# A Lattice-Based Group Signature Scheme with Message-Dependent Opening

Benoît Libert, Fabrice Mouhartem, Khoa Nguyen

# A Lattice-Based Group Signature Scheme with Message-Dependent Opening

Benoît Libert[1], Fabrice Mouhartem[1], and Khoa Nguyen[2]

[1] Ecole Normale Supérieure de Lyon (France)
[2] Nanyang Technological University (Singapore)

**Abstract** Group signatures are an important anonymity primitive allowing users to sign messages while hiding in a crowd. At the same time, signers remain accountable since an authority is capable of de-anonymizing signatures via a process called *opening*. In many situations, this authority is granted too much power as it can identify the author of any signature. Sakai *et al.* proposed a flavor of the primitive, called *Group Signature with Message-Dependent Opening* (GS-MDO), where opening operations are only possible when a separate authority (called "admitter") has revealed a trapdoor for the corresponding message. So far, all existing GS-MDO constructions rely on bilinear maps, partially because the message-dependent opening functionality inherently implies identity-based encryption. This paper proposes the first GS-MDO candidate based on lattice assumptions. Our construction combines the group signature of Ling, Nguyen and Wang (PKC'15) with two layers of identity-based encryption. These components are tied together using suitable zero-knowledge argument systems.

**Keywords:** Group signatures, anonymity, lattice assumptions.

## 1 Introduction

Group signatures. Group signatures were introduced by Chaum and van Heyst in 1991 [16] as a technique allowing users to sign messages while retaining anonymity within a crowd of users they belong to. At the same, misbehaving group members cannot remain unpunished as an *authority*, called *opening authority*, is capable of tracing a signature to the user who generated it [5]. While such a tracing mechanism is necessary to ensure user accountability, it arguably grants excessive power to the opening authority which can retrieve the identity of any well-behaved user from his signature. To address this issue, Sakai *et al.* [42] suggested an extension, named *group signature with message dependent opening* (GS-MDO), which provides a refined balance between accountability and privacy. In GS-MDO systems, as formalized in [42], the identity of a signer can only be determined from two pieces of information: the opening authority's secret key and a message-specific token delivered by a separate authority called the *admitter*. Importantly, neither authority is able to trace any signature alone. Each opening operation has to be approved by the admitter and, at the same time, the latter

cannot identify signers by itself as it is denied access to the opening authority's secret key.

A different way to avoid centralizing the opening capability would be to split the opening authority's private key into several shares scattered among multiple servers using techniques from threshold cryptography [17]. This approach, however, requires all shareholders to run a distributed decryption protocol (indeed, any group signature implies a public-key encryption scheme [1]) at every single opening operation, even for identical messages. The GS-MDO primitive comes in handy when many signatures have to be opened on the same message. As a motivating example, we can think of access control gates in public transportation. In order to enter a metro station, the user can generate a signature (i.e., on a message specifying the date and time or his ride) proving his possession of a valid subscription without betraying his identity nor leaking any information on his habits (e.g., the frequency of his rides). If an accident occurs or a crime is committed, the police – which embodies the opening authority in this case – can request the opening tokens for to the time period of the incident and determine who was nearby at that time. In such a situation, the threshold opening approach would incur a substantial overhead to open all the signatures generated by commuters in a given time interval. In contrast, the GS-MDO primitive allows de-anonymizing all signatures corresponding to a given message – no matter how many users signed this message – without having the police interact any further with the public transportation company once the latter has revealed a message-specific token.

As another motivating application, we can think of anonymous comments posted on a blog engine, where a moderator can use a token to open all signatures related to forbidden messages. Yet another example consists of anonymous auctions where bidders sign the amount of their bid: in case of equalities, a single token allows identifying the multiple winners of the auction.

As such, message-dependent openings are relevant when the number of signatures to be opened is potentially high. Moreover, it can be seen as providing the dual functionality of *traceable* signatures [28]. As introduced by Kiayias, Tsiounis and Yung [28], traceable signatures allow the group manager to release a user-specific trapdoor using which all the signatures that user created can be identified. This extended capability allows delegating the tracing operation to parallel tracing agents who can detect all the transactions where a misbehaving user is involved without affecting the anonymity of honest users. Group signatures with message-dependent opening can be motivated in a similar way in that the distributed tracing process can be made with respect to the message rather than the users. If a signed message contains information about a specific suspicious transaction, releasing a message-specific trapdoor makes it possible to trace all parties involved in a given transaction determined by the signed message.

Lattice-based cryptography. Since the results of Regev [41] and Gentry-Peikert-Vaikuntanathan [20], lattice-based cryptography has emerged (see [39] and references therein) as a promising alternative to discrete-logarithm or factoring-based technologies. This trend can be explained by the fact that lattices provide

appealing advantages like simple arithmetic operations, their better asymptotic efficiency or their potential as candidates for post-quantum cryptography: indeed, quantum algorithms are not known to perform any better than classical ones for well-studied problems like *Learning With Errors* (LWE) or *Short Integer Solution* (SIS). Moreover, many advanced cryptographic functionalities (like full homomorphism [19]), which are elusive in the discrete logarithm setting, are enabled by these assumptions.

In this paper, we describe the first lattice-based realization of group signatures with message-dependent opening.

Related Work. The pioneering work of Chaum an Van Heyst [16] inspired many group signature candidates in the nineties but practical and scalable constructions only came out in 2000. The first group signature that was both scalable and collusion-resistant was proposed by Ateniese, Camenisch, Joye and Tsudik [3] under the Strong RSA assumption. At that time, however, there was no precise definition of what it meant for a group signature to be secure. Security analyses were indeed conducted with respect to lists of sometimes redundant requirements. This state-of-affairs changed with the work of Bellare, Micciancio and Warinschi [5] who proposed a model synthesizing the security requirements into two properties named *anonymity* and *traceability*. In this model, Boneh, Boyen and Shacham [7] put forth a practical construction with very short signatures based on pairing-related assumptions. While the solution of [7] was in the random oracle model, constructions in the standard model came out in several works [10,11,24] inspired by the Groth-Sahai methodology [25].

Sakai *et al.* introduced the message-dependent opening functionality [42] in 2012. In their work, they provided evidence that GS-MDO schemes imply identity-based encryption (IBE) [43,8]. In the random oracle model, Ohara *et al.* [37] subsequently designed efficient GS-MDO schemes [37] based on non-standard assumptions in groups with a bilinear map. Libert and Joye [30] appealed to the same tools and the machinery of Groth-Sahai proofs [25] to build a GS-MDO system in the standard model.

While group signatures have attracted much attention in cryptography for many years, the first lattice-based proposal only appeared in 2010 in the work of Gordon, Katz and Vaikuntanathan [22]. While a simple counting argument suggests that no group signature can contain less than $\log N$ bits (where $N$ is the number of group members), the Gordon *et al.* [22] construction had signatures of linear size in $N$. The desired logarithmic size was reached by Laguillaumie *et al.* [29] whose solution still remained quite costly. Although several substantial improvements were recently achieved [34,36,32], lattice-based group signatures are not yet competitive with pairing-based solutions. One of the cited reasons explaining this efficiency gap is the fact that *zero-knowledge proofs* [21] for lattice-related languages [21,33,6] remain less effective than those in groups with a bilinear map, where the rich underlying algebraic structure has proven very useful [25]. An illustration of the limited amount of algebraic structure of lattices is the absence of non-interactive zero knowledge (NIZK) proofs outside the random oracle model in the lattice setting (except for very specific languages [40]).

Even in the random oracle model, the design of lattice-based group signatures with extra properties remains a non-trivial problem. In particular, no GS-MDO system has been proposed so far. In fact, except the theoretical construction of Sakai *et al.* [42], all existing solutions [42,37,30] rely on bilinear maps. For the sake of not putting all one's eggs in the same basket, it is thus important to seek constructions based on different assumptions.

Our Contribution. We propose the first GS-MDO realization based on standard lattice assumptions. The security of our scheme is proved in the random oracle model under SIS and LWE assumptions. We design this scheme by extending the group signature scheme of Ling, Nguyen and Wang [34], which is recalled in Appendix C.1. Not only does this scheme provide one of the most efficient candidates so far, its built-in zero-knowledge arguments turn out to be sufficiently flexible to accommodate our statements in the setting of message-dependent openings. Like [34], our construction proceeds by having each group member's signing key consist of a Boyen [9] signature for his identity $d \in \{0,1\}^{\ell}$. To sign a message $M$, the user encrypt his identity $d$ using an IND-CCA encryption scheme derived from the Gentry-Peikert-Vaikuntanathan (GPV) IBE [20] via the Canetti-Halevi-Katz (CHK) paradigm [14]. Then, the user provides a ZK argument of possession of a Boyen signature for the message encrypted by the ciphertext, the message being embedded in the Fiat-Shamir challenge to make the proof non-interactive. Our scheme takes advantage of the fact that Ling *et al.* [34] used an IBE to encrypt the group member's identifier. We add a second encryption layer in order to encrypt the ciphertext under the identity $M$, which is the message to be signed. Therefore, the GS-MDO functionality can be achieved by combining two instances of the GPV IBE (one for the admitter and the second one for the opening authority). To reveal a message-specific token $\mathrm{t}_M$, the admitter can simply output a private key for the identity $M$, then allowing the opener to retrieve the ciphertext hiding the identity. Then, using the encryption layer as in the Ling *et al.* scheme [34] allows us to adapt the underlying argument system to our purpose.

Now, the challenge is to prove that the entire double-encryption process was conducted properly. To this end, we can leverage the properties of Stern-like protocols [44] and translate the statements to be proved so as to apply the recently proposed framework of [31]. Our argument system, while addressing a more elaborate relation than in [34], is constructed in a simpler and more modular manner. In short, we reduce the entire statement into an assertion of the form $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$, where $\mathbf{P}$ is a public matrix that depends on the group public key and the outer ciphertext layer, while $\mathbf{x}$ is a short vector which is constructed from the witness and has a special structure.

We can also notice that our technique can be used to enable message-dependent opening in the case of *dynamically growing groups* as well. For instance, the two-layer encryption method can be straightforwardly adapted to the dynamic group signature scheme from Libert *et al.* [31] which is also built upon the Ling *et al.* scheme [34] and also relies on Stern-like ZK arguments.

ROADMAP. To present our results, the rest of the paper is organized as follows. In Section 2, we first recall the necessary definitions and security notions. The supporting zero-knowledge argument system is constructed in Section 3. In Section 4, we present our lattice-based GS-MDO scheme.

## 2 Background

NOTATIONS. Matrices are denoted with bold upper-case letters $\mathbf{A}$ and vectors in bold lower-case letters $\mathbf{x}$. We assume that all vectors are column vectors. The concatenation of vectors $\mathbf{x} \in \mathbb{R}^k$ and $\mathbf{y} \in \mathbb{R}^m$ is denoted by $(\mathbf{x}\|\mathbf{y}) \in \mathbb{R}^{k+m}$. We denote the column concatenation of matrices $\mathbf{A} \in \mathbb{R}^{n\times k}$ and $\mathbf{B} \in \mathbb{R}^{n\times m}$ by $[\mathbf{A}|\mathbf{B}]$. If dimensions are compatible, $\langle \mathbf{u}, \mathbf{v}\rangle$ denote the inner product of vectors $\mathbf{u}$ and $\mathbf{v}$. The identity matrix of order $k$ is denoted by $\mathbf{I}_k$, and $\mathbf{0}_\ell$ stands for the zero vector of dimension $\ell$. If $\mathbf{A}$ is a full column rank matrix, we let $\widetilde{\mathbf{A}}$ denote its Gram-Schmidt orthogonalization. If $\mathbf{u} \in \mathbb{R}^n$, its Euclidean norm is denoted by $\|\mathbf{b}\|$ and this notation is extended to matrices $\mathbf{A} \in \mathbb{R}^{n\times m}$ with columns $(\mathbf{a}_i)_{i\leq m}$ by $\|\mathbf{A}\| = \max_{i\leq m} \|\mathbf{a}_i\|$. Finally, PPT stands for *Probabilistic Polynomial-Time*.

### 2.1 Lattices

A lattice $\Lambda$ is a discrete subgroup of some space $\mathbb{R}^n$, which can be seen as the set of integer linear combinations of linearly independent vectors $(\mathbf{b}_i)_{i\leq n}$. Over a lattice $\Lambda$, and given a parameter $\sigma \in \mathbb{R}_+^*$, we define the Gaussian distribution of support $\Lambda$ and parameter $\sigma$ by $D_{\Lambda,\sigma}[\mathbf{b}] \sim \exp\big(-\pi\|\mathbf{b}\|^2/\sigma^2\big)$, for all $\mathbf{b} \in \Lambda$. We will use the fact that samples from $D_{\Lambda,\sigma}$ are short with overwhelming probability.

**Lemma 1 ([4, Le. 1.5]).** *For any lattice $\Lambda \subseteq \mathbb{R}^n$ and positive real number $\sigma$, we have $\Pr_{\mathbf{b}\hookleftarrow D_{\Lambda,\sigma}}[\|\mathbf{b}\| \leq \sqrt{n}\sigma] \geq 1 - 2^{-\Omega(n)}$.*

Gentry, Peikert and Vaikuntanathan [20] show that it is possible to efficiently sample from a Gaussian distribution on a lattice support given a sufficiently short basis of this lattice.

**Lemma 2 ([12, Le. 2.3]).** *There exists a PPT algorithm GPVSample that takes as inputs a basis $\mathbf{B}$ of a lattice $\Lambda \subseteq \mathbb{Z}^n$ and rational $\sigma \geq \|\widetilde{\mathbf{B}}\| \cdot \Omega(\sqrt{\log n})$, and outputs vectors $\mathbf{b} \in \Lambda$ with distribution $D_{\Lambda,\sigma}$.*

**Definition 1.** *Let $m \geq n \geq 1$ and $q \geq 2$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, define $\Lambda_q(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m : \exists \mathbf{s} \in \mathbb{Z}^n \text{ s.t. } \mathbf{A}^T \cdot \mathbf{s} = \mathbf{x} \bmod q\}$ and*

$$\Lambda_q^\perp(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q\}, \quad \Lambda_q^{\mathbf{u}}(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q\}.$$

We also use an algorithm that jointly samples an uniform matrix $\mathbf{A}$ and a short basis of the lattice $\Lambda_q^\perp(\mathbf{A})$.

**Lemma 3 ([2, Th. 3.2]).** *There exists a* PPT *algorithm* GenTrap *that takes as inputs* $1^n$, $1^m$ *and an integer* $q \geq 2$ *with* $m \geq \Omega(n \log q)$, *and outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *such that* $\mathbf{A}$ *is within statistical distance* $2^{-\Omega(n)}$ *to* $U(\mathbb{Z}_q^{n \times m})$, *and* $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{n \log q})$.

