Block-wise P-Signatures and Non-Interactive Anonymous Credentials with Efficient Attributes

Malika Izabachène[1] *, Benoît Libert[2] **, and Damien Vergnaud[3] ***

[1] Université de Versailles and LSV - ENS Cachan/CNRS/INRIA (France)
[2] Université catholique de Louvain (Belgium)
[3] École Normale Supérieure – C.N.R.S. - INRIA (France)

Abstract. Anonymous credentials are protocols in which users obtain certificates from organizations and subsequently demonstrate their possession in such a way that transactions carried out by the same user cannot be linked. We present an anonymous credential scheme with non-interactive proofs of credential possession where credentials are associated with a number of attributes. Following recent results of Camenisch and Groß (CCS 2008), the proof simultaneously convinces the verifier that certified attributes satisfy a certain predicate. Our construction relies on a new kind of P-signature, termed *block-wise P-signature*, that allows a user to obtain a signature on a committed vector of messages and makes it possible to generate a short witness that serves as a proof that the signed vector satisfies the predicate. A non-interactive anonymous credential is obtained by combining our *block-wise* P-signature scheme with the Groth-Sahai proof system. It allows efficiently proving possession of a credential while simultaneously demonstrating that underlying attributes satisfy a predicate corresponding to the evaluation of inner products (and therefore disjunctions or polynomial evaluations). The security of our scheme is proved in the standard model under non-interactive assumptions.

Keywords. P-signatures, anonymous credentials, efficient attributes, non-interactive proofs, standard model.

1 Introduction

Introduced by Chaum [20] and extensively studied in the last two decades (*e.g.* [17–19, 4, 3] and references therein) anonymous credential systems enable users to authenticate themselves in a privacy-preserving manner. In such a protocol, a user can prove that an organization has supplied him with a certificate in such

* The work of this author was supported by the French ANR-07-TLCOM-04-COPRIM Project.

** This author acknowledges the Belgian Fund for Scientific Research for his "Chargé de recherches" fellowship and the BCRYPT Interuniversity Attraction Pole.

*** The work of this author was supported by the French ANR-07-TCOM-013-04 PACE Project and by the European Commission through the ICT Program under Contract ICT-2007-216676 ECRYPT II.

a way that the request for a certificate cannot be linked to any of its proofs of possession and multiple proofs involving the same credential cannot be linked to each other. In many realistic applications, it is desirable to augment digital credentials with a number of user attributes (such as their citizenship, their birth date, their obtained degrees, ...) while allowing users to selectively disclose some of their attributes or efficiently prove properties about them without disclosing any other information. This problem was addressed by Camenisch and Groß [14] who showed how to conveniently extend the Camenisch-Lysyanskaya construction [17, 18] into an anonymous credential system with efficient attributes. In this paper, we consider similar problems in the context of *non-interactive* anonymous credentials in the standard model, as formalized in [4].

Anonymous credential systems usually combine two essential components. The first one is a protocol allowing a user to obtain a signature from an organization on a committed value (which is typically the user's private key) by sending a commitment to the signer and eventually obtaining a signature on the message without leaking useful information on the latter. The second component is a proof of knowledge of a signature on a committed value. Namely, the prover holds a pair (m, σ), reveals a commitment c to m and demonstrates his possession of σ as a valid signature on m.

Prior Work. Camenisch and Lysyanskaya [17, 18] used groups of hidden order and Fujisaki-Okamoto commitments [26] to build the first practical realizations 10 years ago. Their approach was subsequently extended to groups of public order using bilinear maps [19, 2].

Until recently, all anonymous credential systems required users to engage in an interactive conversation with the verifier to convince him of their possession of a credential. While interaction can be removed using the Fiat-Shamir paradigm [23] and the random oracle model [6], this methodology is limited to only give heuristic arguments in terms of security [28]. This motivated Belenkiy, Chase, Kohlweiss and Lysyanskaya [4] to design non-interactive[1] anonymous credentials in the standard model – assuming a common reference string – using an underlying primitive named *P-signature* (as a shorthand for signatures with efficient Protocols). Their results were extended by [5] (and, more recently, in [25]) into non-interactive anonymous credential schemes supporting credential delegation.

Credentials Supporting Efficient Attributes. Users holding a number of certified attributes may be willing to selectively disclose a restricted number of their attributes while preserving their privacy and the secrecy of their other attributes. A natural approach is to extend classical anonymous credentials such as [17, 19] using generalizations of the Pedersen commitment [36] allowing to commit to n attributes at once in groups of hidden order. However, disclosing a single specific attribute entails to commit to $n - 1$ attributes so as to prove that one attribute matches the disclosed value and committed attributes are the remaining certified ones. The drawback of this technique is that each proof has

[1] The protocol for obtaining a signature on a committed message still demands interaction but the proving phase, which is usually more frequently executed, consists of one message from the prover to the verifier.

linear size in the overall number of attributes.

To address this concern, Camenisch and Groß [14] suggested a completely different technique consisting in encoding attributes as prime numbers. Basically, users first obtain a signature on two committed messages: the first one is the user's private key and the second one consists of the product of all users' attributes. Later on, when the user wants to prove his ownership of a credential containing a certain attribute, he just has to prove that this attribute divides the second committed message. Camenisch and Groß also showed how users can prove that they hold an attribute appearing in some public attribute list and how to handle negated statements (namely, prove that a certain attribute is not contained in their attribute set). They also showed how to extend their techniques and prove the conjunction or the disjunction of simple such atomic statements. Unfortunately, their techniques cannot be applied in the setting of non-interactive anonymous credentials as they inherently rely on groups of hidden order, which makes them hardly compatible with the Groth-Sahai proof systems [29] used in [4, 5]. It turns out that efficiently handling attributes in this context requires new techniques to be worked out.

In [39], Shahandashti and Safavi-Naini used threshold attribute-based signatures [35] to construct attribute-based anonymous credentials where users can prove threshold predicates (*i.e.*, the ownership of t-out-of-n public attributes). However, their construction requires interaction and is not meant to provide compact proofs, which is the focus of this paper.

Our Contribution. This paper presents an anonymous credential scheme allowing to non-interactively prove the possession of a credential associated with attributes that satisfy a given predicate without leaking any further information. To this end, we extend the approach of [4] by introducing a new kind of P-signature termed *block-wise* P-signature. In a nutshell, such a primitive is a P-signature allowing a user to obtain a signature on a committed vector of messages (similarly to the multi-block P-signature of [5]). Unlike [5] however, our P-signature makes it possible for the user to generate a short NIZK argument (*i.e.*, the size of which does not depend on the vector size) that serves as evidence that the signed vector satisfies a certain predicate.

Inspired by the work of Katz, Sahai, Waters [33], we present a block-wise P-signature for predicates corresponding to the zero or non-zero evaluation of inner products (and therefore disjunctions or polynomial evaluations). By combining our block-wise P-signature with the Groth-Sahai methodology [29] as in [4], we readily obtain an efficient non-interactive anonymous credential supporting efficient attributes. By appropriately using the inner product with suitable attribute encodings, we notably obtain (1) an efficient way for users to prove that specific attributes appear in their attribute set; (2) a method for concisely proving the inclusion of one of their attributes in a public list; (3) short proofs that the certified attribute set contains a certain (exact or inexact) threshold of binary attributes (in a similar way, we can prove that a subset of the certified set is at most t binary attributes away from some public attribute set). Using a very small amount of interaction (namely, verifiers just have to send a challenge

consisting of a short random value in $\mathbb{Z}_p$, where p is the group order), we can also handle conjunctions of atomic conditions and even more complex formulas such as CNF or DNF in two rounds. The non-interactivity property is unfortunately lost when we want to deal with CNF/DNF formulas but our solution still decreases the number of rounds w.r.t. traditional interactive constructions. Indeed, at least 3 rounds are needed in interactive proofs using Σ protocols.

The security of our scheme is proved in the standard model under non-interactive assumptions. Although our scheme does not perform as well as the Camenisch-Groß system (notably because, unlike [14], we cannot prevent the public key size from depending on the number n of attributes), this yields the first result on non-interactive anonymous credentials with efficient attributes in the standard model. Like [4, 5], we rely on a common reference string and only need interaction in the protocol allowing users to obtain their credentials (except for predicates involving conjunctions).

Organization. In section 2, we first give formal definitions of block-wise F-unforgeable signatures (similarly to [4], we can only prove a relaxed form of unforgeability which suffices in this context) and block-wise P-signatures. Our realization for inner product relations is described in section 3. Its application to the realization of anonymous credentials with efficient attributes is detailed in the full version of the paper, where we also discuss the efficiency of the scheme and the kind of predicates that can be expressed using inner products.

2 Background and Definitions

Notations. We say that a function $\nu : \mathbb{N} \to [0, 1[$ is negligible if for, any polynomial $p(.)$, we have $\nu(\lambda) < |1/p(\lambda)|$ for any sufficiently large $\lambda \in \mathbb{N}$. If $A(x) \leftrightarrows B(y)$ denotes an interactive protocol between A and B on input x and y, respectively, and if participant A (resp. B) outputs a bit $b \in \{0, 1\}$ after the execution of the protocol, we write $b \Leftarrow A(x) \leftrightarrows B(y)$ (resp. $A(x) \leftrightarrows B(y) \Rightarrow b$).

2.1 Bilinear Maps and Complexity Assumptions

We consider bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p with a mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that: (1) $e(g^a, h^b) = e(g, h)^{ab}$ for any $(g, h) \in \mathbb{G} \times \mathbb{G}$ and $a, b \in \mathbb{Z}$; (2) $e(g, h) \neq 1_{\mathbb{G}_T}$ whenever $g, h \neq 1_{\mathbb{G}}$.

Definition 1 ([9]). *In a group $\mathbb{G}$ of prime order p, the* **Decision Linear Problem** *(DLIN) is to distinguish the distributions $(g, g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ and $(g, g^a, g^b, g^{ac}, g^{bd}, g^z)$, with $a, b, c, d, z \xleftarrow{R} \mathbb{Z}_p$. The* **Decision Linear Assumption** *is the intractability of DLIN for any PPT distinguisher $\mathcal{D}$.*

This problem is to decide if three vectors $\vec{g_1} = (g^a, 1, g)$, $\vec{g_2} = (1, g^b, g)$ and $\vec{g_3} = (g^{ac}, g^{bd}, g^z)$ are linearly dependent (*i.e.*, if $z = c + d$).

