

Multimodal Private Signatures

Khoa Nguyen^[0000-0001-8555-638X], Fuchun Guo^[0000-0001-6939-7710],
Willy Susilo^[0000-0002-1562-5105], and Guomin Yang^[0000-0002-4949-7738]

Institute of Cybersecurity and Cryptology,
School of Computing and Information Technology,
University of Wollongong, Northfields Avenue, Wollongong NSW 2522, Australia
khoa@uow.edu.au, fuchun@uow.edu.au, wsusilo@uow.edu.au, gyang@uow.edu.au

Abstract. We introduce Multimodal Private Signature (MPS) - an anonymous signature system that offers a novel accountability feature: it allows a designated opening authority to learn *some partial information* op about the signer's identity id , and nothing beyond. Such partial information can flexibly be defined as $\text{op} = \text{id}$ (as in group signatures), or as $\text{op} = \mathbf{0}$ (like in ring signatures), or more generally, as $\text{op} = G_j(\text{id})$, where $G_j(\cdot)$ is a certain disclosing function. Importantly, the value of op is known in advance by the signer, and hence, the latter can decide whether she/he wants to disclose that piece of information. The concept of MPS significantly generalizes the notion of tracing in traditional anonymity-oriented signature primitives, and can enable various new and appealing privacy-preserving applications.

We formalize the definitions and security requirements for MPS. We next present a generic construction to demonstrate the feasibility of designing MPS in a modular manner and from commonly used cryptographic building blocks (ordinary signatures, public-key encryption and NIZKs). We also provide an efficient construction in the standard model based on pairings, and a lattice-based construction in the random oracle model.

Keywords: new models, anonymous authentications, accountability, fine-grained information disclosure, modular constructions, zero-knowledge, lattices, pairings

1 Introduction

Privacy is a fundamental human right and is an interdisciplinary area of study [54]. In the digital era, where most of our daily communications are done over computer networks, the problem of privacy protection has become increasingly important and challenging. On the other hand, the development of information technology and cryptography also brings new technical solutions for privacy protection. Since the 1980s [16], various privacy-preserving cryptographic protocols have been proposed for this purpose. This essential area gets a lot of traction not only because of growing practical demands, but also due to its great theoretical interests. Indeed, designing these advanced systems is highly challenging, as they

typically require not only basic algorithms but also non-trivial and specially-developed tools such as zero-knowledge proofs [27] - a beautiful tool allowing to prove the truth of given statements without revealing any additional information.

While privacy-sensitive users want to protect their anonymity as much as possible, excessive privacy could be abused for illegal or inappropriate activities. Hence, from the system authorities' viewpoint, all users who carry out problematic activities should be kept accountable. Thus, there is an uneasy "privacy vs accountability" tension corresponding to the incentives of users and authorities. In privacy-preserving cryptosystems focusing on anonymity that had been proposed before the year 2021, either the users are granted absolute anonymity and can never be traced [50,44,3], or there exists an authority who can break users' privacy without their consent [17,36,24,18]. In other words, these systems always lean rigidly, either in favour of the users or of the authorities. A breakthrough in tackling the "privacy vs accountability" tension was recently put forward in [42], which introduced Bifurcated Anonymous Signatures (BiAS) - a novel primitive in which whether the signer of a given signature can have absolute anonymity or can be traced is made context-dependent and is known to the signer at the time of signature generation. As a result, tracing can only be done with users' consent on the one hand, and no traceable signature can escape being traced on the other hand. This primitive provides a reasonably fair setting for both authorities and users and seems to have offered a satisfying resolution for the discussed tension.

However, a crucial disadvantage of BiAS and of previous proposals is that accountability is realized via a total tracing procedure, during which all the personal identifying information of the traced users must be disclosed to the authorities. This level of accountability is indeed a serious violation of users' privacy. Note that, while privacy is a complicated notion that has differed throughout history [54], in its purest sense, it can be defined as the right of an individual to control which information about herself or himself can be disclosed [45]. Furthermore, in many real-life situations, it is not necessarily the authorities' highest priority to perform a total tracing. For instance, the authorities could only be interested in learning whether an anonymous user is over 18 years old, or works in a given organization, or lives in a particular area, or has been fully vaccinated against COVID-19, or has an annual income exceeding certain threshold. In the following, let us discuss several concrete examples.

Consider the scenario where an anonymous financial transaction (such as the privacy-preserving cryptocurrency Monero [47]) with a hidden amount of money is used to conduct online transactions. When an amount less than \$100 is transferred, then the transaction will be anonymous to everyone, including the authority. However, when an amount between \$100 and \$1,000 is transferred, the authority will be able to evaluate partial information about the sender, namely which country the sender originated from. When an amount between \$1,000 and \$10,000 is transferred, then the authority will be able to identify the country and the organization where the transfer originated from. When an amount larger than \$10,000 is transferred, then the identity of the individual from the organization in that specific country will be identified. In other words, depending

on the underlying transaction amounts, the authority can learn different pieces of information about the sender. There are four different levels of information disclosure in the above scenario.

A more simplistic scenario can be related to an IP address, which is of the form of $w.x.y.z$. When a small transaction is issued, then the authority will not be able to learn any information about the address. However, when a medium range of transactions is issued, then the authority will be able to compute $w.x.*.*$, which denote the range of IP addresses within an organization. Finally, then a large transaction is issued, then the full IP address can be identified.

As another example, imagine the situation where a data broker company quietly sells people’s personal information to others. While this activity is illegitimate, especially with the introduction of GDPR (General Data Protection Regulation), this kind of activity remains happening in the wild. Suppose a whistleblower who works in the data broker company, wants to “leak” this information to the authority. The purpose is to allow the authority to trace the data broker company while protecting the whistleblower’s identity. Therefore, it is essential that the whistleblower can still sell the data from the data broker correctly. Those data eventually will trigger the authority to find some partial identity information from the whistleblower, which points to the data broker company.

Unfortunately, all existing cryptographic methods fail to offer such type of balance between privacy and accountability, i.e., a setting in which authorities can only learn the piece of partial information about the user that the latter would like to disclose - and nothing else. Providing such fine-grained accountable privacy is a highly important and desirable research goal, and addressing it would likely require truly innovative technical ideas and approaches.

OUR CONTRIBUTIONS. We put forward the concept of “Multimodal Private Signatures” (MPS), which provides a novel approach for private information disclosure in anonymity-oriented authentication systems. In an MPS scheme, registered users can generate signatures that remain anonymous to the public, but can be opened by the authority to some partial information op on the identity of the signer. Such partial information can flexibly be defined as $\text{op} = \text{id}$ (as in group signatures), or as $\text{op} = \mathbf{0}$ (like in ring signatures), or more generally, as $\text{op} = G(\text{id})$, where $G(\cdot)$ is certain “disclosing function”. Importantly, the value of op is known in advance by the signer, and hence, the latter can decide whether she/he wants to disclose that piece of information.

In group signature, the disclosing function $G(\cdot)$ is basically the identity function, and in BiAS, $G(\cdot)$ is an all-or-nothing function. However, as mentioned in the examples motivating MPS, a set of more flexible and fine-grained disclosing functions are demanded to balance privacy and accountability in different applications. In MPS, this is achieved via two steps: first, we introduce a signing function F that determines whether a message M is valid (e.g., the transaction amount is below the limit set by the monetary authority), and if so, the critical level j of M ; secondly, we define a family of disclosing functions $\mathcal{G} = \{G_j(\cdot)\}$ that discloses the appropriate level of identity information (i.e., $G_j(\text{id})$) to the

opening authority based on the critical level of M . It is worth noting that, for privacy purposes, we also want to hide the critical level j , meaning M could be a transformation of the real “message”, which we call the witness w of M . Looking ahead, in the pairing-based and lattice-based constructions presented in this paper, we use $M = \text{COM}(w)$ where $\text{COM}(\cdot)$ denotes a secure commitment scheme. Clearly, the privacy against the opening authority in MPS is more intricate than that in other traceable anonymous signatures. Specifically, we require that the opening authority learns only $G_j(\text{id})$ from a valid signature but nothing else.

More formally, an MPS system is associated with a collection of signing functions \mathcal{F} and a collection of disclosing functions $\{G_1, \dots, G_K\}$. When user id would like to sign a message M with respect to signing function $F \in \mathcal{F}$, it computes $j = F(M, w, \text{id}) \in [0, K]$, where w is a “witness” - a context-dependent piece of information available to the user (that intuitively explains why $F(M, w, \text{id}) = j$). The value of j governs the signability of (M, w, id) as well as the accountability of the resulting signature. Specifically, if $j = 0$, then id is not allowed to sign. Otherwise, then id should be able to obtain a valid signature that can be opened by the authority to the value $G_j(\text{id})$.

The concept of MPS could enable various appealing applications that previously have not been considered or realized. Apart from the examples we discussed above, let us provide a few more illustrating scenarios.

In the context of anonymous surveys, one may implement an MPS system allowing the survey conductor to learn some specific piece of information (e.g., age, gender, location) about participants who provided answers that meet certain conditions (with the participant’s consensus). As for private access to buildings or to online systems, the administrator may also use an MPS system so that to gain certain statistics about the characteristics or activities of the anonymous visitors. From another perspective, the signers may also use MPS to purposely send some information to the authorities, e.g., for claiming the financial incentives of releasing the signed messages.

Let us consider another hypothetical scenario concerning paper submissions and reviews for a conference. An MPS system can help to keep both the authors and the reviewers anonymous to the PC chair, yet allowing the latter to check for CoI (Conflict of Interest). To this end, when submitting a paper, the author signs the paper together with a commitment c to her identity id . The chair can set up the system so that he can open the author’s affiliation z based on the signature. If the paper is later accepted, the author can open c to reveal id and claim authorship. Meanwhile, PC members can anonymously post comments on the paper, yet disclose their affiliation z' to the chair. The latter hence can oversee if a CoI has occurred. Moreover, if a PC member would like to post a negative comment on the paper, such as “I previously reviewed this paper and the authors did not take my comments into account.”, then it should be backed up with a legitimate witness w . Such a setting therefore can provide a much higher level of privacy protection than contemporary conference management systems.

Having convinced ourselves that MPS is a highly interesting concept, we come to the next steps: formal definitions and technical constructions for MPS.

Formalizations of MPS. To formalize MPS, we follow the setting of dynamic group signatures [5,33], that was also employed in [42]. Namely, an MPS scheme is a tuple of algorithms (**Setup**, \langle **Join**;**Issue** \rangle , **Sign**, **Verify**, **Open**). The main difference here is that MPS additionally relies on signing functions $F \in \mathcal{F}$ to control signability of (M, w, id) and disclosing functions G_1, \dots, G_K to realize partial information disclosure. Correctness ensures that as long as $j = F(M, w, \text{id}) \neq 0$ where id is a joined user, then the resulting signature Σ should be accepted by the verification algorithm and should be opened to $G_j(\text{id})$. Regarding security, we demand two major requirements: privacy and unforgeability. For each of these notions, we consider two types of adversary.

Regarding privacy, the first type of adversary can corrupt everyone in the system, except the opening authority. This adversary acts like the CCA2-anonymity adversary in group signatures. It is given the secret key of the group manager (GM) - who is in charge of user enrolments, as well as signing keys of all users. It is not allowed to corrupt the opening authority (OA), but it can adaptively query the opening oracle. Roughly speaking, we require that it should be infeasible for this adversary to learn any information about the signer id beyond the fact that M is signable for id . The second type of privacy adversary is even stronger, as it is even allowed to corrupt the OA. For this adversary, we require that no additional information beyond $G_j(\text{id})$ can be learned. (Note that the OA can always learn $G_j(\text{id})$.)

As for unforgeability, we would like to capture several requirements. First, it should be infeasible for signer with identifier id to generate a valid signature Σ associated with (M, F) if $F(M, w, \text{id}) = 0$. Second, it should also be infeasible to “mislead” the signature opening: if **Open** outputs op , then we expect that there exists a registered id whose valid signing key was used by the signer as well as a witness w such that $\text{op} = G_{F(M,w,\text{id})}(\text{id})$. Third, we demand that, no one, even a coalition of a corrupted GM and a corrupted OA, can issue signatures on behalf of honest user id . Note that the last two requirements resemble the notions of full-traceability and non-frameability in dynamic group signatures [5,33].

However, formally defining unforgeability for MPS is a considerably non-trivial task. The main reason is that the original algorithms in the system do not provide a rigorous mechanism to determine whether a tuple (M^*, F^*, Σ^*) forms a valid forgery. In particular, invoking **Open** only provides us with a value op , which does not allow us to answer crucial questions such as: (i) Is message M^* actually signable with respect to F^* and some id ? (ii) Is this the case that $\text{op} = G_{F(M^*,w',\text{id}')}(\text{id}')$ for some (id', w') ? Therefore, for definitional purposes, we would need to introduce certain auxiliary algorithms, namely, **SimSetup** and **Extract**, that allow us to extract additional information, e.g., some identity id' and some witness w' , so that we can meaningfully explain whether and how a forgery has occurred. We note that, previous works such as [3,42] also had to overcome similar situations.