The description of our scheme also uses an algorithm that extends a trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ to a trapdoor of any $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ whose left $n \times m$ submatrix is $\mathbf{A}$.

**Lemma 4 ([15, Le. 3.2]).** *There exists a* PPT *algorithm* ExtBasis *that takes as inputs a matrix* $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ *whose first* $m$ *columns span* $\mathbb{Z}_q^n$, *and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *where* $\mathbf{A}$ *is the left* $n \times m$ *submatrix of* $\mathbf{B}$, *and outputs a basis* $\mathbf{T_B}$ *of* $\Lambda_q^{\perp}(\mathbf{B})$ *with* $\|\widetilde{\mathbf{T_B}}\| \leq \|\widetilde{\mathbf{T_A}}\|$.

### 2.2 Hardness assumptions

We prove the security of our scheme in the ROM among the assumption that both algorithmic problems below are hard, in the sense that they cannot be solved by any PPT algorithm with non-negligible probability nor advantage respectively.

**Definition 2.** *Let* $m, q, \beta$ *be functions of a parameter* $n$. *The Short Integer Solution problem* $\mathsf{SIS}_{m,q,\beta}$ *is as follows: Given* $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$, *find* $\mathbf{x} \in \Lambda_q^{\perp}(\mathbf{A})$ *with* $0 < \|\mathbf{x}\| \leq \beta$.

**Definition 3.** *Let* $q, \alpha$ *be functions of a parameter* $n$. *For* $\mathbf{s} \in \mathbb{Z}_q^n$ *(a secret), the distribution* $A_{q,\alpha,\mathbf{s}}$ *over* $\mathbb{Z}_q^n \times \mathbb{Z}_q$ *is obtained by sampling* $\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n)$ *and (a noise)* $e \hookleftarrow D_{\mathbb{Z},\alpha q}$, *and returning* $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$. *The Learning With Errors problem* $\mathsf{LWE}_{q,\alpha}$ *is as follows: For* $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, *distinguish between arbitrarily many independent samples from* $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ *and the same number of independent samples from* $A_{q,\alpha,\mathbf{s}}$.

If $q \geq \sqrt{n}\beta$ and $m, \beta \leq \mathsf{poly}(n)$, then standard worst-case lattice problems with approximation factors $\gamma = \widetilde{\mathcal{O}}(\beta\sqrt{n})$ reduce to $\mathsf{SIS}_{m,q,\beta}$ (see for instance [20, Se. 9]). Similarly, if $\alpha q = \Omega(\sqrt{n})$, then standard worst-case lattice problems with approximation factors $\gamma = \mathcal{O}(\alpha/n)$ quantumly reduce to $\mathsf{LWE}_{q,\alpha}$ (see [41] as well as [38,12] for classical analogues).

### 2.3 Group Signature with Message Dependent Opening

We use the syntax of Sakai *et al.* [42] to describe a GS-MDO, which extends the model of Bellare, Micciancio and Warinschi [5].

**Definition 4 (GS-MDO).** *A* group signature with message-dependent opening *is a tuple of algorithms* (Keygen, Sign, Verify, TrapGen, Open) *such that:*

**Keygen**$(1^\lambda, 1^N)$**:** *Given a security parameter* $\lambda$ *and the number of group members* $N$, *outputs the group public key* gpk, *the opening key* ok, *the the admitter's private key* $\mathsf{msk}_{\mathsf{ADM}}$, *and a vector of user secret keys* $\mathbf{gsk} = (\mathbf{gsk}[d])_{d=0}^{N-1}$.

**Sign**$(\mathsf{gpk}, \mathbf{gsk}[d], M)$**:** *Given an user $d$ secret key $\mathbf{gsk}[d]$ and a message $M$, issue a signature $\Sigma$ for the message $M$.*

**Verify**$(\mathsf{gpk}, M, \Sigma)$**:** *Given a message $M$ and a signature $\Sigma$, output $0$ or $1$.*

**TrapGen**$(\mathsf{gpk}, \mathsf{msk}_{\mathsf{ADM}}, M)$**:** *Given the admitter key $\mathsf{msk}_{\mathsf{ADM}}$, and a message $M$, output a token $\mathrm{t}_M$.*

**Open**$(\mathsf{gpk}, \mathsf{ok}, \mathrm{t}_M, M, \Sigma)$**:** *Given the opening key $\mathsf{ok}$, a message $M$, a token $\mathrm{t}_M$ for this message, and a signature $\Sigma$, return either $d \in \mathbb{N}$, or $\perp$.*

These algorithms must also verify the correctness property, meaning that for all $(\mathsf{gpk}, \mathbf{gsk}, \mathsf{ok}, \mathsf{msk}_{\mathsf{ADM}}) \leftarrow \mathsf{Keygen}(1^\lambda, 1^N)$, for all $d \in \{0, \ldots, N-1\}$, and for all $M \in \{0,1\}^*$, we have w.h.p. $\mathsf{Verify}(\mathsf{gpk}, M, \mathsf{Sign}(\mathsf{gpk}, \mathbf{gsk}[d], M)) = 1$ and $\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathsf{TrapGen}(\mathsf{gpk}, \mathsf{msk}_{\mathsf{ADM}}, M), M, \mathsf{Sign}(\mathsf{gpk}, \mathbf{gsk}[d], M)) = d$.

Like in a classical group signature, the scheme must verify *Traceability* and *Anonymity*, but since the opening capability is split in two entities, namely the admitter and the opening authority (also known as the group manager), they therefore are two anonymity definitions: the *Opener Anonymity* and the *Admitter Anonymity*, which are formalized as follows.

**Definition 5 (Traceability).** *A GS-MDO scheme provides full traceability if, for any $\lambda \in \mathbb{N}$, any $N \in \mathsf{poly}(\lambda)$ and any* PPT *adversary $\mathcal{A}$ involved in the experiment below, it holds that $\mathbf{Adv}^{\mathrm{trace}}_{\mathcal{A}}(\lambda) = \Pr[\mathbf{Exp}^{\mathrm{trace}}_{\mathcal{A}}(\lambda, N) = 1] \in \mathsf{negl}(\lambda)$.*

$\underline{\boldsymbol{Exp}^{\mathit{trace}}_{\mathcal{A}}(\lambda, N)}$
$(\mathsf{gpk}, \mathsf{ok}, \mathsf{msk}_{\mathsf{ADM}}, \mathbf{gsk}) \leftarrow \mathsf{Keygen}(\lambda, N)$
$\mathtt{st} \leftarrow (\mathsf{ok}, \mathsf{msk}_{\mathsf{ADM}}, \mathsf{gpk})$ ; $\mathcal{C} \leftarrow \emptyset$ ; $K \leftarrow \varepsilon$ ; $\mathit{Cont} \leftarrow \mathtt{true}$
while $(\mathit{Cont} = \mathtt{true})$ do
  $(\mathit{Cont}, \mathtt{st}, j) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathbf{gsk}[\cdot], \cdot)}(\mathit{choose}, \mathtt{st}, K)$
  if $\mathit{Cont} = \mathtt{true}$ then $\mathcal{C} \leftarrow \mathcal{C} \cup \{j\}$ ; $K \leftarrow K \cup \{\mathbf{gsk}[j]\}$ end if
$(M^\star, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathbf{gsk}[\cdot], \cdot)}(\mathit{guess}, \mathtt{st})$
if $\mathsf{Verify}(\mathsf{gpk}, M^\star, \sigma^\star) = 0$ then *Return* $0$
if $\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathsf{TrapGen}(\mathsf{gpk}, \mathsf{msk}_{\mathsf{ADM}}, M^\star), M^\star, \sigma^\star) = \perp$ then *Return* $1$
if $\exists j^\star \in \{0, \ldots, N-1\}$ *such that*
  $(\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, t_{M^\star}, M^\star, \sigma^\star) = j^\star) \wedge (j^\star \notin \mathcal{C}) \wedge ((j^\star, M^\star)$ *not queried by* $\mathcal{A})$
  *with* $t_{M^\star} \leftarrow \mathsf{TrapGen}(\mathsf{gpk}, \mathsf{msk}_{\mathsf{ADM}}, M^\star)$
then *Return* $1$ else *Return* $0$

**Definition 6 (Admitter Anonymity).** *A GS-MDO scheme provides full anonymity against the admitter if, for any $\lambda \in \mathbb{N}$, any $N \in \mathsf{poly}(\lambda)$ and any* PPT *adversary $\mathcal{A}$ involved in the experiment hereunder, we have*

$$\mathbf{Adv}^{\text{anon-adm}}_{\mathcal{A}}(\lambda) = |\Pr[\mathbf{Exp}^{\text{anon-adm}}_{\mathcal{A}}(\lambda, N) = 1] - 1/2| \in \mathsf{negl}(\lambda).$$

$\underline{\boldsymbol{Exp}^{\mathit{anon-adm}}_{\mathcal{A}}(\lambda, N)}$
$(\mathsf{gpk}, \mathsf{ok}, \mathsf{msk}_{\mathsf{ADM}}, \mathbf{gsk}) \leftarrow \mathsf{Keygen}(\lambda, N)$
$(\mathtt{st}, j_0, j_1, M^\star) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ok}}}(\mathit{choose}, \mathsf{gpk}, \mathbf{gsk}, \mathsf{msk}_{\mathsf{ADM}})$
$b \hookleftarrow \{0,1\}$; $\quad \sigma^\star \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathbf{gsk}[j_b], M^\star)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ok}}}(\mathit{guess}, \mathtt{st}, \sigma^\star)$
*Return* $1$ *if* $b' = b$ *and* $0$ *otherwise*

*Here, $\mathcal{O}_{\mathsf{ok}}$ is an oracle that takes as input an arbitrary signature $\sigma \neq \sigma^\star$ and uses $\mathsf{ok}$ and $\mathsf{msk}_{\mathsf{ADM}}$ to return the identity of the signer.*

**Definition 7 (Opener Anonymity).** *A GS-MDO scheme provides full anonymity against the opener if, for any $\lambda \in \mathbb{N}$, any $N \in \mathsf{poly}(\lambda)$ and any $\mathsf{PPT}$ adversary $\mathcal{A}$ involved in the experiment below, it holds that*

$$\mathbf{Adv}^{\text{anon-oa}}_{\mathcal{A}}(\lambda) = |\Pr[\mathbf{Exp}^{\text{anon-oa}}_{\mathcal{A}}(\lambda, N) = 1] - 1/2| \in \mathsf{negl}(\lambda).$$

$\underline{\boldsymbol{Exp}^{anon-oa}_{\mathcal{A}}(\lambda, N)}$
$(\mathsf{gpk}, \mathsf{ok}, \mathsf{msk}_{\mathsf{ADM}}, \mathbf{gsk}) \leftarrow \mathsf{Keygen}(\lambda, N)$
$(\mathtt{st}, j_0, j_1, M^\star) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{msk}_{\mathsf{ADM}}}}(choose, \mathsf{gpk}, \mathbf{gsk}, \mathsf{ok})$
$b \hookleftarrow \{0,1\}; \quad \sigma^\star \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathbf{gsk}[j_b], M^\star)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{msk}_{\mathsf{ADM}}}}(guess, \mathtt{st}, \sigma^\star)$
*Return 1 if $b' = b$ and 0 otherwise*

*In the above notation, $\mathcal{O}_{\mathsf{msk}_{\mathsf{ADM}}}(.)$ is an oracle that returns trapdoors for arbitrary messages $M \neq M^\star$ chosen by the adversary.*

### 2.4 Zero-Knowledge Arguments of Knowledge

We will work with statistical zero-knowledge argument systems, namely, interactive protocols where the zero-knowledge property holds against *any* cheating verifier, while the soundness property only holds against *computationally bounded* cheating provers. More formally, let the set of statements-witnesses $\mathrm{R} = \{(y, w)\} \in \{0,1\}^* \times \{0,1\}^*$ be an $\mathsf{NP}$ relation. A two-party game $\langle \mathcal{P}, \mathcal{V} \rangle$ is called an interactive argument system for the relation R with soundness error $e$ if the following two conditions hold:

- $\mathsf{Completeness}$. If $(y, w) \in \mathrm{R}$ then $\Pr\big[\langle \mathcal{P}(y, w), \mathcal{V}(y) \rangle = 1\big] = 1$.
- $\mathsf{Soundness}$. For any $\mathsf{PPT}$ $\widehat{\mathcal{P}}$, if $(y, w) \notin \mathrm{R}$, then $\Pr[\langle \widehat{\mathcal{P}}(y, w), \mathcal{V}(y) \rangle = 1] \leq e$.

An argument system is called statistical zero-knowledge if for any $\widehat{\mathcal{V}}(y)$, there exists a $\mathsf{PPT}$ simulator $\mathcal{S}(y)$ producing a simulated transcript that is statistically close to the one of the real interaction between $\mathcal{P}(y, w)$ and $\widehat{\mathcal{V}}(y)$. A related notion is argument of knowledge, which requires the witness-extended emulation property. For protocols consisting of 3 moves (*i.e.*, commitment-challenge-response), witness-extended emulation is implied by *special soundness* [23], where the latter assumes that there exists a $\mathsf{PPT}$ extractor which takes as input a set of valid transcripts with respect to all possible values of the 'challenge' to the same 'commitment', and outputs $w'$ such that $(y, w') \in \mathrm{R}$.

Our statistical zero-knowledge arguments of knowledge ($\mathsf{sZKAoK}$) are Stern-type [44]. In particular, they are $\Sigma$-protocols in the generalized sense defined in [26,6] (where 3 valid transcripts are needed for extraction, instead of just 2).

## 3 The Underlying Zero-Knowledge Argument System

First of all, we recall that the protocol from [34] allows prover $\mathcal{P}$ to convince verifier $\mathcal{V}$ in $\mathsf{ZK}$ that $\mathcal{P}$ knows a valid message-signature pair $(d, \mathbf{z})$ for Boyen's signature scheme [9], and that the binary representation of $d$ is honestly encrypted to a given ciphertext pair $(\mathbf{c}_1, \mathbf{c}_2)$. The strategy in [34] was to extend Stern's protocol [44] (via the Decomposition-Extension technique [33]) to prove the statement in a *ad-hoc* manner. However, their argument system was rather complicated, which makes it somewhat inflexible to be used as a sub-protocol in designing more advanced constructions.

The goal of this section is to construct the statistical zero-knowledge argument of knowledge ($\mathsf{sZKAoK}$) underlying the GS-MDO scheme of Section 4. In our setting, the ciphertext component $\mathbf{c}_2$ is hidden, and $\mathcal{P}$ can additionally prove that the secret bits representing $\mathbf{c}_2$ are correctly encrypted to another given ciphertext pair $(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$. By using the new strategy for Stern-like protocols, recently proposed in [31], we can handle the extended relation, yet the resulting argument system is obtained in a simpler and more modular manner than in [34].