Like several previous P-signatures, our scheme uses the Hidden Strong Diffie-Hellman assumption [11] that strengthens a "q-type" assumption from [8].

Definition 2 ([11]). *The* q**-Hidden Strong Diffie-Hellman problem** *(q-HSDH) consists in, given* $(g, u, g^\omega) \in \mathbb{G}^3$ *and a set of* q *tuples* $(g^{1/(\omega+c_i)}, g^{c_i}, u^{c_i})$ *with* $c_1, \ldots, c_q \xleftarrow{R} \mathbb{Z}_p^*$*, finding* $(g^{1/(\omega+c)}, g^c, u^c)$ *such that* $c \neq c_i$ *for* $i = 1, \ldots, q$*.*

We also use the following problem, which is not easier than the problem, used in [32], of finding a pair $(g^\mu, g^{\mu ab}) \in (\mathbb{G}\backslash\{1_\mathbb{G}\})^2$ given $(g, g^a, g^b) \in \mathbb{G}^3$.

Definition 3 ([32]). *The* **Flexible Diffie-Hellman problem** *(FlexDH) in* $\mathbb{G}$ *is, given* $(g, g^a, g^b) \in \mathbb{G}^3$*, where* $a, b \xleftarrow{R} \mathbb{Z}_p^*$*, to find a triple* $(g^\mu, g^{\mu a}, g^{\mu ab})$ *such that* $\mu \neq 0$*.*

The paper will make use of two other problems. The first one was introduced – in a potentially easier variant – in [10].

Definition 4 ([10]). *Let* $\mathbb{G}$ *be a group of prime order* p*. The* n**-Diffie-Hellman Exponent** *(n-DHE) problem is, given* $(g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}) \in \mathbb{G}^{2n}$ *such that* $g_i = g^{(\alpha^i)}$ *for each* $i \in [1, 2n]\backslash\{n+1\}$ *and where* $\alpha \xleftarrow{R} \mathbb{Z}_p^*$*, to compute the missing element* $g_{n+1} = g^{(\alpha^{n+1})}$*.*

We finally need an assumption that strengthens the n-DHE assumption in the same way as the FlexDH assumption is a strengthening of the Diffie-Hellman assumption.

Definition 5. *Let* $\mathbb{G}$ *be a group of prime order* p*. The* **Flexible** n**-Diffie-Hellman Exponent** *(n-FlexDHE) problem is, given* $(g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n})$ *in* $\mathbb{G}^{2n}$ *such that* $g_i = g^{(\alpha^i)}$ *for each* $i \in [1, 2n]\backslash\{n+1\}$ *and where* $\alpha \xleftarrow{R} \mathbb{Z}_p^*$*, to compute a non-trivial triple* $(g^\mu, g_{n+1}^\mu, g_{2n}^\mu) \in (\mathbb{G}\backslash\{1_\mathbb{G}\})^3$*, for some* $\mu \in \mathbb{Z}_p^*$ *and where* $g_{n+1} = g^{(\alpha^{n+1})}$*.*

Evidence of the generic intractability of the n-FlexDHE assumption is provided in the full version of the paper.

2.2 Commitments to Vectors

We consider perfectly hiding commitments (VecCom, VecOpen) allowing to commit to vectors. In the following, we denote by $V = \text{VecCom}(\vec{m}; r)$ the result of committing to $\vec{m} = (m_1, \ldots, m_n) \in \mathbb{Z}_p^n$ using randomness $r \xleftarrow{R} \mathbb{Z}_p$. In addition, we require that commitments be openable in a coordinate-wise manner and call $W = \text{VecOpen}(\vec{m}, r, i)$ the opening of V in position $i \in [1, n]$. Such a pairing-based Pedersen-like commitment [36], based on ideas from [10, 16], was described in [34]. The commitment key is $(g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}) \in \mathbb{G}^{2n}$ where $g_i = g^{(\alpha^i)}$ for each i. To commit to a vector $\vec{m} = (m_1, \ldots, m_n)$, the committer picks $r \xleftarrow{R} \mathbb{Z}_p$ and computes $V = g^r \cdot \prod_{j=1}^n g_{n+1-j}^{m_j}$. Thanks to the specific choice of $\{g_i\}_{i\in[1,2n]\backslash\{n+1\}}$, $W_i = g_i^r \cdot \prod_{j=1, j\neq i}^n g_{n+1-j+i}^{m_j}$ serves as evidence that m_i is the i-th component of $\vec{m}$ as it satisfies the relation $e(g_i, V) = e(g, W_i) \cdot e(g_1, g_n)^{m_i}$. The opening $W_i = \text{VecOpen}(V, \vec{m}, r, i)$ at position i is easily seen not to reveal anything about other components of $\vec{m}$. Moreover, the infeasibility of opening a commitment to two distinct messages for some coordinate $i \in [1, n]$ relies on the n-DHE assumption.

2.3 Block-wise F-Unforgeable Signatures and P-Signatures

We begin by defining block-wise F-unforgeable signatures. As introduced in [4], F-unforgeability refers to the infeasibility for the adversary to craft a valid signature on some message $m \in Z_p$ while only outputting $F(m)$, for some injective function F, instead of m itself. The need for such a relaxation stems from the limited extractability of Groth-Sahai proofs: in a nutshell, only g^m is efficiently extractable from a commitment to $m \in \mathbb{Z}_p$ using the trapdoor of the CRS.

For reasons that will become apparent later on, block-wise F-unforgeable signatures will be associated with two families of relations that we call $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. These two families are not explicitly used in the following definition but they are handy when it comes to formalize security properties.

Definition 6. *Let $\mathcal{D}$ be a domain and let $\mathcal{R}_1$ and $\mathcal{R}_2$ be families of efficiently computable relations such that each $R \in \mathcal{R}_1 \cup \mathcal{R}_2$ is of the form $R : [0,n] \times \mathcal{D}^n \times \mathcal{D}^n \to \{0,1\}$ for some $n \in \mathbb{N}$. A block-wise signature for $(\mathcal{R}_1, \mathcal{R}_2, \mathcal{D})$ consists of a tuple $\Sigma = (\mathsf{Setup}, \mathsf{SigSetup}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Witness\text{-}Gen}, \mathsf{Witness\text{-}Verify})$ of algorithms with the following specifications.*

Setup(λ)**:** *takes as input a security parameter λ and outputs a set of public parameters* params*.*

SigSetup(λ, n)**:** *takes as input a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathsf{poly}(\lambda)$ denoting the length of message vectors to be signed. It outputs a key pair $(\mathsf{pk}, \mathsf{sk})$.*

Sign$(\mathsf{sk}, \vec{m})$**:** *is a (possibly randomized) algorithm that takes as input a private key* sk *and a vector $\vec{m} = (m_1, \ldots, m_n)$ of messages where $m_i \in \mathcal{D}$ for $i = 1$ to n. It outputs a signature σ on $\vec{m}$.*

Verify$(\mathsf{pk}, \vec{m}, \sigma)$**:** *is a deterministic algorithm that takes as input a public key* pk*, a signature σ and a message vector $\vec{m} = (m_1, \ldots, m_n)$. It outputs 1 if σ is deemed valid for $\vec{m}$ or 0 otherwise.*

Witness-Gen$(\mathsf{pk}, R, i, \vec{m}, \vec{X}, \sigma)$**:** *takes as input a public key* pk*, a relation $R \in \mathcal{R}_1 \cup \mathcal{R}_2$, an integer $i \in [0,n]$, two distinct vectors $\vec{m} = (m_1, \ldots, m_n) \in \mathcal{D}^n$ and $\vec{X} = (x_1, \ldots, x_n) \in \mathcal{D}^n$, and a signature σ. If $\mathsf{Verify}(\mathsf{pk}, \vec{m}, \sigma) = 0$ or $R(i, \vec{m}, \vec{X}) = 0$, it outputs $\perp$. Otherwise, it returns a witness W proving that σ is a signature on some $\vec{m} \in \mathcal{D}^n$ s.t. $R(i, \vec{m}, \vec{X}) = 1$.*

Witness-Verify$(\mathsf{pk}, R, i, \vec{X}, W, \sigma)$**:** *is a deterministic algorithm that takes in a public key* pk*, a relation $R \in \mathcal{R}_1 \cup \mathcal{R}_2$, an integer $i \in [0,n]$, a vector $\vec{X} \in \mathcal{D}^n$, a witness W and a signature σ. It outputs 1 if W is deemed as convincing evidence that σ is a valid signature on some vector $\vec{m} = (m_1, \ldots, m_n) \in \mathcal{D}^n$ such that $R(i, \vec{m}, \vec{X}) = 1$.*

Except Setup, these algorithms all implicitly take public parameters params as additional inputs. To lighten notations, we omit to explicitly write them.

The following security definitions consider two kinds of forger. The first one – which corresponds to case (i) in the definition – refers to attacks where the adversary outputs a new signature that was not legally obtained by invoking the signing oracle. The second one – captured by case (ii) – relates to forgeries

where the adversary re-uses a signature (say σ_j for some $j \in \{1, \ldots, q\}$) that *was* produced by the signing oracle but manages to prove a property that is *not* satisfied by the signed vector $\vec{m}_j$.

In case (ii), we need to consider two families of relations. The first one is called $\mathcal{R}_1$ and includes relations R_1 for which the adversary illegitimately proves that $R_1(i, \vec{m}_j, \vec{X}^\star) = 1$ and only outputs $F(\vec{X}^\star) = (F(x_1^\star), \ldots, F(x_n^\star))$. The second relation family $\mathcal{R}_2$ comprises relations R_2 for which the adversary tricks the verifier into believing that $R_2(i, \vec{m}_j, \vec{X}^\star) = 1$ and explicitly outputs $\vec{X}^\star = (x_1^\star, \ldots, x_n^\star)$ instead of $F(\vec{X}^\star)$. We cannot consider a single relation family unifying both $\mathcal{R}_1$ and $\mathcal{R}_2$ because, for technical reasons, our security proof ceases to work if the adversary only outputs $F(\vec{X}^\star)$ in the case of relations $R_2 \in \mathcal{R}_2$ (as explained in the full version of the paper). At the same time, we also need relations $R_1 \in \mathcal{R}_1$ because of the limited extractability properties of Groth-Sahai proofs.