Generic Constructions. Next, to demonstrate the feasibility of building MPS based on standard assumptions and in a modular manner, we provide a generic construction. The construction makes use of common cryptographic tools including ordinary signatures, public-key encryption, and non-interactive zero-knowledge (NIZK) proofs/arguments with two (indistinguishable) modes: a hiding mode with statistical zero-knowledge, and a binding mode with statistical soundness and extractability. At a high level, the construction follows the classical sign-then-encrypt-then-prove paradigm that is typically used for building group signatures [17,4]. The main difference here is that we do not encrypt the signer’s identity id (as in group signatures) or “ id or 0” (as in BiAS [42]). Instead, we let the signer encrypt the function value $\text{op} = G_j(\text{id})$ and prove the well-formedness of the resulting ciphertext - which includes proving knowledge of (id, w) such that $\text{op} = G_{F(M,w,\text{id})}(\text{id})$ is contained in the ciphertext. While such involved statements can be proved in zero-knowledge using well-known NIZK systems for NP such as [29,49], the resulting proofs/arguments would likely have sizes depending on the sizes of the circuits computing functions F, G_1, \dots, G_K .

Theoretically speaking, the dependency of the proof size (and hence, of the signature size) can potentially be reduced by using advanced techniques such as fully-homomorphic encryption (FHE) [25,42], for which the main idea is to compute over encrypted data so that to (publicly) obtain a ciphertext that will decrypt to $G_{F(M,w,\text{id})}(\text{id})$. Nevertheless, using FHEs in that manner would require significant computation costs and/or a large number of initial ciphertexts, and could end up being less efficient than the usual sign-then-encrypt-then-prove approach. We also investigate the potential of efficiently constructing MPS based on functional encryption (FE) [7], since the idea that decryption reveals a function of id is closely related to the spirit of FE. However, we have been unable to progress in this direction: the main obstacle is to ensure that only $G_j(\text{id})$ can be revealed via opening. For instance, giving the opening authority all the decryption keys corresponding to (G_1, \dots, G_K) would not work well, as the authority may additionally learn the index j . We therefore stick with the usual design approach, and leave efficient FHE-based and FE-based constructions of MPS as appealing open questions.

Our sign-then-encrypt-then-prove construction can also have efficiency advantage when we instantiate the system with concrete signing and disclosing functions, the correct evaluations of which can be efficiently proved in zero-knowledge. As illustrations, we provide a relatively efficient pairing-based construction in the standard model, as well as a lattice-based scheme in the random oracle model (ROM) that potentially enjoys post-quantum security. To be more specific, in both instantiations, we consider \mathcal{G} to be a family of linear transformation functions on id , which are sufficient for many of the motivating applications.

Pairing-Based Constructions in the Standard Model. We present an instantiation of the generic construction under pairing groups. The core components of the construction include the Groth-Sahai proof system [30], a structure-preserving signature (SPS) scheme [35], the Boneh-Boyen (BB) signature [6] and a tag-based PKE [34].

We can apply the aforementioned tools to construct efficient group signatures without random oracles, as shown in [28]. The main challenge to construct an MPS is handling the disclosed identity information $G_j(\text{id})$. Different from group signature, we need to ensure not only the encrypted $G_j(\text{id})$ matches the real id but also the disclosing function $G_j(\cdot)$ is the correct function to be applied.

In our construction, we consider a message $M \in \mathbb{G}_1$ being signed to be a Pedersen commitment [48] for some value $v \in \mathbb{Z}_p$. The disclosing function to be applied when signing a message depends on the value of v . For simplicity, in our instantiation, we consider 4 possible ranges $[A_{j-1}, A_j]$ ($1 \leq j \leq 4$), and for each range, we define a disclosing function $G_j(\text{id})$. When generating a signature on M , the signer needs to compute a ciphertext ct of $G_j(\text{id})$ under the OA's public key opk and then prove that

$$M = \text{COM}(v) \wedge A_{j-1} \leq v < A_j \wedge \text{ct} = \text{Enc}(\text{opk}, G_j(\text{id})).$$

To ensure the correct extraction of $G_j(\text{id})$, we need to extract the value v from the NIZK proof. However, the Groth-Sahai proof does not support the extraction of a random value in \mathbb{Z}_p . To address this issue, we convert the above statement by utilizing the homomorphic property of the Pedersen commitment. Instead of proving $A_j \leq v < A_{j+1}$, we let the signer represent the value committed in M/g^{A_j} , i.e., $v - A_j$, as a k -bit binary number, so that each bit can be extracted. In addition, to extract the specific range, among all the possible ranges, the value v actually falls in, we add two additional bits and express the proof statement in the form of an OR-statement, where the additional bits point to the real statement being proved.

Lattice-Based Constructions. While it is feasible to instantiate MPS in the standard model via the lattice-based NIZK techniques of Peikert and Shiehian [49], such a construction would expectedly be extremely inefficient. Here, our goal is to build more efficient constructions in the ROM, where we can employ concrete techniques for obtaining interactive ZK arguments for lattice-based relations, and then remove interaction via the Fiat-Shamir transformation [22].

Similar to our pairing-based construction, here we consider the setting with 1 signing function F and 4 disclosing functions. We also let message M be a commitment to witness w and define $j = F(M, w) \in [0, 4]$ based on integer ranges. We consider 4 disclosing functions, and for each $j \in [1, 4]$ define G_j as a linear endomorphism over \mathbb{Z}_2^k . Specifically, let $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \mathbf{G}_4 \in \mathbb{Z}_2^{k \times k}$ be public matrices, then let $G_j(\text{id}) := \mathbf{G}_j \cdot \text{id}$. This definition is quite general and expressive, in the sense that it captures many natural ways to disclose partial information about id . For instance, we can set $\mathbf{G}_1 = \mathbf{0}^{k \times k}$ and $\mathbf{G}_4 = \mathbf{I}_k$, so that $G_1(\text{id}) = \mathbf{0}$ (i.e., non-traceable case) and $G_4(\text{id}) = \text{id}$ (i.e., fully traceable case). We can also easily define $\mathbf{G}_2, \mathbf{G}_3$ so that $G_2(\text{id}), G_3(\text{id})$ each reveals a specific subset of coordinates of id .

Our construction is proven secure under the Learning With Errors (LWE) and the Short Integer Solutions (SIS) assumptions. The construction employs the following lattice-based building blocks: (i) the KTX SIS-based commitment scheme [31]; (ii) the SIS-based signature scheme from [37], which admits efficient

zero-knowledge arguments of knowledge of a valid message-signature pair; (iii) an LWE-based CCA2-secure PKE scheme obtained from the GPV IBE [26] and the CHK transformation [14]; (iv) a SIS-based one-way function [1]; and (v) an interactive statistical ZK argument system that can handle relatively sophisticated linear and quadratic relations with respect to two moduli ($q_1 = 2$ and $q_2 > 2$) and that is compatible with the signature scheme from [37]. Indeed, we need to prove in ZK that a plaintext \mathbf{y} , encrypted under the GPV IBE scheme, is exactly the value $G_j(\text{id})$, which is the major technical difficulty in our design process. To this end, we adapt the Stern-like [53] framework from [40] and then, employ several dedicated techniques to capture the relation $\mathbf{y} = G_j(\text{id})$ by equations modulo 2, that are compatible with the framework. We note that, there are more efficient systems, such as [56,9,21,20], however, they are not known to be applicable to the two-moduli setting here.

RELATED WORK. There has been a vast body of work on anonymity-oriented signature systems. One of the most prominent examples is group signature [17], in which registered users are allowed to anonymously sign any message, but are fully traceable by the opening authority. Group signature thus can be viewed as a special case of MPS with a single disclosing function $G(\text{id}) = \text{id}$. Ring signature [51], another well-known primitive, provides anonymity with no tracing, yet can also be seen as an MPS system with $G(\text{id}) = \mathbf{0}$. Accountable ring signature [55,8] offers either the ring-signature functionality or the group-signature functionality, but the two modes are separated and distinguishable. Bifurcated anonymous signature (BiAS) [42], a recently proposed concept, simultaneously provide both “ring-signature mode” and “group-signature mode”, as well as indistinguishability between the two modes. BiAS is therefore a special case of MPS, with two disclosing functions $G_1(\text{id}) = \mathbf{0}$ and $G_2(\text{id}) = \text{id}$ (but no signing functions).

There have also been various attempts to increase the privacy of signers against the opening authorities in group signatures, such as traceable signatures [32], group signatures with message-dependent opening [52], accountable tracing signatures [36] or threshold group signatures [11]. In the reverse direction are proposals that aim to increase signers’ accountability, such as traceable ring signatures [23], e-cash-related primitives [12,13] and traceable attribute-based signatures [19]. However, in all these systems, the disclosing functions, once activated, would reveal the full identity, i.e., $G(\text{id}) = \text{id}$.

Attribute-based signature [44] and predicate signature [2,46] provide fine-grained controls on “who can sign”, while policy-based signature [3] and functional signature [10] govern “which messages can be signed”. These controls of signability can also be viewed as instances of MPS’s signing functions $F(M, w, \text{id})$ (with restricted function range $\{0, 1\}$, rather than $[0, K]$).

As a summary, MPS does capture the appealing features of the primitives listed above, and does further generalize and empower them in several dimensions. In particular, the attractive generalization from all-or-nothing tracing of signer’s identity to fine-grained disclosure of signer’s partial information could have a great impact in this research area.

At a high level, our conception of MPS based on group signature somewhat resembles the revolutionizing conception of functional encryption [7] over ordinary PKE, in the sense that the decrypting/opening procedure can only reveal a function of the plaintext/identity, rather than the whole plaintext/identity. However, from a more technical perspective, there could be some crucial difference: while it is known how to build group signatures from PKE in a modular manner, the connection between MPS and functional encryption is still unclear.

ORGANIZATION. The rest of the paper is organized as follows. In Section 2, we provide our definitions of MPS, describe its syntax and formalize the security requirements. Then, in Section 3, we give a generic construction of MPS satisfying our model, based on commonly used cryptographic primitives. A pairing-based instantiation is then presented in Section 4. A lattice-based construction then follows in Section 5. We finally list several interesting open questions in Section 6.

Due to space restrictions, the reminders on the cryptographic building blocks employed in our constructions and most of the security analyses have to be deferred to the full version.

2 Multimodal Private Signatures

2.1 Syntax

Let $\lambda \in \mathbb{N}$ be a security parameter. Any Multimodal Private Signature system is associated with natural numbers $N, K \in \text{poly}(\lambda)$; a message space \mathcal{M} ; a witness space \mathcal{W} ; an identity space \mathcal{ID} ; an opening space \mathcal{OP} ; together with a collection \mathcal{F} of N signing functions and a collection of K disclosing functions $\mathcal{G} = \{G_1, \dots, G_K\}$, where

$$F : \mathcal{M} \times \mathcal{W} \times \mathcal{ID} \rightarrow [0, K], \forall F \in \mathcal{F}; \quad G_j : \mathcal{ID} \rightarrow \mathcal{OP}, \forall j \in [1, K].$$

The parties involving in an MPS system are similar to those of dynamic group signatures [5,33], namely, a trusted authority (TA), a group manager (GM), an opening authority (OA), signers and verifiers. The job of TA consists of setting up the system, announcing the public parameters and providing a secret key for each of GM and OA. Eligible signers are enrolled to the system via an interactive protocol with GM - who records the registration information into a table. A registered signer with personal identifiable information $\text{id} \in \mathcal{ID}$ can issue a signature Σ on a message $M \in \mathcal{M}$ and with respect to function $F \in \mathcal{F}$, if the signer possesses a witness $w \in \mathcal{W}$ such that $j = F(M, w, \text{id}) \neq 0$, i.e., $j \in [1, K]$. Here, the witness w is a context-dependent string that (intuitively) serves as an evidence for the signability of id on M and w.r.t F , and how w comes into the signer's possession is outside of the model (see also discussions in [3,42]). A legitimate signature Σ should be publicly verifiable by any verifier, and could be opened by OA - who would then learn the value of $G_j(\text{id}) \in \mathcal{OP}$.

Formally, an MPS scheme associated with $(N, K, \mathcal{M}, \mathcal{W}, \mathcal{ID}, \mathcal{OP}, \mathcal{F}, \mathcal{G})$ is a tuple of polynomial-time algorithms (**Setup**, **Join; Issue**, **Sign, Verify, Open**), defined as follows.

Setup(λ) \rightarrow (pp , msk , osk , reg). On input security parameter λ , this probabilistic algorithm generates public parameters pp , a secret key msk for the Group Manager (GM) and a secret key osk for the Opening Authority (OA). It also initializes a registration table $\text{reg} := \emptyset$.

