a new handle h and stored in an entry $(h, (\tau_{l_1}, \vec{v}), \sigma)$ of the table T.
- $l_1 = j^\star$: in this situation, $\mathcal{B}$ sets $s = -\langle \vec{\gamma}, \vec{v} \rangle$, chooses a handle h and stores $\big(h, m, (.,.,s)\big)$ in the table T.

Derivation queries: whenever $\mathcal{A}$ queries $\big((h_1, \ldots, h_k), m' = (\tau, \vec{v}')\big)$ to the $\mathsf{SignDerive}$ oracle, $\mathcal{B}$ returns $\perp$ if the handles $h_1, \ldots, h_k$ do not all correspond to queries involving τ. Otherwise, let $\vec{v}_1, \ldots, \vec{v}_k$ be the queried vectors and let $\beta_1, \ldots, \beta_k \in \mathbb{Z}_p$ be the coefficients such that $\vec{v}' = \sum_{i=1}^k \beta_i \cdot \vec{v}_i$. Then, $\mathcal{B}$ considers two cases.

- If τ is the $j^\star$-th tag appearing in signing queries, the handles $h_1, \ldots, h_k$ must correspond to entries of the form $(h_i, (\tau, \vec{v}_i), (.,.,s_i))$, for $i \in \{1, \ldots, k\}$, in the table T. In this case, $\mathcal{B}$ computes $s = \sum_{i=1}^k \beta_i \cdot s_i$, chooses a new handle h and stores $\big(h, (\tau, \vec{v}'), (.,.,s)\big)$ in the table T.
- If τ is *not* the $j^\star$-th tag involved in signing queries, $\mathcal{B}$ proceeds as the real $\mathsf{SignDerive}$ oracle.

Reveal queries: When $\mathcal{A}$ supplies a handle h, $\mathcal{B}$ returns $\perp$ if no entry of the form $(h, (\tau, \vec{v}), .)$ is found in T. Otherwise, we distinguish two cases.

- If τ is the $j^\star$-th tag showing up in signing queries, the handle h necessarily appears in an entry of the form $(h, (\tau, \vec{v}), (.,.,s))$. Then, $\mathcal{B}$ picks $r \xleftarrow{R} \mathbb{Z}_p$ and returns $\sigma = (\sigma_1, \sigma_2, s)$, where

$$(\sigma_1, \sigma_2) = \Big((g^a)^{\langle \vec{\rho}, \vec{v} \rangle} \cdot H_{\mathbb{G}}(\tau_{l_1})^r,\ g^r \Big).$$

It also adds $\big((\tau, \vec{v}), \sigma\big)$ in the list Q. Note that $\mathcal{A}$ can make at most $n-1$ reveal queries $(\tau_{j^\star}, \vec{v}_{l_2})$. For this reason, it will be given at most $n-1$ values $s_1, \ldots, s_{n-1} \in \mathbb{Z}_p$ such that $s_{l_2} = -\langle \vec{\gamma}, \vec{v}_{l_2} \rangle$ for each $l_2 \in \{1, \ldots, n-1\}$. Since the vector $\vec{\gamma}$ is independent of $\mathcal{A}$'s view, the values $\{s_{l_2}\}_{l_2=1}^{n-1}$ look independent to $\mathcal{A}$.

- If τ is *not* the $j^\star$-th distinct tag appearing in the game, the handle h must be in an entry of the form $(h, (\tau, \vec{v}), (\sigma_1, \sigma_2, s))$. In this case, $\mathcal{B}$ simply returns $\sigma = (\sigma_1, \sigma_2, s)$ and adds $\big((\tau, \vec{v}), \sigma\big)$ in the list Q.

Forgery: Eventually, the adversary $\mathcal{A}$ outputs a Class II forgery $(\tau^\star, \vec{y}^\star, \sigma^\star)$, where $\vec{y}^\star = (y_1^\star, \ldots, y_n^\star)$ and $\sigma^\star = (\sigma_1^\star, \sigma_2^\star, s^\star) \in \mathbb{G}^2 \times \mathbb{Z}_p$ satisfies the verification equation (18). At this point, $\mathcal{A}$ reports failure if $\tau^\star \neq \tau_{j^\star}$. It also evaluates $J(\tau^\star)$ and also fails if $J(\tau^\star) \neq 0$. However, we know that $\tau^\star = \tau_{j^\star}$ with probability $1/q$. The same analysis as in [40] also shows that, with probability $1/(8q(L+1))$, we have $J(\tau^\star) = 0$ and $J(\tau_{l_1}) \neq 0$ for each $l_1 \in \{1, \ldots, q\}$. It comes that $\mathcal{B}$'s probability not to fail during the whole game is at least $1/(8q^2(L+1))$.

If $\mathcal{B}$ does not fail, we have $H_{\mathbb{G}}(\tau^\star) = g^{K(\tau^\star)}$, so that $\mathcal{B}$ can compute

$$\eta^\star := \sigma_1^\star / \sigma_2^{\star K(\tau^\star)} = (\prod_{i=1}^{n} g_i^{y_i^\star} \cdot v^{s^\star})^a = \big((g^b)^{\langle \vec{\gamma}, \vec{y}^\star \rangle + s^\star} \cdot g^{\langle \vec{\gamma}, \vec{y}^\star \rangle}\big)^a$$

From $\eta^\star$, $\mathcal{B}$ can compute $g^{ab} = \big(\eta^\star / (g^a)^{\langle \vec{\gamma}, \vec{y}^\star \rangle}\big)^{1/(\langle \vec{\gamma}, \vec{y}^\star \rangle + s^\star)}$ as long as $s^\star \neq -\langle \vec{\gamma}, \vec{y}^\star \rangle$. By hypothesis, we know that $\vec{y}^\star \notin \mathrm{span}(\vec{v}_1, \ldots, \vec{v}_{n-1})$. Since $\mathcal{A}$ obtained at most $n-1$ values $\{\langle \vec{\gamma}, \vec{v}_{l_2} \rangle\}_{l_2=1}^{n-1}$, this implies that $\langle \vec{\gamma}, \vec{y}^\star \rangle$ is completely independent of $\mathcal{A}$'s view. It this case, we only have $s^\star = -\langle \vec{\gamma}, \vec{y}^\star \rangle$ with probability $1/p$. □

The weak context hiding property of the scheme can be proved using exactly the same argument as in [22][Appendix C.1]. Namely, the only information that could help the adversary in the game of Definition 10 is the signature component s. However, since the group order $p = |\mathbb{G}|$ is public, the value s is uniformly chosen in $\mathbb{Z}_p$ in original signatures. When signatures are being derived, the resulting s remains uniformly distributed in $\mathbb{Z}_p$. Consequently, when the original signatures are not available, a derived $s \in \mathbb{Z}_p$ carries no information about the original messages.