In the notations of Definition 7, $\Upsilon \subset \{1, \ldots, n\}$ denotes the smallest subset such that values $\{F(x_t)\}_{t \in \Upsilon}$ make it possible to verify that $\vec{X} = (x_1, \ldots, x_n)$ satisfies $R_1(i, \vec{m}, \vec{X}) = 1$.

Definition 7. *Let $\mathcal{R}_1, \mathcal{R}_2$ be families of relations over $[0, n] \times \mathcal{D}^n \times \mathcal{D}^n$ for some domain $\mathcal{D}$. A block-wise signature scheme Σ is said to be $(F, \mathcal{R}_1, \mathcal{R}_2)$-***unforgeable** *for some efficiently computable injective function $F(.)$, if any PPT adversary has negligible advantage in the following game:*

1. *The challenger runs $\mathsf{SigSetup}(\lambda, n)$, obtains $(\mathsf{pk}, \mathsf{sk})$ and sends pk to $\mathcal{A}$.*
2. *Adversary $\mathcal{A}$ adaptively queries a signing oracle on up to $q \in \mathsf{poly}(\lambda)$ occasions. At each query $j \in [1, q]$, $\mathcal{A}$ chooses a vector $\vec{m} = (m_1, \ldots, m_n)$ and obtains $\sigma_j = \mathsf{Sign}(\mathsf{sk}, \vec{m})$.*
3. *Eventually, $\mathcal{A}$ outputs a tuple $(\mathsf{Pred}^\star, W^\star, \sigma^\star)$ consisting of a predicate $\mathsf{Pred}^\star$, a witness $W^\star$ and a signature $\sigma^\star$. The predicate $\mathsf{Pred}^\star$ consists of a triple which is either of the form $(R_1, i, \{F(x_t^\star)\}_{t \in \Upsilon})$, for some subset $\Upsilon \subset \{1, \ldots, n\}$ such that $i \in \Upsilon$, or $(R_2, i, \vec{X}^\star)$ where $i \in [0, n]$ is an index, $R_1 \in \mathcal{R}_1$ and $R_2 \in \mathcal{R}_1 \cup \mathcal{R}_2$ are relations and $\vec{X}^\star = (x_1^\star, \ldots, x_n^\star) \in \mathcal{D}^n$ is a vector. The adversary wins if: (a) $\mathsf{Witness\text{-}Verify}(\mathsf{pk}, R, i, \vec{X}^\star, W^\star, \sigma^\star) = 1$. (b) It holds that either:*

   (i) $\sigma^\star$ was not the output of any signing query;

   (ii) $\sigma^\star = \sigma_j$, for some query $j \in [1, q]$, but the queried $\vec{m}_j = (m_{j,1}, \ldots, m_{j,n})$ was such that $R_1(i, \vec{m}_j, \vec{X}^\star) = 0$ (resp. $R_2(i, \vec{m}_j, \vec{X}^\star) = 0$) while the predicate $\mathsf{Pred}^\star$ is of the form $(R_1, i, \{F(x_t^\star)\}_{t \in \Upsilon})$ (resp. $(R_2, i, \vec{X}^\star)$).

   The advantage of adversary $\mathcal{A}$ is its probability of being successful, taken over all random coins.

From a block-wise F-unforgeable signature, a full-fledged block-wise P-signature is obtained as specified by Definition 8.

Definition 8. *A block-wise P-signature combines a $(F, \mathcal{R}_1, \mathcal{R}_2)$-unforgeable block-wise signature with a vector commitment* $(\mathrm{VecCom}, \mathrm{VecOpen})$*, a perfectly binding commitment* $(\mathsf{Com}, \mathsf{Open})$ *and:*

1. *An algorithm* $\mathsf{SigProve}_1\big(\mathsf{pk}, R_1, i, \Upsilon, \sigma, \vec{m} = (m_1, \ldots, m_n), \vec{X} = (x_1, \ldots, x_n)\big)$ *that, for some relation* $R_1 \in \mathcal{R}_1$ *and some subset* $\Upsilon \subset \{1, \ldots, n\}$ *such that* $i \in \Upsilon$*, generates commitments* $\{C_{x_t}\}_{t\in\Upsilon}$*,* C_W*,* C_σ *and a NIZK proof*

$$\pi \leftarrow \mathsf{NIZPK}\big(\{x_t \text{ in } C_{x_t}\}_{t\in\Upsilon},\ W \text{ in } C_W,\ \sigma \text{ in } C_\sigma \mid \{(W, \{F(x_t)\}_{t\in\Upsilon}, \sigma) :$$
$$\exists\ \vec{m} \text{ s.t. } \mathsf{Verify}(\mathsf{pk}, \vec{m}, \sigma) = 1 \ \wedge\ \mathsf{Witness\text{-}Verify}(\mathsf{pk}, R_1, i, \vec{X}, W, \sigma) = 1\}\big),$$

and the corresponding $\mathsf{VerifyProof}_1(\mathsf{pk}, R_1, i, \pi, C_\sigma, C_W, \{C_{x_t}\}_{t\in\Upsilon})$ *algorithm.*

2. *An algorithm* $\mathsf{SigProve}_2(\mathsf{pk}, R, i, \sigma, \vec{m}, \vec{X})$ *that, for some relation* $R \in \mathcal{R}_1 \cup \mathcal{R}_2$*, generates commitments* C_W*,* C_σ *and a proof*

$$\pi \leftarrow \mathsf{NIZPK}\big(W \text{ in } C_W,\ \sigma \text{ in } C_\sigma \mid \{(W, \sigma) : \exists\ \vec{m} \text{ s.t. } \mathsf{Verify}(\mathsf{pk}, \vec{m}, \sigma) = 1$$
$$\wedge\ \mathsf{Witness\text{-}Verify}(\mathsf{pk}, R, i, \vec{X}, W, \sigma) = 1\}\big)$$

with its corresponding $\mathsf{VerifyProof}_2(\mathsf{pk}, R, i, \pi, C_\sigma, C_W, \vec{X})$ *algorithm.*

3. *A NIZK proof that two perfectly binding commitments open to the same value, i.e., an algorithm* $\mathsf{EqComProve}$ *outputting a proof of membership for the language*

$$L = \{(C, D) \text{ s.t. } \exists\ (x, y), (open_x, open_y) \mid$$
$$C = \mathsf{Com}(x, open_x) \ \wedge\ D = \mathsf{Com}(y, open_y) \ \wedge\ x = y\}.$$

4. $\mathsf{SigIssue}\big(\mathsf{sk}, V', (m_{n_1+1}, \ldots, m_n)\big) \leftrightarrows \mathsf{SigObtain}(\mathsf{pk}, \vec{m}_{|_{n_1}}, open_{\vec{m}_{|_{n_1}}})$ *is an interactive protocol allowing a user to obtain a signature* σ *on the partially committed vector* $\vec{m} = (m_1, \ldots, m_{n_1}, m_{n_1+1}, \ldots, m_n)$ *without letting the signer – whose input consists of* $V' = \mathrm{VecCom}(\vec{m}_{|_{n_1}}, r')$*, for some* r'*, and an integer* $n_1 \in [1, n]$*, and public messages* $(m_{n_1+1}, \ldots, m_n)$ *– learn anything about* $\vec{m}_{|_{n_1}} = (m_1, \ldots, m_{n_1})$*.*

In this definition, $\Upsilon \subset \{1, \ldots, n\}$ is the smallest subset such that commitments $\{C_{x_t}\}_{t\in\Upsilon}$ allow verifying the proof that the underlying vector $\vec{X}$ satisfies $R_1(i, \vec{m}, \vec{X}) = 1$.

Unforgeability of P-signatures. To define the unforgeability of block-wise P-signatures, we shall assume that $\mathsf{SigIssue} \leftrightarrows \mathsf{SigObtain}$ starts with the user $\mathcal{U}$ committing to a vector $(m_1, \ldots, m_{n_1})$ and interactively proving to the issuer his knowledge of an opening of the commitment. We require the existence of a knowledge extractor $\mathcal{E}^{\mathcal{A}}_{\mathsf{SigObtain}}$ that can extract the committed vector $(m_1, \ldots, m_{n_1})$ by rewinding the prover $\mathcal{A}$. Since (VecCom, VecOpen) is a perfectly hiding commitment, this will be necessary to formalize the unforgeability of our P-signatures. We note that a similar approach was taken in [15] to define specific security properties of e-cash systems.

