Provably Secure Group Signature Schemes From Code-Based Assumptions

Martianus Frederic Ezerman[ID], Hyung Tae Lee[ID], San Ling[ID], Khoa Nguyen, and Huaxiong Wang[ID]

Abstract— We solve an open question in code-based cryptography by introducing two provably secure group signature schemes from code-based assumptions. Our basic scheme satisfies the **CPA**-anonymity and traceability requirements in the random oracle model, assuming the hardness of the McEliece problem, the Learning Parity with Noise problem, and a variant of the Syndrome Decoding problem. The construction produces smaller key and signature sizes than the previous group signature schemes from lattices, as long as the cardinality of the underlying group does not exceed 2^{24}, which is roughly comparable to the current population of the Netherlands. We develop the basic scheme further to achieve the strongest anonymity notion, *i.e.*, **CCA**-anonymity, with a small overhead in terms of efficiency. The feasibility of two proposed schemes is supported by implementation results. Our two schemes are the first in their respective classes of provably secure groups signature schemes. Additionally, the techniques introduced in this work might be of independent interest. These are a new verifiable encryption protocol for the randomized McEliece encryption and a novel approach to design formal security reductions from the Syndrome Decoding problem.

Index Terms— Code-based group signature, zero-knowledge protocol, McEliece encryption, syndrome decoding.

I. INTRODUCTION

A. Background and Motivation

GROUP signature [1] is a fundamental cryptographic primitive with two intriguing features. The first one is *anonymity*. It allows users of a group to anonymously sign documents on behalf of the whole group. The second one

is *traceability*. There exists a tracing authority that can tie a given signature to the signer's identity should the need arise. These two properties make group signatures highly useful in various real-life scenarios such as controlled anonymous printing services, digital right management systems, e-bidding and e-voting schemes. Theoretically, designing secure and efficient group signature schemes is of deep interest since doing so typically requires a sophisticated combination of carefully chosen cryptographic ingredients. Numerous constructions of group signatures have been proposed. Most of them, *e.g.*, the respective schemes in [2]–[5], and [6], are based on classical number-theoretic assumptions.

While number-theoretic-based group signatures, such as those in [3] and [4], can be very efficient, they would become insecure once the era of scalable quantum computing arrives [7]. Prior to our work, the search for group signatures that have the potential to be secure against quantum computers, as a preparation for the future, has been quite active, with at least six published schemes [8]–[13]. All of them are based on computational assumptions from lattices. Despite their theoretical interest, the schemes require significantly large key and signature sizes. None of them has been supported by implementation results. Our evaluation, in Section I-B below, shows that these lattice-based schemes are indeed very far from being practical. This somewhat unsatisfactory situation highlights two interesting challenges. The first one is to push group signatures from quantum-resistant assumptions closer to practice. The second one is to bring more diversity in, with schemes from other candidates for post-quantum cryptography, *e.g.*, code-based, hash-based, and multivariate-based. An easy-to-implement and competitively efficient code-based group signature scheme, for instance, would be highly desirable.

A code-based group signature, in the strongest security model for static groups as discussed in [14], typically requires the following three cryptographic layers.

1) The first layer requires a **secure (standard) signature scheme** to sign messages. Note that in most schemes based on the model in [14], a standard signature is also employed to issue the users' secret keys. However, this is not necessary. The scheme constructed in this paper is an illustrative example. We observe that existing code-based signatures fall into two categories.
   The "hash-and-sign" category consists of the CFS signature [15] and its modified versions [16]–[18]. The known security proofs for schemes in this category, however, should be viewed with skepticism. The assumption

used in [16], for example, had been invalidated by the distinguishing attacks detailed in [19], while the new assumption proposed in [18] lies on a rather fragile ground.

The "Fiat-Shamir" category consists of schemes derived from Stern's identification protocol in [20] and its variants in [21]–[23] via the Fiat-Shamir transformation from [24]. Although these schemes produce relatively large signatures, as the underlying protocol has to be repeated many times to make the soundness error negligibly small, their provable security, in the random oracle model, is well-understood.

2) The second layer demands a **semantically secure encryption scheme** to enable the tracing feature. The signer is constrained to encrypt its identifying information and to send the ciphertext as part of the group signature, so that the tracing authority can decrypt if and when necessary. This ingredient is also available in code-based cryptography, thanks to various CPA-secure and CCA-secure variants of the McEliece [25] and the Niederreiter [26] cryptosystems available in, *e.g.*, [27]–[30].

3) The third layer requires a **zero-knowledge (ZK) protocol** that connects the previous two layers. This is essentially the bottleneck in realizing secure code-based group signatures. Specifically, the protocol should demonstrate that a given signature is generated by a certain certified group user who honestly encrypts its identifying information. Constructing such a protocol is quite challenging. There have been ZK protocols involving the CFS and Stern's signatures, which yield identity-based identification schemes in [31]–[33] and threshold ring signatures in [34] and [35]. There have also been ZK proofs of plaintext knowledge for the McEliece and the Niederreiter cryptosystems [36]. Yet we are unaware of any efficient ZK protocol that *simultaneously* deals with both code-based signature and encryption schemes in the above sense.

Designing provably secure group signature schemes has been a long-standing open question in code-based cryptography, as was also discussed in [37].

B. Our Contributions

This work introduces two group signature schemes which are provably secure under code-based assumptions. Specifically, our basic scheme achieves the CPA-anonymity [4] and the traceability requirements in [14] in the random oracle model. We assume the hardness of the McEliece problem, the Learning Parity with Noise problem, and a variant of the Syndrome Decoding problem. The basic scheme is then extended to achieve anonymity in the strongest sense [14], *i.e.*, CCA-anonymity, for which the adversary is allowed to adaptively query for the opening of group signatures. Our two schemes are the first of their respective classes.

Contributions to Code-Based Cryptography. By introducing provably secure code-based group signature schemes, we solve the open problem discussed earlier. Along the way, we introduce the following two new techniques for code-based cryptography, which might be of independent interest.

1) We design a ZK protocol for the randomized McEliece encryption scheme. The protocol allows the prover to convince the verifier that a given ciphertext is well-formed and that the hidden plaintext satisfies an additional condition. Such *verifiable encryption protocols* are useful, not only in constructing group signatures, but also in much broader contexts [38]. It is worth noting that, prior to our work, verifiable encryption protocols for code-based cryptosystems only exist in a very basic form where the plaintext is publicly given [36], restricting their applications.

2) In our security proof of the traceability property, to obtain a reduction from the hardness of the Syndrome Decoding (SD) problem, we come up with an approach that, to the best of our knowledge, has not been considered in the literature before. Let us recall the (average-case) SD problem with parameters m, r, ω. Given a *uniformly random* matrix $\widetilde{\mathbf{H}} \in \mathbb{F}_2^{r \times m}$ and a *uniformly random* syndrome $\tilde{\mathbf{y}} \in \mathbb{F}_2^r$, the problem asks to find a vector $\mathbf{s} \in \mathbb{F}_2^m$ of Hamming weight ω, denoted by $\mathbf{s} \in \mathsf{B}(m, \omega)$, such that $\widetilde{\mathbf{H}} \cdot \mathbf{s}^\top = \tilde{\mathbf{y}}^\top$. In our scheme, the key generation algorithm produces a public key that contains a matrix $\mathbf{H} \in \mathbb{F}_2^{r \times m}$ and syndromes $\mathbf{y}_j \in \mathbb{F}_2^r$, while users are given secret keys of the form $\mathbf{s}_j \in \mathsf{B}(m, \omega)$ such that $\mathbf{H} \cdot \mathbf{s}_j^\top = \mathbf{y}_j^\top$. In the security proof, since we would like to embed an SD challenge instance $(\widetilde{\mathbf{H}}, \tilde{\mathbf{y}})$ into the public key without being noticed, except with negligible probability, by the adversary, we have to require that $\mathbf{H}$ and the $\mathbf{y}_j$'s produced by the key generation are indistinguishable from uniform.

One method to generate these keys is to employ the "hash-and-sign" technique from the CFS signature [15]. Unfortunately, while the syndromes $\mathbf{y}_j$'s could be made uniformly random, as the outputs of the random oracle, the assumption that the CFS matrix $\mathbf{H}$ is *computationally* close to uniform for practical parameters is invalidated by the distinguishing attacks from [19].

Another method, pioneered by Stern [20], is to pick $\mathbf{H}$ and the $\mathbf{s}_j$'s uniformly at random. The corresponding syndromes $\mathbf{y}_j$'s could be made *computationally* close to uniform provided that the parameters are set to make ω slightly smaller than the value ω_0 given by the Gilbert-Varshamov bound, *i.e.*, ω_0 is such that $\binom{m}{\omega_0} \approx 2^r$. In such a case, the function $f_{\mathbf{H}}(\mathbf{s}_j) = \mathbf{H} \cdot \mathbf{s}_j^\top$ acts as a pseudorandom generator [39]. However, for these parameters, it is not guaranteed with high probability that a uniformly random SD instance $(\widetilde{\mathbf{H}}, \tilde{\mathbf{y}})$ has solutions, which would affect the success probability of the reduction algorithm.