$\langle \text{Join}(\text{pp}); \text{Issue}(\text{pp}, \text{msk}, \text{reg}) \rangle$. This is an interactive protocol run by a user who wishes to become a group member and the GM. If it completes successfully, then:

- Algorithm **Join** outputs user’s signing key $\text{sk}_{\text{id}} = (\text{id}, \text{sec}_{\text{id}}, \text{cert}_{\text{id}})$, where $\text{id} \in \mathcal{ID}$ is a unique identifier, sec_{id} is a membership secret (that is known only by the user), and cert_{id} is a membership certificate.
- Algorithm **Issue** stores the transcript of the protocol in the registration table $\text{reg} := \text{reg} \cup \text{trans}_{\text{id}}$.

Sign($\text{pp}, \text{sk}_{\text{id}}, M, w, F$) $\rightarrow \Sigma / \perp$. Given pp , signing key $\text{sk}_{\text{id}} = (\text{id}, \text{sec}_{\text{id}}, \text{cert}_{\text{id}})$, message $M \in \mathcal{M}$, witness $w \in \mathcal{W}$, and a signing function $F \in \mathcal{F}$, this probabilistic algorithm outputs a signature Σ or a symbol \perp indicating failure.

Verify(pp, M, F, Σ) $\rightarrow 1/0$. This deterministic algorithm checks the validity of the signature Σ on message $M \in \mathcal{M}$ with respect to signing function $F \in \mathcal{F}$. It outputs a bit indicating the validity or invalidity of Σ .

Open($\text{pp}, \text{osk}, \Sigma, M, F$) $\rightarrow \text{op} / \perp$. This algorithm takes as inputs the public parameters pp , the OA’s secret key osk , a signature Σ on message $M \in \mathcal{M}$ with respect to signing function $F \in \mathcal{F}$. It outputs either an opening result $\text{op} \in \mathcal{OP}$ or symbol \perp to indicate failure.

2.2 Correctness and Security

The requirements that any Multimodal Private Signature system should satisfy are *correctness*, *privacy* and *unforgeability*.

CORRECTNESS. Correctness of MPS guarantees that honest signers can join the group, and when $j = F(M, w, \text{id}) \neq 0$, signer id should be able to issue an accepted signature Σ on message M and with respect to signing function F , and that Σ should be opened to the value $G_j(\text{id})$. More formally, correctness of MPS is defined as follows.

Definition 1 (Correctness). *An MPS system associated with $(N, K, \mathcal{M}, \mathcal{W}, \mathcal{ID}, \mathcal{OP}, \mathcal{F}, \mathcal{G})$, where $\mathcal{G} = \{G_1, \dots, G_K\}$, is called correct, if for all $\lambda \in \mathbb{N}$, all $(\text{pp}, \text{msk}, \text{osk}, \text{reg}) \leftarrow \text{Setup}(\lambda)$, the following conditions hold with overwhelming probability in λ .*

1. If $\langle \text{Join}(\text{pp}); \text{Issue}(\text{pp}, \text{msk}, \text{reg}) \rangle$ is run by two honest parties, then it completes successfully, and the signer obtains $\text{sk}_{\text{id}} = (\text{id}, \text{sec}_{\text{id}}, \text{cert}_{\text{id}})$.
2. If $M \in \mathcal{M}$, $F \in \mathcal{F}$, $\text{id} \in \mathcal{ID}$, $w \in \mathcal{W}$ and if $j = F(M, w, \text{id}) \in [1, K]$, then algorithm **Sign**($\text{pp}, \text{sk}_{\text{id}}, M, w, F$) does not fail and

$$\begin{aligned} \text{Verify}(\text{pp}, M, F, \text{Sign}(\text{pp}, \text{sk}_{\text{id}}, M, w, F)) &= 1 \\ \text{Open}(\text{pp}, \text{osk}, \text{Sign}(\text{pp}, \text{sk}_{\text{id}}, M, w, F), M, F) &= G_j(\text{id}). \end{aligned}$$

SECURITY. We require two main security properties for MPS, namely, privacy and unforgeability. Informally, these properties capture the following intuitions.

Privacy roughly ensures that each party in the system can only learn the piece of signer’s information which the signer intends to disclose. Given a valid signature $\Sigma \leftarrow \text{Sign}(\text{pp}, \text{sk}_{\text{id}}, M, w, F)$, it should be infeasible for everyone - excluding the OA - to learn anything about the signer’s private information, apart from the fact that M is signable, i.e., $j = F(M, w, \text{id}) \neq 0$. Furthermore, even the OA should be able to additionally learn only the value $G_j(\text{id})$, and should remain oblivious about j and id .

Unforgeability captures several requirements. First, it should be infeasible for signer with identifier id to generate a valid signature Σ associated with (M, F) if $F(M, w, \text{id}) = 0$. Second, it should also be infeasible to “mislead” the signature opening: if $\text{Open}(\text{pp}, \text{osk}, \Sigma, M, F)$ outputs $\text{op} \in \mathcal{OP}$, then we expect that there exist a registered id whose valid signing key was used by the signer as well as a witness $w \in \mathcal{W}$ such that $\text{op} = G_{F(M, w, \text{id})}(\text{id})$. Third, we demand that, without the knowledge of membership secret sec_{id} , no one, even a coalition of corrupted GM and OA, can issue signatures on behalf of honest user id . Note that the last two requirements resemble the notions of full-traceability and non-frameability in dynamic group signatures [5,33].

For each of the above security properties, we therefore will consider two types of adversaries, whose goals and powers are related but different from each other. For formalization, we will follow the definitional approach used by Libert et al. [42], which was first put forward by Kiayias and Yung [33].

We will consider experiments in which the adversary interacts with a stateful interface \mathcal{I} that maintains the following variables:

- $\text{state}_{\mathcal{I}}$: is a data structure representing the state of the interface as the adversary invokes the various oracles available in the attack games. It is initialized as $\text{state}_{\mathcal{I}} = (\text{pp}, \text{msk}, \text{osk}, \text{reg})$, where reg is initially empty and later will store all transcripts of $\langle \text{Join}; \text{Issue} \rangle$.
- SIGS: is a database of honestly generated signatures created by the signing oracle. Each entry consists of a tuple $(\Sigma, \text{id}, M, w, F)$ indicating that signature Σ was returned in response to a signing query involving identity id , message M , witness w and signing function F .
- HUL: is an initially empty list of honest users introduced in the system by the adversary acting as a dishonest GM. For these users, the adversary obtains the transcript of $\langle \text{Join}; \text{Issue} \rangle$ but not the user’s membership secret.
- CUL: is an initially empty list of corrupted users that are introduced by the adversary in the system in an execution of the join protocol.

In attack games, adversaries are granted access to the following oracles:

- \mathcal{O}_{CU} : allows the adversary to introduce users under its control in the group. A $\langle \text{Join}; \text{Issue} \rangle$ protocol is run, in which the adversary plays the role of

- the prospective user. If the protocol successfully completes, a new user id is added to CUL and the protocol transcript trans_{id} is added to reg .
- \mathcal{O}_{HU} : allows the adversary, acting as a corrupted GM, to introduce new honest group members of its choice. A $\langle \text{Join}; \text{Issue} \rangle$ protocol is run, in which the adversary plays the role of the GM. If the protocol successfully completes, a new user id is added to HUL and protocol transcript trans_{id} is added to reg . The interface stores the membership certificate cert_{id} and the membership secret sec_{id} in a *private* part of $\text{state}_{\mathcal{I}}$.
 - \mathcal{O}_{sig} : given a tuple (M, w, F) and an identifier id , the interface returns \perp if $F(M, w, \text{id}) = 0$ or if $\text{id} \notin \text{HUL}$. Otherwise, the private area of $\text{state}_{\mathcal{I}}$ must contain a certificate cert_{id} and a membership secret sec_{id} . The interface outputs a signature Σ on behalf of user id and also updates $\text{SIGS} \leftarrow \text{SIGS} \parallel (\Sigma, \text{id}, M, w, F)$.
 - $\mathcal{O}_{\text{open}}$: when this oracle is invoked on input of a valid triple (M, Σ, F) , the interface runs algorithm Open using osk . When S is a set of tuples of the form (M, Σ, F) , $\mathcal{O}_{\text{open}}^{-S}$ denotes a restricted oracle that only applies the opening algorithm to tuples (M, Σ, F) which are not in S .
 - $\mathcal{O}_{\text{read}}$ and $\mathcal{O}_{\text{write}}$: are used by the adversary to read and write the content of reg . At each invocation, $\mathcal{O}_{\text{read}}$ outputs the current records in reg . Meanwhile, $\mathcal{O}_{\text{write}}$ enables the adversary to modify reg as long as the table remains well-formed.

PRIVACY. We say that an MPS scheme is private if it satisfies **computational privacy** against Type-1-Adversary and **computational/statistical privacy** against Type-2-Adversary.

Privacy against Type-1 Adversary. This captures the power of the CCA2-anonymity adversary in group signatures [4,5,33]. The adversary is allowed to corrupt the GM, corrupt all users, and is allowed to make queries to various oracles, including adaptive queries to the opening oracle.

In the challenge phase, adversary returns a function $F^* \in \mathcal{F}$, a message $M^* \in \mathcal{M}$, together with two valid signing keys $\text{sk}_{\text{id}_0} = (\text{id}_0, \text{sec}_{\text{id}_0}, \text{cert}_{\text{id}_0})$, $\text{sk}_{\text{id}_1} = (\text{id}_1, \text{sec}_{\text{id}_1}, \text{cert}_{\text{id}_1})$, as well as witnesses $w_0, w_1 \in \mathcal{W}$. Here, by “valid signing keys”, we mean that the keys have been formed correctly via certain legitimate executions of $\langle \text{Join}; \text{Issue} \rangle$, initiated by the adversary. Furthermore, for the challenge to be meaningful, (M^*, F^*) should be signable by both id_0 and id_1 , i.e.,

$$(j_0 = F^*(M^*, w_0, \text{id}_0) \neq 0) \quad \wedge \quad (j_1 = F^*(M^*, w_1, \text{id}_1) \neq 0).$$

Receiving a challenge signature $\Sigma^* \leftarrow \text{Sign}(\text{pp}, \text{sk}_{\text{id}_b}, M^*, w_b, F^*)$, where $b \xleftarrow{\$} \{0, 1\}$, the adversary can continue making non-trivial opening queries, i.e., those that do not involve (M^*, Σ^*, F^*) . Eventually, it outputs a guess $b' \in \{0, 1\}$ and wins if the guess is correct with non-negligible advantage.

Privacy against Type-2 Adversary. This strong adversary can potentially be computationally unbounded and can corrupt everyone in the system: GM, all users and even OA. It is also allowed to make unrestricted queries to all available oracles. Privacy against this adversary roughly demands that, apart

from the opening result $G_j(\text{id})$ (and, obviously, the fact that the underlying message-function pair is signable by id), the adversary can learn no additional information about j or id .

In the challenge phase, when adversary returns $(F^*, M^*, \text{sk}_{\text{id}_0}, w_0, \text{sk}_{\text{id}_1}, w_1)$, we additionally require that $G_{j_0}(\text{id}_0) = G_{j_1}(\text{id}_1)$, namely, the opening information corresponding to both choices of the challenger must be the same. This restriction is necessary (as the adversary knows osk) and also sufficient to capture the requirement that signature opening only reveals $G_j(\text{id})$.

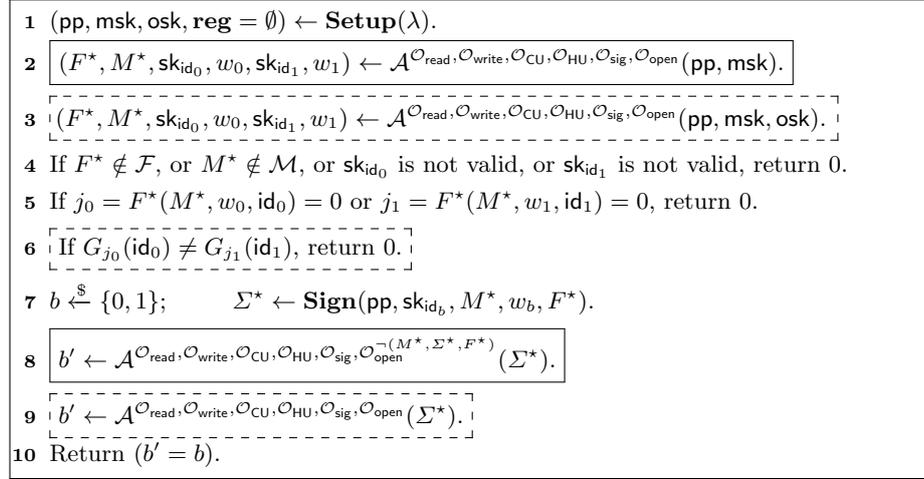


Fig. 1: Experiment $\text{Exp}_{\mathcal{A}}^{\text{privacy}-1}(\lambda)$ (*resp.*, $\text{Exp}_{\mathcal{A}}^{\text{privacy}-2}(\lambda)$) excluding the dotted (*resp.*, solid) boxes.