More formally, let $n, m, \ell, q, \beta, b$ be positive integers and $k = \lceil \log q \rceil$. Let $\mathbf{H} = \mathbf{I}_\ell \otimes \left(1 \mid 2 \mid 4 \mid \cdots \mid 2^{k-1}\right) \in \mathbb{Z}_q^{\ell \times \ell k}$, and let $\mathsf{bin} : \mathbb{Z}_q^\ell \to \{0,1\}^{\ell k}$ be the function mapping $\mathbf{w}$ to its component-wise binary decomposition $\mathsf{bin}(\mathbf{w})$. (Note that for all $\mathbf{w} \in \mathbb{Z}_q^\ell$, we have $\mathbf{H} \cdot \mathsf{bin}(\mathbf{w}) = \mathbf{w}$.) We define as well the binary decomposition function for integer $\mathsf{bin} : \mathbb{N} \to \{0,1\}^\star$.
The relation $\mathrm{R}_{\mathrm{gsmdo}}$ associated with our protocol is then defined as follows.

**Definition 8.** *Define*

$$\mathrm{R}_{\mathrm{gsmdo}} = \left\{(\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^{\ell}, \mathbf{B}, \mathbf{C}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{u}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2), \mathbf{d}, \mathbf{z}, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{e}_1, \hat{\mathbf{e}}_1, \mathbf{e}_2, \hat{\mathbf{e}}_2, \mathbf{c}_2\right\}$$

*as a relation where*

$$\begin{cases} \mathbf{A}, \{\mathbf{A}_i\}_{i=0}^{\ell}, \mathbf{B}, \mathbf{C} \in \mathbb{Z}_q^{n \times m}; \mathbf{G} \in \mathbb{Z}_q^{n \times \ell}; \hat{\mathbf{G}} \in \mathbb{Z}_q^{n \times \ell k}; \mathbf{u} \in \mathbb{Z}_q^n; \mathbf{c}_1, \hat{\mathbf{c}}_1 \in \mathbb{Z}_q^m; \hat{\mathbf{c}}_2 \in \mathbb{Z}_q^{\ell k}; \\ \mathbf{d} = (d_1, \ldots, d_\ell) \in \{0,1\}^\ell; \mathbf{z} \in [-\beta, \beta]^{2m}; \mathbf{s}, \hat{\mathbf{s}} \in [-b, b]^n; \mathbf{e}_1, \hat{\mathbf{e}}_1 \in [-b, b]^m; \\ \mathbf{e}_2 \in [-b, b]^\ell; \hat{\mathbf{e}}_2 \in [-b, b]^{\ell k}; \mathbf{c}_2 \in \mathbb{Z}_q^\ell \end{cases}$$

*satisfy*

$$\begin{cases} \left[\mathbf{A} | \mathbf{A}_0 + \sum_{i=1}^{\ell} d_i \cdot \mathbf{A}_i\right] \cdot \mathbf{z} = \mathbf{u} \bmod q & (1) \\ \mathbf{c}_1 = \mathbf{B}^\top \cdot \mathbf{s} + \mathbf{e}_1 \bmod q; \;\; \mathbf{c}_2 = \mathbf{G}^\top \cdot \mathbf{s} + \mathbf{e}_2 + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{d} \bmod q & (2) \\ \hat{\mathbf{c}}_1 = \mathbf{C}^\top \cdot \hat{\mathbf{s}} + \hat{\mathbf{e}}_1 \bmod q; \; \hat{\mathbf{c}}_2 = \hat{\mathbf{G}}^\top \cdot \hat{\mathbf{s}} + \hat{\mathbf{e}}_2 + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathsf{bin}(\mathbf{c}_2) \bmod q. & (3) \end{cases}$$

In Section 3.1, we present Stern's protocol from a high-level point of view, according to the abstraction of [31]. Then, via the transformations performed in Section 3.2, we show how to obtain a $\mathsf{ZKAoK}$ for $\mathrm{R}_{\mathrm{gsmdo}}$ based on this abstract protocol.

### 3.1 Stern's Protocol, from a High-Level Viewpoint

Let $D, L, q \geq 2$ be positive integers and let $\mathsf{VALID}$ be a subset of $\{-1, 0, 1\}^L$. Suppose that $\mathcal{S}$ is a finite set such that one can associate every $\pi \in \mathcal{S}$ with a permutation $T_\pi$ of $L$ elements, satisfying the following condition:

$$\mathbf{x} \in \mathsf{VALID} \iff T_\pi(\mathbf{x}) \in \mathsf{VALID}. \tag{4}$$

We aim to construct a $\mathsf{sZKAoK}$ for the following abstract relation:

$$\mathrm{R}_{\mathrm{abstract}} = \big\{(\mathbf{P}, \mathbf{v}), \mathbf{x} \in \mathbb{Z}_q^{D \times L} \times \mathbb{Z}_q^D \times \mathsf{VALID} : \mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q.\big\}$$

Note that, Stern's original protocol corresponds to the special case when $\mathsf{VALID} = \{\mathbf{x} \in \{0,1\}^L : \mathsf{wt}(\mathbf{x}) = k\}$ (where $\mathsf{wt}(\cdot)$ denotes the Hamming weight and $k < L$ is a given integer), $\mathcal{S} = \mathcal{S}_L$ - hereunder the set of all permutations of $L$ elements, and $T_\pi(\mathbf{x}) = \pi(\mathbf{x})$.

The equivalence in (4) plays a crucial role in proving in $\mathsf{ZK}$ that $\mathbf{x} \in \mathsf{VALID}$: To do so $\mathcal{P}$ samples $\pi \hookleftarrow U(\mathcal{S})$ and let $\mathcal{V}$ check that $T_\pi(\mathbf{x}) \in \mathsf{VALID}$, while the later cannot learn any additional information about $\mathbf{x}$ thanks to the randomness of $\pi$. Furthermore, to prove in $\mathsf{ZK}$ that the linear equation holds, $\mathcal{P}$ samples a masking vector $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^L)$, sends $\mathbf{y} = \mathbf{x} + \mathbf{r} \bmod q$, and convinces $\mathcal{V}$ instead that $\mathbf{P} \cdot \mathbf{y} = \mathbf{P} \cdot \mathbf{r} + \mathbf{v} \bmod q$.

The interactive protocol between $\mathcal{P}(\mathbf{P}, \mathbf{v}, \mathbf{x})$ and $\mathcal{V}(\mathbf{P}, \mathbf{v})$, which employs a statistically hiding and computationally binding string commitment scheme $\mathsf{COM}$ (e.g., the $\mathsf{SIS}$-based one from [27]), is described in Figure 1.

1. **Commitment:** $\mathcal{P}$ samples $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^L)$, $\pi \hookleftarrow U(\mathcal{S})$ and randomness $\rho_1, \rho_2, \rho_3$ for $\mathsf{COM}$. Then $\mathcal{P}$ sends the commitment $\mathrm{CMT} = \big(C_1, C_2, C_3\big)$ to $\mathcal{V}$, where

   $$C_1 = \mathsf{COM}(\pi, \mathbf{P} \cdot \mathbf{r}; \rho_1), \;\; C_2 = \mathsf{COM}(T_\pi(\mathbf{r}); \rho_2), \;\; C_3 = \mathsf{COM}(T_\pi(\mathbf{x} + \mathbf{r}); \rho_3).$$

2. **Challenge:** $\mathcal{V}$ sends a challenge $Ch \hookleftarrow U(\{1, 2, 3\})$ to $\mathcal{P}$.
3. **Response:** Depending on $Ch$, $\mathcal{P}$ sends the response RSP computed as follows:
   - $Ch = 1$: Let $\mathbf{t}_x = T_\pi(\mathbf{x})$, $\mathbf{t}_r = T_\pi(\mathbf{r})$, and $\mathrm{RSP} = (\mathbf{t}_x, \mathbf{t}_r, \rho_2, \rho_3)$.
   - $Ch = 2$: Let $\pi_2 = \pi$, $\mathbf{y} = \mathbf{x} + \mathbf{r}$, and $\mathrm{RSP} = (\pi_2, \mathbf{y}, \rho_1, \rho_3)$.
   - $Ch = 3$: Let $\pi_3 = \pi$, $\mathbf{r}_3 = \mathbf{r}$, and $\mathrm{RSP} = (\pi_3, \mathbf{r}_3, \rho_1, \rho_2)$.

**Verification:** Receiving RSP, the verifier proceeds as follows:

- $Ch = 1$: Check that $\mathbf{t}_x \in \mathsf{VALID}$ and $C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2)$, $C_3 = \mathsf{COM}(\mathbf{t}_x + \mathbf{t}_r; \rho_3)$.
- $Ch = 2$: Check that $C_1 = \mathsf{COM}(\pi_2, \mathbf{P} \cdot \mathbf{y} - \mathbf{v}; \rho_1)$, $C_3 = \mathsf{COM}(T_{\pi_2}(\mathbf{y}); \rho_3)$.
- $Ch = 3$: Check that $C_1 = \mathsf{COM}(\pi_3, \mathbf{P} \cdot \mathbf{r}_3; \rho_1)$, $\;C_2 = \mathsf{COM}(T_{\pi_3}(\mathbf{r}_3); \rho_2)$.

In each case, $\mathcal{V}$ outputs 1 if and only if all the conditions hold. Otherwise, it outputs 0.

**Figure 1:** A $\mathsf{ZKAoK}$ for the relation $\mathrm{R}_{\mathrm{abstract}}$.

The properties of the given protocol is summarized in the following lemma.

**Lemma 5.** *The protocol in Figure 1 is a* sZKAoK *for the relation* $\mathrm{R}_{\mathrm{abstract}}$ *with perfect completeness, soundness error* $2/3$*, and communication cost* $\widetilde{\mathcal{O}}(L \log q)$*. In particular:*

- *There exists an efficient simulator that, on input* $(\mathbf{P}, \mathbf{v})$*, outputs an accepted transcript which is statistically close to that produced by the real prover.*
- *There exists an efficient knowledge extractor that, on input a commitment* CMT *and* 3 *valid responses* $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ *to all* 3 *possible values of the challenge* $Ch$*, outputs* $\mathbf{x}' \in \mathsf{VALID}$ *such that* $\mathbf{P} \cdot \mathbf{x}' = \mathbf{v} \bmod q$*.*

The proof of Lemma 5 employs standard simulation and extraction techniques for Stern-type protocols [27,33,34,18,32]. We defer it to Appendix B.

### 3.2 From $\mathbf{R}_{\mathbf{gsmdo}}$ to $\mathbf{R}_{\mathbf{abstract}}$

We show that a sZKAoK for relation $\mathrm{R}_{\mathrm{gsmdo}}$ in Definition 8 can be derived from the one for relation $\mathrm{R}_{\mathrm{abstract}}$ from Section 3.1. In the process, we employ the Decomposition-Extension technique from [33], which we will formalize as follows.

- For any positive integer $i$, denote by $\mathsf{B}_{2i}$ the set of all vectors in $\{0,1\}^{2i}$ having exactly $i$ coordinates equal to 1, and denote by $\mathsf{B}_{3i}$ the set of all vectors in $\{-1,0,1\}^{3i}$ having exactly $i$ coordinates equal to $j$, for every $j \in \{-1,0,1\}$.
- Define, for any integer $B > 0$, the number $\delta_B := \lfloor \log B \rfloor + 1$ and the sequence $B_1, \ldots, B_{\delta_B}$, where $B_j = \left\lfloor \frac{B+2^{j-1}}{2^j} \right\rfloor$ for all $j \in [\delta_B]$. As noted in [33], this sequence satisfies $\sum_{j=1}^{\delta_B} B_j = B$, and any integer in $[-B, B]$ can be expressed as a linear combination of the $B_j$'s with coefficients in $\{-1,0,1\}$.
- Define the following matrices for any positive integers $\mathfrak{m}, B$:

$$\mathbf{H}_{\mathfrak{m},B} = \begin{bmatrix} B_1 \ldots B_{\delta_B} & & & \\ & B_1 \ldots B_{\delta_B} & & \\ & & \ddots & \\ & & & B_1 \ldots B_{\delta_B} \end{bmatrix} \in \mathbb{Z}^{\mathfrak{m} \times \mathfrak{m}\delta_B},$$

  and $\mathbf{H}^*_{\mathfrak{m},B} = \left[\mathbf{H}_{\mathfrak{m},B} \middle| \mathbf{0}^{\mathfrak{m} \times 2\mathfrak{m}\delta_B}\right] \in \mathbb{Z}^{\mathfrak{m} \times 3\mathfrak{m}\delta_B}$.

**Lemma 6 (Decomposition-Extension).** *Let* $\mathfrak{m}, B$ *be positive integers. Then, there exists an efficient algorithm that on input vector* $\mathbf{v} \in [-B, B]^{\mathfrak{m}}$*, outputs vector* $\mathbf{v}^* \in \mathsf{B}_{3m\delta_B}$ *such that* $\mathbf{H}^*_{\mathfrak{m},B} \cdot \mathbf{v}^* = \mathbf{v}$*.*

*Proof.* Let $\mathbf{v} = (v_1, \ldots, v_{\mathfrak{m}})$, where $v_i \in [-B, B]$ for all $i \in [\mathfrak{m}]$. For each $i$, one can efficiently find $v_{i,1}, \ldots, v_{i,\delta_B} \in \{-1,0,1\}$ such that $\sum_{j=1}^{\delta_B} B_j \cdot v_{i,j} = v_i$.