Our work considers the case when ω is moderately larger than ω_0, so that two conditions hold. First, the uniform distribution over the set $\mathsf{B}(m, \omega)$ has sufficient min-entropy to apply the Left-over Hash Lemma from [40]. Second, the SD problem with parameters (m, r, ω) admits solutions with high probability, yet remains intractable against the best known attacks

TABLE I

EFFICIENCY COMPARISON BETWEEN OUR SCHEMES AND [13]

	Anonymity	N	Public Key Size		Signature Size	
Ours	CPA	$2^8 = 256$	5.13×10^6 bits	(642 KB)	8.57×10^6 bits	(1.07 MB)
		$2^{16} = 65,536$	4.10×10^7 bits	(5.13 MB)	1.77×10^7 bits	(2.21 MB)
		$2^{24} = 16,777,216$	9.23×10^9 bits	(1.16 GB)	2.36×10^9 bits	(294 MB)
	CCA	$2^8 = 256$	8.60×10^6 bits	(1.08 MB)	9.99×10^6 bits	(1.25 MB)
		$2^{16} = 65,536$	4.45×10^7 bits	(5.56 MB)	1.91×10^7 bits	(2.39 MB)
		$2^{24} = 16,777,216$	9.23×10^9 bits	(1.16 GB)	2.36×10^9 bits	(294 MB)
[13]	CPA	$\leq 2^{24}$	1.18×10^{10} bits	(1.48 GB)	4.63×10^9 bits	(579 MB)

from [42] and [43]. Note that the variant of the SD problem considered in this work are not widely believed to be the hardest one [20], [41], but suitable parameters can be chosen such that the best known attacks run in exponential time. Further treatment on how to decide on the parameters will be given in Section VI. This approach gives us a new method to generate uniformly random vectors $\mathbf{s}_j \in \mathsf{B}(m,\omega)$ and a matrix $\mathbf{H} \in \mathbb{F}_2^{r \times m}$ so that the syndromes $\mathbf{y}_j$'s corresponding to the $\mathbf{s}_j$'s are *statistically* close to uniform. The approach, which somewhat resembles the technique used in [44] for the Inhomogeneous Small Integer Solution problem, is helpful in our security proof and, generally, in designing formal security reductions from the SD problem.

Contributions to Group Signatures from Quantum-Resistant Assumptions. Although we are yet to obtain security proofs in the quantum random oracle model (QROM), our constructions provide the first non-lattice-based alternatives to provably secure group signatures from quantum-resistant assumptions. Our schemes feature public key and signature sizes linear in the number of group users N, which are *asymptotically* not as efficient as the previously published lattice-based counterparts [10]–[13]. However, when instantiated with practical parameters, our schemes behave much more efficiently than the scheme proposed in [13]. The latter is arguably the current most efficient lattice-based group signature in the asymptotic sense. Indeed, our estimation shows that our basic scheme, which achieves the CPA-anonymity notion, gives public key and signature sizes that are $2,300$ times and 540 times smaller, respectively, for an average-size group of $N = 2^8$ users. As N grows, the advantage lessens, but our basic scheme remains more efficient even for a huge group of $N = 2^{24}$ users, a number which is roughly comparable to the current population of the Netherlands. Our extended scheme, which achieves the strongest anonymity notion, *i.e.*, CCA-anonymity, introduces only a small overhead of about 434 KB and 177 KB in public key and signature sizes, respectively, compared to the basic scheme. Table I gives the details of our estimation. The parameters for our schemes are set as in Section VI. For the scheme in [13], we choose the commonly used lattice dimension $n = 2^8$ and set the parameters $m = 2^9 \times 150$ and $q = 2^{150}$ to satisfy the requirements given in [13, Section 5.1]. While our basic scheme and the scheme in [13] achieve the CPA-anonymity

notion [4], our extended scheme achieves the CCA-anonymity notion [14]. All schemes have soundness error 2^{-80}.

We give actual implementation results for our proposed schemes to support their claim of feasibility. In our implementations, as presented later in Section VI, the actual signature sizes can be reduced thanks to an additional technique. Our schemes are the first group signature from quantum-resistant assumptions that comes supported with an actual deployment analysis. The results, while not yielding a truly practical scheme, certainly help in bringing this new class of group signatures closer to practice.

C. Overview of Our Techniques

Let m, r, ω, n, k, t and ℓ be positive integers. We consider a group of size $N = 2^\ell$, where each user is indexed by an integer $j \in [0, N-1]$. The secret signing key of user j is a vector $\mathbf{s}_j$ chosen uniformly at random from the set $\mathsf{B}(m,\omega)$. A uniformly random matrix $\mathbf{H} \in \mathbb{F}_2^{r \times m}$ and N syndromes $\mathbf{y}_0, \ldots, \mathbf{y}_{N-1} \in \mathbb{F}_2^r$, such that $\mathbf{H} \cdot \mathbf{s}_j^\top = \mathbf{y}_j^\top$, for all j, are made public. Let us now explain the development of the three ingredients used in our basic scheme.

The Signature Layer. User j can run Stern's ZK protocol [20] to prove the possession of a vector $\mathbf{s} \in \mathsf{B}(m,\omega)$ such that $\mathbf{H} \cdot \mathbf{s}^\top = \mathbf{y}_j^\top$. The constraint $\mathbf{s} \in \mathsf{B}(m,\omega)$ is proved in ZK by randomly permuting the entries of $\mathbf{s}$ and showing that the permuted vector belongs to $\mathsf{B}(m,\omega)$. The protocol is then transformed into a Fiat-Shamir signature [24]. However, such a signature is publicly verifiable only if the index j is given to the verifier.

The user can further hide its index j to achieve unconditional anonymity among all N users. This, incidentally, yields a *ring signature* [45] on the way, à la [46]. Let $\mathbf{A} = \left[\mathbf{y}_0^\top | \cdots | \mathbf{y}_j^\top | \cdots | \mathbf{y}_{N-1}^\top \right] \in \mathbb{F}_2^{r \times N}$. Let $\mathbf{x} = \delta_j^N$ be the N-dimensional unit vector with entry 1 at the j-th position and 0 elsewhere. Observe that $\mathbf{A} \cdot \mathbf{x}^\top = \mathbf{y}_j^\top$, and thus, the equation $\mathbf{H} \cdot \mathbf{s}^\top = \mathbf{y}_j^\top$ can be written as

$$\mathbf{H} \cdot \mathbf{s}^\top \oplus \mathbf{A} \cdot \mathbf{x}^\top = \mathbf{0}, \qquad (1)$$

where $\oplus$ denotes addition modulo 2. Stern's framework allows the user to prove in ZK the possession of $(\mathbf{s}, \mathbf{x})$ satisfying this equation, where the condition $\mathbf{x} = \delta_j^N$ can be justified using a random permutation.

The Encryption Layer. To enable the tracing capability of the scheme, we let user j encrypt the binary representation

of j via the randomized McEliece encryption scheme [27]. Specifically, we represent j as

$$\mathsf{I2B}(j) = (j_0, \ldots, j_{\ell-1}) \in \{0,1\}^\ell, \text{ where } \sum_{i=0}^{\ell-1} j_i 2^{\ell-1-i} = j.$$

Given a public encrypting key $\mathbf{G} \in \mathbb{F}_2^{k \times n}$, a ciphertext of $\mathsf{I2B}(j)$ is of the form

$$\mathbf{c} = \big(\mathbf{u} \,\|\, \mathsf{I2B}(j)\big) \cdot \mathbf{G} \oplus \mathbf{e} \in \mathbb{F}_2^n, \tag{2}$$

where $(\mathbf{u}, \mathbf{e})$ is the encryption randomness, with $\mathbf{u} \in \mathbb{F}_2^{k-\ell}$, and $\mathbf{e} \in \mathsf{B}(n,t)$, *i.e.*, $\mathbf{e}$ is a vector of weight t in $\mathbb{F}_2^n$.

Connecting the Signature and Encryption Layers. User j must demonstrate that it does not cheat, *e.g.*, by encrypting some string that does not point to j, without revealing j. Thus, we need a ZK protocol that allows the user to prove that the vector $\mathbf{x} = \delta_j^N$ in (1) and the plaintext hidden in (2) both correspond to the same secret $j \in [0, N-1]$. The crucial challenge is to establish a connection, which must be verifiable in ZK, between the "index representation" δ_j^N and the binary representation $\mathsf{I2B}(j)$. We show how to handle this challenge well.