The respective experiments, $\text{Exp}_{\mathcal{A}}^{\text{privacy}-1}(\lambda)$ and $\text{Exp}_{\mathcal{A}}^{\text{privacy}-2}(\lambda)$, are described in Fig. 1. We hence come to the following formal definition of privacy for MPS.

Definition 2 (Privacy). *An MPS system associated with $(N, K, \mathcal{M}, \mathcal{W}, \mathcal{ID}, \mathcal{OP}, \mathcal{F}, \mathcal{G})$ is called private if the following conditions hold.*

1. **Computational privacy against Type-1 adversary:** For any PPT adversary \mathcal{A} , one has

$$\text{Adv}_{\mathcal{A}}^{\text{privacy}-1}(\lambda) := |\Pr[\text{Exp}_{\mathcal{A}}^{\text{privacy}-1}(\lambda) = 1] - 1/2| \in \text{negl}(\lambda).$$

2. **Statistical (*resp.*, computational) privacy against Type-2 adversary:** For any adversary \mathcal{A} (*resp.*, any PPT adversary \mathcal{A}), one has

$$\text{Adv}_{\mathcal{A}}^{\text{privacy}-2}(\lambda) := |\Pr[\text{Exp}_{\mathcal{A}}^{\text{privacy}-2}(\lambda) = 1] - 1/2| \in \text{negl}(\lambda).$$

UNFORGEABILITY. Defining unforgeability for MPS is a considerably non-trivial task. The main reason is that the original algorithms in the system do not provide

a rigorous mechanism to determine whether a tuple (M^*, F^*, Σ^*) forms a valid forgery. Therefore, for definitional purposes, we would need to introduce certain auxiliary algorithms that allow us to extract additional information, e.g., some identity id' and some witness w' , so that we can meaningfully explain whether and how a forgery has occurred.

To that end, we assume the existence of the following two auxiliary algorithms, namely, **SimSetup** and **Extract**.

SimSetup (λ) Given the security parameter λ , this algorithm generates simulated $(\text{pp}, \text{msk}, \text{osk}, \text{reg})$, together with an extraction trapdoor τ_{ext} .

Extract $(\tau_{\text{ext}}, (\text{pp}, \Sigma, M, F))$ Given trapdoor τ_{ext} , a *valid* signature Σ on message M and with respect to signing function F , i.e., $\text{Verify}(\text{pp}, M, F, \Sigma) = 1$, this extraction algorithm returns a pair $\zeta = (\text{id}', w') \in \mathcal{ID} \times \mathcal{W}$.

Naturally, we demand that the outputs of **SimSetup** and **Setup** are indistinguishable to the adversary. Next, we require that (id', w') outputted by **Extract** is compatible with the value op outputted by **Open**. Specifically, w' should be a valid witness for the signability of identity id' w.r.t. (M^*, F^*) , i.e., $j' = F(M^*, w', \text{id}') \neq 0$, and, furthermore, $G_{j'}(\text{id}')$ should coincide with op . Formally, we define *extractability* as a “supporting” security property for unforgeability. The definition uses experiment $\text{Exp}_{\mathcal{A}}^{\text{extract}}(\lambda)$ described in **Fig. 2**.

Definition 3 (Extractability). *An MPS system with auxiliary algorithms **SimSetup**, **Extract** is called extractable if the following conditions hold.*

1. *The distribution of simulated $(\text{pp}, \text{msk}, \text{osk}, \text{reg}) \leftarrow \text{SimSetup}(\lambda)$ is computationally close to the distribution of a real output of **Setup**.*
2. *For any PPT adversary \mathcal{A} involving in the experiment of **Fig. 2**, the advantage $\text{Adv}_{\mathcal{A}}^{\text{extract}}(\lambda) := \Pr[\text{Exp}_{\mathcal{A}}^{\text{extract}}(\lambda) = 1]$ is negligible in λ .*

- 1 $((\text{pp}, \text{msk}, \text{osk}, \text{reg} = \emptyset), \tau_{\text{ext}}) \leftarrow \text{SimSetup}(\lambda)$.
- 2 $(F, M, \Sigma) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{read}}, \mathcal{O}_{\text{write}}, \mathcal{O}_{\text{CU}}, \mathcal{O}_{\text{HU}}, \mathcal{O}_{\text{sig}}, \mathcal{O}_{\text{open}}}(\text{pp}, \text{msk}, \text{osk})$,
- 3 If $F \notin \mathcal{F}$, or $M \notin \mathcal{M}$, or $\text{Verify}(\text{pp}, M, F, \Sigma) = 0$, return 0.
- 4 $(\text{id}', w') \leftarrow \text{Extract}(\tau_{\text{ext}}, (\text{pp}, \Sigma, M, F))$; $j' = F(M, w', \text{id}')$;
- 5 If $j' = 0$, return 1.
- 6 $\text{op} \leftarrow \text{Open}(\text{pp}, \text{osk}, \Sigma, M, F)$;
- 7 If $G_{j'}(\text{id}') \neq \text{op}$, return 1.
- 8 Return 0.

Fig. 2: Experiment $\text{Exp}_{\mathcal{A}}^{\text{extract}}(\lambda)$.

Now, we are ready for the definitions of unforgeability. An MPS scheme is said to satisfy unforgeability if it is extractable and has computational security against Type-1-Forgery and Type-2-Forgery.

1	$((\text{pp}, \text{msk}, \text{osk}, \text{reg} = \emptyset), \tau_{\text{ext}}) \leftarrow \mathbf{SimSetup}(\lambda).$
2	$(M^*, F^*, \Sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{read}}, \mathcal{O}_{\text{write}}, \mathcal{O}_{\text{CU}}, \mathcal{O}_{\text{HU}}, \mathcal{O}_{\text{sig}}, \mathcal{O}_{\text{open}}}(\text{pp}, \text{osk}).$
3	$(M^*, F^*, \Sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{read}}, \mathcal{O}_{\text{write}}, \mathcal{O}_{\text{CU}}, \mathcal{O}_{\text{HU}}, \mathcal{O}_{\text{sig}}, \mathcal{O}_{\text{open}}}(\text{pp}, \text{osk}, \text{msk}).$
4	If $F^* \notin \mathcal{F}$, or $M^* \notin \mathcal{M}$, or $\mathbf{Verify}(\text{pp}, M^*, F^*, \Sigma^*) = 0$, return 0.
5	$(\text{id}', \cdot) \leftarrow \mathbf{Extract}(\tau_{\text{ext}}, (\text{pp}, \Sigma^*, M^*, F^*)).$
6	If $(\Sigma^*, \text{id}', M^*, \cdot, F^*) \in \text{SIGS}$, return 0.
7	If $\text{id}' \notin \text{CUL}$, return 1.
8	If $\text{id}' \in \text{HUL}$, return 1.
9	Return 0.

Fig. 3: Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{unforge-1}}(\lambda)$ (*resp.*, $\mathbf{Exp}_{\mathcal{A}}^{\text{unforge-2}}(\lambda)$) excluding the dotted (*resp.*, solid) boxes.

- **Type-1-Forger** roughly captures the traceability adversary in group signatures. It can fully corrupt the OA, corrupt a number of users and can make various oracle queries. Its goal is to output a valid forgery (Σ^*, M^*, F^*) such that the extraction points to some identity id' which it has not previously corrupted.
- **Type-2-Forger** is similar to the non-frameability adversary in group signatures, whose goal is to point the opening/extraction to an innocent user. The adversary can corrupt everyone else in the system, i.e., GM, OA and all other users. It succeeds if it can output a valid forgery that is extracted to some honest identity id' .

In **Fig. 3**, we formalize the respective security experiments, i.e., $\mathbf{Exp}_{\mathcal{A}}^{\text{unforge-1}}(\lambda)$ and $\mathbf{Exp}_{\mathcal{A}}^{\text{unforge-2}}(\lambda)$. The formal definition of unforgeability follows.

Definition 4 (Unforgeability). *An MPS system associated is called unforgeable if it satisfies extractability, and for any PPT adversary \mathcal{A} , one has*

$$\begin{aligned} \mathbf{Adv}_{\mathcal{A}}^{\text{unforge-1}}(\lambda) &:= \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{unforge-1}}(\lambda) = 1] \in \text{negl}(\lambda); \\ \mathbf{Adv}_{\mathcal{A}}^{\text{unforge-2}}(\lambda) &:= \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{unforge-2}}(\lambda) = 1] \in \text{negl}(\lambda). \end{aligned}$$

3 Generic Constructions

In this section, we present a generic construction of MPS for arbitrary signing functions F 's and arbitrary disclosing functions G_1, \dots, G_K . The construction satisfies the correctness and security properties defined in Section 2, and employs cryptographic building blocks that are commonly used for designing advanced privacy-preserving primitives: ordinary (one-time) signatures, public-key encryption and non-interactive zero-knowledge (NIZK) proofs/arguments for some NP-relations. For the latter ingredient, we additionally require the dual-mode

property, i.e., we will use a NIZK system that operates in two modes: hiding mode (for the real scheme and simulation) and binding mode (for simulated setup and extraction).

Our construction can serve as a proof of feasibility of designing MPS based on standard assumptions and in a modular manner. In particular, it can be realized in the standard model from pairings and from lattices, using the techniques for obtaining NIZKs for NP by Groth-Ostrovsky-Sahai [29] and by Peikert-Shiehian [49], respectively.

At a high level, the construction follows the classical sign-then-encrypt-then-prove paradigm. The main difference here is that we do not encrypt the signer’s identity id (as in group signatures) or “ id or 0 ” (as in BiAS [42]). Instead, we let the signer encrypt the function value $\text{op} = G_j(\text{id})$ and prove the well-formedness of the resulting ciphertext - which includes proving knowledge of (id, w) such that $\text{op} = G_{F(M,w,\text{id})}(\text{id})$ is contained in the ciphertext. While such involved statements can be proved in zero-knowledge using well-known NIZK systems for NP such as [29,49], the resulting proofs/arguments would likely have sizes depending on the sizes of the circuits computing functions F, G_1, \dots, G_K .

Our construction can also have efficiency advantage when we instantiate the system with *concrete* signing and disclosing functions, the correct evaluations of which can be efficiently proved in zero-knowledge. As illustrations, we will later present relatively efficient pairing-based and lattice-based constructions of MPS for some specific functions F, G_1, \dots, G_K , in Section 4 and Section 5, respectively.

In the following, we will give a technical overview of our generic construction in Section 3.1, then describe it in detail in Section 3.2 and provide its analyses in Section 3.3.

3.1 Technical Overview

The construction employs the following technical building blocks.

- A secure digital signature scheme $\mathcal{S} = (\text{S.Kg}, \text{S.Sign}, \text{S.Ver})$;
- A secure one-time signature scheme $\mathcal{OTS} = (\text{O.Kg}, \text{O.Sign}, \text{O.Ver})$;
- A secure public-key encryption scheme $\mathcal{E} = (\text{E.Kg}, \text{E.Enc}, \text{E.Dec})$;
- A dual-mode NIZK argument system $\mathcal{NIZK} = (\text{ZK.Setup}, \text{ZK.ExtSetup}, \text{ZK.Prove}, \text{ZK.Ver}, \text{ZK.Sim}, \text{ZK.Extr})$ for the NP-relation \mathcal{R} defined below.

The main ideas underlying the construction are as follows. The GM is associated with a signing-verification key-pair (msk, mpk) for \mathcal{S} , while the OA is associated with a decryption-encryption key-pair (osk, opk) for \mathcal{E} . When joining, a perspective user generates a signature key-pair $(\text{sec}_{\text{id}}, \text{upk})$, sends upk together with its personal identifiable information id to GM. The latter certifies $(\text{id} \parallel \text{upk})$ in the form of a signature cert_{id} . When signing, the signer first generates a one-time signature key-pair (otk, ovk) , uses its secret key sec_{id} to certify ovk as signature s . Then it evaluates $j = F(M, w, \text{id})$ and encrypts $G_j(\text{id})$ under opk

with randomness r , obtaining ciphertext \mathbf{c} . The signer then generates a NIZK argument π for the relation \mathcal{R} defined as follows

$$\mathcal{R} := \left\{ (\text{mpk}, \text{opk}, \mathbf{c}, M, F, \text{ovk}), (\text{id}, \text{upk}, \text{cert}_{\text{id}}, s, w, j, r) : \right. \\ \left. (\text{S.Ver}(\text{upk}, \text{ovk}, s) = 1) \wedge (\text{S.Ver}(\text{mpk}, (\text{id} \parallel \text{upk}), \text{cert}_{\text{id}}) = 1) \wedge \right. \\ \left. (F(M, w, \text{id}) = j) \wedge (j \in [1, K]) \wedge (\mathbf{c} = \text{E.Enc}(\text{opk}, G_j(\text{id}); r)) \right\}.$$

Next, the signer uses otk to one-time sign (M, F, \mathbf{c}, π) as sig , and outputs the final signature as $\Sigma = (\text{ovk}, \mathbf{c}, \pi, \text{sig})$. Verification of Σ basically consists of verifying sig and π . Meanwhile, opening of Σ is done via decrypting \mathbf{c} with key osk .