Let $\mathbf{v}' = (v_{1,1}, \ldots, v_{1,\delta_B}, v_{2,1}, \ldots, v_{2,\delta_B}, \ldots, v_{\mathfrak{m},1}, \ldots, v_{\mathfrak{m},\delta_B}) \in \{-1,0,1\}^{\mathfrak{m}\delta_B}$, then $\mathbf{H}_{\mathfrak{m},B} \cdot \mathbf{v}' = \mathbf{v}$. By appending $2\mathfrak{m}\delta_B$ suitable coordinates to $\mathbf{v}'$, one can obtain a vector $\mathbf{v}^* \in \mathsf{B}_{3m\delta_B}$ such that $\mathbf{H}^*_{\mathfrak{m},B} \cdot \mathbf{v}^* = \mathbf{v}$. □

We now transform equations in Definition 8 into a unified equation of the form $\mathbf{P}\cdot\mathbf{x} = \mathbf{v} \bmod q$. Regarding Equation (1), if we write $\mathbf{z}$ as $\mathbf{z} = (\mathbf{z}_1 \| \mathbf{z}_2)$, where $\mathbf{z}_1, \mathbf{z}_2 \in [-\beta, \beta]^m$, and let $\mathbf{z}_1^*, \mathbf{z}_2^* \in \mathsf{B}_{3m\delta_\beta}$ be the vectors obtained by applying Lemma 6 to $\mathbf{z}_1, \mathbf{z}_2$, respectively, then we have:

$$\begin{aligned}\mathbf{u} &= \left[\mathbf{A}|\mathbf{A}_0 + \sum_{i=1}^{\ell} d_i \cdot \mathbf{A}_i\right] \cdot \mathbf{z} = \mathbf{A}\cdot\mathbf{z}_1 + \mathbf{A}_0\cdot\mathbf{z}_2 + \sum_{i=1}^{\ell} d_i\cdot\mathbf{A}_i\cdot\mathbf{z}_2 \bmod q \\ &= (\mathbf{A}\cdot\mathbf{H}^*_{m,\beta})\cdot\mathbf{z}_1^* + (\mathbf{A}_0\cdot\mathbf{H}^*_{m,\beta})\cdot\mathbf{z}_2^* + \sum_{i=1}^{\ell}(\mathbf{A}_i\cdot\mathbf{H}^*_{m,\beta})\cdot(d_i\cdot\mathbf{z}_2^*) \bmod q \\ &= \overline{\mathbf{A}}\cdot\bar{\mathbf{z}} \bmod q,\end{aligned}$$

where

$$\begin{cases}\overline{\mathbf{A}} = \left[\mathbf{A}\cdot\mathbf{H}^*_{m,\beta}|\mathbf{A}_0\cdot\mathbf{H}^*_{m,\beta}|\mathbf{A}_1\cdot\mathbf{H}^*_{m,\beta}|\ldots|\mathbf{A}_\ell\cdot\mathbf{H}^*_{m,\beta}\right] \in \mathbb{Z}_q^{n\times(\ell+2)3m\delta_\beta} \\ \bar{\mathbf{z}} = \left(\mathbf{z}_1^*\|\mathbf{z}_2^*\|d_1\cdot\mathbf{z}_2^*\|\ldots\|d_\ell\cdot\mathbf{z}_2^*\right) \in \{-1,0,1\}^{(\ell+2)3m\delta_\beta}.\end{cases}$$

Next, we extend $\mathbf{d} = (d_1,\ldots,d_\ell)$ to $\mathbf{d}^* = (d_1,\ldots,d_\ell,d_{\ell+1},\ldots,d_{2\ell}) \in \mathsf{B}_{2\ell}$, and let $\mathbf{z}^* = \left(\bar{\mathbf{z}}\|d_{\ell+1}\cdot\mathbf{z}_2^*\|\ldots\|d_{2\ell}\cdot\mathbf{z}_2^*\right)$ and $\mathbf{A}^* = \left[\overline{\mathbf{A}}\,|\mathbf{0}^{n\times\ell 3m\delta_\beta}\right] \in \mathbb{Z}_q^{n\times(2\ell+2)3m\delta_\beta}$, then we have the following equation:

$$\mathbf{A}^*\cdot\mathbf{z}^* = \mathbf{u} \bmod q. \tag{5}$$

Meanwhile, we observe that (2) and (3) can be unified in the following form:

$$\left(\begin{array}{c}\mathbf{0}\\\hline \lfloor\frac{q}{2}\rfloor\mathbf{I}_\ell\\\hline \mathbf{0}\\\hline \mathbf{0}\end{array}\right)\mathbf{d} + \left(\begin{array}{c}\mathbf{0}\\\hline -\mathbf{H}\\\hline \mathbf{0}\\\hline \lfloor\frac{q}{2}\rfloor\mathbf{I}_{\ell k}\end{array}\right)\mathsf{bin}(\mathbf{c}_2) + \left(\begin{array}{c|c|c}\mathbf{B}^\top & \mathbf{I}_{m+\ell} & \mathbf{0}\\\hline \mathbf{G}^\top & & \\\hline \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}} & \mathbf{C}^\top\\\cline{3-3} \multicolumn{2}{c|}{} & \hat{\mathbf{G}}^\top \end{array}\right.\left.\begin{array}{c}\\\\\hline\\\hline \mathbf{I}_{m+\ell k}\end{array}\right)\left(\begin{array}{c}\mathbf{s}\\\mathbf{e}_1\\\mathbf{e}_2\\\hat{\mathbf{s}}\\\hat{\mathbf{e}}_1\\\hat{\mathbf{e}}_2\end{array}\right) = \left(\begin{array}{c}\mathbf{c}_1\\\hline \mathbf{0}^\ell\\\hline \hat{\mathbf{c}}_1\\\hline \hat{\mathbf{c}}_2\end{array}\right).$$

For simplicity, we define $n_1 = 2m + \ell + \ell k$ and $m_1 = 2m + 2n + \ell + \ell k$. In the above unified equation, let $\mathbf{F}_1 \in \mathbb{Z}_q^{n_1\times\ell}$, $\mathbf{F}_2 \in \mathbb{Z}_q^{n_1\times\ell k}$, and $\mathbf{F}_3 \in \mathbb{Z}_q^{n_1\times m_1}$ be the matrices associated with $\mathbf{d}$, $\mathsf{bin}(\mathbf{c}_2)$, and $\mathbf{e} = \left(\mathbf{s}\|\mathbf{e}_1\|\mathbf{e}_2\|\hat{\mathbf{s}}\|\hat{\mathbf{e}}_1\|\hat{\mathbf{e}}_1\right) \in [-b,b]^{m_1}$, respectively. Let $\mathbf{c} = \left(\mathbf{c}_1\|\mathbf{0}^\ell\|\hat{\mathbf{c}}_1\|\hat{\mathbf{c}}_2\right) \in \mathbb{Z}_q^{n_1}$, then the equation becomes:

$$\mathbf{F}_1\cdot\mathbf{d} + \mathbf{F}_2\cdot\mathsf{bin}(\mathbf{c}_2) + \mathbf{F}_3\cdot\mathbf{e} = \mathbf{c} \bmod q.$$

We then extend $\mathsf{bin}(\mathbf{c}_2) \in \{0,1\}^{\ell k}$ to vector $\mathsf{bin}^*(\mathbf{c}_2) \in \mathsf{B}_{2\ell k}$, and apply Lemma 6 to vector $\mathbf{e}$ to obtain $\mathbf{e}^* \in \mathsf{B}_{3m_1\delta_b}$. Furthermore, let $\mathbf{y}^* = \left(\mathbf{d}^*\|\mathsf{bin}^*(\mathbf{c}_2)\|\mathbf{e}^*\right)$, and $\mathbf{F}^* = \left[\mathbf{F}_1|\mathbf{0}^{n_1\times\ell}|\mathbf{F}_2|\mathbf{0}^{n_1\times nk}|\mathbf{F}_3\cdot\mathbf{H}^*_{m_1,b}\right] \in \mathbb{Z}_q^{n_1\times(2\ell+2\ell k+3m_1\delta_b)}$, then we have:

$$\mathbf{F}^*\cdot\mathbf{y}^* = \mathbf{c} \bmod q. \tag{6}$$

In the last step of our transformations, we let $L = (2\ell+2)3m\delta_\beta + 2\ell + 2\ell k + 3m_1\delta_b$ and $D = n + n_1$, and define matrix $\mathbf{P} = \left(\begin{array}{c|c}\mathbf{A}^* & \mathbf{0}\\\hline \mathbf{0} & \mathbf{F}^*\end{array}\right) \in \mathbb{Z}_q^{D\times L}$, vector

$\mathbf{x} = \begin{pmatrix} \mathbf{z}^* \\ \mathbf{y}^* \end{pmatrix} \in \{-1, 0, 1\}^L$, vector $\mathbf{v} = \begin{pmatrix} \mathbf{u} \\ \mathbf{c} \end{pmatrix} \in \mathbb{Z}_q^D$.
Equations (5) and (6) are now unified as:

$$\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q. \tag{7}$$

Having obtained the desired equation (7), we now specify the set $\mathsf{VALID}$ to which $\mathbf{x}$ belongs, the set $\mathcal{S}$ and permutations of $L$ elements $\{T_\pi : \pi \in \mathcal{S}\}$ for which the equivalence (4) holds.

- $\mathsf{VALID}$: the set of all vectors $\mathbf{t} \in \{-1, 0, 1\}^L$ having the form:

$$\mathbf{t} = \big(\mathbf{t}_1 \| \mathbf{t}_2 \| g_1 \cdot \mathbf{x}_2 \| \ldots \| g_{2\ell} \cdot \mathbf{t}_2 \| \mathbf{g} \| \mathbf{t}_3 \| \mathbf{t}_4\big)$$

  for some $\mathbf{t}_1, \mathbf{t}_2 \in \mathsf{B}_{3m\delta_\beta}$, $\mathbf{g} = (g_1, \ldots, g_{2\ell}) \in \mathsf{B}_{2\ell}$, $\mathbf{t}_3 \in \mathsf{B}_{2\ell k}$, $\mathbf{t}_4 \in \mathsf{B}_{3m_1\delta_b}$.
- $\mathcal{S} = \mathcal{S}_{3m\delta_\beta} \times \mathcal{S}_{3m\delta_\beta} \times \mathcal{S}_{2\ell} \times \mathcal{S}_{2\ell k} \times \mathcal{S}_{3m_1\delta_b}$.
- For $\pi = (\phi, \psi, \tau, \sigma, \eta) \in \mathcal{S}$ and $\mathbf{w} = \big(\hat{\mathbf{w}} \| \tilde{\mathbf{w}} \| \mathbf{w}_1 \| \ldots \| \mathbf{w}_{2\ell} \| \bar{\mathbf{w}} \| \ddot{\mathbf{w}} \| \breve{\mathbf{w}}\big) \in \mathbb{Z}_q^L$, where $\hat{\mathbf{w}}, \tilde{\mathbf{w}}, \mathbf{w}_1, \ldots, \mathbf{w}_{2\ell} \in \mathbb{Z}_q^{3m\delta_\beta}$, $\bar{\mathbf{w}} \in \mathbb{Z}_q^{2\ell}$, $\ddot{\mathbf{w}} \in \mathbb{Z}_q^{2\ell k}$, $\breve{\mathbf{w}} \in \mathbb{Z}_q^{3m_1\delta_b}$, we define:

$$T_\pi(\mathbf{w}) = \big(\phi(\hat{\mathbf{w}}) \| \psi(\tilde{\mathbf{w}}) \| \psi(\mathbf{w}_{\tau(1)}) \| \ldots \| \psi(\mathbf{w}_{\tau(2\ell)}) \| \tau(\bar{\mathbf{w}}) \| \sigma(\ddot{\mathbf{w}}) \| \eta(\breve{\mathbf{w}}) \ \big)$$

  as the permutation that transforms $\mathbf{w}$ as follows:
  1. It rearranges the order of the $2\ell$ blocks $\mathbf{w}_1, \ldots, \mathbf{w}_{2\ell}$ according to $\tau$.
  2. It then permutes block $\hat{\mathbf{w}}$ according to $\phi$, blocks $\tilde{\mathbf{w}}$, $\{\mathbf{w}_i\}_{i=1}^{2\ell}$ according to $\psi$, block $\bar{\mathbf{w}}$ according to $\tau$, block $\ddot{\mathbf{w}}$ according to $\sigma$, and block $\breve{\mathbf{w}}$ via $\eta$.

By inspection, it can be seen that

$$\mathbf{x} = \big(\mathbf{z}_1^* \| \mathbf{z}_2^* \| \mathbf{d}_1 \cdot \mathbf{z}_2^* \| \ldots \| \mathbf{d}_{2\ell} \cdot \mathbf{z}_2^* \| \mathbf{d}^* \| \mathsf{bin}^*(\mathbf{c}_2) \| \mathbf{e}^*\big) \in \mathsf{VALID},$$

and that the property (4) is satisfied, as desired. As a result, we can obtain a $\mathsf{sZKAoK}$ for $\mathrm{R}_{\mathrm{gsmdo}}$ by running the protocol in Figure 1 with common input $(\mathbf{P}, \mathbf{v})$ and prover's input $\mathbf{x}$.

Putting everything together, we have the following theorem.

**Theorem 1.** *There exists a Stern-type* $\mathsf{ZKAoK}$ *for the relation* $\mathrm{R}_{\mathrm{gsmdo}}$ *with perfect completeness, soundness error* $2/3$*, and communication cost* $\mathcal{O}(L \log q)$*. In particular:*

- *There exists an efficient simulator that, on input* $(\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^{\ell}, \mathbf{B}, \mathbf{C}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{u}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$*, outputs an accepted transcript which is statistically close to that produced by the real prover.*
- *There exists an efficient knowledge extractor that, on input a commitment* CMT *and* 3 *valid responses* $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ *to all* 3 *possible values of the challenge* $Ch$*, outputs a tuple* $(\mathbf{d}', \mathbf{z}', \mathbf{s}', \hat{\mathbf{s}}', \mathbf{e}_1', \hat{\mathbf{e}}_1', \mathbf{e}_2', \hat{\mathbf{e}}_2', \mathbf{c}_2')$ *such that:*

$$\big((\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^{\ell}, \mathbf{B}, \mathbf{C}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{u}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2), \mathbf{d}', \mathbf{z}', \mathbf{s}', \hat{\mathbf{s}}', \mathbf{e}_1', \hat{\mathbf{e}}_1', \mathbf{e}_2', \hat{\mathbf{e}}_2', \mathbf{c}_2'\big) \in \mathrm{R}_{\mathrm{gsmdo}}.$$

The proof of Theorem 1 is straightforward. For simulation, we run the simulator of Lemma 5. For extraction, we run the knowledge extractor of Lemma 5, and then "backtrack" the described above transformations to obtain a satisfying witness for $\mathrm{R}_{\mathrm{gsmdo}}$. We thus omit the details.

## 4 A GS-MDO Scheme based on Lattice Assumptions

Our scheme is described and analyzed in the model of Sakai *et al.* [42], which is described in Section 2.3.

Our GS-MDO scheme builds on the Ling *et al.* [34] group signature, which is recalled in Appendix C.1. In order to enable message-dependent openings, we add an encryption layer to the previous scheme using an IBE where the signed message serves as the receiver's identity. The *admitter*, which holds the master secret key for this IBE, is able to derive a message-specific token consisting of an IBE private key for this "identity". By itself, this information is insufficient to open the signature as it uncovers a second ciphertext embedded in the message space of the initial encryption layer. At the same time, the opening authority only has access to the external encryption layer which prevents it from identifying the signer without the message-specific token.

Now, the challenge is to prove that the entire double-encryption process was conducted properly while proving the knowledge of a Boyen signature at the same time. As demonstrated in Section 3, we solve this challenge by leveraging the properties of Stern-like protocols [44] and translating the statements to be proved so as to apply the technique of Section 3.