Instead of working with $\mathsf{I2B}(j) = (j_0, \ldots, j_{\ell-1})$, let us consider an extension of $\mathsf{I2B}(j)$, defined as

$$\begin{aligned}\mathsf{Encode}(j) = \big(1 - j_0, j_0, \ldots, 1 - j_i, j_i, \ldots, \\ 1 - j_{\ell-1}, j_{\ell-1}\big) \in \mathbb{F}_2^{2\ell}.\end{aligned}$$

We then suitably insert ℓ zero-rows into $\mathbf{G}$ to obtain $\widehat{\mathbf{G}} \in \mathbb{F}_2^{(k+\ell) \times n}$ such that

$$\big(\mathbf{u} \,\|\, \mathsf{Encode}(j)\big) \cdot \widehat{\mathbf{G}} = \big(\mathbf{u} \,\|\, \mathsf{I2B}(j)\big) \cdot \mathbf{G}.$$

Letting $\mathbf{f} = \mathsf{Encode}(j)$, we rewrite (2) as

$$\mathbf{c} = \big(\mathbf{u} \,\|\, \mathbf{f}\big) \cdot \widehat{\mathbf{G}} \oplus \mathbf{e} \in \mathbb{F}_2^n. \tag{3}$$

Now, let $\mathsf{B2I} : \{0,1\}^\ell \to [0, N-1]$ be the inverse function of $\mathsf{I2B}(\cdot)$. For every $\mathbf{b} \in \{0,1\}^\ell$, we carefully design two classes of permutations

$$T_{\mathbf{b}} : \mathbb{F}_2^N \to \mathbb{F}_2^N \text{ and } T'_{\mathbf{b}} : \mathbb{F}_2^{2\ell} \to \mathbb{F}_2^{2\ell},$$

such that, for any $j \in [0, N-1]$,

$$\mathbf{x} = \delta_j^N \iff T_{\mathbf{b}}(\mathbf{x}) = \delta_{\mathsf{B2I}(\mathsf{I2B}(j) \oplus \mathbf{b})}^N \text{ and}$$
$$\mathbf{f} = \mathsf{Encode}(j) \iff T'_{\mathbf{b}}(\mathbf{f}) = \mathsf{Encode}(\mathsf{B2I}(\mathsf{I2B}(j) \oplus \mathbf{b})).$$

Given the equivalences, the protocol's user samples a uniformly random vector $\mathbf{b} \in \{0,1\}^\ell$ and sends $\mathbf{b}_1 = \mathsf{I2B}(j) \oplus \mathbf{b}$. The verifier, seeing that

$$T_{\mathbf{b}}(\mathbf{x}) = \delta_{\mathsf{B2I}(\mathbf{b}_1)}^N \text{ and } T'_{\mathbf{b}}(\mathbf{f}) = \mathsf{Encode}(\mathsf{B2I}(\mathbf{b}_1)),$$

should be convinced that $\mathbf{x}$ and $\mathbf{f}$ correspond to the same $j \in [0, N-1]$, yet the value of j is completely hidden from its view since $\mathbf{b}$ acts essentially as a one-time pad.

The technique extending $\mathsf{I2B}(j)$ into $\mathsf{Encode}(j)$ and then permuting $\mathsf{Encode}(j)$ in a "one-time pad" fashion is inspired by a method originally proposed by Langlois *et al.* in [11] in a seemingly unrelated context. There, the goal is to prove that the message being signed under the Bonsai tree signature [47]

is of the form $\mathsf{I2B}(j)$, for some $j \in [0, N-1]$. Here, we adapt and develop their method to *simultaneously* prove two facts. First, the plaintext being encrypted under the randomized McEliece encryption is of the form $\mathsf{I2B}(j)$. Second, the unit vector $\mathbf{x} = \delta_j^N$ is used in the signature layer.

By embedding the above technique into Stern's framework, we obtain an interactive ZK argument system, in which, given the public input $(\mathbf{H}, \mathbf{A}, \mathbf{G})$, the user is able to prove the possession of a secret tuple $(j, \mathbf{s}, \mathbf{x}, \mathbf{u}, \mathbf{f}, \mathbf{e})$ that satisfies (1) and (3). The protocol is repeated many times to achieve negligible soundness error, and then made non-interactive, resulting in a non-interactive ZK argument of knowledge Π. The final group signature is of the form $(\mathbf{c}, \Pi)$, where $\mathbf{c}$ is the ciphertext. In the random oracle model, the anonymity of the scheme relies on the zero-knowledge property of Π and the CPA-security of the randomized McEliece encryption scheme, while its traceability is based on the hardness of the variant of the SD problem discussed earlier.

Achieving CCA-Anonymity. Our basic group signature scheme makes use of the randomized McEliece encryption scheme that achieves only CPA-security. Hence, it only satisfies CPA-anonymity for which the adversary is not granted access to the signature opening oracle. To achieve the strongest notion of anonymity put forward in [14], *i.e.*, CCA-anonymity, we would need a CCA2-secure encryption scheme so that we can respond to adaptive opening queries from the adversary by invoking the decryption oracle associated with the encryption mechanism. There are a number of known CCA2-secure code-based encryption schemes, *e.g.*, [28]–[30]. They are, however, either too inefficient, say with ciphertext size quadratic in the security parameter, or incompatible with zero-knowledge protocols for proving the well-formedness of ciphertexts. Hence, they are unsuitable for our purpose. Instead, we exploit the fact that our verifiable encryption protocol for the randomized McEliece scheme is a simulation-sound ZK argument of knowledge. We then upgrade the encryption system further to a CCA2-secure one via the Naor-Yung twin-encryption paradigm [51]. The protocol operates in Stern's framework and satisfies the "quasi-unique responses" property in [48], deriving simulation-soundness from soundness. This fact was recently exploited by several group signature schemes, *e.g.*, [49] and [50], which are based on Stern-like protocols.

Specifically, we will work with two public keys $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$ of the randomized McEliece encryption scheme. The user now encrypts $\mathsf{I2B}(j)$ under each of the keys to obtain ciphertexts $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$, respectively, and extend the verifiable encryption protocol discussed above to prove that these ciphertexts are well-formed and correspond to the same plaintext $\mathsf{I2B}(j)$, which is the binary representation of the user's index j. This extension is quite smooth, since the same techniques for handling ciphertext $\mathbf{c}$ can be used to handle $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$. In the proof of CCA-anonymity, we then employ the strategy of [52] that makes use of the CPA-security of the underlying encryption scheme and the zero-knowledge, soundness and simulation-soundness of the resulting non-interactive argument. In terms of efficiency,

our CCA-anonymous construction only has a small and reasonable overhead compared to its CPA-anonymous version, with one more McEliece encrypting matrix in the group public key and one more ciphertext equipped with its supporting ZK sub-protocol in the group signature.

D. Related Works

The present paper is the full extension of our earlier work [53], which was published in the proceedings of ASI-ACRYPT 2015. Achieving CCA-anonymity for code-based group signatures was raised as an open question in [53]. We are able to fully address the problem in this work.

In a work concurrent to and independent of [53], Alamélou *et al.* also proposed a code-based group signature scheme in [54] and, later on, in [55]. Their scheme considers the setting of dynamic groups. It does not use any encryption mechanism to enable traceability. Instead, the authors rely on a modified version of Stern's protocol that allows the opening authority to test whether each protocol execution is generated using a secret key of a given user. Unfortunately, such approach does not yield a secure group signature. Recall that Stern's protocol admits a soundness error of $2/3$ in each execution. It has to be repeated $\kappa = \omega(\log \lambda)$ times, where λ is the security parameter, to make the error negligibly small. Then, a valid signature is generated by an honest user j if and only if the tests for *all* κ executions of the protocol yield the *same* user j. Unfortunately, the testing mechanism used in their scheme fails to capture this crucial point. When running through protocol execution numbers $1, 2, \ldots, \kappa$, it stops and outputs user j when it sees the *first* execution that points to j. This shortcoming opens a room for cheating users to break the traceability and non-frameability of the scheme. Specifically, a cheating user j', who wants to mislead the opening result to an innocent user j, can simulate the first several protocol executions. The simulation can be done with noticeable probability using the transcript simulator associated with the protocol, because each execution admits a soundness error of $2/3$. If the opening algorithm is run, it would return j with noticeable probability. The remaining protocol executions are done faithfully with secret key for user j'. Thus, the construction in [54] and [55] is not secure. We note that a very similar testing mechanism for Stern-like protocols was used in [11] to avoid the use of encryption in their group signature. This had been broken. In [56], which is the corrected version of [11], the authors eventually had to rely on an encryption-like mechanism to enable traceability.

Nguyen *et al.* [57] recently proposed a number of new code-based privacy-preserving cryptographic constructions, including the first code-based group signature scheme with logarithmic signature size - which resolves an interesting question we left open in [53]. In their scheme, group users are associated with leaves in a code-based Merkle tree supported by a zero-knowledge argument of tree inclusion, which has communication cost linear in the tree depth (and hence, logarithmic in the number of users). Although their scheme achieves better asymptotic efficiency than ours, it yields signature size larger than 2.5 MB even for small groups.

In particular, for groups of size up to 2^{16}, the signatures are longer than those produced by our CPA-anonymous and CCA-anonymous schemes.

Subsequent to the publication of [53], a number of lattice-based group signatures have been proposed, bringing group signatures from quantum-resistant assumptions much closer to practice. Examples include the works done in [49], [50], [58]–[60]. We believe that this interesting research direction will continue to attract attention from the community. The hope is that some provably secure and truly practical schemes can be realized in the near future.

II. PRELIMINARIES

NOTATIONS. Let λ be the security parameter and $\mathsf{negl}(\lambda)$ denote a negligible function in λ. We use $a \xleftarrow{\$} A$ if a is chosen uniformly at random from the finite set A. The symmetric group of all permutations of k elements is denoted by S_k. Bold capital letters, *e.g.*, $\mathbf{A}$, denote matrices. Bold lowercase letters, *e.g.*, $\mathbf{x}$, denote row vectors. We use $\mathbf{x}^\top$ to denote the transpose of $\mathbf{x}$ and $wt(\mathbf{x})$ to denote the (Hamming) weight of $\mathbf{x}$. We denote by $\mathsf{B}(m, \omega)$ the set of all vectors $\mathbf{x} \in \mathbb{F}_2^m$ such that $wt(\mathbf{x}) = \omega$. Throughout the paper, we define a function I2B which takes a non-negative integer a as an input, and outputs the binary representation $(a_0, \cdots, a_{\ell-1}) \in \{0, 1\}^\ell$ of a such that $a = \sum_{i=0}^{\ell-1} a_i 2^{\ell-1-i}$, and a function B2I which takes as an input the binary representation $(a_0, \cdots, a_{\ell-1}) \in \{0, 1\}^\ell$ of a, and outputs a. All logarithms are in base 2.