Roughly speaking, the correctness of the obtained MPS scheme is based on the correctness/completeness of the underlying building blocks. Privacy is achieved as long as \mathcal{E} is IND-CCA2 secure and \mathcal{NIZK} has the ZK property. Meanwhile, unforgeability is based on the soundness of \mathcal{NIZK} , the unforgeability of \mathcal{S} and the strong unforgeability of \mathcal{OTS} .

3.2 Description

Let $\lambda \in \mathbb{N}$ be a security parameter. Our generic construction of an MPS system associated with $(N, K, \mathcal{M}, \mathcal{W}, \mathcal{ID}, \mathcal{OP}, \mathcal{F}, \mathcal{G})$ works as follows.

Setup(λ) \rightarrow ($\text{pp}, \text{msk}, \text{osk}, \text{reg}$). On input security parameter λ , this probabilistic algorithm performs the following steps:

1. Run $\text{S.Kg}(\lambda)$ to obtain a signing-verification key-pair (msk, mpk) .
2. Run $\text{E.Kg}(\lambda)$ to obtain a decryption-encryption key-pair (osk, opk) .
3. Run $\text{ZK.Setup}(\lambda)$ to obtain a common reference string crs (and a simulation trapdoor τ_{sim} - which is discarded) for the NIZK system.

Then, it sets $\text{pp} := (\text{crs}, \text{mpk}, \text{opk})$, GM's secret key as msk and OA's secret key as osk , and initializes $\text{reg} := \emptyset$.

Join(pp); **Issue**($\text{pp}, \text{msk}, \text{reg}$). A user with personal identifiable information id , who would like to join the group, interacts with the GM as follows.

1. User runs $\text{S.Kg}(\lambda)$ to obtain a signing-verification key-pair (usk, upk) . Then it generates $\text{sig}_{\text{id}} \leftarrow \text{S.Sign}(\text{usk}, (\text{id} \parallel \text{upk}))$, and sends $(\text{id}, \text{upk}, \text{sig}_{\text{id}})$ to GM.
2. GM verifies that $\text{S.Ver}(\text{upk}, (\text{id} \parallel \text{upk}), \text{sig}_{\text{id}}) = 1$, and checks that id has not been registered in table reg . If any of these conditions does not hold, GM aborts. Otherwise, GM issues a signature $\sigma_{\text{id}} \leftarrow \text{S.Sign}(\text{msk}, (\text{id} \parallel \text{upk}))$, sends σ_{id} to the user, sets $\text{trans}_{\text{id}} := (\text{id}, \text{upk}, \text{sig}_{\text{id}}, \sigma_{\text{id}})$ and updates the registration table $\text{reg} := \text{reg} \cup \text{trans}_{\text{id}}$.
3. The user verifies that $\text{S.Ver}(\text{mpk}, \text{id} \parallel \text{upk}, \sigma_{\text{id}}) = 1$, and aborts if it is not the case. Otherwise, user sets $\text{sk}_{\text{id}} = (\text{id}, \text{sec}_{\text{id}}, \text{cert}_{\text{id}})$, where $\text{sec}_{\text{id}} = \text{usk}$ and $\text{cert}_{\text{id}} = (\sigma_{\text{id}}, \text{upk})$.

Sign($\text{pp}, \text{sk}_{\text{id}}, M, w, F$) $\rightarrow \Sigma / \perp$. Let $\text{sk}_{\text{id}} = (\text{id}, \text{sec}_{\text{id}}, \text{cert}_{\text{id}})$, where $\text{sec}_{\text{id}} = \text{usk}$ and $\text{cert}_{\text{id}} = (\sigma_{\text{id}}, \text{upk})$. The signing algorithm then proceeds as follows.

1. Compute $j = F(M, w, \text{id}) \in [0, K]$. Return \perp if $j = 0$.
2. Generate a one-time signature key-pair $(otk, ovk) \leftarrow \text{O.Kg}(\lambda)$.
3. Use usk to certify ovk as signature $s \leftarrow \text{S.Sign}(\text{usk}, ovk)$.
4. Encrypt $G_j(\text{id})$ under public key opk as $\mathbf{c} = \text{E.Enc}(\text{opk}, G_j(\text{id}); r)$, where r is the encryption randomness.
5. Generate an NIZK proof

$$\pi \leftarrow \text{ZK.Prove}\left(\text{crs}, ((\text{mpk}, \text{opk}, \mathbf{c}, M, F, ovk), (\text{id}, \text{upk}, \text{cert}_{\text{id}}, s, w, j, r))\right)$$

to prove that $((\text{mpk}, \text{opk}, \mathbf{c}, M, F, ovk), (\text{id}, \text{cert}_{\text{id}}, s, w, j, r)) \in \mathcal{R}$, where \mathcal{R} is the NP-relation defined above.

6. Use otk to issue a one-time signature $sig \leftarrow \text{O.Sign}(otk, (M, F, \mathbf{c}, \pi))$.
7. Return the signature $\Sigma := (ovk, \mathbf{c}, \pi, sig)$.

Verify(pp, M, F, Σ) $\rightarrow 0/1$. Given a purported signature $\Sigma = (ovk, \mathbf{c}, \pi, sig)$ on message M and with respect to signing function F , the verification algorithm proceeds as follows.

1. If $\text{O.Ver}(ovk, (M, F, \mathbf{c}, \pi), sig) = 0$, then return 0.
2. If $\text{ZK.Ver}(\text{crs}, (\text{mpk}, \text{opk}, \mathbf{c}, M, F, ovk), \pi) = 0$, then return 0.
3. Return 1.

Open($\text{pp}, \text{osk}, \Sigma, M, F$). Given $\Sigma = (ovk, \mathbf{c}, \pi, sig)$, the opening algorithm proceeds as follows.

1. Use osk to decrypt \mathbf{c} and obtain $z \leftarrow \text{E.Dec}(\text{osk}, \mathbf{c}) \in \mathcal{OP} \cup \{\perp\}$.
2. Return \perp if $z = \perp$. Otherwise, return $\text{op} = z \in \mathcal{OP}$.

AUXILIARY ALGORITHMS. Let us describe the auxiliary algorithms **SimSetup** and **Extract** associated with the above MPS system, which are required by the security model and are helpful for the security analyses.

SimSetup(λ). This algorithm is almost the same as the real setup algorithm presented above. The only difference is that, at Step 3, instead of generating $\text{crs} \leftarrow \text{ZK.Setup}(\lambda)$, one runs $\text{ZK.ExtSetup}(\lambda)$ to obtain a common reference string crs together with an extraction trapdoor τ_{ext} . The simulated public parameters are then set as $\text{pp} = (\text{crs}, \text{mpk}, \text{opk})$.

Extract($\tau_{\text{ext}}, (\text{pp}, \Sigma, M, F)$). Given the extraction trapdoor τ_{ext} , a *valid* signature $\Sigma = (ovk, \mathbf{c}, \pi, sig)$ on message M and with respect to signing function F , this algorithm runs $\text{ZK.Extr}(\text{crs}, \tau_{\text{ext}}, \pi)$ to obtain a witness $(\text{id}', \text{cert}'_{\text{id}}, s', w', j', r')$ for the relation \mathcal{R} . It then outputs $\zeta = (\text{id}', w')$.

3.3 Analyses

Theorem 1 states that the correctness and security properties of the presented MPS system can be based on the completeness/correctness and security features of the employed cryptographic building blocks.

Theorem 1. *Assume that \mathcal{S} is an unforgeable signature scheme under adaptive chosen-message attacks, \mathcal{OTS} is a strongly unforgeable one-time signature scheme, \mathcal{E} is an IND-CCA2-secure public-key encryption scheme and \mathcal{NIZK} is a dual-model non-interactive zero-knowledge argument system for relation \mathcal{R} . Then, the described MPS system satisfies correctness, privacy and unforgeability.*

We prove Theorem 1 via Lemma 1–6. The proofs of Lemma 2–6 are provided in the full version.

Lemma 1 (Correctness). *If \mathcal{S} , \mathcal{OTS} and \mathcal{E} are correct, and \mathcal{NIZK} is complete, then the presented MPS scheme satisfies correctness.*

Proof. The proof is straightforward. It follows from the correctness of \mathcal{S} that

$$\text{S.Ver}(\text{upk}, \text{id}, \text{S.Sign}(\text{usk}, \text{id})) = 1, \quad \text{S.Ver}(\text{mpk}, \text{id}, \text{S.Sign}(\text{msk}, \text{id})) = 1.$$

Hence, an honest signer should be able to enrol in the group and obtain a legitimate signing key $\text{sk}_{\text{id}} = (\text{id}, \text{sec}_{\text{id}}, \text{cert}_{\text{id}})$.

Next, thanks to the correctness of \mathcal{S} as well as the completeness of \mathcal{NIZK} , the signer should be able to obtain a valid witness $(\text{id}, \text{cert}_{\text{id}}, s, w, j, r)$ for the relation \mathcal{R} , and proof π should be accepted by ZK.Ver . Furthermore, one-time signature sig should pass the verification algorithm O.Ver . Therefore, as long as $j = F(M, w, \text{id}) \neq 0$, one should have $\text{Verify}(\text{pp}, M, F, \Sigma) = 1$.

Finally, the correctness of \mathcal{E} guarantees that $\text{E.Dec}(\text{osk}, \text{E.Enc}(\text{opk}, G_j(\text{id}), r))$ returns $G_j(\text{id})$, and so does $\text{Open}(\text{pp}, \text{osk}, \Sigma, M, F)$. \square

Lemma 2 (Type-1 Privacy). *The described MPS system satisfies computational privacy against Type-1 adversary if (i) \mathcal{E} has IND-CCA2 security; (ii) \mathcal{NIZK} has (computational/statistical) zero-knowledge property.*

Lemma 3 (Type-2 Privacy). *The described MPS system satisfies statistical (resp. computational) privacy against Type-2 adversary if \mathcal{NIZK} has statistical (resp. computational) zero-knowledge property.*

Lemma 4 (Extractability). *The described MPS scheme is extractable if \mathcal{NIZK} has CRS indistinguishability and extractability in the binding mode, and if \mathcal{E} is correct.*

Lemma 5 (Type-1 Unforgeability). *The described MPS system satisfies unforgeability against Type-1 forger if (i) the conditions of Lemma 4 hold; (ii) \mathcal{S} is unforgeable under chosen-message attacks; (iii) \mathcal{OTS} is a strongly unforgeable one-time signature; (iv) \mathcal{NIZK} is computationally sound.*

Lemma 6 (Type-2 Unforgeability). *The described MPS system satisfies unforgeability against Type-2 forger if (i) the conditions of Lemma 4 hold; (ii) \mathcal{S} is unforgeable under chosen-message attacks; (iii) \mathcal{OTS} is a strongly unforgeable one-time signature; (iv) \mathcal{NIZK} is computationally sound.*

4 A Construction from Pairings

4.1 Notations and Parameters

Let $(\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T)$ denote a non-degenerate bilinear map over pairing groups \mathbb{G}_1 and \mathbb{G}_T of prime order p and $\mathbb{G}_1^* := \mathbb{G}_1 \setminus \{1\}$. Let g, h be random generators of \mathbb{G}_1 . Our construction assumes the following parameter spaces:

- an identity space $\mathcal{ID} = (\mathbb{G}_1^*)^2$ where each user identity is encoded as $\text{id} = (\text{id}_1, \text{id}_2) \in (\mathbb{G}_1^*)^2$;
- a user public key space $\mathcal{UPK} = \mathbb{G}_1^*$;
- a message space $\mathcal{M} = \mathbb{G}_1$ for the signers where each $M \in \mathcal{M}$ is a Pedersen commitment for an integer value $w_1 \in \mathbb{Z}_p$ with randomness $w_2 \in \mathbb{Z}_p$;
- a witness space $\mathcal{W} = \mathbb{Z}_p^2$ where a witness $w = (w_1, w_2)$ for $M \in \mathcal{M}$ consists of the opening for M ;
- a function index space $\mathcal{J} = [1, 4]$;
- the valid ranges of w_1 , denoted by $[A_{i-1}, A_i)$ for $(1 \leq i \leq 4)$;
- a signing function F defined as

$$F(M, w = (w_1, w_2)) := \begin{cases} 1 & \text{iff } (M = g^{w_1} h^{w_2} \wedge A_0 \leq w_1 < A_1) \\ 2 & \text{iff } (M = g^{w_1} h^{w_2} \wedge A_1 \leq w_1 < A_2) \\ 3 & \text{iff } (M = g^{w_1} h^{w_2} \wedge A_2 \leq w_1 < A_3) \\ 4 & \text{iff } (M = g^{w_1} h^{w_2} \wedge A_3 \leq w_1 < A_4) \\ 0 & \text{otherwise} \end{cases}$$

- an opening space $\mathcal{OP} = \mathbb{G}_1^2$;
- a family of disclosing functions $\mathcal{G} = \{G_j : (\mathbb{G}_1^*)^2 \rightarrow \mathbb{G}_1^2\}$ ($j \in [1, 4]$) such that for an identity $\text{id} = (\text{id}_1, \text{id}_2) \in (\mathbb{G}_1^*)^2$

$$G_1(\text{id}) = (1_{\mathbb{G}_1}, 1_{\mathbb{G}_1}), G_2(\text{id}) = (1_{\mathbb{G}_1}, \text{id}_2), G_3(\text{id}) = (\text{id}_1, 1_{\mathbb{G}_1}), G_4(\text{id}) = (\text{id}_1, \text{id}_2).$$

4.2 Technical Overview

In our pairing-based MPS, a message $M \in \mathbb{G}_1$ is in the form of a Pedersen commitment [48], i.e., $M = g^v h^r$ where v represents a value (e.g., a transaction amount) and r is the randomness. The construction follows the same paradigm as the generic construction, but we change/adapt some of the building blocks by following the design of an efficient group signature scheme by Groth [28], which makes the construction more efficient. Specifically, we apply the following tools in our construction:

- The structure-preserving digital signature scheme by Kiltz et al. [35] $\mathcal{SPS} = (\mathcal{SPS.Kg}, \mathcal{SPS.Sign}, \mathcal{SPS.Ver})$ with message space $\mathcal{ID} \times \mathcal{UPK} = (\mathbb{G}_1^*)^3$ and signature space \mathbb{G}_1^{10} .