Definition 9. *A block-wise P-signature* Σ *is* $(F, \mathcal{R}_1, \mathcal{R}_2)$-**unforgeable**, *for relation families* $\mathcal{R}_1, \mathcal{R}_2$*, if there are efficient algorithms* $(\mathsf{ExtractSetup}, \mathsf{Extract})$ *s.t. (i) the output distributions of* Setup *and* $\mathsf{ExtractSetup}$ *are statistically close; (ii) any PPT algorithm* $\mathcal{A}$ *has negligible advantage in the following game:*

1. *The challenger runs* $\mathsf{params} \leftarrow \mathsf{ExtractSetup}(\lambda)$ *and* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{SigSetup}(\lambda, n)$*, for some integer* $n \in \mathsf{poly}(\lambda)$*, and hands* pk *to* $\mathcal{A}$*.*
2. *On up to* $q \in \mathsf{poly}(\lambda)$ *occasions,* $\mathcal{A}$ *triggers an execution of* $\mathsf{SigIssue} \leftrightarrows \mathsf{SigObtain}$ *and acts as a user interacting with the* $\mathsf{SigIssue}$*-executing challenger. At each such execution* $j \in [1, q]$*, the challenger runs* $\mathcal{E}^{\mathcal{A}}_{\mathsf{SigObtain}}$ *so as to extract* $\mathcal{A}$*'s vector* $\vec{m}_j = (m_{j,1}, \ldots, m_{j,n})$ *(or, more precisely, the restriction* $(m_{j,1}, \ldots, m_{j,n_1})$ *to its first* n_1 *coordinates, for some* $n_1 \in [1, n]$*) and bookkeeps it. We denote by* σ_j *the signature obtained by* $\mathcal{A}$ *at the end of the* j*-th execution of* $\mathsf{SigObtain}$*.*
3. $\mathcal{A}$ *outputs commitments* C_σ*,* C_W*, a proof* π *and a statement* claim *consisting of a triple which is either of the form* $(R_1, i, \{C_{x_t}\}_{t\in\Upsilon})$ *or* $(R_2, i, \vec{X})$*, for some integer* $i \in [0, n]$*, some relations* $R_1 \in \mathcal{R}_1$ *or* $R_2 \in \mathcal{R}_1 \cup \mathcal{R}_2$*, some vector* $\vec{X} = (x_1, \ldots, x_n) \in \mathcal{D}^n$ *or some commitments* $\{C_{x_t}\}_{t\in\Upsilon}$ *– for some subset* $\Upsilon \subset \{1, \ldots, n\}$ *– to elements* $x_t \in \mathcal{D}$*. The adversary is successful if:*
   a. *Exactly one of the following conditions is satisfied.*
      1. $\mathsf{claim} = (R_1, i, \{C_{x_t}\}_{t\in\Upsilon})$ *and*
      $$\mathsf{VerifyProof}_1(\mathsf{pk}, R_1, i, \pi, C_\sigma, C_W, \{C_{x_t}\}_{t\in\Upsilon}) = 1.$$
      2. $\mathsf{claim} = (R_2, i, \vec{X})$ *and* $\mathsf{VerifyProof}_2(\mathsf{pk}, R_2, i, \pi, C_\sigma, C_W, \vec{X}) = 1$*.*
   b. *If we define* Pred *to be* $(R, i, \{\mathsf{Extract}(C_{x_t})\}_{t\in\Upsilon})$ *in situation 1 and simply* claim *in situation 2, the triple* $\big(\mathsf{Pred}, \mathsf{Extract}(C_W), \mathsf{Extract}(C_\sigma)\big)$ *forms a successful forgery in the game of Definition 7 where the vectors* $\vec{m}_1, \ldots, \vec{m}_q$ *are those queried for signature.*

The advantage of $\mathcal{A}$ *is its success probability, taken over all coin tosses.*

Belenkiy *et al.* [4] formalized other security notions named signer privacy, user privacy and zero-knowledge that P-signatures ought to satisfy (formal definitions are given in the full version of the paper).

Signer Privacy. As formalized in [4], this notion captures that, during its interaction with the honest issuer, an adversary acting as a malicious user should not gain any side information beyond the obtained signature on a vector $\vec{m} = (\vec{m}_{|_{n_1}} | (m_{n_1+1}, \ldots, m_n)) \in \mathcal{D}^n$.

More precisely, there must exist an efficient simulator $\mathsf{SimIssue}$ such that no PPT adversary $\mathcal{A}$ can tell whether it is running $\mathsf{SigIssue} \leftrightarrows \mathsf{SigObtain}$ in interaction with a real issuer or if it is interacting with $\mathsf{SimIssue}$ that only has access to a signing oracle. As insisted in [4], $\mathsf{SimIssue}$ is allowed to rewind $\mathcal{A}$ if necessary.

User Privacy. User privacy is also defined following [4]. It requires that any malicious signer interacting with an honest user be unable to learn anything about the user's private messages $\vec{m}_{|_{n_1}} \in \mathcal{D}^{n_1}$. As previously, there must exist an efficient simulator $\mathsf{SimObtain}$ – which is allowed to rewind the adversary $\mathcal{A}$ – such that a dishonest signer $\mathcal{A}$ cannot distinguish a conversation with a real user from an interaction with $\mathsf{SimObtain}$.

ZERO KNOWLEDGE. To explain the zero-knowledge property, we introduce a simulator $\mathsf{Sim} = (\mathsf{SimSetup}, \mathsf{SimSigProve}_1, \mathsf{SimSigProve}_2, \mathsf{SimEqComProve})$ that implements P-signature algorithms for generating parameters, proving statements involving some relation family $\mathcal{R}$ and proving the equality of commitment openings without using any secret.

If for all outputs $(\mathsf{params}_s, \tau)$ of $\mathsf{SimSetup}$, it holds that $\mathsf{Com}(\mathsf{params}_s, \cdot)$ is now perfectly hiding, if params_s are computationally indistinguishable from those produced by Setup, and if any PPT adversary cannot tell whether it is interacting with real algorithms $(\mathsf{SigProve}_1, \mathsf{SigProve}_2, \mathsf{EqComProve})$ or simulators $(\mathsf{SimSigProve}_1, \mathsf{SimSigProve}_2, \mathsf{SimEqComProve})$, the scheme is said *zero-knowledge* and it is guaranteed not to leak useful information about secret values.

2.4 Groth-Sahai Proofs

In the following notation, for equal-dimension vectors $\vec{A}$ and $\vec{B}$ containing exponents or group elements, $\vec{A} \odot \vec{B}$ stands for their component-wise product.

To simplify the description, our scheme uses Groth-Sahai proofs based on the DLIN assumption although instantiations based on the symmetric external Diffie-Hellman assumption are also possible. In the DLIN setting, the Groth-Sahai (GS) proof systems [29] use a common reference string comprising 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 $f_1, f_2, g \in \mathbb{G}$. To commit to $X \in \mathbb{G}$, one sets $\vec{C} = (1, 1, X) \odot \vec{f}_1^{\,r} \odot \vec{f}_2^{\,s} \odot \vec{f}_3^{\,t}$ with $r, s, t \xleftarrow{R} \mathbb{Z}_p$. When proofs should be perfectly sound, $\vec{f}_3$ is set as $\vec{f}_3 = \vec{f}_1^{\,\xi_1} \odot \vec{f}_2^{\,\xi_2}$ with $\xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p^*$. 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 Boneh-Boyen-Shacham (BBS) ciphertexts [9] that can be decrypted using $\alpha_1 = \log_g(f_1)$, $\alpha_2 = \log_g(f_2)$. In the perfect witness indistinguishability (WI) setting, defining $\vec{f}_3 = \vec{f}_1^{\,\xi_1} \odot \vec{f}_2^{\,\xi_2} \odot (1, 1, g)^{-1}$ gives linearly independent $(\vec{f}_1, \vec{f}_2, \vec{f}_3)$ and $\vec{C}$ is a perfectly hiding commitment. Under the DLIN assumption, the two settings are indistinguishable. In either case, the commitment is denoted by $\vec{C} = \mathsf{GSCom}(X, open_X)$ and $open_X = (r, s, t)$ is its opening.

To commit to an exponent $x \in \mathbb{Z}_p$, one computes $\vec{C} = \vec{\varphi}^x \odot \vec{f}_1^{\,r} \odot \vec{f}_2^{\,s}$, with $r, s \xleftarrow{R} \mathbb{Z}_p^*$, using a CRS comprising vectors $\vec{\varphi}, \vec{f}_1, \vec{f}_2$. The commitment and its opening are denoted by $\vec{C} = \mathsf{GSCom}(x, open_x)$ and $open_x = (r, s)$, respectively. In the soundness setting $\vec{\varphi}, \vec{f}_1, \vec{f}_2$ are linearly independent vectors (typically, one chooses $\vec{\varphi} = \vec{f}_3 \odot (1, 1, g)$ where $\vec{f}_3 = \vec{f}_1^{\,\xi_1} \odot \vec{f}_2^{\,\xi_2}$) whereas, in the WI setting, choosing $\vec{\varphi} = \vec{f}_1^{\,\xi_1} \odot \vec{f}_2^{\,\xi_2}$ gives a perfectly hiding commitment since $\vec{C}$ is always a BBS encryption of $1_{\mathbb{G}}$. On a perfectly sound CRS (where $\vec{f}_3 = \vec{f}_1^{\,\xi_1} \odot \vec{f}_2^{\,\xi_2}$ and $\vec{\varphi} = \vec{f}_3 \odot (1, 1, g)$), commitments to exponents are not fully extractable since the trapdoor (α_1, α_2) only allows recovering g^x from $\vec{C} = \vec{\varphi}^x \odot \vec{f}_1^{\,r} \odot \vec{f}_2^{\,s}$. In order to commit to $x \in \mathbb{Z}_p$, we will sometimes commit to the group element g^x. The result of this process will be denoted by $\vec{C} = \mathsf{GSCom}'(x, open_x) = \mathsf{GSCom}(g^x, open_x)$

with $open_x = (r, s, t)$.

To prove that committed variables satisfy a set of relations, the Groth-Sahai techniques require one commitment per variable and one proof element (made of a constant number of group elements) per relation. Such proofs are available for pairing-product relations, which are of the type

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

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{ij} \in \mathbb{G}$, for $i, j \in [1, n]$. Efficient proofs also exist for multi-exponentiation equations

$$\prod_{i=1}^{m} \mathcal{A}_i^{y_i} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{b_j} \cdot \prod_{i=1}^{m} \cdot \prod_{j=1}^{n} \mathcal{X}_j^{y_i \gamma_{ij}} = T,$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$, $y_1, \ldots, y_m \in \mathbb{Z}_p$ and constants $T, \mathcal{A}_1, \ldots, \mathcal{A}_m \in \mathbb{G}$, $b_1, \ldots, b_n \in \mathbb{Z}_p$ and $\gamma_{ij} \in \mathbb{G}$, for $i \in [1, m], j \in [1, n]$.

Multi-exponentiation equations admit zero-knowledge proofs at no additional cost. On a simulated CRS (prepared for the WI setting), the trapdoor (ξ_1, ξ_2) makes it is possible to simulate proofs without knowing witnesses and simulated proofs are perfectly indistinguishable from real proofs. As for pairing-product equations, NIZK proofs are often possible (this is typically the case when the target element t_T has the special form $t_T = \prod_{i=1}^{t} e(S_i, T_i)$, for constants $\{(S_i, T_i)\}_{i=1}^{t}$ and some $t \in \mathbb{N}$) but usually come at some expense.