A. Background on Code-Based Cryptography

We first recall the Syndrome Decoding problem. It is well-known to be NP-complete [61], and is widely believed to be intractable in the average case for appropriate choice of parameters [20], [41].

Definition 1 (The Syndrome Decoding Problem): The SD (m, r, ω) problem asks, given a uniformly random matrix $\mathbf{H} \in \mathbb{F}_2^{r \times m}$ and a uniformly random syndrome $\mathbf{y} \in \mathbb{F}_2^r$, for a vector $\mathbf{s} \in \mathsf{B}(m, \omega)$ such that $\mathbf{H} \cdot \mathbf{s}^\top = \mathbf{y}^\top$. When $m = m(\lambda)$, $r = r(\lambda)$ and $\omega = \omega(\lambda)$, the SD(m, r, ω) problem is hard if the success probability of any PPT algorithm in solving the problem is at most $\mathsf{negl}(\lambda)$.

In our security reduction, the following variant of the Left-over Hash Lemma for matrix multiplication over $\mathbb{F}_2$ is used.

Lemma 1 (Left-Over Hash Lemma, Adapted From [40]): Let D be a distribution over $\mathbb{F}_2^m$ with min-entropy e. For $\epsilon > 0$ and $r \le e - 2\log(1/\epsilon) - \mathcal{O}(1)$, the statistical distance between the distribution of $(\mathbf{H}, \mathbf{H} \cdot \mathbf{s}^\top)$, where $\mathbf{H} \xleftarrow{\$} \mathbb{F}_2^{r \times m}$ and $\mathbf{s} \in \mathbb{F}_2^m$ is drawn from distribution D, and the uniform distribution over $\mathbb{F}_2^{r \times m} \times \mathbb{F}_2^r$ is at most ϵ.

In particular, if $\omega < m$ is an integer such that

$$r \le \log \binom{m}{\omega} - 2\lambda - \mathcal{O}(1)$$

and D is the uniform distribution over $\mathsf{B}(m, \omega)$, *i.e.*, D has min-entropy $\log \binom{m}{\omega}$, then the statistical distance between the distribution of $(\mathbf{H}, \mathbf{H} \cdot \mathbf{s}^\top)$ and the uniform distribution over $\mathbb{F}_2^{r \times m} \times \mathbb{F}_2^r$ is at most $2^{-\lambda}$.

The Randomized McEliece Encryption Scheme. We employ the following randomized variant, suggested in [27], of the McEliece encryption scheme [25], where a uniformly random vector is concatenated to the plaintext.

1) ME.Setup(1^λ): Select $n = n(\lambda)$, $k = k(\lambda)$, and $t = t(\lambda)$ as the parameters for a binary $[n, k, 2t+1]$ Goppa code. Choose integers k_1 and k_2 such that $k = k_1 + k_2$. Set the plaintext space as $\mathbb{F}_2^{k_2}$.

2) ME.KeyGen(n, k, t): Perform the following steps.

   a) Produce a generator matrix $\mathbf{G}' \in \mathbb{F}_2^{k \times n}$ of a randomly selected $[n, k, 2t+1]$ Goppa code. Choose a random invertible matrix $\mathbf{S} \in \mathbb{F}_2^{k \times k}$ and a random permutation matrix $\mathbf{P} \in \mathbb{F}_2^{n \times n}$. Let
   $$\mathbf{G} = \mathbf{S}\mathbf{G}'\mathbf{P} \in \mathbb{F}_2^{k \times n}.$$

   b) Output encrypting key $\mathsf{pk}_{\mathsf{ME}} = \mathbf{G}$ and decrypting key $\mathsf{sk}_{\mathsf{ME}} = (\mathbf{S}, \mathbf{G}', \mathbf{P})$.

3) ME.Enc$(\mathsf{pk}_{\mathsf{ME}}, \mathbf{m})$: To encrypt a message $\mathbf{m} \in \mathbb{F}_2^{k_2}$, get $\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^{k_1}$ and $\mathbf{e} \xleftarrow{\$} \mathsf{B}(n, t)$, then output the ciphertext
   $$\mathbf{c} = (\mathbf{u}\|\mathbf{m}) \cdot \mathbf{G} \oplus \mathbf{e} \in \mathbb{F}_2^n.$$

4) ME.Dec$(\mathsf{sk}_{\mathsf{ME}}, \mathbf{c})$: Perform the following steps.

   a) Compute $\mathbf{c} \cdot \mathbf{P}^{-1} = ((\mathbf{u}\|\mathbf{m}) \cdot \mathbf{G} \oplus \mathbf{e}) \cdot \mathbf{P}^{-1}$, and, then,
   $$\mathbf{m}' \cdot \mathbf{S} = Decode_{\mathbf{G}'}(\mathbf{c} \cdot \mathbf{P}^{-1}),$$
   where $Decode$ is an error-correcting algorithm with respect to $\mathbf{G}'$. If $Decode$ fails, then return $\perp$.

   b) Compute $\mathbf{m}' = (\mathbf{m}'\mathbf{S}) \cdot \mathbf{S}^{-1}$, parse $\mathbf{m}' = (\mathbf{u}\|\mathbf{m})$, where $\mathbf{u} \in \mathbb{F}_2^{k_1}$ and $\mathbf{m} \in \mathbb{F}_2^{k_2}$, and return $\mathbf{m}$.

The scheme described above is CPA-secure in the standard model, assuming the hardness of the $\mathsf{DMcE}(n, k, t)$ problem and the $\mathsf{DLPN}(k_1, n, \mathsf{B}(n, t))$ problem [27], [62]. We now recall these two problems.

Definition 2 (The Decisional McEliece Problem): The $\mathsf{DMcE}(n, k, t)$ problem is to distinguish if a given $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ is a uniformly random matrix over $\mathbb{F}_2^{k \times n}$ or is generated by ME.KeyGen(n, k, t) above. When $n = n(\lambda)$, $k = k(\lambda)$, and $t = t(\lambda)$, the $\mathsf{DMcE}(n, k, t)$ problem is hard if the success probability of any PPT distinguisher is at most $1/2 + \mathsf{negl}(\lambda)$.

Definition 3 (The Decisional Learning Parity With (Fixed-Weight) Noise Problem): The $\mathsf{DLPN}(k, n, \mathsf{B}(n, t))$ problem, given a pair $(\mathbf{A}, \mathbf{v}) \in \mathbb{F}_2^{k \times n} \times \mathbb{F}_2^n$, is to distinguish whether $(\mathbf{A}, \mathbf{v})$ is a uniformly random pair over $\mathbb{F}_2^{k \times n} \times \mathbb{F}_2^n$ or is obtained by choosing $\mathbf{A} \xleftarrow{\$} \mathbb{F}_2^{k \times n}$, $\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^k$, $\mathbf{e} \xleftarrow{\$} \mathsf{B}(n, t)$ and outputting $(\mathbf{A}, \mathbf{u} \cdot \mathbf{A} \oplus \mathbf{e})$. When $k = k(\lambda)$, $n = n(\lambda)$, and $t = t(\lambda)$, the $\mathsf{DLPN}(k, n, \mathsf{B}(n, t))$ problem is hard if the success probability of any PPT distinguisher is at most $1/2 + \mathsf{negl}(\lambda)$.

B. Group Signatures

We follow the definition of group signatures provided in [14] for the case of static groups.

Definition 4: A *group signature*

$$\mathcal{GS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$$

is a tuple of the following four polynomial-time algorithms.

1) KeyGen: This randomized algorithm takes as input $(1^\lambda, 1^N)$, where $N \in \mathbb{N}$ is the number of group users, and outputs $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$, where gpk is the group public key, gmsk is the group manager's secret key, and $\mathsf{gsk} = \{\mathsf{gsk}[j]\}_{j \in [0, N-1]}$ with $\mathsf{gsk}[j]$ being the secret key for the group user of index j.

2) Sign: This randomized algorithm takes as input a secret signing key $\mathsf{gsk}[j]$ for some $j \in [0, N-1]$ and a message M and returns a group signature Σ on M.

3) Verify: This deterministic algorithm takes as input the group public key gpk, a message M, and a signature Σ on M. The output is either 1 (Accept) or 0 (Reject).

4) Open: This deterministic algorithm takes as input the group manager's secret key gmsk, a message M, and a signature Σ on M. It outputs either $j \in [0, N-1]$, which is an index associated with a particular user, or $\perp$, indicating failure.

A **correct** group signature scheme requires that, for all $\lambda, N \in \mathbb{N}$, all $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$ produced by KeyGen$(1^\lambda, 1^N)$, all $j \in [0, N-1]$, and all messages $M \in \{0, 1\}^*$, we have

$$\mathsf{Verify}(\mathsf{gpk}, M, \mathsf{Sign}(\mathsf{gsk}[j], M)) = 1 \text{ and}$$
$$\mathsf{Open}(\mathsf{gmsk}, M, \mathsf{Sign}(\mathsf{gsk}[j], M)) = j.$$

A **secure** group signature scheme must meet two conditions.