- The (weak) Boneh-Boyen (BB) digital signature scheme [6] $\mathcal{BBS} = (\text{BBS.Kg}, \text{BBS.Sign}, \text{BBS.Ver})$ with public key space $\mathcal{UPK} = \mathbb{G}_1^*$ and signature space \mathbb{G}_1^* .
- The Pedersen commitment scheme [48] $\mathcal{CM} = (\text{CM.Setup}, \text{CM.Cmt}, \text{CM.Open}, \text{CM.Ver})$ with witness space $\mathcal{W} = \mathbb{Z}_p^2$ and commitment space \mathbb{G}_1 .
- The tag-based PKE by Kiltz [34] $\mathcal{E} = (\text{E.Kg}, \text{E.Enc}, \text{E.Ver}, \text{E.Dec})$ with message space \mathbb{G}_1 and ciphertext space \mathbb{G}_1^5 . Note that E.Ver allows public verification of a ciphertext w.r.t. an encryption tag.
- The DLIN-based instantiation of the Groth-Sahai proof system [30] $\mathcal{GS} = (\text{GS.Setup}, \text{GS.Prove}, \text{GS.Ver}, \text{GS.SimSetup}, \text{GS.SimProve}, \text{GS.Extract})$ which includes two DLIN-based commitment schemes $\mathcal{GSCM}_i = (\text{GSCM}_i.\text{Cmt}, \text{GSCM}_i.\text{Open}, \text{GSCM}_i.\text{Ver})$ $i \in \{1, 2\}$ for committing elements in \mathbb{G}_1 and \mathbb{Z}_p , respectively. Both commitments use $\text{crs}_{\text{GS}} \leftarrow \text{GS.Setup}(\lambda)$ as the commitment key.
- A strongly unforgeable one-time digital signature scheme $\mathcal{OTS} = (\text{O.Kg}, \text{O.Sign}, \text{O.Ver})$.

In the center of our construction is the Groth-Sahai Proof system [30] that enables efficient non-interactive proofs for statements expressed in the forms of pairing product equations, multi-exponentiation equations and quadratic equations. To be compatible with Groth-Sahai proof, we adopt a Structure Preserving Signature [35] for the issuing of cert_{id} w.r.t. (id, upk) for a signer. To sign a message, the signer randomly generates a one-time key pair (ovk, otk) , certifies ovk using usk , and employs the Groth-Sahai Proof system to prove that there is a valid certification chain $\text{mpk} \rightarrow \text{upk} \rightarrow \text{ovk}$, without revealing id, upk or cert_{id} . The one-time key is used to generate the final signature. However, since the Groth-Sahai Proof system does not have NIZK for general pairing product equations, we replace the NIZK proof by NIWI proof, as in [28].

Proving $G_j(\text{id})$. The main difference between our construction and [28] is in dealing with the disclosed identity information $G_j(\text{id})$. In [28], the disclosing function is the identity function, i.e., $G(\text{id}) = \text{id}$, so the opening authority OA's secret key is the same as the extraction key for the Groth-Sahai proof system in the binding mode. In MPS, we need to separate OA's secret key from the extraction key as even the OA should not learn more than $G_j(\text{id})$. Moreover, we need to ensure extractability, meaning that the correct identity information $G_j(\text{id})$ is encrypted by the signer and $j = F(M, w) \in [1, 4]$ is correctly computed based on the witness w of the message M .

The Groth-Sahai Proof system allows perfect extraction of committed group elements in \mathbb{G}_1 , but not arbitrary elements in \mathbb{Z}_p . To achieve extractability in MPS, we need to ensure the correct extraction of not only $G_j(\text{id})$ but also $j = F(M, w)$ where $w \in \mathbb{Z}_p$. Meanwhile, we observe that if a committed value in \mathbb{Z}_p is a binary value, then we can also perfectly extract it from the commitment. This motivates us to convert the witness w into binary form, and prove in zero-knowledge that $j = F(M, w)$.

To do so, we first observe that the Pedersen commitment is homomorphic. Hence, to prove that the witness v of a message $M = g^v h^r$ falls in a range, say

$[A_i, A_{i+1})$ where $A_{i+1} - A_i = 2^k$, we only need to prove that M/g^{A_i} is a correct commitment for $v - A_i \in [0, 2^k)$. To do so, we follow the standard range proof approach by converting $v - A_i$ into a k -bit binary string and proving that each position has a binary value. We then employ additional bits (2 bits in our concrete instantiation in this paper) to specify which range, among all the possible ranges, the value v actually falls in. This is achieved by expressing the proof statement in the form of an OR-proof, where the additional bits point to the real statement being proved.

As a result, we convert the NIZK proof for $G_j(\text{id})$ into a collection of multi-exponentiation equations in \mathbb{G}_1 and quadratic equations (for binary values) in \mathbb{Z}_p , which can be proved using the Groth-Sahai Proof system while achieving extractability.

It is worth noting that the above approach allows us to support a larger identity space, e.g., $\text{id} \in (\mathbb{G}_1^*)^n$, accompanied with a variety of disclosing functions.

Due to the space limit, the details are presented in the full version.

5 A Construction from Lattices

In this section, we present a concrete construction of MPS which is proven secure under lattice-based assumptions in the random oracle model (ROM).

Let integers $n, m, q, k = 3n \lceil \log q \rceil, L$ and $0 < A_1 < A_2 < A_3 < 2^L - 1$ be the system parameters. Let $(\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{n \times L} \times \mathbb{Z}_q^{n \times m}$ be a commitment key for the KTX commitment scheme [31], which is statistically hiding and computationally binding under the SIS assumption.

Similar to the pairing-based construction presented in Section 4, this lattice-based scheme is also associated with $N = 1$ signing function F and $K = 4$ disclosing functions G_1, \dots, G_4 , and also consider the setting where $\mathbf{m} = \text{com}(\mathbf{w}_1, \mathbf{w}_2)$, with \mathbf{m} is a message to be signed, and $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$ is a witness.

Let $\mathcal{M} = \mathbb{Z}_q^n$, $\mathcal{W} = \{0, 1\}^L \times \{0, 1\}^m$, $\mathcal{ID} = \mathcal{OP} = \{0, 1\}^k$. Define the signing function $F : \mathcal{M} \times \mathcal{W} \rightarrow [0, 4]$ as follows.

$$F(\mathbf{m}, \mathbf{w}) := \begin{cases} 0 & \text{if } \mathbf{m} \neq \mathbf{C}_1 \cdot \mathbf{w}_1 + \mathbf{C}_2 \cdot \mathbf{w}_2 \pmod{q}, \text{ else} \\ 1 & \text{if } (0 \leq W_1 < A_1), \text{ else} \\ 2 & \text{if } (A_1 \leq W_1 < A_2), \text{ else} \\ 3 & \text{if } (A_2 \leq W_1 < A_3), \text{ else} \\ 4 & \text{if } (A_3 \leq W_1 \leq 2^L - 1), \end{cases}$$

where $W_1 = \text{int}(\mathbf{w}_1)$ - the integer in $[0, 2^L - 1]$ whose binary representation is \mathbf{w}_1 .

Disclosing functions. For each $j \in [1, 4]$ define G_j as a linear endomorphism over \mathbb{Z}_2^k . Specifically, let $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \mathbf{G}_4 \in \mathbb{Z}_2^{k \times k}$ be public matrices, then let

$$G_j(\text{id}) = \mathbf{G}_j \cdot \text{id}.$$

The definition is quite general and expressive, in the sense that it captures many natural ways to disclose partial information about id . For instance, we can set $\mathbf{G}_1 = \mathbf{0}^{k \times k}$ and $\mathbf{G}_4 = \mathbf{I}_k$, so that $G_1(\text{id}) = \mathbf{0}$ (i.e., non-traceable case) and $G_4(\text{id}) = \text{id}$ (i.e., fully traceable case). We can also easily define $\mathbf{G}_2, \mathbf{G}_3$ so that $G_2(\text{id}), G_3(\text{id})$ each reveals a specific subset of coordinates of id .

5.1 Technical Overview

While it is feasible to instantiate MPS in the standard model via the lattice-based NIZK techniques of Peikert and Shiehian [49], such a construction would expectedly be extremely inefficient. Here, our goal is to build more efficient constructions in the ROM, where we can employ concrete techniques for obtaining interactive ZK arguments for lattice-based relations, and then remove interaction via the Fiat-Shamir transformation [22].

Regarding lattice-based building blocks, apart from the KTX SIS-based commitment scheme [31] which we mentioned above, we employ the following ingredients:

- The SIS-based signature scheme from [37], which admits efficient zero-knowledge arguments of knowledge of a valid message-signature pair. This signature scheme will be used by the GM to issue users' certificates.
- An LWE-based CCA2-secure PKE scheme obtained from the GPV IBE [26] and the CHK transformation [14]. This encryption scheme will be used to encrypt $G_j(\text{id})$, and ciphertexts will be decryptable by the OA.
- A SIS-based one-way function [1]. In the ROM, since the NIZK argument π included in Σ can be viewed as a signature of knowledge [15] of the signer's membership secret sec_{id} , we can slightly depart from the generic construction of Section 3, by equipping users with a one-way function rather than an ordinary signature scheme.
- We also need a statistical ZK argument system that can handle relatively sophisticated linear and quadratic relations with respect to two moduli ($q_1 = 2$ and $q_2 > 2$) and that is compatible with the signature scheme from [37]. To this end, we choose to employ the Stern-like [53] framework from [40].

Proving in ZK that $\mathbf{y} = G_j(\text{id})$. The major technical difficulty that we have to overcome is to prove in ZK that a plaintext \mathbf{y} , encrypted under the GPV IBE scheme, is exactly the value $G_j(\text{id})$. To this end, we first would need to show that the index $j = F(\mathbf{m}, \mathbf{w}) \in [1, 4]$ is computed correctly. Our techniques are as follows.

We first “extract” the position of $W_1 = \text{int}(\mathbf{w}_1) \in [0, 2^L - 1]$ relative to A_1, A_2, A_3 by defining bits $b_1, b_2, b_3 \in \{0, 1\}$ such that

$$\begin{aligned} 0 \leq W_1 < A_1 &\iff (b_1, b_2, b_3) = (0, 0, 0) \iff (1 - b_1)(1 - b_2)(1 - b_3) = 1; \\ A_1 \leq W_1 < A_2 &\iff (b_1, b_2, b_3) = (1, 0, 0) \iff b_1(1 - b_2)(1 - b_3) = 1; \\ A_2 \leq W_1 < A_3 &\iff (b_1, b_2, b_3) = (1, 1, 0) \iff b_1 b_2(1 - b_3) = 1; \\ A_3 \leq W_1 < 2^L - 1 &\iff (b_1, b_2, b_3) = (1, 1, 1) \iff b_1 b_2 b_3 = 1. \end{aligned}$$

This can be realized by viewing inequalities under the lens of integer additions, in the following way. Suppose that there exist (non-negative) L -bit integers $Y_0, Y_1, Z_0, Z_1, T_0, T_1$ and bits b_1, b_2, b_3 such that:

$$\begin{aligned} W_1 + (1 - b_1) \cdot Y_1 + (1 - b_1) &= A_1 + b_1 \cdot Y_0, \\ W_1 + (1 - b_2) \cdot Z_1 + (1 - b_2) &= A_2 + b_2 \cdot Z_0, \\ W_1 + (1 - b_3) \cdot T_1 + (1 - b_3) &= A_3 + b_3 \cdot Y_0. \end{aligned} \quad (1)$$

Observe that, when $b_1 = 0$, we have $W_1 + Y_1 + 1 = A_1$, implying that $W_1 < A_1$ since $Y_1 \geq 0$. On the other hand, if $b_1 = 1$, we have $W_1 = A_1 + Y_0$, and as $Y_0 \geq 0$, we can deduce that $W_1 \geq A_1$. In other words, b_1 captures the predicate $(W_1 \geq A_1)$. Similarly, we have $b_2 = (W_1 \geq A_2)$ and $b_3 = (W_1 \geq A_3)$.