From an efficiency standpoint, quadratic pairing product equations cost 9 elements to prove whereas linear ones (when $a_{ij} = 0$ for all i, j) take 3 group elements. Linear multi-exponentiation equations of the type $\prod_{i=1}^{m} \mathcal{A}_i^{y_i} = T$ demand 2 group elements.

3 A Construction for Inner Product Relations

As noted in [33], many predicates can be expressed in terms of the inner product of two vectors of attributes. In this section, we describe a P-signature scheme for families $(\mathcal{R}_1, \mathcal{R}_2)$ where $\mathcal{R}_1$ encompasses (in)-equality relations and $\mathcal{R}_2$ relates to inner products. Namely, we set $\mathcal{R}_1 = \{R^{\mathsf{EQ}}, R^{\neg\mathsf{EQ}}\}$ and $\mathcal{R}_2 = \{R^{\mathsf{IP}}, R^{\neg\mathsf{IP}}\}$, which are specified as follows. We let $\mathcal{D} = \mathbb{Z}_p$, for some prime p and, for vectors $\vec{m} \in \mathbb{Z}_p^n$, $\vec{X} \in \mathbb{Z}_p^n$, the relations R^{IP} and $R^{\neg\mathsf{IP}}$ are only defined for $i = 0$ in such a way that $R^{\mathsf{IP}}(0, \vec{m}, \vec{X}) = 1$ (resp. $R^{\neg\mathsf{IP}}(0, \vec{m}, \vec{X}) = 1$) if and only if $\vec{m} \cdot \vec{X} = 0$ (resp. $\vec{m} \cdot \vec{X} \neq 0$). As for $\mathcal{R}_1$, we define relations R^{EQ} and $R^{\neg\mathsf{EQ}}$ for $i \in [1, n]$ and so that $R^{\mathsf{EQ}}(i, \vec{m}, \vec{X}) = 1$ (resp. $R^{\neg\mathsf{EQ}}(i, \vec{m}, \vec{X}) = 1$) if and only if $m_i = x_i$ (resp. $m_i \neq x_i$).

The construction is based on the commitment scheme of section 2.2 and a signature scheme suggested in [21] to sign group elements. The intuition is to sign a commitment to a vector $\vec{m}$ using a signature scheme for group elements such as [21, 24, 1]. Here, a lightweight version of the scheme can be used since,

in the proof, the simulator knows the discrete logarithms of the group elements that are signed (hence, there is no need to combine the scheme with a trapdoor commitment to group elements as in [21]). In this simplified version, the signer holds a public key comprising $(\Omega = g^{\omega}, A = g^{\gamma}, u, U_0, U_1 = g^{\beta_1}) \in \mathbb{G}^5$, for private elements $(\omega, \gamma, \beta_1)$. To sign a vector $\vec{m}$, the signer first computes a commitment V to $\vec{m}$, chooses $c \xleftarrow{R} \mathbb{Z}_p$ and computes $\sigma_1 = (g^{\gamma})^{1/(\omega+c)}$, $\sigma_2 = g^c$, $\sigma_3 = u^c$, $\sigma_4 = (U_0 \cdot V^{\beta_1})^c$, $\sigma_5 = V^c$ and also sets $\sigma_6 = V$ as part of the signature.

The construction handles inner products using the properties of the commitment scheme recalled in section 2.2. More precisely, we use the property that this scheme allows the committer to generate a short non-interactive argument allowing to convince the verifier that the committed vector $\vec{m}$ is orthogonal to a public vector $\vec{X} = (x_1, \ldots, x_n)$ without revealing anything else. Concretely, given a commitment $C = g^r \cdot \prod_{j=1}^{n} g_{n+1-j}^{m_j}$ to $\vec{m} = (m_1, \ldots, m_n)$, for each $i \in [1, n]$, we know that the witness $W_i = g_i^r \cdot \prod_{j=1, j\neq i}^{n} g_{n+1-j+i}^{m_j}$ satisfies

$$e(g_i, C) = e(g_1, g_n)^{m_i} \cdot e(g, W_i), \tag{1}$$

For each i, if we raise both members of (1) to the power x_i and multiply the resulting n equations altogether, we find

$$e\big(\prod_{i=1}^{n} g_i^{x_i}, C\big) = e(g_1, g_n)^{\vec{m}\cdot\vec{X}} \cdot e\big(g, \prod_{i=1}^{n} W_i^{x_i}\big), \tag{2}$$

which implies $e\big(\prod_{i=1}^{n} g_i^{x_i}, C\big) = e\big(g, \prod_{i=1}^{n} W_i^{x_i}\big)$ whenever $\vec{m} \cdot \vec{X} = 0$. As it turns out, a single group element $W = \prod_{i=1}^{n} W_i^{x_i}$ suffices to convince the verifier that $\vec{m} \cdot \vec{X} = 0$. It can be showed (as in detailed in the full version of the paper) that, after the commitment phase, if the committer is able to produce a witness W satisfying $e\big(\prod_{i=1}^{n} g_i^{x_i}, C\big) = e\big(g, W\big)$ and subsequently open the commitment C to a vector $\vec{m}$ such that $\vec{m} \cdot \vec{X} \neq 0$, the n-DHE assumption can be broken.

Likewise, the committer can also convince the verifier that $\vec{m} \cdot \vec{X} \neq 0$ by proving knowledge of group elements $W = \prod_{i=1}^{n} W_i^{x_i}$, $W_1 = g_1^{\vec{m}\cdot\vec{X}} \in \mathbb{G}$ such that

$$e\big(\prod_{i=1}^{n} g_i^{x_i}, C\big) = e(W_1, g_n) \cdot e(g, W). \tag{3}$$

To convince the verifier that $W_1 \neq 1_{\mathbb{G}}$, the prover demonstrates knowledge of another group element $W_0 = g^{1/\vec{m}\cdot\vec{X}}$ for which $e(W_0, W_1) = e(g, g_1)$. We would like to argue that a malicious committer cannot open a commitment C to a vector $\vec{m}$ such that $\vec{m} \cdot \vec{X} = 0$ and also produce $(W, W_0, W_1) \in \mathbb{G}$ such that the equalities $e(W_0, W_1) = e(g, g_1)$ and (3) are both satisfied. Unfortunately, this is not true since a cheating prover can commit to $\vec{m} = \vec{0}$ (which is orthogonal to everything). Since the commitment $C = g^r$ and the value $W = \prod_{i=1}^{n} g_i^{r \cdot x_i}$ satisfy $e(\prod_{i=1}^{n} g_i^{x_i}, C) = e(g, W)$, the prover can fool the verifier by revealing $(W_0, W_1, W') = \big(g_1^{1/\mu}, g^{\mu}, W/g_n^{\mu}\big)$, with $\mu \xleftarrow{R} \mathbb{Z}_p$, which satisfies the equalities

$e(W_0, W_1) = e(g, g_1)$ and $e(\prod_{i=1}^{n} g_i^{x_i}, C) = e(W_1, g_n) \cdot e(g, W')$.

To address this problem, we require the prover to additionally reveal the pair $(W_2, W_3) = (g^{\vec{m}\cdot\vec{X}}, g_{2n}^{\vec{m}\cdot\vec{X}})$ when stating that $\vec{m} \cdot \vec{X} \neq 0$. The extra checks $e(W_1, g) = e(g_1, W_2)$ and $e(W_1, g_{2n}) = e(g_1, W_3)$ then suffice to convince the verifier. Under the n-FlexDHE assumption, we can show (as detailed in the full version of the paper) that the prover cannot generate (W_0, W_1, W_2, W_3, W) and subsequently open the commitment to a vector $\vec{m}$ that contradicts the assertion.

In details, the F- unforgeable block-wise signature scheme is as follows.

Setup(λ)**:** chooses bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ with a generator $g \xleftarrow{R} \mathbb{G}$. It generates a perfectly sound Groth-Sahai CRS $\mathbf{f} = (\vec{f_1}, \vec{f_2}, \vec{f_3})$. Public parameters consist of $\mathsf{params} := \big((\mathbb{G}, \mathbb{G}_T),\ g,\ \mathbf{f}\big)$.

SigSetup(λ, n)**:** picks $\gamma, \omega, \alpha, \beta_1 \xleftarrow{R} \mathbb{Z}_p$, $u, U_0 \xleftarrow{R} \mathbb{G}$ at random and computes $\Omega = g^{\omega}$, $A = g^{\gamma}$, $U_1 = g^{\beta_1}$ as well as $g_i = g^{(\alpha^i)}$ for each $i \in [1, n] \cup [n+2, 2n]$. The private key is $\mathsf{sk} = (\gamma, \omega, \beta_1)$ and the corresponding public key is defined to be $\mathsf{pk} = \big(u,\ \Omega = g^{\omega},\ A = g^{\gamma},\ U_0,\ U_1,\ \{g_i\}_{i\in[1,2n]\setminus\{n+1\}}\big)$.

Sign$(\mathsf{sk}, \vec{m})$**:** to sign $\vec{m} = (m_1, \ldots, m_n)$, conduct the following steps.

1. Pick $r \xleftarrow{R} \mathbb{Z}_p$ and compute $V = g^r \cdot \prod_{j=1}^{n} g_{n+1-j}^{m_j} = g_n^{m_1} \cdots g_1^{m_n} \cdot g^r$.
2. Choose $c \xleftarrow{R} \mathbb{Z}_p$ and compute

$$\sigma_1 = g^{\gamma/(\omega+c)}, \qquad \sigma_2 = g^c, \qquad \sigma_3 = u^c, \qquad \sigma_4 = (U_0 \cdot V^{\beta_1})^c,$$
$$\sigma_5 = V^c, \qquad \sigma_6 = V$$

and output $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, r)$.

Verify$(\mathsf{pk}, \vec{m}, \sigma)$**:** parse σ as $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, r)$ and $\vec{m}$ as $(m_1, \ldots, m_n)$.