1) *Traceability*: all signatures, even those produced by a coalition of group users and the group manager, can be traced back to a member of the coalition.

2) *Anonymity*: signatures generated by two distinct group users are computationally indistinguishable to an adversary who knows all of the user secret keys. In Bellare *et al.*'s model [14], the anonymity adversary is granted access to an opening oracle (CCA-anonymity). A relaxed notion, where the adversary cannot query the opening oracle (CPA-anonymity), was later proposed by Boneh *et al.* [4].

We now give the formal definitions of CPA-anonymity, CCA-anonymity and traceability.

Definition 5: A group signature $\mathcal{GS}$ is CPA-anonymous if, for all polynomial $N(\cdot)$ and any PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following experiment is negligible in λ.

1) Run $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^N)$ and, then, send $(\mathsf{gpk}, \mathsf{gsk})$ to $\mathcal{A}$.

2) $\mathcal{A}$ outputs two identities $j_0, j_1 \in [0, N-1]$ with a message M^*. Choose a random bit b and give $\Sigma^* \leftarrow \mathsf{Sign}(\mathsf{gsk}[j_b], M^*)$ to $\mathcal{A}$. Then, $\mathcal{A}$ outputs a bit b'.

If $b' = b$, then $\mathcal{A}$ succeeds. The advantage of $\mathcal{A}$ is defined to be $|\Pr[\mathcal{A} \text{ succeeds}] - \frac{1}{2}|$.

A group signature $\mathcal{GS}$ is CCA-anonymous if for all polynomial $N(\cdot)$ and any PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following experiment is negligible in λ.

1) Run $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^N)$ and send $(\mathsf{gpk}, \mathsf{gsk})$ to $\mathcal{A}$.

2) $\mathcal{A}$ can make queries to the opening oracle. On input a message M and a signature Σ, the oracle returns $\mathsf{Open}(\mathsf{gmsk}, M, \Sigma)$ to $\mathcal{A}$.

3) $\mathcal{A}$ outputs two identities $j_0, j_1 \in [0, N-1]$ with a message M^*. Choose a random bit b and give $\Sigma^* \leftarrow \mathsf{Sign}(\mathsf{gsk}[j_b], M^*)$ to $\mathcal{A}$.

4) $\mathcal{A}$ can make further queries to the opening oracle, with the exception that it cannot query for the opening of (M^*, Σ^*).

5) Finally, $\mathcal{A}$ outputs a bit b'.

$\mathcal{A}$ succeeds if $b' = b$. The advantage of $\mathcal{A}$ is defined to $\left| \Pr[\mathcal{A} \text{ succeeds}] - \frac{1}{2} \right|$.

A group signature $\mathcal{GS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ is traceable if, for all polynomial $N(\cdot)$ and any PPT adversaries $\mathcal{A}$, the success probability of $\mathcal{A}$ in the following experiment is negligible in λ.

1) Run $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^N)$ and, then, send $(\mathsf{gpk}, \mathsf{gmsk})$ to $\mathcal{A}$.

2) $\mathcal{A}$ may query the following oracles adaptively and in any order.

   a) An $\mathcal{O}^{\mathsf{Corrupt}}$ oracle that, on input $j \in [0, N-1]$, outputs $\mathsf{gsk}[j]$.

   b) An $\mathcal{O}^{\mathsf{Sign}}$ oracle that, on input j and a message M, returns $\mathsf{Sign}(\mathsf{gsk}[j], M)$.

3) Finally, $\mathcal{A}$ outputs a message M^* and a signature Σ^*.

If CU is the set of identities queried to $\mathcal{O}^{\mathsf{Corrupt}}$, then $\mathcal{A}$ succeeds if

$$\mathsf{Verify}(\mathsf{gpk}, M^*, \Sigma^*) = 1$$

and $\mathsf{Sign}(\mathsf{gsk}[j], M^*)$ was never queried for $j \notin CU$, yet

$$\mathsf{Open}(\mathsf{gmsk}, M^*, \Sigma^*) \notin CU.$$

III. THE UNDERLYING ZERO-KNOWLEDGE ARGUMENT SYSTEMS

A statistical zero-knowledge argument system is an interactive protocol where the soundness property holds for *computationally bounded* cheating provers, while the zero-knowledge property holds against *any* cheating verifier. In this section we present statistical zero-knowledge argument systems which will serve as building blocks in our CPA-anonymous and CCA-anonymous group signature schemes in Section IV and Section V, respectively.

Before describing the protocols, we introduce several supporting notations and techniques. Let ℓ be a positive integer and $N = 2^\ell$. For $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1}) \in \mathbb{F}_2^N$ and for $j \in [0, N-1]$, we write $\mathbf{x} = \delta_j^N$ if $x_j = 1$ and $x_i = 0$ for all $i \neq j$. An encoding function $\mathsf{Encode} : [0, N-1] \to \mathbb{F}_2^{2\ell}$ maps an integer $j \in [0, N-1]$, whose binary representation is $\mathsf{I2B}(j) = (j_0, \ldots, j_{\ell-1})$, to the vector

$$\mathsf{Encode}(j) = (1-j_0, j_0, \ldots, 1-j_i, j_i, \ldots, 1-j_{\ell-1}, j_{\ell-1}).$$

Given a vector $\mathbf{b} = (b_0, \ldots, b_{\ell-1}) \in \{0, 1\}^\ell$, we define two pertmutations. The first permutation $T_\mathbf{b} : \mathbb{F}_2^N \to \mathbb{F}_2^N$ transforms $\mathbf{x} = (x_0, \ldots, x_{N-1})$ to $(x_0', \ldots, x_{N-1}')$, where for each $i \in [0, N-1]$, we have $x_i = x_{i^*}'$ with $i^* = \mathsf{B2I}(\mathsf{I2B}(i) \oplus \mathbf{b})$. The second permutation $T_\mathbf{b}' : \mathbb{F}_2^{2\ell} \to \mathbb{F}_2^{2\ell}$ maps

$$\mathbf{f} = (f_0, f_1, \ldots, f_{2i}, f_{2i+1}, \ldots, f_{2(\ell-1)}, f_{2(\ell-1)+1}) \mapsto$$
$$T_\mathbf{b}'(\mathbf{f}) = (f_{b_0}, f_{1-b_0}, \ldots, f_{2i+b_i}, f_{2i+(1-b_i)}, \ldots,$$
$$f_{2(\ell-1)+b_{\ell-1}}, f_{2(\ell-1)+(1-b_{\ell-1})}).$$

Observe that, for any $j \in [0, N-1]$ and any $\mathbf{b} \in \{0, 1\}^\ell$, we have

$$\mathbf{x} = \delta_j^N \iff T_\mathbf{b}(\mathbf{x}) = \delta_{\mathsf{B2I}(\mathsf{I2B}(j) \oplus \mathbf{b})}^N \text{ and} \qquad (4)$$
$$\mathbf{f} = \mathsf{Encode}(j) \iff T_\mathbf{b}'(\mathbf{f}) = \mathsf{Encode}(\mathsf{B2I}(\mathsf{I2B}(j) \oplus \mathbf{b})). \qquad (5)$$

Example 1: Given $N = 2^4$ and $j = 6$, we quickly obtain $\mathsf{I2B}(j) = (0, 1, 1, 0)$ and $\mathsf{Encode}(j) = (1, 0, 0, 1, 0, 1, 1, 0)$. If $\mathbf{b} = (1, 0, 1, 0)$, then $\mathsf{B2I}(\mathsf{I2B}(j) \oplus \mathbf{b}) = \mathsf{B2I}(1, 1, 0, 0) = 12$. We have $T_\mathbf{b}(\delta_6^{16}) = \delta_{12}^{16}$ and

$$T_\mathbf{b}'(\mathsf{Encode}(6)) = (0, 1, 0, 1, 1, 0, 1, 0) = \mathsf{Encode}(12).$$

A. The Interactive Protocol Underlying the CPA-Anonymous Group Signature

We now present the interactive zero-knowledge argument of knowledge (ZKAoK) that will be used in the CPA-anonymous group signature scheme of Section IV. Let $n, k, t, m, r, \omega, \ell$ be positive integers, and $N = 2^\ell$. The public input consists of matrices $\mathbf{G} \in \mathbb{F}_2^{k \times n}$, $\mathbf{H} \in \mathbb{F}_2^{r \times m}$, as well as N syndromes $\mathbf{y}_0, \ldots, \mathbf{y}_{N-1} \in \mathbb{F}_2^r$, and vector $\mathbf{c} \in \mathbb{F}_2^n$. The protocol allows the prover $\mathcal{P}$ to *simultaneously* convince the verifier $\mathcal{V}$ in zero-knowledge that $\mathcal{P}$ possesses a vector $\mathbf{s} \in \mathsf{B}(m, \omega)$ corresponding to certain syndrome $\mathbf{y}_j \in \{\mathbf{y}_0, \ldots, \mathbf{y}_{N-1}\}$ with hidden index j, *and* that $\mathbf{c}$ is a correct encryption of $\mathsf{I2B}(j)$ via the randomized McEliece encryption. Specifically, the secret witness of $\mathcal{P}$ is a tuple

$$(j, \mathbf{s}, \mathbf{u}, \mathbf{e}) \in [0, N-1] \times \mathbb{F}_2^m \times \mathbb{F}_2^{k-\ell} \times \mathbb{F}_2^n$$

that satisfies

$$\begin{cases} \mathbf{H} \cdot \mathbf{s}^\top = \mathbf{y}_j^\top \;\wedge\; \mathbf{s} \in \mathsf{B}(m, \omega), \\ (\mathbf{u} \| \mathsf{I2B}(j)) \cdot \mathbf{G} \oplus \mathbf{e} = \mathbf{c} \;\wedge\; \mathbf{e} \in \mathsf{B}(n, t). \end{cases} \qquad (6)$$