Next, let us consider bits $f_0, f_1 \in \{0, 1\}$ such that

$$\begin{aligned} \mathbf{y} = G_j(\text{id}) = \mathbf{G}_j \cdot \text{id} &= (1 - f_0)(1 - f_1) \cdot \mathbf{G}_1 \cdot \text{id} + (1 - f_0)f_1 \cdot \mathbf{G}_2 \cdot \text{id} \\ &+ f_0 \cdot (1 - f_1) \cdot \mathbf{G}_3 \cdot \text{id} + f_0 \cdot f_1 \cdot \mathbf{G}_4 \cdot \text{id} \pmod{2}. \end{aligned} \quad (2)$$

In other words, f_0, f_1 are such that $j = 1, 2, 3, 4$ if and only if $(f_0, f_1) = (0, 0), (0, 1), (1, 0), (1, 1)$, respectively.

Now, observe that f_0, f_1 and b_1, b_2, b_3 are connected via the following equation:

$$\begin{aligned} (f_0, f_1) &= (1 - b_1)(1 - b_2)(1 - b_3) \cdot (0, 0) + b_1(1 - b_2)(1 - b_3) \cdot (0, 1) \\ &+ b_1b_2(1 - b_3) \cdot (1, 0) + b_1b_2b_3 \cdot (1, 1) \pmod{2}. \end{aligned} \quad (3)$$

As a summary of the above ideas, we have reduced the problem of proving that $\mathbf{y} = G_j(\text{id})$ to the equivalent problem of proving knowledge of bits b_1, b_2, b_3, f_0, f_1 and L -bit integers $Y_0, Y_1, Z_0, Z_1, T_0, T_1$ satisfying equations in (1), (2) and (3).

We note that equations in (1) can be proved in zero-knowledge using the techniques from [41], which, in a nutshell, translate integer additions into binary adders with carries, and hence obtain a system of equations modulo 2. Combining with equations in (2) and (3), we can obtain an equation of the form

$$\mathbf{M}_2 \cdot \mathbf{p}_2 = \mathbf{u}_2 \pmod{2}, \quad (4)$$

where matrix \mathbf{M}_2 and vector \mathbf{u}_2 are public, and \mathbf{p}_2 is a binary vector that encodes all the information of vector id , bits b_1, b_2, b_3, f_0, f_1 and integers $Y_0, Y_1, Z_0, Z_1, T_0, T_1$.

The main ZK argument system. Our construction will make use of a ZK argument system that allows to prove knowledge of a tuple $(\text{id}, \mathbf{z}, \mathbf{x}, \text{cert}_{\text{id}} = (\tau, \mathbf{v}, \mathbf{s}), \mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2), \mathbf{y}, (\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2))$ satisfying the following conditions:

- (i) (\mathbf{z}, \mathbf{x}) is a preimage-image pair of a SIS-based one-way function;
- (ii) $\text{cert}_{\text{id}} = (\tau, \mathbf{v}, \mathbf{s})$ is a signature on message $(\text{id} \parallel \mathbf{x})$, with respect to the signature scheme from [37];
- (iii) A GPV IBE ciphertext is a correct encryption of plaintext \mathbf{y} , with randomness $(\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2)$;
- (iv) \mathbf{m} is a KTX commitment of \mathbf{w}_1 with randomness \mathbf{w}_2 ;

(v) $\mathbf{y} = \mathbf{G}_j(\text{id})$, where $j = F(\mathbf{m}, \mathbf{w}) \in [1, 4]$.

Recall that item (v) can be handled using the ideas we discussed above. As for (i), (ii), (iii), (iv) we can use the techniques from [43,37] to obtain a vector \mathbf{p}_1 that has coordinates in $\{-1, 0, 1\}$ and that encodes the information of $(\mathbf{z}, \mathbf{x}, \tau, \mathbf{v}, \mathbf{s}, \text{id}, \mathbf{r}, \mathbf{e}_1, \mathbf{e}_2, \mathbf{w}_1, \mathbf{w}_2)$, and that satisfies an equation of the form

$$\mathbf{M}_1 \cdot \mathbf{p}_1 = \mathbf{u}_1 \pmod{q}, \quad (5)$$

where matrix \mathbf{M}_1 and vector \mathbf{u}_1 are public.

Now, our task is to prove that equations (4) and (5) hold for the constructed vectors \mathbf{p}_1 and \mathbf{p}_2 , both of which contain encoded information of id . To this end, we can employ dedicated Stern-like permuting techniques [39,38] to reduce the underlying relation to an instance of the abstract relation considered in [40]. (An adaptation of [40], where there are two moduli $q_1 = q$ and $q_2 = 2$, is presented in detail in the full version.) As a result, we can obtain a statistical ZK argument of knowledge for the considered relation.

5.2 Description of the Scheme

The scheme can be seen as an extension of the dynamic GS from [37]. The scheme works with lattice parameter $n \in \mathcal{O}(\lambda)$, parameter $\ell = \mathcal{O}(\log n)$, prime modulus $q = \tilde{\mathcal{O}}(n^4)$, dimensions $m = 2n \lceil \log q \rceil$, $k = 3n \lceil \log q \rceil$, $L = \mathcal{O}(n)$, Gaussian parameter $\sigma = \Omega(\sqrt{n \log q} \log n)$ and infinity norm bounds $\beta = \sigma \omega(\log m)$. Let $\text{bin}(\cdot)$ be a function mapping vectors over \mathbb{Z}_q to their binary representations.

The main ZK protocol of the scheme is for the relation $\mathcal{R}_{\text{Imps}}$, defined below.

Definition 5. *Define*

$$\mathcal{R}_{\text{Imps}} = \left\{ \left((\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{u}, \mathbf{F}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{m}, \{\mathbf{G}_j\}_{j=1}^4, \mathbf{B}, \mathbf{G}, \mathbf{c}_1, \mathbf{c}_2), \right. \right. \\ \left. \left. (\text{id}, \mathbf{z}, \mathbf{x}, \text{cert}_{\text{id}} = (\tau, \mathbf{v}, \mathbf{s}), \mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2), \mathbf{y}, (\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2)) \right) \right\}$$

as a relation, where

- $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_\ell, \mathbf{D}, \mathbf{B}, \mathbf{F}, \mathbf{C}_2 \in \mathbb{Z}_q^{n \times m}$, $\mathbf{D}_0, \mathbf{D}_1 \in \mathbb{Z}_q^{2n \times 2m}$, $\mathbf{C}_1 \in \mathbb{Z}_q^{n \times L}$, $\mathbf{G} \in \mathbb{Z}_q^{n \times k}$, $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{c}_1 \in \mathbb{Z}_q^m$, $\mathbf{c}_2 \in \mathbb{Z}_q^k$, $\mathbf{G}_1, \dots, \mathbf{G}_4 \in \mathbb{Z}_2^{k \times k}$.
- $\text{id}, \mathbf{y} \in \{0, 1\}^k$, $\mathbf{z} \in \{0, 1\}^m$, $\mathbf{x} \in \{0, 1\}^{n \lceil \log q \rceil}$, $\tau \in \{0, 1\}^\ell$, $\mathbf{v}, \mathbf{s} \in [-\beta, \beta]^{2m}$, $\mathbf{w}_1 \in \{0, 1\}^L$, $\mathbf{w}_2 \in \{0, 1\}^m$, $\mathbf{r} \in [-B, B]^n$, $\mathbf{e}_1 \in [-B, B]^m$, $\mathbf{e}_2 \in [-B, B]^k$.
- $\mathbf{x} = \text{bin}(\mathbf{F} \cdot \mathbf{z} \pmod{q})$.
- $[\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau_j \cdot \mathbf{A}_j] \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \text{bin}(\mathbf{D}_0 \cdot \mathbf{s} + \mathbf{D}_1 \cdot (\text{id} \parallel \mathbf{x})) \pmod{q}$.
- $\mathbf{c}_1 = \mathbf{B}^\top \cdot \mathbf{r} + \mathbf{e}_1 \pmod{q}$, $\mathbf{c}_2 = \mathbf{G}^\top \cdot \mathbf{r} + \mathbf{e}_2 + \lfloor q/2 \rfloor \cdot \mathbf{y} \pmod{q}$.
- $\mathbf{m} = \mathbf{C}_1 \cdot \mathbf{w}_1 + \mathbf{C}_2 \cdot \mathbf{w}_2 \pmod{q}$.
- $\mathbf{y} = G_j(\text{id})$.

Using the techniques discussed above, we can obtain a statistical ZK argument for $\mathcal{R}_{\text{Imps}}$. The protocol, has soundness error $2/3$. It is repeated $\kappa = \mathcal{O}(\lambda)$ times in parallel to make the error negligibly small, and then made non-interactive via the Fiat-Shamir heuristic. Our lattice-based MPS scheme works as follows.

Setup(λ) \rightarrow (**pp**, **msk**, **osk**, **reg**). This algorithm performs the following steps.

1. Generate verification key

$$(\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_\ell, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{u}) \in (\mathbb{Z}_q^{n \times m})^{\ell+3} \times (\mathbb{Z}_q^{2n \times 2m})^2 \times \mathbb{Z}_q^n$$

and signing key $\mathbf{T}_\mathbf{A}$ for the signature scheme from [37].

2. Generate master public key $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ and master secret key $\mathbf{T}_\mathbf{B}$ for the GPV IBE scheme [26].
3. Choose uniformly random matrices $\mathbf{F} \in \mathbb{Z}_q^{n \times m}$, and $\mathbf{C}_1 \in \mathbb{Z}_q^{n \times \ell}$, $\mathbf{C}_2 \in \mathbb{Z}_q^{n \times m}$. Looking ahead, \mathbf{F} will define a SIS-based one-way function, while $(\mathbf{C}_1, \mathbf{C}_2)$ will serve as a KTX commitment key for L -bit messages.
4. Let χ be a B -bounded distribution.
5. Choose a one-time signature scheme $\mathcal{OTS} = (\text{O.Kg}, \text{O.Sign}, \text{O.Ver})$.
6. Choose hash functions $H_{FS} : \{0, 1\}^* \rightarrow \{1, 2, 3\}^\kappa$ and $H_{GPV} : \{0, 1\}^* \rightarrow \mathbb{Z}_q^{n \times k}$ that will be modeled as random oracles.

Output **msk** = $\mathbf{T}_\mathbf{A}$, **osk** = $\mathbf{T}_\mathbf{B}$, **reg** = \emptyset and

$$\text{pp} = (\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_\ell, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{u}, \mathbf{B}, \mathbf{F}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{F}, \chi, \mathcal{OTS}, H_{FS}, H_{GPV}).$$

Join(pp); **Issue**(pp, **msk** = $\mathbf{T}_\mathbf{A}$, **reg**). A prospective user with identity $\text{id} \in \mathbb{Z}_q^k$ interacts with the GM as follows.

1. User selects $\mathbf{z} \xleftarrow{\$} \{0, 1\}^m$, computes $\mathbf{x} = \text{bin}(\mathbf{F} \cdot \mathbf{z}) \in \{0, 1\}^{n \lceil \log q \rceil}$. User then signs $(\text{id} \parallel \mathbf{x}) \in \{0, 1\}^{2m}$ using an ordinary signature and sends $(\text{id} \parallel \mathbf{x})$ together with the signature sig_{id} to the GM.
2. GM verifies sig_{id} and then certifies $(\text{id} \parallel \mathbf{x})$ using $\mathbf{T}_\mathbf{A}$. The certificate has the form $\text{cert}_{\text{id}} = (\tau, \mathbf{v}, \mathbf{s}) \in \{0, 1\}^\ell \times [-\beta, \beta]^{2m} \times [-\beta, \beta]^{2m}$, satisfying

$$[\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau_j \cdot \mathbf{A}_j] \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \text{bin}(\mathbf{D}_0 \cdot \mathbf{s} + \mathbf{D}_1 \cdot (\text{id} \parallel \mathbf{x})) \pmod{q}. \quad (6)$$

3. User id verifies the validity of cert_{id} and outputs $\text{sk}_{\text{id}} = (\text{id}, \text{sec}_{\text{id}}, \text{cert}_{\text{id}})$, where $\text{sec}_{\text{id}} = \mathbf{z}$.
4. GM computes $\text{trans}_{\text{id}} = (\text{id}, \mathbf{x}, \text{sig}_{\text{id}}, \text{cert}_{\text{id}})$ and updates the registration table $\text{reg} := \text{reg} \cup \text{trans}_{\text{id}}$.