1. Return 0 if the following equalities do not hold

$$e(A, g) = e(\sigma_1, \Omega \cdot \sigma_2), \qquad e(u, \sigma_2) = e(\sigma_3, g), \tag{4}$$
$$e(g, \sigma_4) = e(U_0, \sigma_2) \cdot e(U_1, \sigma_5), \qquad e(g, \sigma_5) = e(\sigma_6, \sigma_2). \tag{5}$$

2. Return 1 if $\sigma_6 = g^r \cdot \prod_{j=1}^{n} g_{n+1-j}^{m_j}$ and 0 otherwise.

Witness-Gen$(\mathsf{pk}, R, i, \vec{m}, \vec{X}, \sigma)$**:** parse σ as $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, r)$. Parse $\vec{m}$ and $\vec{X}$ as $(m_1, \ldots, m_n)$ and $(x_1, \ldots, x_n)$, respectively. Return $\perp$ if it turns out that $\mathsf{Verify}(\mathsf{pk}, \vec{m}, \sigma) = 0$. Otherwise,

a. If $R = R^{\mathsf{EQ}}$ (and $i \in [1, n]$), return $\perp$ if $m_i \neq x_i$. Otherwise, compute and output the witness $W = g_i^r \cdot \prod_{j=1, j\neq i}^{n} g_{n+1-j+i}^{m_j}$.

b. If $R = R^{\neg\mathsf{EQ}}$ (and $i \in [1, n]$), return $\perp$ if $m_i = x_i$. Otherwise, compute $W_0 = g^{1/(m_i - x_i)}$, $W_1 = g_1^{m_i - x_i}$, $W_2 = g^{m_i - x_i}$, $W_3 = g_{2n}^{m_i - x_i}$ and finally $W_4 = g_i^r \cdot \prod_{j=1, j\neq i}^{n} g_{n+1-j+i}^{m_j}$. Return the witness consisting of the tuple $W = (W_0, W_1, W_2, W_3, W_4)$.

c. If $R = R^{\mathsf{IP}}$ (and $i = 0$), return $\perp$ if $\vec{m} \cdot \vec{X} \neq 0$. Otherwise, compute $W_i = g_i^r \cdot \prod_{j=1, j\neq i}^{n} g_{n+1-j+i}^{m_j}$ for $i = 1$ to n. Then, compute and output the witness $W = \prod_{i=1}^{n} W_i^{x_i}$.

d. If $R = R^{\neg\mathsf{IP}}$ (and $i = 0$), return $\perp$ if $\vec{m} \cdot \vec{X} = 0$. Otherwise, compute $W_0 = g^{1/(\vec{m}\cdot\vec{X})}$, $W_1 = g_1^{\vec{m}\cdot\vec{X}}$, $W_2 = g^{\vec{m}\cdot\vec{X}}$, $W_3 = g_{2n}^{\vec{m}\cdot\vec{X}}$. For $i = 1$ to n, compute $W_{4,i} = g_i^r \cdot \prod_{j=1, j\neq i}^{n} g_{n+1-j+i}^{m_j}$ and finally set $W_4 = \prod_{i=1}^{n} W_{4,i}^{x_i}$. Return the witness $W = (W_0, W_1, W_2, W_3, W_4)$.

Witness-Verify$(\mathsf{pk}, R, i, \vec{X}, W, \sigma)$**:** parse σ as $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, r)$ and $\vec{X}$ as $(x_1, \ldots, x_n)$. Return 0 if equations (4)-(5) are not satisfied. Otherwise, two cases are distinguished.

a. If $R = R^{\mathsf{EQ}}$ (and $i \in [1, n]$), return 1 iff $e(g_i, \sigma_6) = e(g_1, g_n)^{x_i} \cdot e(g, W)$.

b. If $R = R^{\neg\mathsf{EQ}}$ (and $i \in [1, n]$), parse W as $(W_0, W_1, W_2, W_3, W_4) \in \mathbb{G}^5$ and return $\perp$ if it does not parse properly. Otherwise, return 1 if and only if $e\big(g_i, \sigma_6 \cdot g_{n+1-i}^{-x_i}\big) = e(W_1, g_n) \cdot e(g, W_4)$ and[2]

$$e(W_0, W_1) = e(g, g_1), \qquad e(W_1, g) = e(g_1, W_2), \qquad e(W_1, g_{2n}) = e(g_1, W_3). \tag{6}$$

c. If $R = R^{\mathsf{IP}}$ (and $i = 0$), parse the witness W as a group element $W \in \mathbb{G}$ and return 1 if and only if $e(g, W) = e\big(\prod_{i=1}^{n} g_i^{x_i}, \sigma_6\big)$.

d. If $R = R^{\neg\mathsf{IP}}$ (and $i = 0$), parse W as $(W_0, W_1, W_2, W_3, W_4) \in \mathbb{G}^5$ and return $\perp$ if it does not parse properly. Return 1 if and only if $e\big(\prod_{i=1}^{n} g_i^{x_i}, \sigma_6\big) = e(W_1, g_n) \cdot e(g, W_4)$ and

$$e(W_0, W_1) = e(g, g_1), \qquad e(W_1, g) = e(g_1, W_2), \qquad e(W_1, g_{2n}) = e(g_1, W_3). \tag{7}$$

The correctness of algorithms Sign and Verify is almost straightforward and that of Witness-Gen and Witness-Verify follows from the properties of the commitment scheme in section 2.2.

P-Signature Protocols. To obtain a complete P-signature, the scheme is augmented with algorithms $\mathsf{SigProve}_i$, for $i \in \{1, 2\}$, and EqComProve.

SigProve$_1(\mathsf{pk}, R, i, \Upsilon = \{i\}, \sigma, \vec{m}, \vec{X})$**:** parse σ as $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, r)$, $\vec{m}$ as $(m_1, \ldots, m_n)$ and $\vec{X}$ as $(x_1, \ldots, x_n)$. Then, compute Groth-Sahai commitments $\{\vec{C}_{x_t,j} = \mathsf{GSCom}(X_{t,j}, open_{x_t,j})\}_{t\in\Upsilon, j\in\{1,2,3\}}$ to the variables

$$\{(X_{t,1}, X_{t,2}, X_{t,3}) = (g_1^{x_t}, g^{x_t}, g_{2n}^{x_t})\}_{t\in\Upsilon}.$$

For $j = 1$ to 6, compute $\vec{C}_{\sigma_j} = \mathsf{GSCom}(\sigma_j, open_{\sigma_j})$ and generate a NIZK proof that committed variables $\{\sigma_j\}_{j=1}^{6}$ satisfy (4)-(5). This requires to introduce auxiliary variables $\sigma_7 \in \mathbb{G}$, $\theta_1 \in \mathbb{Z}_p$ with their own commitments

[2] Looking ahead, W_0 will be useful to convince the verifier (via the first relation of (7)) that $W_1 \neq 1_{\mathbb{G}}$ when (W_1, W_2, W_3, W_4) will appear in committed form within Groth-Sahai proofs produced by $\mathsf{SigProve}_2$. Although W_0 is not strictly necessary in Witness-Verify in the cases $R = R^{\neg\mathsf{IP}}$ and $R = R^{\neg\mathsf{EQ}}$ (since the algorithm can directly check that $W_1 \neq 1_{\mathbb{G}}$), we included it among the outputs of Witness-Gen for ease of explanation.

$\vec{C}_{\sigma_7} = \mathsf{GSCom}(\sigma_7, open_{\sigma_7})$, $\vec{C}_{\theta_1} = \mathsf{GSCom}(\theta_1, open_{\theta_1})$ and to prove that

$$e(\sigma_7, g) = e(\sigma_1, \Omega \cdot \sigma_2), \qquad e(u, \sigma_2) = e(\sigma_3, g), \tag{8}$$
$$e(g, \sigma_4) = e(U_0, \sigma_2) \cdot e(U_1, \sigma_5), \qquad e(g, \sigma_5) = e(\sigma_6, \sigma_2), \tag{9}$$
$$\theta_1 = 1, \qquad e(A/\sigma_7, g^{\theta_1}) = 1_{\mathbb{G}_T} \tag{10}$$

Let π_σ be the proof for (8)-(10). Then, the algorithm considers two cases.

- If $R = R^{\mathsf{EQ}}$, let $\vec{C}_W = \mathsf{GSCom}(W, open_W)$, where the witness W is obtained as $W = \mathsf{Witness\text{-}Gen}(\mathsf{pk}, R^{\mathsf{EQ}}, i, \vec{m}, \vec{X}, \sigma)$. Generate proofs π_{x_i}, $\{\pi_{X_t,j}\}_{t\in\Upsilon, j=1,2}$ that committed variables σ_6, W and $X_{i,1}$ satisfy

$$e(g_i, \sigma_6) = e(X_{i,1}, g_n) \cdot e(g, W), \tag{11}$$
$$e(X_{i,2}, g_1) = e(X_{i,1}, g), \qquad e(X_{i,2}, g_{2n}) = e(X_{i,3}, g). \tag{12}$$

  The final proof is

$$\pi = \big(\{\vec{C}_{x_t,j}\}_{t\in\Upsilon, j\in\{1,2,3\}}, \{\vec{C}_{\sigma_j}\}_{j=1}^7, \vec{C}_W, \vec{C}_{\theta_1}, \pi_\sigma, \pi_{x_i}, \{\pi_{X_t,j}\}_{t\in\Upsilon, j=1,2}\big).$$