Let $\mathbf{A} = [\mathbf{y}_0^\top | \cdots | \mathbf{y}_j^\top | \cdots | \mathbf{y}_{N-1}^\top] \in \mathbb{F}_2^{r \times N}$ and $\mathbf{x} = \delta_j^N$. We have $\mathbf{A} \cdot \mathbf{x}^\top = \mathbf{y}_j^\top$ and rewrite

$$\mathbf{H} \cdot \mathbf{s}^\top = \mathbf{y}_j^\top \text{ as } \mathbf{H} \cdot \mathbf{s}^\top \oplus \mathbf{A} \cdot \mathbf{x}^\top = \mathbf{0}.$$

Let $\widehat{\mathbf{G}} \in \mathbb{F}_2^{(k+\ell) \times n}$ be the matrix obtained from $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ by replacing its last ℓ rows $\mathbf{g}_{k-\ell+1}, \mathbf{g}_{k-\ell+2}, \ldots, \mathbf{g}_k$ by the 2ℓ rows $\mathbf{0}^n, \mathbf{g}_{k-\ell+1}, \mathbf{0}^n, \mathbf{g}_{k-\ell+2}, \ldots, \mathbf{0}^n, \mathbf{g}_k$. We then observe that $(\mathbf{u} \| \mathsf{I2B}(j)) \cdot \mathbf{G} = (\mathbf{u} \| \mathsf{Encode}(j)) \cdot \widehat{\mathbf{G}}$.

Letting $\mathbf{f} = \mathsf{Encode}(j)$, we can equivalently rewrite (6) as

$$\begin{cases} \mathbf{H} \cdot \mathbf{s}^\top \oplus \mathbf{A} \cdot \mathbf{x}^\top = \mathbf{0} \;\wedge\; \mathbf{x} = \delta_j^N \;\wedge\; \mathbf{s} \in \mathsf{B}(m, \omega), \\ (\mathbf{u} \| \mathbf{f}) \cdot \widehat{\mathbf{G}} \oplus \mathbf{e} = \mathbf{c} \;\wedge\; \mathbf{f} = \mathsf{Encode}(j) \;\wedge\; \mathbf{e} \in \mathsf{B}(n, t). \end{cases} \qquad (7)$$

To obtain a ZKAoK for relation (7) in Stern's framework [20], $\mathcal{P}$ proceeds as follows.

To prove that $\mathbf{x} = \delta_j^N$ *and* $\mathbf{f} = \mathsf{Encode}(j)$ while keeping j secret, $\mathcal{P}$ samples a uniformly random vector $\mathbf{b} \in \{0, 1\}^\ell$, sends $\mathbf{b}_1 = \mathsf{I2B}(j) \oplus \mathbf{b}$, and shows that

$$T_\mathbf{b}(\mathbf{x}) = \delta_{\mathsf{B2I}(\mathbf{b}_1)}^N \;\wedge\; T_\mathbf{b}'(\mathbf{f}) = \mathsf{Encode}(\mathsf{B2I}(\mathbf{b}_1)).$$

By the equivalences observed in (4) and (5), the verifier will be convinced about the facts to prove. Furthermore,

1) **Commitment:** $\mathcal{P}$ samples the uniformly random objects

$$\mathbf{b} \xleftarrow{\$} \{0,1\}^\ell, \ \pi \xleftarrow{\$} \mathsf{S}_m, \ \sigma \xleftarrow{\$} \mathsf{S}_n, \ \rho_1, \rho_2, \rho_3 \xleftarrow{\$} \{0,1\}^\lambda, \ \mathbf{r_s} \xleftarrow{\$} \mathbb{F}_2^m, \ \mathbf{r_x} \xleftarrow{\$} \mathbb{F}_2^N, \ \mathbf{r_u} \xleftarrow{\$} \mathbb{F}_2^{k-\ell}, \ \mathbf{r_f} \xleftarrow{\$} \mathbb{F}_2^{2\ell}, \ \mathbf{r_e} \xleftarrow{\$} \mathbb{F}_2^n.$$

It then sends the commitment $\mathsf{CMT} := (c_1, c_2, c_3)$ to $\mathcal{V}$, where

$$c_1 = \mathsf{COM}\big(\mathbf{b}, \ \pi, \ \sigma, \ \mathbf{H} \cdot \mathbf{r_s}^\top \oplus \mathbf{A} \cdot \mathbf{r_x}^\top, \ \big(\mathbf{r_u} \,\|\, \mathbf{r_f}\big) \cdot \widehat{\mathbf{G}} \oplus \mathbf{r_e}; \ \rho_1\big),$$
$$c_2 = \mathsf{COM}\big(\pi(\mathbf{r_s}), T_\mathbf{b}(\mathbf{r_x}), T'_\mathbf{b}(\mathbf{r_f}), \sigma(\mathbf{r_e}); \ \rho_2\big),$$
$$c_3 = \mathsf{COM}\big(\pi(\mathbf{s} \oplus \mathbf{r_s}), T_\mathbf{b}(\mathbf{x} \oplus \mathbf{r_x}), T'_\mathbf{b}(\mathbf{f} \oplus \mathbf{r_f}), \sigma(\mathbf{e} \oplus \mathbf{r_e}); \ \rho_3\big).$$

2) **Challenge:** Upon receiving CMT, $\mathcal{V}$ sends a challenge $\mathsf{Ch} \xleftarrow{\$} \{1, 2, 3\}$ to $\mathcal{P}$.
3) **Response:** $\mathcal{P}$ responds based on Ch.
   a) If $\mathsf{Ch} = 1$: Reveal c_2 and c_3. Let

   $$\mathbf{b}_1 = \mathsf{I2B}(j) \oplus \mathbf{b}, \ \mathbf{v_s} = \pi(\mathbf{r_s}), \ \mathbf{w_s} = \pi(\mathbf{s}), \ \mathbf{v_x} = T_\mathbf{b}(\mathbf{r_x}), \ \mathbf{v_f} = T'_\mathbf{b}(\mathbf{r_f}), \ \mathbf{v_e} = \sigma(\mathbf{r_e}), \ \text{and} \ \mathbf{w_e} = \sigma(\mathbf{e}).$$

   Send $\mathsf{RSP} := \big(\mathbf{b}_1, \ \mathbf{v_s}, \ \mathbf{w_s}, \ \mathbf{v_x}, \ \mathbf{v_f}, \ \mathbf{v_e}, \ \mathbf{w_e}; \ \rho_2, \rho_3\big)$ to $\mathcal{V}$.
   b) If $\mathsf{Ch} = 2$: Reveal c_1 and c_3. Let

   $$\mathbf{b}_2 = \mathbf{b}, \ \pi_2 = \pi, \ \sigma_2 = \sigma, \ \mathbf{z_s} = \mathbf{s} \oplus \mathbf{r_s}, \ \mathbf{z_x} = \mathbf{x} \oplus \mathbf{r_x}, \ \mathbf{z_u} = \mathbf{u} \oplus \mathbf{r_u}, \ \mathbf{z_f} = \mathbf{f} \oplus \mathbf{r_f}, \ \text{and} \ \mathbf{z_e} = \mathbf{e} \oplus \mathbf{r_e}.$$

   Send $\mathsf{RSP} := \big(\mathbf{b}_2, \ \pi_2, \ \sigma_2, \ \mathbf{z_s}, \ \mathbf{z_x}, \ \mathbf{z_u}, \ \mathbf{z_f}, \ \mathbf{z_e}; \ \rho_1, \rho_3\big)$ to $\mathcal{V}$.
   c) If $\mathsf{Ch} = 3$: Reveal c_1 and c_2. Let

   $$\mathbf{b}_3 = \mathbf{b}, \ \pi_3 = \pi, \ \sigma_3 = \sigma, \ \mathbf{y_s} = \mathbf{r_s}, \ \mathbf{y_x} = \mathbf{r_x}, \ \mathbf{y_u} = \mathbf{r_u}, \ \mathbf{y_f} = \mathbf{r_f}, \ \text{and} \ \mathbf{y_e} = \mathbf{r_e}.$$