To encrypt the user's identity $d \in \{0,1\}^\ell$, we apply a multi-bit variant of the dual Regev system [20] and obtain a first-layer encryption

$$(\mathbf{c}_1, \mathbf{c}_2) = \big(\mathbf{B}^T\mathbf{s} + \mathbf{e}_1, \mathbf{G}^T\mathbf{s} + \mathbf{e}_2 + \lfloor q/2 \rfloor \cdot \mathsf{bin}(d)\big),$$

where $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ is the master public key of the underlying IBE, $\mathbf{e}_1, \mathbf{e}_2$ are small noise vectors and $\mathbf{G} \in \mathcal{H}_1(\mathsf{ovk}) \in \mathbb{Z}_q^{n\times \ell}$ is derived by hashing a one-time signature verification key (recall that, as in [34], we achieve anonymity in the CCA2 sense by applying the CHK paradigm [14] using $\mathsf{ovk}$ as the receiver's identity). Then, we use a second IBE layer to encrypt the binary decomposition of $\mathbf{c}_2 \in \mathbb{Z}_q^\ell$. In this second IBE instance, we use a matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ and compute

$$(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2) = \big(\mathbf{C}^T\hat{\mathbf{s}} + \hat{\mathbf{e}}_1, \hat{\mathbf{G}}^T\hat{\mathbf{s}} + \hat{\mathbf{e}}_2 + \lfloor q/2 \rfloor \cdot \mathsf{bin}(\mathbf{c}_2)\big),$$

for suitable noise vectors $\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2$ and where $\hat{\mathbf{G}} = \mathcal{H}_2(M) \in \mathbb{Z}_q^{n\times \ell\lceil \log q\rceil}$ is an IBE public key obtained by hashing the "identity" $M$. (Note that the two IBE layers use distinct random oracles $\mathcal{H}_1$ and $\mathcal{H}_2$.)

Now, the problem is to demonstrate the proper computation of $(\mathbf{c}_1, \mathbf{c}_2)$ and $(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$. This can be achieved by proving knowledge of $\mathsf{bin}(\mathbf{c}_2) \in \{0,1\}^{\ell\lceil \log q\rceil}$, $\mathbf{s}, \hat{\mathbf{s}} \in \mathbb{Z}^n$, $\mathbf{e}_1, \hat{\mathbf{e}}_1 \in \mathbb{Z}^m$, $\mathbf{e}_2 \in \mathbb{Z}^\ell$, $\mathbf{e}_2 \in \mathbb{Z}^{\ell\lceil \log q\rceil}$ satisfying:

$$\left(\begin{array}{c|c|c|c|c|c|c|c}
\mathbf{B}^T & \mathbf{I}_m & \mathbf{0} & & & & \mathbf{0} & \mathbf{0} \\ \hline
-\mathbf{G}^T & \mathbf{0} & -\mathbf{I}_\ell & & & & \mathbf{H} & -\lfloor q/2 \rfloor \cdot \mathbf{I}_\ell \\ \hline
 & & & \mathbf{C}^T & \mathbf{I}_m & & \mathbf{0} & \mathbf{0} \\ \hline
 & & & \hat{\mathbf{G}}^T & & \mathbf{I}_{\ell\lceil \log q\rceil} & \lfloor q/2 \rfloor \cdot \mathbf{I}_{\ell\lceil \log q\rceil} & \mathbf{0}
\end{array}\right) \cdot \begin{pmatrix} \mathbf{s} \\ \mathbf{e}_1 \\ \mathbf{e}_2 \\ \hat{\mathbf{s}} \\ \hat{\mathbf{e}}_1 \\ \hat{\mathbf{e}}_2 \\ \mathsf{bin}(\mathbf{c}_2) \\ \mathsf{bin}(d) \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{0}_\ell \\ \hat{\mathbf{c}}_1 \\ \hat{\mathbf{c}}_2 \end{pmatrix},$$

where $\mathbf{H}$ is defined as in Section 3. The second and fourth block relations ensure that that $\mathbf{c}_2$ is the message encrypted by $\hat{\mathbf{c}}_2$ while this hidden $\mathbf{c}_2$ encrypts $\mathsf{bin}(d)$.

We are left with arguing knowledge of a Boyen signature on $\mathsf{bin}(d) \in \{0,1\}^\ell$, which can be achieved as in [34].

### 4.1 Description of the Scheme

The parameters are set in such a way that the Boyen signature and the GPV IBE scheme function properly and are secure. Let $n = \mathcal{O}(\lambda)$ be the lattice parameter, $N = 2^\ell = \mathsf{poly}(\lambda)$ be the number of group members, $q = \mathcal{O}(\ell \cdot n^2)$ be a prime modulus, $\beta = \widetilde{\mathcal{O}}(\sqrt{\ell n})$ be the infinity norm bound for signatures generated by Boyen's scheme [9], and $b$ such that $q/b = \ell \cdot \widetilde{\mathcal{O}}(n)$ be the infinity norm bound for $\mathsf{LWE}$ noises sampled from error distribution $\chi$.

**$\mathsf{Keygen}(1^\lambda, 1^N)$:** This algorithm performs the following steps:

1. Generate a verification key $(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}) \in (\mathbb{Z}_q^{n\times m})^{\ell+2} \times \mathbb{Z}_q^n$ and a private key $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ for Boyen's signature scheme.
   Then for each $d \in \{0, \ldots, 2^\ell - 1\}$, define the corresponding private key $\mathbf{gsk}[d] = (\mathbf{v}_{d,1}^T \mid \mathbf{v}_{d,2}^T)^T \in \mathbb{Z}^{2m}$ to be the Boyen's signature for the message $\mathsf{bin}(d) = (d_1, \ldots, d_\ell) \in \{0,1\}^\ell$ using the trapdoor $\mathbf{T_A}$.
2. Generate two encryption and decryption key pairs for the GPV-IBE scheme: the matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ along with its trapdoor basis $\mathbf{T_B} \in \mathbb{Z}^{m\times m}$ and the matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ with its trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m\times m}$ using the $\mathsf{GenTrap}$ algorithm from Gentry *et al.* [20] described in Lemma 3.
3. Select a strong one-time signature $\Pi^{\mathsf{OTS}} = (\mathsf{OKeygen}, \mathsf{OSign}, \mathsf{OVer})$ and hash functions $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{Z}_q^{n\times\ell}$, $\mathcal{H}_2 : \{0,1\}^* \to \mathbb{Z}_q^{n\times\ell\lceil\log q\rceil}$.
4. Output $\mathsf{ok} = \mathbf{T_B}$, $\mathsf{msk_{ADM}} = \mathbf{T_C}$, $\mathbf{gsk} = \big(\mathbf{gsk}[d]\big)_{d=0}^{N-1}$ and

$$\mathsf{gpk} = \big\{\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^\ell, \mathbf{u}, \mathbf{B}, \mathbf{C}, \Pi^{\mathsf{OTS}}, \mathcal{H}_1, \mathcal{H}_2\big\},$$

**$\mathsf{Sign}(\mathsf{gpk}, \mathbf{gsk}[d], M)$:** To sign $M$ using a group private key $\mathbf{gsk}[d]$,

1. Generate a key pair $(\mathsf{ovk}, \mathsf{osk}) \leftarrow \mathsf{OKeygen}(1^\lambda)$ for the signature $\Pi^{\mathsf{OTS}}$.
2. Encrypt the message $d$ with respect to the "identity" $\mathsf{ovk}$ using the GPV IBE [20]. Namely, let $\mathbf{G} = \mathcal{H}_1(\mathsf{ovk}) \in \mathbb{Z}_q^{n\times\ell}$. Sample $\mathbf{s} \hookleftarrow \chi^n; \mathbf{e}_1 \hookleftarrow \chi^m; \mathbf{e}_2 \hookleftarrow \chi^\ell$, and compute the ciphertext

$$\big(\mathbf{c}_1 = \mathbf{B}^T\mathbf{s} + \mathbf{e}_1, \mathbf{c}_2 = \mathbf{G}^T\mathbf{s} + \mathbf{e}_2 + \lfloor q/2\rfloor \cdot \mathsf{bin}(d)\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^\ell.$$

3. Using the GPV IBE again, encrypt the ciphertext $\mathbf{c}_2$ w.r.t the "identity" $M$. In other words, let $\hat{\mathbf{G}} = \mathcal{H}_2(M) \in \mathbb{Z}_q^{n\times\ell\lceil\log q\rceil}$, then sample $\hat{\mathbf{s}} \hookleftarrow \chi^n; \hat{\mathbf{e}}_1 \hookleftarrow \chi^m, \hat{\mathbf{e}}_2 \hookleftarrow \chi^{\ell\lceil\log q\rceil}$ and compute the ciphertext

$$\big(\hat{\mathbf{c}}_1 = \mathbf{C}^T\hat{\mathbf{s}} + \hat{\mathbf{e}}_1, \hat{\mathbf{c}}_2 = \hat{\mathbf{G}}^T\hat{\mathbf{s}} + \hat{\mathbf{e}}_2 + \lfloor q/2\rfloor \cdot \mathsf{bin}(\mathbf{c}_2)\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{\ell\lceil\log q\rceil}.$$

4. Generate a NIZKAoK $\Pi$ to prove the possession of a valid message-signature pair $(d, \mathbf{z})$ for Boyen's signature, and that $(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$ is a correct encryption of $\mathbf{c}_2$ under the identity $M$, where $(\mathbf{c}_1, \mathbf{c}_2)$ is a correct encryption of $\mathbf{d} = \mathsf{bin}(d)$ under the identity $\mathsf{ovk}$. To do this, run

the interactive argument system for the relation $\mathrm{R}_{\mathrm{gsmdo}}$ in Section 3 with public input $(\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^{\ell}, \mathbf{B}, \mathbf{C}, \mathbf{G}, \hat{\mathbf{G}}, \mathbf{u}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$ and prover's input $(\mathbf{d}, \mathbf{z}, \mathbf{s}, \hat{\mathbf{s}}, \mathbf{e}_1, \hat{\mathbf{e}}_1, \mathbf{e}_2, \hat{\mathbf{e}}_2, \mathbf{c}_2)$.

The protocol is repeated $t = \omega(\log n)$ times to get a negligible soundness error, and then made non-interactive using the Fiat-Shamir heuristic, which gives $\Pi = \big(\{\mathsf{Comm}_j\}_{j=1}^{t}, \mathsf{Chall}, \{\mathsf{Resp}_j\}_{j=1}^{t}\big)$, where

$$\mathsf{Chall} = \mathcal{H}(M, \mathsf{ovk}, \{\mathsf{Comm}_j\}_{j=1}^{t}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2) \in \{1,2,3\}^t.$$

5. Compute a one-time signature $sig = \mathsf{OSign}(\mathsf{osk}; \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \Pi)$.
6. Output $\Sigma = \big(\mathsf{ovk}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \Pi, sig\big)$.

**Verify**$(\mathsf{gpk}, M, \Sigma)$**:** $\Sigma = \big(\mathsf{ovk}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \Pi, sig\big)$ is verified w.r.t. $M$ as follows:

1. If $\mathsf{OVer}(\mathsf{ovk}; sig; \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \Pi) = 0$, return 0.
2. Verify the validity of the proof $\Pi$, if it fails, return 0.
3. If everything went correctly, then return 1.

**TrapGen**$(\mathsf{gpk}, \mathsf{msk}_{\mathsf{ADM}}, M)$**:** To generate a token $\mathrm{t}_M$.

1. If a token for a message $M$ was already queried, answer consistently.
2. Otherwise, derive a key for the identity $M$ using the master secret key $\mathbf{T}_{\mathbf{C}} \in \mathbb{Z}^{m \times m}$. Namely compute $\hat{\mathbf{G}} = \mathcal{H}_2(M)$, then using $\mathsf{SamplePre}$, compute a small-norm matrix $\mathbf{E}_M \in \mathbb{Z}^{m \times \ell \lceil \log q \rceil}$ such that $\mathbf{C} \cdot \mathbf{E}_M = \hat{\mathbf{G}}$.
3. Output $\mathrm{t}_M = \mathbf{E}_M$.

**Open**$(\mathsf{gpk}, \mathsf{ok}, \mathrm{t}_M, \Sigma, M)$**:** To open $\Sigma(\mathsf{ovk}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \Pi, sig\big)$ using the opening key $\mathsf{ok}$ and the token for the message $\mathrm{t}_M$, do the following:

1. Decrypt $(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$ using $\mathrm{t}_M$: $\mathbf{c}_2 = \mathbf{H} \cdot \big\lfloor (\hat{\mathbf{c}}_2 - {\mathrm{t}_M}^T \cdot \hat{\mathbf{c}}_1) \cdot (q/2) \big\rceil$.
2. Decrypt $(\mathbf{c}_1, \mathbf{c}_2)$ using $\mathsf{ok} = \mathbf{T}_{\mathbf{B}} \in \mathbb{Z}^{m \times m}$, namely compute $\mathbf{G} = \mathcal{H}_1(\mathsf{ovk})$, and using $\mathsf{SamplePre}$ to get a short-norm matrix $\mathbf{F} \in \mathbb{Z}^{m \times \ell}$ such that $\mathbf{B} \cdot \mathbf{F} = \mathbf{G}$, and finally compute

$$d = \big(1 \mid 2 \mid 4 \mid \cdots \mid 2^{\ell-1}\big) \cdot \big\lfloor (\mathbf{c}_2 - \mathbf{F}^T \cdot \mathbf{c}_1) \cdot (q/2) \big\rceil .$$

3. Verify that $d$ belongs to a valid user, if not return $\perp$, otherwise return $d$.

### 4.2 Security

The security of the above construction has been proven in the ROM under LWE and SIS assumptions as evidenced in the following theorems. The proofs of Theorems 2, 3 and 4 are available in Appendix A.

**Theorem 2.** *In the random oracle model, the above group signature scheme is fully traceable under the assumption that the* SIS *problem is hard.*

**Theorem 3.** *The above group signature scheme is fully anonymous against the admitter under the* LWE *assumption, and assuming that the one-time signature scheme* $\Pi^{\mathsf{OTS}}$ *is strongly unforgeable.*

**Theorem 4.** *The above group signature scheme is fully anonymous against the opener under the* LWE *assumption.*

## Acknowledgements

The first author was funded by the "Programme Avenir Lyon Saint-Etienne de l'Université de Lyon" in the framework of the programme "Investissements d'Avenir" (ANR-11-IDEX-0007). Khoa Nguyen was supported by the "Singapore Ministry of Education under Research Grant MOE2013-T2-1-041".