- If $R = R^{\neg\mathsf{EQ}}$, generate commitments $\{C_{W_j}\}_{j=0}^4$ to the components of the witness $(W_0, W_1, W_2, W_3, W_4) \leftarrow \mathsf{Witness\text{-}Gen}(\mathsf{pk}, R^{\neg\mathsf{EQ}}, i, \vec{m}, \vec{X}, \sigma)$. Generate proofs π_{x_i}, π_W for relations (13) and (14)

$$e(g_i, \sigma_6) \cdot e(X_{i,1}, g_n)^{-1} = e(W_1, g_n) \cdot e(g, W_4) \tag{13}$$
$$e(W_0, W_1) = e(g, g_1) \tag{14}$$
$$e(W_1, g) = e(g_1, W_2)$$
$$e(W_1, g_{2n}) = e(g_1, W_3),$$

  and proofs $\{\pi_{X_t,j}\}_{t\in\Upsilon, j=1,2}$ that $\{(X_{t,1}, X_{t,2}, X_{t,3})\}_{t\in\Upsilon}$ satisfy (12). The final proof consists of

$$\pi = \big(\{\vec{C}_{x_t,j}\}_{t\in\Upsilon, j\in\{1,2,3\}} \{\vec{C}_{\sigma_j}\}_{j=1}^7, \{\vec{C}_{W_i}\}_{j=0}^4, \vec{C}_{\theta_1}, \pi_\sigma, \pi_{x_i}, \pi_W, \{\pi_{X_t,j}\}_{t\in\Upsilon, j=1,2}\big).$$

$\mathsf{SigProve}_2(\mathsf{pk}, R, i, \sigma, \vec{m}, \vec{X})$: parse σ and $\vec{m}$ as previously and $\vec{X}$ as $(x_1, \ldots, x_n)$. For $i = 1$ to 6, compute commitments $\vec{C}_{\sigma_i} = \mathsf{GSCom}(\sigma_i, open_{\sigma_i})$. Using extra variables $\sigma_7 \in \mathbb{G}$, $\theta_1 \in \mathbb{Z}_p$ and their commitments $\vec{C}_{\sigma_7} = \mathsf{GSCom}(\sigma_7, open_{\sigma_7})$, $\vec{C}_{\theta_1} = \mathsf{GSCom}(\theta_1, open_{\theta_1})$, generate a NIZK proof that $\{\sigma_i\}_{i=1}^6$ satisfy (4)-(5). We call π_σ the proof for (8)-(10). Then, consider the two following cases.

- If $R = R^{\mathsf{IP}}$, set $\vec{C}_W = \mathsf{GSCom}(W, open_W)$, where the witness W is computed as $W = \mathsf{Witness\text{-}Gen}(\mathsf{pk}, R^{\mathsf{IP}}, 0, \vec{m}, \vec{X}, \sigma)$. Then, generate a proof $\pi_{\vec{X}}$ that W and σ_6 satisfy

$$e(\prod_{j=1}^n g_j^{x_j}, \sigma_6) = e(g, W). \tag{15}$$

  The NIZK proof is $\pi = \big(\{\vec{C}_{\sigma_j}\}_{j=1}^7, \vec{C}_W, \vec{C}_{\theta_1}, \pi_\sigma, \pi_{\vec{X}}\big)$.

- If $R = R^{\neg\mathsf{IP}}$, define the auxiliary variable $\Theta = g \in \mathbb{G}$ and generate $\vec{C}_\Theta = \mathsf{GSCom}(\Theta, open_\Theta)$, $\{\vec{C}_{W_j} = \mathsf{GSCom}(W_j, open_{W_j})\}_{j=0}^4$, where $\{W_j\}_{j=0}^4 \leftarrow \mathsf{Witness\text{-}Gen}(\mathsf{pk}, R^{\neg\mathsf{IP}}, 0, \vec{m}, \vec{X}, \sigma)$. Then, generate a proof $\pi_{\vec{X}}^{\mathsf{NOT}}$ that Θ and $\{W_i\}_{i=0}^4$ satisfy

$$e(W_0, W_1) = e(\Theta, g_1), \qquad e(W_1, g) = e(g_1, W_2), \tag{16}$$
$$e(W_1, g_{2n}) = e(g_1, W_3), \qquad e(\Theta/g, g^{\theta_1}) = 1_{\mathbb{G}_T}. \tag{17}$$
$$e(\prod_{j=1}^{n} g_j^{x_j}, \sigma_6) = e(g, W_4) \cdot e(W_1, g_n). \tag{18}$$

The NIZK proof consists of $\pi = \big(\{\vec{C}_{\sigma_j}\}_{j=1}^7, \{\vec{C}_{W_j}\}_{j=0}^4, \vec{C}_{\theta_1}, \pi_\sigma, \pi_{\vec{X}}^{\mathsf{NOT}}\big)$.

- If $R = R^{\mathsf{EQ}}$ (and $i \in [1, n]$), generate $\vec{C}_W = \mathsf{GSCom}(W, open_W)$ for the witness $W \leftarrow \mathsf{Witness\text{-}Gen}(\mathsf{pk}, R^{\mathsf{EQ}}, i, \vec{m}, \vec{X}, \sigma)$, introduces a commitment $\vec{C}_{X_i} = \mathsf{GSCom}(X_i, open_{X_i})$ to the auxiliary variable $X_i = g_1^{x_i}$ and compute proofs π_W and π_{x_i} that

$$e(g_i, \sigma_6) = e(X_i, g_n) \cdot e(g, W) \qquad e(X_i/g_1^{x_i}, g^{\theta_1}) = 1_{\mathbb{G}_T} \tag{19}$$

The proof is $\pi = \big(\{\vec{C}_{\sigma_j}\}_{j=1}^7, \vec{C}_W, \vec{C}_{X_i}, \vec{C}_{\theta_1}, \pi_\sigma, \pi_W, \pi_{x_i}\big)$.

- If $R = R^{\neg\mathsf{EQ}}$ (and thus $i \in [1, n]$), compute Groth-Sahai commitments $\{\vec{C}_{W_j} = \mathsf{GSCom}(W_j, open_{W_j})\}_{j=0}^4$ to the components of the witness $\{W_j\}_{j=0}^4 \leftarrow \mathsf{Witness\text{-}Gen}(\mathsf{pk}, R^{\neg\mathsf{EQ}}, i, \vec{m}, \vec{X}, \sigma)$. Then, introduce a commitment $\vec{C}_{X_i} = \mathsf{GSCom}(X_i, open_{X_i})$ to the auxiliary variable $X_i = g_1^{x_i}$ and generate proofs $(\pi_{X_i,W}, \{\pi_{W_j}\}_{j=1}^3, \pi_{X_i}, \pi_\Theta)$ for

$$e(g_i, \sigma_6) \cdot e(X_i, g_n)^{-1} = e(W_1, g_n) \cdot e(g, W), \tag{20}$$
$$e(W_0, W_1) = e(\Theta, g_1), \tag{21}$$
$$e(W_1, g) = e(g_1, W_2),$$
$$e(W_1, g_{2n}) = e(g_1, W_3),$$
$$e(X_i/g_1^{x_i}, g^{\theta_1}) = e(\Theta/g, g^{\theta_1}) = 1_{\mathbb{G}_T} \tag{22}$$

The proof is

$$\pi = \big(\{\vec{C}_{\sigma_j}\}_{j=1}^7, \{\vec{C}_{W_j}\}_{j=0}^4, \vec{C}_{X_i}, \vec{C}_{\theta_1}, \pi_\sigma, \pi_{X_i,W}, \{\pi_{W_j}\}_{j=1}^3, \pi_{X_i}, \pi_\Theta\big).$$

$\mathsf{SigIssue}\big(\mathsf{sk}, V', (m_{n_1+1}, \ldots, m_n)\big) \leftrightarrows \mathsf{SigObtain}(\mathsf{pk}, \vec{m}_{|_{n_1}}, open_{\vec{m}_{|_{n_1}}})$: the user $\mathcal{U}$ and the issuer interact with each other in the following way.

1. $\mathcal{U}$ commits to $\vec{m}_{|_{n_1}} = (m_1, \ldots, m_{n_1})$ and computes $V' = g^{r'} \cdot \prod_{j=1}^{n_1} g_{n+1-j}^{m_j}$, where $r' \xleftarrow{R} \mathbb{Z}_p$, retains $open_{\vec{m}_{|_{n_1}}} = (m_1, \ldots, m_{n_1}, r')$ and provides the issuer with an interactive WI proof of knowledge of $(m_1, \ldots, m_{n_1}, r')$ such that $V' = g^{r'} \cdot \prod_{j=1}^{n_1} g_{n+1-j}^{m_j}$.

2. The issuer sets $V = V' \cdot \prod_{j=n_1+1}^{n} g_{n+1-j}^{m_j}$. Then, it randomly chooses $c, r'' \xleftarrow{R} \mathbb{Z}_p$, computes $\sigma_1 = g^{\gamma/(\omega+c)}$, $\sigma_2 = g^c$, $\sigma_3 = u^c$ and

$$\sigma_4 = \big(U_0 \cdot (V \cdot g^{r''})^{\beta_1}\big)^c, \qquad \sigma_5 = (V \cdot g^{r''})^c, \qquad \sigma_6 = V \cdot g^{r''}$$

and returns $\tilde{\sigma} = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)$ and r''.

3. $\mathcal{U}$ outputs $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, r)$, where $r = r' + r''$.

The algorithm $\mathsf{EqComProve}$ is standard: given two distinct Groth-Sahai commitments $\vec{C_X} = \mathsf{GSCom}(X, open_X)$ and $\vec{C_Y} = \mathsf{GSCom}(Y, open_Y)$ such that $X = Y \in \mathbb{G}$, the NIZK proof can be a proof that $\vec{C_X} \odot \vec{C_Y}^{-1}$ is a commitment that opens to $1_{\mathbb{G}}$. If we write $\vec{f_1} = (f_1, 1, g)$, $\vec{f_2} = (1, f_2, g)$ and $\vec{f_3} = (f_{31}, f_{32}, f_{33})$, this amounts to proving the existence of $(\rho_1, \rho_2, \rho_3) \in \mathbb{Z}_p^3$ such that $\vec{C_X} \odot \vec{C_Y}^{-1} = (f_1^{\rho_1} \cdot f_{31}^{\rho_3}, f_2^{\rho_2} \cdot f_{32}^{\rho_3}, g^{\rho_1+\rho_2} \cdot f_{33}^{\rho_3})$. On a simulated CRS, this relation can always be proved in NIZK since it is a linear multi-exponentiation equation.