   Send $\mathsf{RSP} := \big(\mathbf{b}_3, \ \pi_3, \ \sigma_3, \ \mathbf{y_s}, \ \mathbf{y_x}, \ \mathbf{y_u}, \ \mathbf{y_f}, \ \mathbf{y_e}; \ \rho_1, \rho_2\big)$ to $\mathcal{V}$.
4) **Verification:** Upon receiving RSP, $\mathcal{V}$ proceeds based on Ch.
   a) If $\mathsf{Ch} = 1$: Let $\mathbf{w_x} = \delta_{\mathsf{B2I}(\mathbf{b}_1)}^N \in \mathbb{F}_2^N$ and $\mathbf{w_f} = \mathsf{Encode}(\mathsf{B2I}(\mathbf{b}_1)) \in \mathbb{F}_2^{2\ell}$.
   Check that $\mathbf{w_s} \in \mathsf{B}(m, \omega)$, $\mathbf{w_e} \in \mathsf{B}(n, t)$,

   $$c_2 = \mathsf{COM}\big(\mathbf{v_s}, \ \mathbf{v_x}, \ \mathbf{v_f}, \ \mathbf{v_e}; \ \rho_2\big), \ \text{and} \ c_3 = \mathsf{COM}\big(\mathbf{v_s} \oplus \mathbf{w_s}, \ \mathbf{v_x} \oplus \mathbf{w_x}, \ \mathbf{v_f} \oplus \mathbf{w_f}, \ \mathbf{v_e} \oplus \mathbf{w_e}; \ \rho_3\big).$$

   b) If $\mathsf{Ch} = 2$: Check that

   $$c_1 = \mathsf{COM}\big(\mathbf{b}_2, \ \pi_2, \ \sigma_2, \ \mathbf{H} \cdot \mathbf{z_s}^\top \oplus \mathbf{A} \cdot \mathbf{z_x}^\top, \ \big(\mathbf{z_u} \,\|\, \mathbf{z_f}\big) \cdot \widehat{\mathbf{G}} \oplus \mathbf{z_e} \oplus \mathbf{c}; \ \rho_1\big) \ \text{and}$$
   $$c_3 = \mathsf{COM}\big(\pi_2(\mathbf{z_s}), \ T_{\mathbf{b}_2}(\mathbf{z_x}), \ T'_{\mathbf{b}_2}(\mathbf{z_f}), \ \sigma_2(\mathbf{z_e}); \ \rho_3\big).$$

   c) If $\mathsf{Ch} = 3$: Check that

   $$c_1 = \mathsf{COM}\big(\mathbf{b}_3, \ \pi_3, \ \sigma_3, \ \mathbf{H} \cdot \mathbf{y_s}^\top \oplus \mathbf{A} \cdot \mathbf{y_x}^\top, \ \big(\mathbf{y_u} \,\|\, \mathbf{y_f}\big) \cdot \widehat{\mathbf{G}} \oplus \mathbf{y_e}; \ \rho_1\big) \ \text{and}$$
   $$c_2 = \mathsf{COM}\big(\pi_3(\mathbf{y_s}), \ T_{\mathbf{b}_3}(\mathbf{y_x}), \ T'_{\mathbf{b}_3}(\mathbf{y_f}), \ \sigma_3(\mathbf{y_e}); \ \rho_2\big).$$

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

Fig. 1. The underlying ZK protocol of the CPA-anonymous group signature.

since $\mathbf{b}$ essentially acts as a one-time pad, the secret j remains perfectly hidden.

To prove in zero-knowledge that $\mathbf{s} \in \mathsf{B}(m, \omega)$, $\mathcal{P}$ samples a uniformly random permutation $\pi \in \mathsf{S}_m$, and shows that $\pi(\mathbf{s}) \in \mathsf{B}(m, \omega)$. Similarly, to prove in zero-knowledge that $\mathbf{e} \in \mathsf{B}(n, t)$, a uniformly random permutation $\sigma \in \mathsf{S}_n$ is employed. Finally, to prove the linear equations in zero-knowledge, $\mathcal{P}$ samples uniformly random "masking" vectors $(\mathbf{r_s}, \mathbf{r_x}, \mathbf{r_u}, \mathbf{r_f}, \mathbf{r_e})$ and shows that

$$\mathbf{H} \cdot (\mathbf{s} \oplus \mathbf{r_s})^\top \oplus \mathbf{A} \cdot (\mathbf{x} \oplus \mathbf{r_x})^\top = \mathbf{H} \cdot \mathbf{r_s}^\top \oplus \mathbf{A} \cdot \mathbf{r_x}^\top \ \text{and}$$
$$\big(\mathbf{u} \oplus \mathbf{r_u} \| \mathbf{f} \oplus \mathbf{r_f}\big) \cdot \widehat{\mathbf{G}} \oplus (\mathbf{e} \oplus \mathbf{r_e}) \oplus \mathbf{c} = \big(\mathbf{r_u} \| \mathbf{r_f}\big) \cdot \widehat{\mathbf{G}} \oplus \mathbf{r_e}.$$
$$\tag{8}$$

Now let $\mathsf{COM} : \{0,1\}^* \to \{0,1\}^\lambda$ be a collision-resistant hash function, to be modelled as a random oracle. The prover

$\mathcal{P}$ and the verifier $\mathcal{V}$ first perform the preparatory steps described above, and then interact as described in Figure 1.

Analysis of the Protocol. The following theorem summarizes the properties of our protocol.

Theorem 2: The interactive protocol described in Section III-A has perfect completeness. Its communication cost is bounded above by $\beta = (N + 3 \log N) + m(\log m + 1) + n(\log n + 1) + k + 5\lambda$ bits. If COM is modelled as a random oracle, then the protocol is statistical zero-knowledge. If COM is a collision-resistant hash function, then the protocol is an argument of knowledge.

The given interactive protocol is perfectly **complete**, *i.e.*, if $\mathcal{P}$ possesses a valid witness $(j, \mathbf{s}, \mathbf{u}, \mathbf{e})$ and follows the protocol, then $\mathcal{V}$ always outputs 1. Indeed, given $(j, \mathbf{s}, \mathbf{u}, \mathbf{e})$ satisfying (6), $\mathcal{P}$ can always obtain $(j, \mathbf{s}, \mathbf{x}, \mathbf{u}, \mathbf{f}, \mathbf{e})$

satisfying (7). Then, as discussed above, the following three assertions hold.

$$\forall\ \pi \in \mathsf{S}_m :\ \pi(\mathbf{s}) \in \mathsf{B}(m,\omega),$$
$$\forall\ \sigma \in \mathsf{S}_n :\ \sigma(\mathbf{e}) \in \mathsf{B}(n,t),$$
$$\forall\ \mathbf{b} \in \{0,1\}^\ell :\ T_{\mathbf{b}}(\mathbf{x}) = \delta^N_{\mathsf{B2I}(\mathsf{I2B}(j)\oplus\mathbf{b})} = \mathbf{w_x} \text{ and}$$
$$T'_{\mathbf{b}}(\mathbf{f}) = \mathsf{Encode}(\mathsf{B2I}(\mathsf{I2B}(j)\oplus \mathbf{b})) = \mathbf{w_f}.$$

Thus, $\mathcal{P}$ always passes $\mathcal{V}$'s checks when $\mathrm{Ch} = 1$. When $\mathrm{Ch} = 2$, $\mathcal{P}$ also passes the verification, since the linear equations in (8) hold true. Finally, for $\mathrm{Ch} = 3$, it suffices to note that $\mathcal{V}$ simply checks for honest computations of c_1 and c_2.

Let us now consider the **communication cost**. The commitment CMT has bit-size 3λ. If $\mathrm{Ch} = 1$, then the response RSP has bit-size

$$3\ell + N + 2(m + n + \lambda).$$

If $\mathrm{Ch} = 2$ or $\mathrm{Ch} = 3$, then RSP has bit-size

$$2\ell + N + m(\log m + 1) + n(\log n + 1) + k + 2\lambda.$$

Therefore, the protocol's total communication cost, in bits, is less than the specified bound β.

The following lemma says that our interactive protocol is **statistically zero-knowledge** if COM is modelled as a random oracle. It employs the standard simulation technique for Stern-type protocols as was done, *e.g.*, in [20], [63], and [64].

Lemma 3: In the random oracle model, there exists an efficient simulator $\mathcal{S}$ interacting with a (possibly cheating) verifier $\widehat{\mathcal{V}}$, such that, given only the public input of the protocol, $\mathcal{S}$ 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.

Proof: Simulator $\mathcal{S}$, given the public input $(\mathbf{H}, \mathbf{A}, \widehat{\mathbf{G}}, \mathbf{c})$, begins by selecting a random $\overline{\mathrm{Ch}} \in \{1, 2, 3\}$. This is a prediction of the challenge value that $\widehat{\mathcal{V}}$ will *not* choose.