## References

1. M. Abdalla and B. Warinschi. On the minimal assumptions of group signature schemes. In *ICICS 2004*, volume 3269 of *LNCS*, pp. 1–13. Springer, 2004.
2. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *STACS'09*, 2009.
3. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Crypto'00*, pp. 255–270, 2000.
4. W. Banaszczyk. New bounds in some transference theorems in the geometry of number. *Mathematische Annalen*, 296:625–635, 1993.
5. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Eurocrypt'03*, number 2656 in LNCS, pp. 614–629, 2003.
6. F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *Asiacrypt'14*, number 8873 in LNCS, pp. 551–572, 2014.
7. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Crypto 2004*, volume 3152 of *LNCS*, pp. 41–55. Springer, 2004.
8. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Crypto 2001*, volume 2139 of *LNCS*, pp. 213–229. Springer, 2001.
9. X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *PKC 2010*, LNCS, pp. 499–517. Springer, 2010.
10. X. Boyen and B. Waters. Compact group signatures without random oracles. In *Eurocrypt'06*, volume 4004 of *LNCS*, pp. 427–444. Springer, 2006.
11. X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC 2010*, volume 4450 of *LNCS*, pp. 1–15. Springer, 2007.
12. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. On the classical hardness of learning with errors. In *STOC'13*, pp. 575–584. ACM, 2013.
13. E. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung. Design validations for discrete logarithm based signature schemes. In *PKC 2000*, volume 1751 of *LNCS*, pp. 276–292. Springer, 2000.
14. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Eurocrypt 2004*, volume 3027 of *LNCS*, pp. 207–222. Springer, 2004.
15. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Eurocrypt'10*, volume 6110 of *LNCS*, pp. 523–552. Springer, 2010.
16. D. Chaum and E. Van Heyst. Group signatures. In *Eurocrypt'91*, volume 547 of *LNCS*, pp. 257–265. Springer, 1991.
17. Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *Crypto 1989*, volume 435 of *LNCS*, pp. 307–315. Springer.
18. M. F. Ezerman, H. T. Lee, S. Ling, K. Nguyen, and H. Wang. A provably secure group signature scheme from code-based assumptions. In *Asiacrypt'15*, LNCS. Springer, 2015. `http://eprint.iacr.org/`.

19. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pp. 169–178, 2009.
20. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, pp. 197–206. ACM, 2008.
21. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC 1985*, pp. 291–304. ACM, 1985.
22. S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *Asiacrypt 2010*, volume 2647 of *LNCS*, pp. 395–412, 2010.
23. J. Groth. Evaluating Security of Voting Schemes in the Universal Composability Framework. In *ACNS*, volume 3089 of *LNCS*, pp. 46–60. Springer, 2004.
24. J. Groth. Fully anonymous group signatures without random oracles. In *Asiacrypt 2007*, volume 4833 of *LNCS*, pp. 164–180. Springer, 2007.
25. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt 2008*, volume 4965 of *LNCS*, pp. 415–432. Springer, 2008.
26. A. Jain, S. Krenn, K. Pietrzak, and A. Tentes. Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise. In *Asiacrypt'12*, volume 7658 of *LNCS*, pp. 663–680. Springer, 2012.
27. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Asiacrypt'08*, volume 5350 of *LNCS*, pp. 372–389. Springer, 2008.
28. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *Eurocrypt'04*, volume 3027 of *LNCS*, pp. 571–589. Springer, 2004.
29. F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *Asiacrypt 2013*, volume 8270 of *LNCS*, pp. 41–61. Springer, 2013.
30. B. Libert and M. Joye. Group signatures with message-dependent opening in the standard model. In *CT-RSA'14*, 2014.
31. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. Cryptology ePrint Archive: Report 2016/101, January 2016.
32. 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*, LNCS. Springer, 2016. To appear.
33. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC 2013*, 2013.
34. S. Ling, K. Nguyen, and H. Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based. In *PKC'15*. Springer, springer edition, 2015.
35. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt'12*, volume 7237 of *LNCS*, pp. 700–718. Springer, 2012.
36. P. Q. Nguyen, J. Zhang, and Z. Zhang. Simpler efficient group signatures from lattices. In *PKC 2015*, volume 9020 of *LNCS*, pp. 401–426. Springer, 2015.
37. K. Ohara, Y. Sakai, K. Emura, and G. Hanaoka. A group signature scheme with unbounded message-dependent opening. In *AsiaCCS'13*, 2013.
38. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC'09*, pp. 333–342. ACM, 2009.
39. C. Peikert. A decade of lattice cryptography. Cryptology ePrint Archive: Report 2015/939, September 2015.
40. C. Peikert and V. Vaikuntanathan. Non-interactive statistical zero-knowledge proofs for lattice problems. In *Crypto'08*, LNCS, pp. 536–553, 2008.
41. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC'05*, pp. 84–93. ACM, 2005.

42. Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote. Group signatures with message-dependent opening. In *Pairing'12*, volume 7708 of *LNCS*, pp. 270–294. Springer, 2012.
43. A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO '84*, volume 196 of *LNCS*, pp. 47–53. Springer, 1985.
44. J. Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

# A Deferred Proofs of the GS-MDO Scheme

## A.1 Proof of Theorem 2: Traceability

*Proof.* Our proof is similar to the proof of traceability in Ling *et al.* scheme [34] since adding an additional encryption layer does not alter the power of the reduction.

Let us assume that there exists a $\mathsf{PPT}$ adversary $\mathcal{A}$ with noticeable advantage $\varepsilon$. By hypothesis, such an adversary can produce a valid signature $(M^\star, \Sigma^\star)$ which opens to $\perp$ or a non-adversarially controlled user who did not sign $M^\star$. We construct a reduction $\mathcal{B}$ that uses $\mathcal{A}$ to produce a forgery for Boyen's signature with advantage polynomially smaller than $\varepsilon$. Since Boyen's signature is secure under $\mathsf{SIS}$ assumption, this completes our proof.

Setup. The reduction $\mathcal{B}$ is given the verification key $(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u})$ for Boyen's signature. It generates two key pairs $(\mathbf{B}, \mathbf{T_B})$, $(\mathbf{C}, \mathbf{T_C})$ for the GPV IBE scheme and interacts with $\mathcal{A}$ by sending it $\mathsf{gpk} = \big\{\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}, \mathbf{B}, \mathbf{C}\big\}$, $\mathsf{ok} = \mathbf{T_B}$ and $\mathsf{msk}_{\mathsf{ADM}} = \mathbf{T_C}$.

Queries. Algorithm $\mathcal{B}$ initializes $\mathcal{Q}_u \leftarrow \emptyset$ and handles $\mathcal{A}$'s queries as follow:

- *Random Oracle Query.* Queries on $\mathcal{H}, \mathcal{H}_1$ and $\mathcal{H}_2$ are handled by consistently returning uniformly random values in the respective ranges. In the following, we denote by $q_k$ the answer to the $k$-th $\mathcal{H}$-query.
- *Private Key Query.* In order to return a private key $\mathbf{gsk}[d]$ for user $d$, the reduction $\mathcal{B}$ invokes its own signing oracle for Boyen's signature of $d$ : $\mathbf{z}_d$. Then, $\mathcal{B}$ updates $\mathcal{Q}_u \leftarrow \mathcal{Q}_u \cup \{d\}$ and sends $\mathbf{z}_d$ to $\mathcal{A}$. It consistently returns the same response in case a given identity is queried more than once.
- *Signing Query.* On signature query on a message $M$ for user $d$, the reduction $\mathcal{B}$ appeals to the simulator for the NIZKAoK protocol in order to produce a signature $\Sigma = \big(\mathsf{ovk}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \Pi', sig\big)$. Namely, $\mathsf{ovk}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, sig$ are generated as in the real protocol while $\Pi'$ is produced without the witnesses by running the HVZK simulator of the underlying NIZKAoK $t$ times and programming the random oracle $\mathcal{H}$ accordingly. Thus, the HVZK property of the argument system ensures that $\Sigma$ is statistically indistinguishable from a legitimate signature.

Main Reduction. At some point, $\mathcal{A}$ outputs a forgery

$$\Big(M^\star, \Sigma^\star = \big(\mathsf{ovk}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, (\{\mathsf{Comm}_j\}_{j=1}^t, \mathsf{Chall}, \{\mathsf{Resp}_j\}_{j=1}^t), sig\big)\Big)$$

which satisfies the requirements of the traceability game in Definition 5.

The reduction $\mathcal{B}$ then exploits the forgery as follows. We can start by noticing that $\mathcal{A}$ must have queried $\mathcal{H}$ on input $(M, \mathsf{ovk}, \{\mathsf{Comm}_j\}_{j=1}^t, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$ since, otherwise, the probability that $(\mathsf{Chall})_{i=1}^t = \mathcal{H}(M, \mathsf{ovk}, \{\mathsf{Comm}_j\}_{j=1}^t, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$ is at most $3^{-t}$. With probability at least $\varepsilon - 3^{-t}$, there thus exists an index $\kappa^* \leq Q_{\mathcal{H}}$ such that the $\kappa^*$-th Oracle query involves the input $(M, \mathsf{ovk}, \{\mathsf{Comm}_j\}_{j=1}^t, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$.

The reduction $\mathcal{B}$ then sets $\kappa^*$ as the forking point: namely, $\mathcal{B}$ replays $\mathcal{A}$ many times with the same random tape and input as in the original run. In each re-played execution, the first $\kappa^* - 1$ queries receive exactly the same responses $r_1, \ldots, r_{\kappa^*-1}$ but, beyond that point, all following queries are replied with fresh random values $r'_{\kappa^*}, \ldots, r'_{Q_{\mathcal{H}}} \hookleftarrow \mathcal{U}\big(\{1,2,3\}^t\big)$.

The Improved Forking Lemma of Brickell *et al.* [13] ensures that with probability at least $1/2$, $\mathcal{B}$ can obtain a 3-fork on $\mathcal{H}$ involving the same input tuple $(M, \mathsf{ovk}, \{\mathsf{Comm}_j\}_{j=1}^t, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)$ with pairwise distinct outputs $r_{\kappa^*}^{(1)}, r_{\kappa^*}^{(2)}, r_{\kappa^*}^{(3)} \in \{1,2,3\}^t$ after less than $32 \cdot Q_{\mathcal{H}}/(\varepsilon - 3^{-t})$ reruns of $\mathcal{A}$. With probability $1-(7/9)^t$, it can be shown that there exists an index $j \in \{1, \ldots, t\}$ for which the $j$-th bits of $r_{\kappa^*}^{(1)}, r_{\kappa^*}^{(2)}, r_{\kappa^*}^{(3)}$ are $\{r_{\kappa^*,j}^{(1)}, r_{\kappa^*,j}^{(2)}, r_{\kappa^*,j}^{(3)}\} = \{1,2,3\}$. Since the corresponding responses $(\mathsf{Resp}_j^{*(1)}, \mathsf{Resp}_j^{*(2)}, \mathsf{Resp}_j^{*(3)})$ must be valid answers with respect to three distinct challenges for the same commitment $\mathsf{Comm}_j$. $\mathcal{B}$ can now use the knowledge extractor of the underlying argument scheme to obtain witnesses

$$\big(d^\star, \mathbf{z}^\star, \mathbf{s}^\star, \mathbf{e}_2^\star, \hat{\mathbf{s}}^\star, \hat{\mathbf{e}}_1^\star, \hat{\mathbf{e}}_2^\star\big) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^n \times \mathbb{Z}^\ell \times \mathbb{Z}^n \times \mathbb{Z}^m \times \mathbb{Z}^{\ell\lceil \log q \rceil},$$

such that the vector $\mathbf{z}^\star$ is a Boyen signature for the message $d^\star$. Moreover, as required by the traceability game, we must have $d^\star \notin \mathcal{Q}_u$, which implies that $(d^\star, \mathbf{z}^\star)$ is a forgery for the Boyen signature. Finally, we can notice that, if $\mathcal{A}$ has non-negligible success probability and runs in polynomial time, the same holds for $\mathcal{B}$. □

### A.2 Proof of Theorem 3: Anonymity against the Admitter

*Proof.* In this proof, the attacker is given the private key of the external encryption layer at the outset of the game. Since this key is also known to the challenger, we can proceed as if this encryption layer did not exist in our proof. The proof thus remains very similar to the proof of anonymity in [34]. Namely, we will proceed via a sequence of games such that $\mathbf{Adv}_{\mathcal{A}}(\mathsf{Game}\ 0) = \varepsilon$ in the first game and $\mathbf{Adv}_{\mathcal{A}}(\mathsf{Game}\ 6) = 0$ in the last one. The indistinguishability of all adjacent games is supported by the strong unforgeability of $\Pi^{\mathsf{OTS}}$, the statistical ZK property of the argument system and the hardness of $\mathsf{LWE}$. For each $i$, we denote by $S_i$ the event that the adversary wins in Game $i$.

**Game** 0**.** This is the real $\mathsf{CCA}$ anonymity game against the admitter: the challenger runs $\mathsf{Keygen}(1^\lambda, 1^N)$ to obtain $(\mathsf{gpk}, \mathsf{ok} = \mathbf{T_B}, \mathsf{msk}_{\mathsf{ADM}} = \mathbf{T_C}, \mathbf{gsk})$, and then sends $(\mathsf{gpk}, \mathsf{msk}_{\mathsf{ADM}}, \mathbf{gsk})$ to $\mathcal{A}$. Using the decryption keys $\mathbf{T_B}$ and $\mathbf{T_C}$, the challenger can answer all signature opening queries. In the challenge phase, $\mathcal{A}$ sends a

message $M$ along with two identities $d_0 \neq d_1 \in \{0,1\}^\ell$. The challenger sends back a challenge signature $\Sigma^\star = \big(\mathsf{ovk}^\star, \mathbf{c}_1^\star, \hat{\mathbf{c}}_1^\star, \hat{\mathbf{c}}_2^\star, \Pi^\star, sig^\star\big) \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathbf{gsk}[d_b], M)$ for $b \in \{0,1\}$. After further queries, the adversary finally outputs a bit $b' \in \{0,1\}$ and wins if $b = b'$. We denote by $S_0$ the latter event. By hypothesis, $\mathbf{Adv}_{\mathcal{A}}(\mathsf{Game}\ 0) = |\Pr[S_0] - 1/2| = \varepsilon$.

**Game** 1**.** In this game, we make a small modification with respect to Game 0. At the outset of the game, the one-time signature key pair $(\mathsf{ovk}^\star, \mathsf{osk}^\star)$ is generated by the challenger. During the game, if $\mathcal{A}$ makes a request to open a valid signature of the form $\big(\mathsf{ovk}^\star, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \Pi, sig\big)$, the challenger outputs a random bit and aborts. However, the strong unforgeability of the one-time signature $\Pi^{\mathsf{OTS}}$ ensures that this is unlikely to happen. Indeed, before the challenge phase, $\mathsf{ovk}^\star$ is independent of $\mathcal{A}$'s view, and the probability that $\mathsf{ovk}^\star$ appears in $\mathcal{A}$'s queries is negligible. Furthermore, after seeing the challenge signature $\Sigma^\star$, if $\mathcal{A}$ comes up with a valid signature of the form $\big(\mathsf{ovk}^\star, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \Pi, sig\big)$, then $sig$ must be a forged one-time signature, which defeats the strong unforgeability of $\Pi^{\mathsf{OTS}}$. Then, the probability that the challenger aborts in this experiment is negligible and we have $|\Pr[S_1] - \Pr[S_0]| \in \mathsf{negl}(\lambda)$. In the upcoming experiments, we will assume without loss of generality that $\mathcal{A}$ does not requests for opening of valid signature that includes $\mathsf{ovk}^\star$.