Efficiency. From an efficiency standpoint, the outputs of $\mathsf{SigProve}_1$ consist of 80 elements of $\mathbb{G}$ for R^{EQ} and 101 group elements for $R^{\neg\mathsf{EQ}}$. Each proof produced by $\mathsf{SigProve}_2$ requires less than 80 group elements for relations R^{EQ} and R^{IP} and at most 107 elements in the case of $R^{\neg\mathsf{EQ}}$ and $R^{\neg\mathsf{IP}}$.

When these proofs are combined to prove the ownership of a credential, they result in non-interactive proofs demanding about 2 kB at the 80-bit security level. A detailed efficiency analysis is provided in the full version of the paper.

We leave it as an open problem to eliminate the dependency on n in the public key size (as was done in [14]) without using interaction or random oracles.

Security. The security of the scheme relies on the assumptions described at the beginning of section 2. The proofs of the following theorems are given in the full version of the paper.

Theorem 1. *If the HSDH, FlexDH and n-FlexDHE assumptions hold in $\mathbb{G}$, the above block-wise P-signature scheme is $(F, \mathcal{R}_1, \mathcal{R}_2)$-unforgeable w.r.t. the injective function $F(m) = (g_1^m, g^m, g_{2n}^m)$ and the relations families $\mathcal{R}_1 = \{R^{\mathsf{EQ}}, R^{\neg\mathsf{EQ}}\}$, $\mathcal{R}_2 = \{R^{\mathsf{IP}}, R^{\neg\mathsf{IP}}\}$.*

Theorem 2. *The block-wise P-signature provides signer and user privacy if the underlying WI proof of knowledge is secure.*

Theorem 3. *The block-wise P-signature is zero-knowledge if the DLIN assumption holds in $\mathbb{G}$.*

4 Non-Interactive Anonymous Credentials with Efficient Attributes

In the full version of the paper, we provide the complete details about how block-wise P-signatures for these relation families can be generically turned into

non-interactive anonymous credentials with efficient attributes. Proper security definitions for these are given in the full paper, where we prove the security of the generic construction in the same way as in [4].

In a nutshell, the construction appeals to $\mathsf{SigProve}_1$ to prove that the first component of the user's certified vector $\vec{m}$ is the same value (*i.e.*, his private key sk_U) as the one contained in the user's pseudonym. Then, $\mathsf{SigProve}_2$ is used to convince the verifier that the certified vector $\vec{X}$ satisfies $\vec{m} \cdot \vec{X} = 0$. The construction is presented without optimizations for the sake of generality. Its optimized variant provides proofs of about 2 kB.

In the full version, we describe in details the predicates that can be expressed using inner product relations and suitable attribute encodings (already used in [33]). For example, when $\vec{m}$ contains the coefficients a polynomial whose roots are the user's attributes, the inclusion (or the non-inclusion) of some attribute $\omega \in \mathbb{Z}_p$ can be selectively demonstrated by setting the coordinates of $\vec{X}$ as $(1, \omega, \ldots, \omega^{n-1})$. A similar technique can be used to prove that some certified attribute ω (this time encoded as a sub-vector $(1, \omega, \ldots, \omega^{n-1})$ of $\vec{m}$) lies in a public list (or not) by proving its orthogonality to some $\vec{X}$ that contains the coefficients of a polynomial.

Using more complex attribute encodings, inner products can also handle disjunctions of a small (e.g., logarithmic in λ) number of atomic conditions. If we assume only two rounds of interaction, conjunctions can also be dealt with: the verifier just has to send a short random challenge in $\mathbb{Z}_p$ which is used to randomize the vector $\vec{X}$ in such a way that the condition $\vec{m} \cdot \vec{X} = 0$ guarantees the validity of assertions $(m_1 = x_1) \wedge \ldots \wedge (m_n = x_n)$ with overwhelming probability. Although the need for interaction seems at odds with the original motivation of P-signatures, we still gain something since only two rounds are necessary.

Finally, as already noted in [33], inner products also provide a method to prove exact threshold statements about sets of binary attributes. For example, if $\vec{m}$ and $\vec{X}$ encode two sets of binary attributes (such as "gender", "graduated", etc.) X and S, the prover can convince the verifier that $|S \cap X| = t$. In addition, by combining the same technique with set membership proofs [12], statements about inexact thresholds $|S \cap X| \leq t$ can also be proved as detailed in the full version.

References

1. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev and M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In *Crypto'10*, LNCS 6223, pp. 209–236, 2010.
2. N. Akagi, Y. Manabe, T. Okamoto. An Efficient Anonymous Credential System. In *Financial Cryptography (FC'08)*, LNCS 5143, pp. 272–286, 2008.
3. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya and H. Shacham. Randomizable Proofs and Delegatable Anonymous Credentials. In *Crypto'09*, LNCS 5677, pp. 108–125, 2009
4. M. Belenkiy, M. Chase, M. Kohlweiss and A. Lysyanskaya. P-signatures and non-interactive anonymous credentials. In *TCC'08*, LNCS 4948, pages 356–374, 2008.

5. —, Compact E-Cash and Simulatable VRFs Revisited. In *Pairing'09*, LNCS 5671, pp. 114–131, 2009.
6. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS'93*, pp. 62–73, 1993.
7. O. Blazy, G. Fuchsbauer, M. Izabachène, A. Jambert, H. Sibert, D. Vergnaud. Batch Groth-Sahai. In *Applied Cryptography and Network Security (ACNS'10)*, LNCS 6123, pp. 218–235, 2010.
8. D. Boneh and X. Boyen. Short signatures without random oracles. In *Eurocrypt'04*, LNCS 3027, pages 56–73, 2004.
9. D. Boneh, X. Boyen and H. Shacham. Short Group Signatures. In *Crypto'04*, *LNCS* 3152, pp. 41–55, 2004.
10. D. Boneh, C. Gentry and B. Waters. Collusion-Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In *Crypto'05*, *LNCS* 3621, pp. 258–275, 2005.
11. X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC'07* , LNCS 4450, pp. 1–15, 2007.
12. J. Camenisch, R. Chaabouni and a. shelat. Efficient Protocols for Set Membership and Range Proofs. In *Asiacrypt'08*, LNCS 5330, pp. 234–252, 2008.
13. J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *Eurocrypt'09*, LNCS 5479, pp. 351–368, 2009.
14. J. Camenisch and T. Groß. Efficient Attributes for Anonymous Credentials. In *ACM-CCS'08*, pp. 345–356, ACM Press, 2008. Extended version available from `http://eprint.iacr.org/2010/496`.
15. J. Camenisch, S. Hohenberger and A. Lysyanskaya. Compact E-Cash. In *Eurocrypt'05*, LNCS 3494, pp. 302–321, 2005.
16. J. Camenisch, M. Kohlweiss and C. Soriente. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In *PKC'09*, *LNCS* 5443, pp. 481–500, 2009.
17. J. Camenisch and A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *Eurocrypt'01*, *LNCS* 2045, pp. 93–118, 2001.
18. —, A Signature Scheme with Efficient Protocols. In *SCN'02*, *LNCS* 2576, pp. 268–289, 2001.
19. —, Signature Schemes and Anonymous Credentials from Bilinear Maps. In *Crypto'04*, *LNCS* 3152, pp. 56–72, 2004.
20. D. Chaum. Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM, 28(10), pp. 1030–1044, 1985.
21. J. Cathalo, B. Libert and M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt'09*, *LNCS* 5912, pp. 179–196, 2009.
22. D.-W. Cheung, N. Mamoulis, W.-K. Wong, S.-M. Yiu and Y. Zhang. Anonymous Fuzzy Identity-based Encryption for Similarity Search. In *ISAAC 2010*, *LNCS* 6506, pp. 61–72, 2010.
23. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto'86*, LNCS 263, pp. 186–194, 1986.
24. G. Fuchsbauer. Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. Cryptology ePrint Archive: Report 2009/320, 2009.
25. G. Fuchsbauer. Commuting Signatures and Verifiable Encryption and an Application to Non-Interactively Delegatable Credentials. In *Eurocrypt'11*, LNCS 6632, pp. 224–245, 2011.

26. E. Fujisaki, T. Okamoto. Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. In *Crypto'97*, *LNCS* 1294, pp. 16–30, 1997.
27. S. Goldwasser, S. Micali, R. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM J. Comput.* 17(2), pp. 281–308, 1988.
28. S. Goldwasser and Y. Tauman-Kalai. On the (In)security of the Fiat-Shamir Paradigm In *FOCS'03*, pages 102–115, 2003.
29. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
30. M. Jakobsson, K. Sako, R. Impagliazzo. Designated Verifier Proofs and Their Applications. In *Eurocrypt'96*, LNCS 1070, pp. 143–154, 1996.
31. J. Katz. Efficient and Non-malleable Proofs of Plaintext Knowledge and Applications. In *Eurocrypt'03*, LNCS 2656, pp. 211–228, 2003.
32. S. Kunz-Jacques and D. Pointcheval. About the security of MTI/C0 and MQV. In *SCN'06*, LNCS 4116, pp. 156–172, 2006.
33. J. Katz, A. Sahai and B. Waters. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In *Eurocrypt'08*, *LNCS* 4965, pp. 146-162, 2008.
34. B. Libert and M. Yung. Concise Mercurial Vector Commitments and Independent Zero-Knowledge Sets with Short Proofs. In *TCC'10*, LNCS 5978, pp. 499–517, 2010.
35. H.K. Maji, M. Prabhakaran, M. Rosulek. Attribute-based signatures. In *CT-RSA'11*, *LNCS* 6558, pp. 376–392, 2011.
36. T. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Crypto'91*, *LNCS* 576, pp. 129–140, 1991.
37. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Eurocrypt'97*, *LNCS* 1233, pp. 256–66, 1997.
38. A. Rial, M. Kohlweiss and B. Preneel. Universally Composable Adaptive Priced Oblivious Transfer. In *Pairing'09*, *LNCS* 5671, pp. 231–247, 2009.
39. S.-F. Shahandashti, R. Safavi-Naini. Threshold Attribute-Based Signatures and Their Application to Anonymous Credential Systems. In *Africacrypt'09*, *LNCS* 5580, pp. 198–216, 2009.
40. J. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. Journal of the ACM 27, pp. 701717, 1980.