Sign(pp, sk_{id} , \mathbf{m} , $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$, F). Given pp, signing key $\text{sk}_{\text{id}} = (\text{id}, \text{sec}_{\text{id}}, \text{cert}_{\text{id}})$, message $\mathbf{m} \in \mathcal{M}$, witness $w \in \mathcal{W}$, and signing function F , this algorithm performs the following steps.

1. Check that $F(\mathbf{m}, \mathbf{w}, \text{id}) \neq 0$, i.e., $\mathbf{w}_1 \in \{0, 1\}^L$, $\mathbf{w}_2 \in \{0, 1\}^m$ and $\mathbf{m} = \mathbf{C}_1 \cdot \mathbf{w}_1 + \mathbf{C}_2 \cdot \mathbf{w}_2 \pmod{q}$. Return \perp if this is not the case.

2. Determine the value of $j = F(\mathbf{m}, \mathbf{w}, \text{id}) \in [1, 4]$. Let $\mathbf{y} = G_j(\text{id}) \in \{0, 1\}^k$.
3. Generate a key-pair $(otk, ovk) \leftarrow \text{O.Kg}(\lambda)$ and encrypt \mathbf{y} with respect to “identity” ovk as follows. Let $\mathbf{G} = H_{GPV}(ovk) \in \mathbb{Z}_q^{n \times k}$. Sample $\mathbf{r} \leftarrow \chi^n$, $\mathbf{e}_1 \leftarrow \chi^m$, $\mathbf{e}_2 \leftarrow \chi^k$, then compute the ciphertext

$$(\mathbf{c}_1 = \mathbf{B}^\top \cdot \mathbf{r} + \mathbf{e}_1, \quad \mathbf{c}_2 = \mathbf{G}^\top \cdot \mathbf{r} + \mathbf{e}_2 + \lfloor q/2 \rfloor \cdot \mathbf{y}) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^k.$$

4. Using witness $(\text{id}, \mathbf{z}, \mathbf{x}, \text{cert}_{\text{id}} = (\tau, \mathbf{v}, \mathbf{s}), \mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2), \mathbf{y}, (\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2))$, generate a NIZKAoK π for the relation $\mathcal{R}_{\text{Imps}}$ (Definition 5). This is done by repeating κ times an interactive ZK argument of knowledge for $\mathcal{R}_{\text{Imps}}$, and made non-interactive as a triple $\pi = (\{\text{CMT}_j\}_{j \in [\kappa]}, \text{CH}, \{\text{RSP}_j\}_{j \in [\kappa]})$, where $\text{CH} = H_{FS}(\mathbf{m}, ovk, \mathbf{c}_1, \mathbf{c}_2, \{\text{CMT}_j\}_{j \in [\kappa]}) \in \{1, 2, 3\}^\kappa$.
 5. Compute a one-time signature $sig \leftarrow \text{O.Sign}(otk, (\mathbf{m}, F, \mathbf{c}_1, \mathbf{c}_2, \pi))$.
- Output the signature

$$\Sigma = (ovk, \mathbf{c}_1, \mathbf{c}_2, \pi, sig). \quad (7)$$

Verify(pp, \mathbf{m}, F, Σ). This algorithm parses Σ as in (7), and returns 1 if:

1. $\text{O.Ver}(ovk, (\mathbf{m}, F, \mathbf{c}_1, \mathbf{c}_2, \pi), sig) = 1$;
2. π is a valid NIZKAoK for $\mathcal{R}_{\text{Imps}}$.

Open(pp, $osk = \mathbf{T}_B, \Sigma, \mathbf{m}, F$). This algorithm proceeds as follows.

1. Compute $\mathbf{G} = H_{GPV}(ovk) \in \mathbb{Z}_q^{n \times k}$, then using \mathbf{T}_B to sample a small-norm matrix \mathbf{E}_{ovk} such that $\mathbf{B} \cdot \mathbf{E}_{ovk} = \mathbf{G} \pmod q$.
2. Using \mathbf{E}_{ovk} to decrypt $(\mathbf{c}_1, \mathbf{c}_2)$ (by computing $\lfloor (\mathbf{c}_2 - \mathbf{E}_{ovk}^\top \cdot \mathbf{c}_1) / (q/2) \rfloor$), so that to obtain $\mathbf{y} \in \{0, 1\}^k$. Output \perp if the decryption fails.
3. Output $\text{op} = \mathbf{y}$.

5.3 Analyses of the Scheme

EFFICIENCY. Let us analyze the asymptotic efficiency of the proposed scheme. The size of pp is dominated by that of the verification key of the signature scheme from [37] and has bit-size $\mathcal{O}(\ell mn \log q) = \tilde{\mathcal{O}}(\lambda^2)$. A signing key sk_{id} consists of a few small-norm vectors and has bit-size $\mathcal{O}(m \log q \log \beta) = \tilde{\mathcal{O}}(\lambda)$. The size of each signature Σ is dominated by that of the NIZKAoK π , which is roughly $\kappa = \mathcal{O}(\lambda)$ times the bit-size of the underlying witness $(\text{id}, \mathbf{z}, \mathbf{x}, \text{cert}_{\text{id}} = (\tau, \mathbf{v}, \mathbf{s}), \mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2), \mathbf{y}, (\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2))$. Overall, Σ has bit-size $\tilde{\mathcal{O}}(\lambda^2)$.

CORRECTNESS. The correctness of the described MPS scheme follows directly from the correctness of the signature scheme from [37], the correctness of the GPV IBE scheme [26] and the perfect completeness of the employed Stern-like argument system [53,40].

SECURITY. The security of the scheme can be proven in the ROM, under the SIS and the LWE assumptions.

Theorem 2. *In the random oracle model, the described MPS system satisfies privacy and unforgeability if (i) the SIS and LWE assumptions hold; (ii) OTS is a strongly unforgeable one-time signature; (iii) The employed argument system is a statistically ZKAoK.*

Due to space restriction, the proof of Theorem 2 is deferred to the full version.

6 Open Questions

As the first work that proposes the brand-new concept of Multimodal Private Signatures, we do not expect to provide a thorough study of this primitive. We leave several fascinating open questions for future investigations:

1. Constructing practically usable MPS schemes with expressive signing and disclosing functions;
2. Studying theoretical connections between MPS and other advanced primitives like functional encryption and fully-homomorphic encryption;
3. Designing efficient MPS schemes with post-quantum security;
4. Equipping MPS with additional functionalities such as verifiable opening and/or user revocation.

ACKNOWLEDGEMENTS. We would like to thank Dung Duong for helpful discussions. We would also like to thank the anonymous reviewers of CRYPTO 2022 for valuable suggestions. F. Guo, W. Susilo and G. Yang are partially supported by the Australian Research Council Discovery Projects DP200100144 and DP220100003.

References

1. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC 1996*, pages 99–108. ACM, 1996.
2. N. Attrapadung, G. Hanaoka, and S. Yamada. Conversions among several classes of predicate encryption and applications to ABE with various compactness tradeoffs. In *ASIACRYPT 2015*. Springer, 2015.
3. M. Bellare and G. Fuchsbaauer. Policy-based signatures. In *PKC 2014*. Springer, 2014.
4. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT 2003*. Springer, 2003.
5. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA 2005*. Springer, 2005.
6. D. Boneh and X. Boyen. Short signatures without random oracles. In *EUROCRYPT 2004*. Springer, 2004.
7. D. Boneh, A. Sahai, and B. Waters. Functional encryption: a new vision for public-key cryptography. *Commun. ACM*, 55(11):56–64, 2012.
8. J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, J. Groth, and C. Petit. Short accountable ring signatures based on DDH. In *ESORICS 2015*. Springer, 2015.
9. J. Bootle, V. Lyubashevsky, and G. Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *CRYPTO 2019*. Springer, 2019.
10. E. Boyle, S. Goldwasser, and I. Ivan. Functional signatures and pseudorandom functions. In *PKC 2014*. Springer, 2014.
11. J. Camenisch, M. Drijvers, A. Lehmann, G. Neven, and P. Towa. Short threshold dynamic group signatures. In *SCN 2020*. Springer, 2020.

12. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *EUROCRYPT 2005*. Springer, 2005.
13. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Balancing accountability and privacy using e-cash. In *SCN 2006*. Springer, 2006.
14. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004*. Springer, 2004.
15. M. Chase and A. Lysyanskaya. On signatures of knowledge. In *CRYPTO 2006*. Springer, 2006.
16. D. Chaum. Security without identification: Transactions system to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.
17. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT 1991*. Springer, 1991.
18. J. Diaz and A. Lehmann. Group signatures with user-controlled and sequential linkability. In *PKC 2021*. Springer, 2021.
19. A. El Kaafarani, E. Ghadafi, and D. Khader. Decentralized traceable attribute-based signatures. In *CT-RSA 2014*. Springer, 2014.
20. M. F. Esgin, N. K. Nguyen, and G. Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *ASIACRYPT 2020*. Springer, 2020.
21. M. F. Esgin, R. Steinfeld, J. K. Liu, and D. Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In *CRYPTO 2019*. Springer, 2019.
22. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO 1986*. Springer, 1987.
23. E. Fujisaki and K. Suzuki. Traceable ring signature. In *PKC, 2007*. Springer, 2007.
24. L. Garms and A. Lehmann. Group signatures with selective linkability. In *PKC, 2019*. Springer, 2019.
25. C. Gentry, J. Groth, Y. Ishai, C. Peikert, A. Sahai, and A. D. Smith. Using fully homomorphic hybrid encryption to minimize non-interactive zero-knowledge proofs. *J. Cryptol.*, 28(4):820–843, 2015.
26. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC, 2008*. ACM, 2008.
27. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC 1985*, pages 291–304. ACM, 1985.
28. J. Groth. Fully anonymous group signatures without random oracles. In *ASIACRYPT 2007*. Springer, 2007.
29. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero-knowledge for NP. In *EUROCRYPT 2006*. Springer, 2006.
30. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT 2008*. Springer, 2008.
31. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT 2008*. Springer, 2008.
32. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *EUROCRYPT 2004*. Springer, 2004.
33. A. Kiayias and M. Yung. Secure scalable group signature with dynamic joins and separable authorities. *Int. J. Secur. Networks*, 1(1/2):24–45, 2006.
34. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC 2006*. Springer, 2006.
35. E. Kiltz, J. Pan, and H. Wee. Structure-preserving signatures from standard assumptions, revisited. In *CRYPTO 2015*. Springer, 2015.

36. M. Kohlweiss and I. Miers. Accountable metadata-hiding escrow: A group signature case study. In *PoPETs*, 2015.
37. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *ASIACRYPT 2016*. Springer, 2016.
38. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In *ASIACRYPT 2016*. Springer, 2016.
39. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT 2016*. Springer, 2016.
40. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based PRFs and applications to e-cash. In *ASIACRYPT 2017*. Springer, 2017.
41. B. Libert, S. Ling, K. Nguyen, and H. Wang. Lattice-based zero-knowledge arguments for integer relations. In *CRYPTO 2018*. Springer, 2018.
42. B. Libert, K. Nguyen, T. Peters, and M. Yung. Bifurcated signatures: Folding the accountability vs. anonymity dilemma into a single private signing scheme. In *EUROCRYPT 2021*. Springer, 2021.
43. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC 2013*. Springer, 2013.
44. H. K. Maji, M. Prabhakaran, and M. Rosulek. Attribute-based signatures. In *CT-RSA 2011*. Springer, 2011.
45. P. K. Masur. *Situational Privacy and Self-Disclosure: Communication Processes in Online Environments*. Springer, 2019.
46. M. Nandi and T. Pandit. Predicate signatures from pair encodings via dual system proof technique. *J. Math. Cryptol.*, 13(3-4):197–228, 2019.
47. S. Noether. Ring signature confidential transactions for monero. *IACR Cryptology ePrint Archive*, 2015:1098, 2015.
48. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO 1991*. Springer, 1991.
49. C. Peikert and S. Shiehian. Non-interactive zero knowledge for NP from (plain) Learning With Errors. In *CRYPTO 2019*. Springer, 2019.
50. R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*. Springer, 2001.
51. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*. Springer, 2001.
52. Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote. Group signatures with message-dependent opening. In *Pairing*, 2012. Springer, 2012.
53. J. Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.
54. B. van der Sloot and A. de Groot, editors. *The Handbook of Privacy Studies: An Interdisciplinary Introduction*. Amsterdam University Press, 2018.
55. S. Xu and M. Yung. Accountable ring signatures: A smart card approach. In *CARDIS*, 2004. Springer, 2004.
56. R. Yang, M.-H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO 2019*. Springer, 2019.