**Game** 2**.** We program the random oracle $\mathcal{H}_1$ in the following way. At the beginning of the game, we choose an uniformly random matrix $\mathbf{G}^\star \in \mathbb{Z}_q^{n \times \ell}$, and set $\mathcal{H}_1(\mathsf{ovk}^\star) = \mathbf{G}^\star$. As for other queries, for each fresh $\mathcal{H}_1$ queries on $\mathsf{ovk}$, the challenger samples a small norm matrix $\mathbf{F}_{\mathsf{ovk}} \hookleftarrow D^m_{\mathbb{Z}^\ell,\sigma}$ and programs the oracle to have $\mathcal{H}_1(\mathsf{ovk}) = \mathbf{B} \cdot \mathbf{F}_{\mathsf{ovk}}$ and subsequent queries are answered consistently. At each $\mathcal{H}_1$-query, the chosen matrix $\mathbf{F}_{\mathsf{ovk}}$ is retained for later use. Note that the value $\mathcal{H}_1(\mathsf{ovk})$ is statistically close to the uniform distribution, as in [20]. From the attacker's view, the output distribution of $\mathcal{H}_1$ thus remains the same as in Game 2.

**Game** 3**.** In this game, we change the opening algorithm. At the beginning of the game, instead of generating $\mathbf{B}$ with a trapdoor, the challenger samples a random $\mathbf{B}^\star \in \mathbb{Z}_q^{n \times m}$. We recall that at each fresh queries, we retain the matrix $\mathbf{F}_{\mathsf{ovk}} \in \mathbb{Z}^{m \times \ell}$ for later use. Upon receiving an adversarial query to open a signature $\Sigma = \big(\mathsf{ovk}, \mathbf{c}_1, \hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, \Pi, sig\big)$, the challenger recovers the corresponding small-norm matrix $\mathbf{F}_{\mathsf{ovk}}$ which was defined when $\mathcal{A}$ first queried $\mathcal{H}_1(\mathsf{ovk})$ (we can assume w.l.o.g. that each opening query on $\mathsf{ovk}$ is preceded by hash query on the same $\mathsf{ovk}$). This matrix is used to "decrypt" $(\mathbf{c}_1, \mathbf{c}_2)$ for the corresponding $\mathbf{G} = \mathcal{H}_1(\mathsf{ovk})$. Thanks to Lemma 3, the distribution of $(\mathbf{B}^\star, \mathbf{G})$ is statistically close to that of Game 2. Therefore, Game 2 and Game 3 are statistically indistinguishable and $\Pr[S_3] = \Pr[S_2]$.

**Game** 4**.** We modify the generation of the challenge signature $\Sigma^\star$. Instead of faithfully generating the NIZKAoK $\Pi^\star$, the challenger simulates it without using the witnesses via the HVZK simulator of the underlying argument system and by programming the random oracle $\mathcal{H}$ as in the proof of Theorem 2 (in the unlikely event that $\mathcal{H}$ has to be programmed on an input for which it is already defined,

the challenger aborts the experiment and outputs 1). Thanks to the statistical zero-knowledge property of the argument scheme, the simulated signature $\Pi^\star$ produced in this manner is statistically indistinguishable from a real signature. Game 3 is then statistically indistinguishable from Game 4 as long as the challenger does not fail in the simulation of $\Pi^\star$. However, such a failure event occurs with negligible probability and we have $|\Pr[S_4] - \Pr[S_3]| \in \mathsf{negl}(\lambda)$.

**Game** 5**.** In this game, we change the generation of the ciphertext $(\mathbf{c}_1^\star, \mathbf{c}_2^\star)$ during the challenge phase in step 2 of the Sign algorithm. Namely, instead of using the real encryption algorithm of GPV IBE, the challenger generates a random ciphertext. In other words, the challenger sets

$$\mathbf{c}_1^\star := \mathbf{z}_1, \qquad \mathbf{c}_2^\star := \mathbf{z}_2 + \lfloor q/2 \rfloor \cdot \mathsf{bin}(d_b),$$

for randomly chosen $(\mathbf{z}_1, \mathbf{z}_2) \leftarrow U(\mathbb{Z}_q^m \times \mathbb{Z}_q^\ell)$. The hardness of the LWE problem implies that Game 5 is computationally indistinguishable from Game 4. Indeed, distinguishing between these two experiments would require $\mathcal{A}$ to distinguish $(\mathbf{B}^\star \mid \mathbf{G}^\star)^T\mathbf{s}_0 + (\mathbf{e}_1^T \mid \mathbf{e}_2^T)^T$ from $(\mathbf{z}_1^T|\mathbf{z}_2^T)^T$, which contradicts the decisional LWE assumption. Under the LWE assumption, we have $|\Pr[S_5] - \Pr[S_4]| \in \mathsf{negl}(\lambda)$.

**Game** 6**.** Finally we make a conceptual change on the previous game. Namely we sample random $(\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^\ell$ and set $(\mathbf{c}_1^\star, \mathbf{c}_2^\star)$ in step 2 of the Sign algorithm to be $(\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2)$. Clearly, Game 6 is statistically indistinguishable from Game 5.

Since Game 6 does no longer depend on the bit $b \in \{0, 1\}$ identifying the target signer, the advantage of $\mathcal{A}$ in Game 6 is then zero, which concludes our proof. □

### A.3 Proof of Theorem 4: Anonymity against the Opening Authority

*Proof.* The proof is very similar to the proof of Theorem 3 described in the previous section. We proceed with a sequence of Games that ends in a game where the adversary's advantage is unconditionally zero. The main idea is that in the security game, the adversary has at disposal the secret key of the inside layer of the proof system, meaning that we can view the ciphertext $(\mathbf{c}_1, \mathbf{c}_2)$ in step 2 of the signing algorithm as a plaintext. For each $i$, we call $S_i$ the event that the adversary wins in Game $i$.

**Game** 0**.** This is the actual security game as described in Definition 6. By hypothesis, the advantage of $\mathcal{A}$ in this game is $\varepsilon = |\Pr[S_0] - 1/2|$.

**Game** 1**.** In this game, we program the random oracle $\mathcal{H}_2$ in the following way. At the beginning of the game, we choose an uniform matrix $\hat{\mathbf{G}}^\star \leftarrow U(\mathbb{Z}_q^{n \times \ell\lceil \log q \rceil})$, and in the challenge phase, we set $\mathcal{H}_2(M^\star) = \hat{\mathbf{G}}_0^\star$. From the attacker's view, the distribution of $\hat{\mathbf{G}}_0^\star$ is as in Game 0. As for other $\mathcal{H}_2$-queries, on a fresh query on the input $M$, the challenger samples a small-norm matrix $\mathbf{E}_M \hookleftarrow D_{\mathbb{Z}^n,\sigma}^{\ell\lceil \log q \rceil}$, sets $\mathcal{H}_2(M) = \mathbf{C} \cdot \mathbf{E}_M \bmod q$ and retains the chosen matrix $\mathbf{E}_M$ for later use.

We notice that the attacker is not allowed to query a token for $M^\star$. Then, we remark that the value of $\mathcal{H}_2(M)$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n\lceil \log q \rceil}$ as in [20]. This game is therefore indistinguishable from Game 0: $|\Pr[S_1] - \Pr[S_0]| \in \mathsf{negl}(\lambda)$.

**Game** 2. In this game, we change the way the challenger handles token queries. First, we start by picking a random matrix $\mathbf{C}^\star \leftarrow U(\mathbb{Z}_q^{n\times m})$ at the outset of the game, without generating a GPV trapdoor for it. Then, in order to answer token queries on a message $M$, the challenger recalls the matrix $\mathbf{E}_M$, and hands it as $\mathrm{t}_M$. Thanks to Lemma 3, the distribution of $\mathrm{t}_M$ remains the same as in the real game. Then, Game 1 remains statistically indistinguishable from Game 2.

**Game** 3. Here, in the challenge phase, we modify the generation of the challenge signature $\Sigma^\star$. Instead of using the witnesses to generate the NIZK proof $\Pi^\star$, we proceed as in the proof of Theorem 2 using the ZK simulator of the underlying argument system to generate the proof $\Pi'$ of the signature $\Sigma^\star = \big(\mathsf{ovk}^\star, \mathbf{c}_1^\star, \hat{\mathbf{c}}_1^\star, \hat{\mathbf{c}}_2^\star, \Pi', sig^\star\big)$ without using the witness. For the same reason as in the proof of Theorem 3, we have that Game 3 is statistically indistinguishable from Game 2.

**Game** 4. In this game, we change the way the challenge signature $\Sigma^\star$ is generated. Instead of faithfully computing the ciphertext $(\hat{\mathbf{c}}_1^\star, \hat{\mathbf{c}}_2^\star)$ of $\mathsf{bin}(\mathbf{c}_2^\star)$, the challenger samples uniformly random $(\hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{\ell\lceil \log q \rceil}$ and sends

$$\Sigma^\star = \big(\mathsf{ovk}^\star, \mathbf{c}_1^\star, \hat{\mathbf{z}}_2^\star, \hat{\mathbf{z}}_2^\star + \lfloor q/2 \rfloor \cdot \mathsf{bin}(\mathbf{c}_2^\star), \Pi', sig^\star\big),$$

as a challenge signature. This game is computationally indistinguishable from Game 3 from the assumed hardness of the decisional LWE problem. The LWE assumption implies that $|\Pr[S_5] - \Pr[S_4]| \in \mathsf{negl}(\lambda)$.

**Game** 5. Finally, we make a conceptual change in the signature. Instead of sending $(\hat{\mathbf{z}}_1^\star, \hat{\mathbf{z}}_2^\star + \lfloor q/2 \rfloor \mathsf{bin}(\mathbf{c}_2^\star))$ in the challenge signature $\Sigma^\star$, the challenger samples $(\tilde{\mathbf{z}}_1^\star, \tilde{\mathbf{z}}_2^\star) \leftarrow U(\mathbb{Z}_q^n \times \mathbb{Z}_q^{\ell\lceil \log q \rceil})$ and uses them in place of $(\hat{\mathbf{z}}_1^\star, \hat{\mathbf{z}}_2^\star + \lfloor q/2 \rfloor \cdot \mathsf{bin}(\mathbf{c}_2^\star))$ to construct $\Sigma^\star$. Game 5 is indistinguishable from Game 4.

The last game is clearly independent of the challenger's bit $b \in \{0,1\}$ that determines the signer's identity and it follows that $\mathbf{Adv}_{\mathcal{A}}(\text{Game } 5) = 0$. □

## B Proof of Lemma 5

We first restate Lemma 5.

**Lemma 7.** *The protocol in Figure 1 is a* sZKAoK *for the relation* $\mathrm{R}_{\mathrm{abstract}}$ *with perfect completeness, soundness error* $2/3$*, and communication cost* $\widetilde{\mathcal{O}}(L \log q)$*. In particular:*

- *There exists an efficient simulator that, on input* $(\mathbf{P}, \mathbf{v})$*, outputs an accepted transcript which is statistically close to that produced by the real prover.*

– *There exists an efficient knowledge extractor that, on input a commitment* CMT *and* 3 *valid responses* $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ *to all* 3 *possible values of the challenge* $Ch$*, outputs* $\mathbf{x}' \in \mathsf{VALID}$ *such that* $\mathbf{P} \cdot \mathbf{x}' = \mathbf{v} \bmod q$.

*Proof.* It can be checked that the protocol has perfect completeness: If an honest prover follows the protocol, then he always gets accepted by the verifier. It is also easy to see that the communication cost is bounded by $\widetilde{\mathcal{O}}(L \log q)$.

We now will prove that the protocol is a statistical zero-knowledge argument of knowledge for the relation $\mathrm{R}_{\mathrm{abstract}}$.

**Zero-Knowledge Property.** We construct a $\mathsf{PPT}$ simulator $\mathsf{SIM}$ interacting with a (possibly dishonest) verifier $\widehat{\mathcal{V}}$, such that, given only the public input, $\mathsf{SIM}$ outputs with probability negligibly close to $2/3$ a simulated transcript that is statistically close to the one produced by the honest prover in the real interaction.

The simulator first chooses a random $\overline{Ch} \in \{1, 2, 3\}$. This is a prediction of the challenge value that $\widehat{\mathcal{V}}$ will *not* choose.

**Case** $\overline{Ch} = 1$: Using basic linear algebra over $\mathbb{Z}_q$, $\mathsf{SIM}$ computes a vector $\mathbf{x}' \in \mathbb{Z}_q^L$ such that $\mathbf{P} \cdot \mathbf{x}' = \mathbf{v} \bmod q$. Next, it samples $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^L)$, $\pi \hookleftarrow U(\mathcal{S})$, and randomness $\rho_1, \rho_2, \rho_3$ for $\mathsf{COM}$. Then it sends the commitment $\mathrm{CMT} = \big(C_1', C_2', C_3'\big)$ to $\widehat{\mathcal{V}}$, where

$$C_1' = \mathsf{COM}(\pi, \mathbf{P} \cdot \mathbf{r}; \rho_1), \;\; C_2' = \mathsf{COM}(T_\pi(\mathbf{r}); \rho_2), \;\; C_3' = \mathsf{COM}(T_\pi(\mathbf{x}' + \mathbf{r}); \rho_3).$$

Receiving a challenge $Ch$ from $\widehat{\mathcal{V}}$, the simulator responds as follows:

– If $Ch = 1$: Output $\perp$ and abort.
– If $Ch = 2$: Send $\mathrm{RSP} = \big(\pi, \mathbf{x}' + \mathbf{r}, \rho_1, \rho_3\big)$.
– If $Ch = 3$: Send $\mathrm{RSP} = \big(\pi, \mathbf{r}, \rho_1, \rho_2\big)$.

**Case** $\overline{Ch} = 2$: $\mathsf{SIM}$ samples $\mathbf{x}' \hookleftarrow U(\mathsf{VALID})$, $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^L)$, $\pi \hookleftarrow U(\mathcal{S})$, and randomness $\rho_1, \rho_2, \rho_3$ for $\mathsf{COM}$. Then it sends the commitment $\mathrm{CMT} = \big(C_1', C_2', C_3'\big)$ to $\widehat{\mathcal{V}}$, where

$$C_1' = \mathsf{COM}(\pi, \mathbf{P} \cdot \mathbf{r}; \rho_1), \;\; C_2' = \mathsf{COM}(T_\pi(\mathbf{r}); \rho_2), \;\; C_3' = \mathsf{COM}(T_\pi(\mathbf{x}' + \mathbf{r}); \rho_3).$$

Receiving a challenge $Ch$ from $\widehat{\mathcal{V}}$, the simulator responds as follows:

– If $Ch = 1$: Send $\mathrm{RSP} = \big(T_\pi(\mathbf{x}'), T_\pi(\mathbf{r}), \rho_2, \rho_3\big)$.
– If $Ch = 2$: Output $\perp$ and abort.
– If $Ch = 3$: Send $\mathrm{RSP} = \big(\pi, \mathbf{r}, \rho_1, \rho_2\big)$.

**Case** $\overline{Ch} = 3$: $\mathsf{SIM}$ samples $\mathbf{x}' \hookleftarrow U(\mathsf{VALID})$, $\mathbf{r} \hookleftarrow U(\mathbb{Z}_q^L)$, $\pi \hookleftarrow U(\mathcal{S})$, and randomness $\rho_1, \rho_2, \rho_3$ for $\mathsf{COM}$. Then it sends the commitment $\mathrm{CMT} = \big(C_1', C_2', C_3'\big)$ to $\widehat{\mathcal{V}}$, where $C_2' = \mathsf{COM}(T_\pi(\mathbf{r}); \rho_2)$, $C_3' = \mathsf{COM}(T_\pi(\mathbf{x}' + \mathbf{r}); \rho_3)$ as in the previous two cases, while

$$C_1' = \mathsf{COM}(\pi, \mathbf{P} \cdot (\mathbf{x}' + \mathbf{r}) - \mathbf{v}; \rho_1),$$

Receiving a challenge $Ch$ from $\widehat{\mathcal{V}}$, it responds as follows:

– If $Ch = 1$: Send RSP computed as in the case $(\overline{Ch} = 2, Ch = 1)$.
– If $Ch = 2$: Send RSP computed as in the case $(\overline{Ch} = 1, Ch = 2)$.
– If $Ch = 3$: Output $\perp$ and abort.

We observe that, in every case we have considered above, since $\mathsf{COM}$ is statistically hiding, the distribution of the commitment CMT and the distribution of the challenge $Ch$ from $\widehat{\mathcal{V}}$ are statistically close to those in the real interaction. Hence, the probability that the simulator outputs $\perp$ is negligibly close to $1/3$. Moreover, one can check that whenever the simulator does not halt, it will provide an accepted transcript, the distribution of which is statistically close to that of the prover in the real interaction. In other words, we have constructed a simulator that can successfully impersonate the honest prover with probability negligibly close to $2/3$.

**Argument of Knowledge.** Suppose that $\mathrm{RSP}_1 = (\mathbf{t}_x, \mathbf{t}_r, \rho_2, \rho_3)$, $\mathrm{RSP}_2 = (\pi_2, \mathbf{y}, \rho_1, \rho_3)$, $\mathrm{RSP}_3 = (\pi_3, \mathbf{r}_3, \rho_1, \rho_2)$ are 3 valid responses to the same commitment $\mathrm{CMT} = (C_1, C_2, C_3)$, with respect to all 3 possible values of the challenge. The validity of these responses implies that:

$$\begin{cases} \mathbf{t}_x \in \mathsf{VALID}; \\ C_1 = \mathsf{COM}(\pi_2, \mathbf{P} \cdot \mathbf{y} - \mathbf{v}; \rho_1) = \mathsf{COM}(\pi_3, \mathbf{P} \cdot \mathbf{r}_3; \rho_1); \\ C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2) = \mathsf{COM}(T_{\pi_3}(\mathbf{r}_3); \rho_2); \\ C_3 = \mathsf{COM}(\mathbf{t}_x + \mathbf{t}_r; \rho_3) = \mathsf{COM}(T_{\pi_2}(\mathbf{y}); \rho_3). \end{cases}$$

Since $\mathsf{COM}$ is computationally binding, we can deduce that:

$$\mathbf{t}_x \in \mathsf{VALID}; \pi_2 = \pi_3; \mathbf{t}_r = T_{\pi_3}(\mathbf{r}_3); \mathbf{t}_x + \mathbf{t}_r = T_{\pi_2}(\mathbf{y}); \mathbf{P} \cdot \mathbf{y} - \mathbf{v} = \mathbf{P} \cdot \mathbf{r}_3 \bmod q.$$

Let $\mathbf{x}' = \mathbf{y} - \mathbf{r}_3$, then we have $T_{\pi_2}(\mathbf{x}') = \mathbf{t}_x \in \mathsf{VALID}$ which implies that $\mathbf{x}' \in \mathsf{VALID}$. Furthermore, we have $\mathbf{P} \cdot \mathbf{x}' = \mathbf{P} \cdot (\mathbf{y} - \mathbf{r}_3) = \mathbf{v} \bmod q$.

This concludes the proof. □

# C Building Blocks

## C.1 The Ling-Nguyen-Wang Group Signature

We build our scheme on the [34] group signature scheme, which is described as follows.

Let $n$ be the security parameter, and $N = \mathsf{Poly}(n)$ be the maximum expected number of group users. The other parameters are chosen such that Boyen's signature scheme [9] and the GPV IBE scheme [20] function properly and are secure, like in Section 4.1.

We choose hash functions $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{Z}_q^{n \times \ell}$, which goal is to produce $\ell$ random syndromes for the multi-bit version of GPV, and $\mathcal{H} : \{0,1\}^* \to \{1,2,3\}^t$ which goal is to provide challenges for the Stern's protocol in the Fiat-Shamir heuristic, and select a one time signature scheme $\Pi^{\mathrm{OTS}} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$. Let $\chi$ denote the $b$-bounded $\mathsf{LWE}$ distribution over $\mathbb{Z}$.

The group signature is described as follows:

**KeyGen**$(1^n, 1^N)$**:** Given a dimension $n$ and a number of group users $N = 2^\ell$:

1. Generate verification key $(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}) \in (\mathbb{Z}_q^{n\times m})^{\ell+1} \times \mathbb{Z}_q^n$ and signing key $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ for Boyen's signature scheme. Then for each $d = (d_1, \ldots, d_\ell) \in \{0,1\}^\ell$, use $\mathbf{T_A}$ to generates $\mathbf{gsk}[d]$ as Boyen's signature on message $d$.
2. Generate encryption and decryption master keys pair $(\mathbf{B}, \mathbf{T_B}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}^{m\times m}$ for the GPV-IBE scheme.
3. Output

$$\mathsf{gpk} = ((\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}), \mathbf{B}); \quad \mathsf{gmsk} = \mathbf{T_B}; \quad \mathbf{gsk} = \{\mathbf{gsk}[\mathsf{id}]\}_{\mathsf{id}\in\{0,1\}^\ell}.$$

**Sign**$(\mathbf{gsk}[d], M)$**:** Given $\mathsf{gpk}$, to sign a message $M \in \{0,1\}^*$ using the secret key $\mathbf{gsk}[d] = \mathbf{z}$, the user generates a key pair $(\mathsf{ovk}, \mathsf{osk}) \leftarrow \mathsf{OKeygen}(1^n)$ for $\Pi^{\mathsf{OTS}}$, and then perform the following steps:

1. Encrypt the index $d$ with respect to "identity" $\mathsf{ovk}$ using the GPV IBE:

$$(\mathbf{c}_1 = \mathbf{B}^T\mathbf{s} + \mathbf{e}_1, \mathbf{c}_2 = \mathbf{G}^T\mathbf{s} + \mathbf{e}_2 + \lfloor q/2 \rfloor \cdot d\,) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^\ell,$$

   with $\mathbf{e}_1, \mathbf{e_2}, \mathbf{s} \hookleftarrow \chi^n \times \chi^\ell \times \chi^m$, and $\mathbf{G} = \mathcal{H}_1(\mathsf{ovk})$.
2. Generate a NIZKPoK $\Pi$ to show the possession of a valid message-signature pair $(\mathsf{id}, \mathbf{z})$ for Boyen's signature and that $(\mathbf{c}_1, \mathbf{c}_2)$ is a correct GPV IBE encryption of $d$ with respect to identity $\mathsf{ovk}$. The argument system is described in [34]. The message is embedded in the challenge of the Fiat-Shamir heuristic to construct NIZK proofs.
3. Compute a one-time signature $sig = \mathsf{OSign}(\mathsf{ovk}; \mathbf{c}_1, \mathbf{c}_2, \Pi)$.
4. Output the signature $\Sigma = (\mathsf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig)$.

**Verify**$(\mathsf{gpk}, M, \Sigma)$**:** Parse $\Sigma$ as before. Check that $\mathsf{OVer}(\mathsf{ovk}; sig, (\mathbf{c}_1, \mathbf{c}_2), \Pi) = 1$, otherwise return 0. Verify that $\Pi$ is a valid proof, otherwise return 0. If all verifications succeed, return 1.

**Open**$(\mathsf{gmsk}, M, \Sigma)$**:** Given $\mathsf{gmsk} = \mathbf{T_B}$ and a signature $\Sigma = (\mathsf{ovk}, (\mathbf{c}_1, \mathbf{c}_2), \Pi, sig)$, this algorithm decrypt $(\mathbf{c}_1, \mathbf{c}_2)$ using GPV-IBE decryption algorithm with respect to the identity $\mathsf{ovk}$ and returns the decrypted identity $\mathsf{id}$.

### C.2 The Boyen and Micciancio-Peikert Signatures

This section recalls a variant [35] of Boyen's signature [9], which we use in our scheme to generate the group user secret key. The scheme is parameterized by $\ell$, denoting the size of the message that can be signed. The difference between the original scheme and [35] is the use of a vector $\mathbf{u} \in \mathbb{Z}_q^m$: instead of computing signatures as short vectors in a message-dependent lattice, Micciancio and Peikert [35] use a shifted lattice.

**Keygen**$(1^\lambda, 1^\ell)$**:** To generate the $\mathsf{sk}$ and $\mathsf{pk}$ from $\lambda, \ell$, this algorithm starts by choosing $n = \Theta(\lambda), q = \mathsf{Poly}(\lambda), \alpha^{-1} = \mathsf{Poly}(\lambda), \sigma = \alpha \cdot q$ and $m \geq \Omega(n \log q)$.

1. Use $\mathsf{GenTrap}(1^n, 1^m, q)$ to generates a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ along with a trapdoor $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$.
2. Sample uniformly $\ell + 1$ matrices $\mathbf{A}_0, \ldots, \mathbf{A}_\ell \in_R \mathbb{Z}_q^{n \times m}$.
3. Sample uniformly a vector $\mathbf{u} \in_R \mathbb{Z}_q^n$
4. Output

$$\mathsf{pk} = \{\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}\} \qquad \mathsf{sk} = \mathbf{T_A}$$

**Sign**$(\mathsf{pk}, \mathsf{sk}, M)$**:** To sign a message $M = m_1 \ldots m_\ell \in \{0,1\}^\ell$ with signing key $\mathsf{sk}$.

1. Use $\mathsf{ExtBasis}$ and $\mathsf{pk}$ to compute a trapdoor basis $\mathbf{T_M}$ for the matrix $\mathbf{M} = \left[\mathbf{A} \mid \mathbf{A}_0 + \sum_{i=1}^{\ell} m_i \cdot \mathbf{A}_i\right] \in \mathbb{Z}_q^{n \times 2m}$.
2. Use $\mathsf{GPVSample}$ to get a vector $\mathbf{x} \in D_{\Lambda_q^{\mathbf{u}}(\mathbf{M}),\sigma}$.
3. Output $\mathbf{x} \in \mathbb{Z}^{2m}$ as the signature.

**Verify**$(\mathsf{pk}, M, \mathbf{x})$**:** To verify a signature $\mathbf{x}$ w.r.t a message $M$ and a public key $\mathsf{pk}$:

1. Verify that $\|\mathbf{x}\| \leq \sigma\sqrt{2m}$, if not, output 0 (corresponding to a rejection).
2. Compute the matrix $\mathbf{M} = \left[\mathbf{A} \mid \mathbf{A}_0 + \sum_{i=1}^{\ell} m_i \cdot \mathbf{A}_i\right] \in \mathbb{Z}_q^{n \times 2m}$ using $\mathsf{pk}$.
3. Verify that $\mathbf{M} \cdot \mathbf{x} = \mathbf{u} \bmod q$. If not, output 0.
4. If everything went correctly, output 1.

### C.3 The Gentry-Peikert-Vaikuntanathan IBE

This section recalls the dual Regev cryptosystem, which was proposed by Gentry *et al.* [20] and readily implies an IBE system due to its "obliviously samplable" public key space.

The multi-bit variant of the GPV IBE uses a hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q^{n \times \ell}$ modeled as a random oracle, where $\ell$ denotes the bit-length of the message. In the following $\chi$ denotes the $\mathsf{LWE}$ distribution.

**Setup**$(1^\lambda)$**:** Using algorithm $\mathsf{GenTrap}(1^n, 1^m, q)$, generate a statistically uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with its trapdoor $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ consisting of a short basis of $\Lambda_q^\perp(\mathbf{A})$. Output the master public key $\mathsf{mpk} := \mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and the master secret $\mathsf{msk} := \mathbf{T_A} \in \mathbb{Z}^{m \times m}$.

**Extract**$(\mathbf{A}, \mathbf{T_A}, \mathsf{id})$**:** If a private key for the identity $\mathsf{id}$ has already been released, return the same $sk_{\mathsf{id}}$. Otherwise, let $\mathbf{G} = \mathcal{H}(\mathsf{id})$ and use $\mathsf{SamplePre}$ to return a small-norm matrix $\mathbf{E} \in \mathbb{Z}^{m \times \ell}$ such that $\mathbf{A} \cdot \mathbf{E} = \mathbf{G} \bmod q$.

**Encrypt**$(\mathbf{A}, \mathsf{id}, \mathbf{m})$**:** To encrypt a message $\mathbf{m} \in \{0,1\}^\ell$ w.r.t. identity $\mathsf{id}$. Let $\mathbf{G} = \mathcal{H}(\mathsf{id})$. We sample $\mathbf{s} \hookleftarrow \chi^n, \mathbf{e}_1 \hookleftarrow \chi^m, \mathbf{e}_2 \hookleftarrow \chi^\ell$, and output

$$\mathbf{c} = \left(\mathbf{A}^T\mathbf{s} + \mathbf{e}_1, \mathbf{G}^T\mathbf{s} + \mathbf{e}_2 + \lfloor q/2 \rfloor \cdot \mathbf{m}\right) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^\ell.$$

**Dec**$(\mathbf{E}, (\mathbf{c}_1, \mathbf{c}_2))$**:** Output $\mathbf{m}' = \left(1 \mid 2 \mid 4 \mid \ldots \mid 2^{\ell-1}\right) \cdot \lfloor (\mathbf{c}_2 - \mathbf{E}^T \cdot \mathbf{c}_1) \cdot (q/2) \rceil$.