1) **Case $\overline{\mathrm{Ch}} = 1$:** $\mathcal{S}$ proceeds as follows.
   a) Compute $\mathbf{s}' \in \mathbb{F}_2^m$ and $\mathbf{x}' \in \mathbb{F}_2^N$ such that $\mathbf{H} \cdot \mathbf{s}'^\top \oplus \mathbf{A} \cdot \mathbf{x}'^\top = \mathbf{0}$. Compute $\mathbf{u}' \in \mathbb{F}_2^{k-\ell}$, $\mathbf{f}' \in \mathbb{F}_2^{2\ell}$, $\mathbf{e}' \in \mathbb{F}_2^n$ such that $(\mathbf{u}' \,\|\, \mathbf{f}') \cdot \widehat{\mathbf{G}} \oplus \mathbf{e}' = \mathbf{c}$. These steps can be done efficiently by using linear algebraic tools.
   b) Sample uniformly random objects, and send a commitment computed in the same manner as of the real prover. More explicitly, $\mathcal{S}$ samples

$$\mathbf{b} \overset{\$}{\leftarrow} \{0,1\}^\ell, \quad \pi \overset{\$}{\leftarrow} \mathsf{S}_m, \quad \sigma \overset{\$}{\leftarrow} \mathsf{S}_n,$$
$$\rho_1, \rho_2, \rho_3 \overset{\$}{\leftarrow} \{0,1\}^\lambda, \quad \mathbf{r_s} \overset{\$}{\leftarrow} \mathbb{F}_2^m, \quad \mathbf{r_x} \overset{\$}{\leftarrow} \mathbb{F}_2^N,$$
$$\mathbf{r_u} \overset{\$}{\leftarrow} \mathbb{F}_2^{k-\ell}, \quad \mathbf{r_f} \overset{\$}{\leftarrow} \mathbb{F}_2^{2\ell}, \quad \mathbf{r_e} \overset{\$}{\leftarrow} \mathbb{F}_2^n.$$

   and sends the commitment $\mathsf{CMT} := (c_1', c_2', c_3')$, where

$$c_1' = \mathsf{COM}\big(\mathbf{b}, \pi, \sigma,\ \mathbf{H} \cdot \mathbf{r_s}^\top \oplus \mathbf{A} \cdot \mathbf{r_x}^\top,$$
$$(\mathbf{r_u} \,\|\, \mathbf{r_f}) \cdot \widehat{\mathbf{G}} \oplus \mathbf{r_e};\ \rho_1\big),$$
$$c_2' = \mathsf{COM}\big(\pi(\mathbf{r_s}), T_{\mathbf{b}}(\mathbf{r_x}),\ T'_{\mathbf{b}}(\mathbf{r_f}), \sigma(\mathbf{r_e});\ \rho_2\big),$$

$$c_3' = \mathsf{COM}\big(\pi(\mathbf{s}'\oplus\mathbf{r_s}), T_{\mathbf{b}}(\mathbf{x}'\oplus\mathbf{r_x}), T'_{\mathbf{b}}(\mathbf{f}'\oplus\mathbf{r_f}),$$
$$\sigma(\mathbf{e}'\oplus\mathbf{r_e});\ \rho_3\big). \tag{9}$$

Upon receiving a challenge Ch from $\widehat{\mathcal{V}}$, the simulator responds accordingly.
   a) If $\mathrm{Ch} = 1$: Output $\bot$ and abort.
   b) If $\mathrm{Ch} = 2$: Send

$$\mathsf{RSP} = \big(\mathbf{b}, \pi, \sigma, \mathbf{s}' \oplus \mathbf{r_s}, \mathbf{x}' \oplus \mathbf{r_x},$$
$$\mathbf{u}'\oplus\mathbf{r_u}, \mathbf{f}'\oplus\mathbf{r_f}, \mathbf{e}'\oplus\mathbf{r_e};\ \rho_1, \rho_3\big).$$

   c) If $\mathrm{Ch} = 3$: Send

$$\mathsf{RSP} = \big(\mathbf{b}, \pi, \sigma, \mathbf{r_s}, \mathbf{r_x}, \mathbf{r_u}, \mathbf{r_f}, \mathbf{r_e};\ \rho_1, \rho_2\big).$$

2) **Case $\overline{\mathrm{Ch}} = 2$:** $\mathcal{S}$ samples

$$j' \overset{\$}{\leftarrow} [0, N-1], \quad \mathbf{s}' \overset{\$}{\leftarrow} \mathsf{B}(m,\omega), \quad \mathbf{e}' \overset{\$}{\leftarrow} \mathsf{B}(n,t),$$
$$\mathbf{b} \overset{\$}{\leftarrow} \{0,1\}^\ell, \quad \pi \overset{\$}{\leftarrow} \mathsf{S}_m, \quad \sigma \overset{\$}{\leftarrow} \mathsf{S}_n,$$
$$\rho_1, \rho_2, \rho_3 \overset{\$}{\leftarrow} \{0,1\}^\lambda, \quad \mathbf{r_s} \overset{\$}{\leftarrow} \mathbb{F}_2^m, \quad \mathbf{r_x} \overset{\$}{\leftarrow} \mathbb{F}_2^N,$$
$$\mathbf{r_u} \overset{\$}{\leftarrow} \mathbb{F}_2^{k-\ell}, \quad \mathbf{r_f} \overset{\$}{\leftarrow} \mathbb{F}_2^{2\ell}, \quad \mathbf{r_e} \overset{\$}{\leftarrow} \mathbb{F}_2^n.$$

It also lets $\mathbf{x}' = \delta_{j'}^N$ and $\mathbf{f}' = \mathsf{Encode}(j')$. Then $\mathcal{S}$ sends the commitment CMT computed as in (9).

Upon receiving a challenge Ch from $\widehat{\mathcal{V}}$, it responds as follows.
   a) If $\mathrm{Ch} = 1$: Send

$$\mathsf{RSP} = \big(\mathsf{I2B}(j') \oplus \mathbf{b}, \pi(\mathbf{r_s}), \pi(\mathbf{s}'),$$
$$T_{\mathbf{b}}(\mathbf{r_x}), T'_{\mathbf{b}}(\mathbf{r_f}), \sigma(\mathbf{r_e}), \sigma(\mathbf{e}');\ \rho_2, \rho_3\big).$$

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

3) **Case $\overline{\mathrm{Ch}} = 3$:** The simulator performs the preparation as in the case $\overline{\mathrm{Ch}} = 2$ above as well as samples $\mathbf{u}' \overset{\$}{\leftarrow} \mathbb{F}_2^{k-\ell}$. It then sends the commitment $\mathsf{CMT} := (c_1', c_2', c_3')$, where c_2', c_3' are computed as in (9) and

$$c_1' = \mathsf{COM}\big(\mathbf{b}, \pi, \sigma, \mathbf{H} \cdot (\mathbf{s}' \oplus \mathbf{r_s})^\top \oplus \mathbf{A} \cdot (\mathbf{x}' \oplus \mathbf{r_x})^\top,$$
$$(\mathbf{u}' \oplus \mathbf{r_u} \,\|\, \mathbf{f}' \oplus \mathbf{r_f}) \cdot \widehat{\mathbf{G}} \oplus (\mathbf{e}' \oplus \mathbf{r_e}) \oplus \mathbf{c}; \rho_1\big).$$

Upon receiving a challenge Ch from $\widehat{\mathcal{V}}$, it responds as follows.
   a) If $\mathrm{Ch} = 1$: Send RSP computed as in the case $(\overline{\mathrm{Ch}} = 2, \mathrm{Ch} = 1)$.
   b) If $\mathrm{Ch} = 2$: Send RSP computed as in the case $(\overline{\mathrm{Ch}} = 1, \mathrm{Ch} = 2)$.
   c) If $\mathrm{Ch} = 3$: Output $\bot$ and abort.

In every case that we have considered above, 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, since the outputs of the random oracle COM are assumed to be uniformly random. Hence, the probability that $\mathcal{S}$ outputs $\bot$ is negligibly close to $1/3$. Moreover, one can check that whenever the simulator does not abort, it provides a successful transcript, whose distribution is statistically close to that of the prover in the real interaction. We have thus constructed a

simulator that can successfully impersonate the honest prover, with probability $2/3$. ∎

The next lemma establishes that our protocol satisfies the special soundness property of Σ-protocols. This implies, as had been shown in [65], that the protocol is indeed an **argument of knowledge**.

Lemma 4: Let COM be a collision-resistant hash function. Given the public input of the protocol, a commitment CMT and three valid responses $\mathsf{RSP}_1, \mathsf{RSP}_2, \mathsf{RSP}_3$ to all three possible values of the challenge Ch, one can efficiently construct a knowledge extractor $\mathcal{E}$ that outputs a tuple $(j', \mathbf{s}', \mathbf{u}', \mathbf{e}') \in [0, N-1] \times \mathbb{F}_2^m \times \mathbb{F}_2^{k-\ell} \times \mathbb{F}_2^n$ that simultaneously satisfies $\mathbf{H} \cdot \mathbf{s}'^\top = \mathbf{y}_{j'}^\top$, $\mathbf{s}' \in \mathsf{B}(m, \omega)$, $(\mathbf{u}' \| \mathsf{I2B}(j')) \cdot \mathbf{G} \oplus \mathbf{e}' = \mathbf{c}$, and $\mathbf{e}' \in \mathsf{B}(n, t)$.

Proof: Assume that we have $\mathsf{CMT} = (c_1, c_2, c_3)$ and the three responses

$$\mathsf{RSP}_1 = (\mathbf{b}_1, \mathbf{v_s}, \mathbf{w_s}, \mathbf{v_x}, \mathbf{v_f}, \mathbf{v_e}, \mathbf{w_e}; \rho_2, \rho_3),$$
$$\mathsf{RSP}_2 = (\mathbf{b}_2, \pi_2, \sigma_2, \mathbf{z_s}, \mathbf{z_x}, \mathbf{z_u}, \mathbf{z_f}, \mathbf{z_e}; \rho_1, \rho_3),$$
$$\mathsf{RSP}_3 = (\mathbf{b}_3, \pi_3, \sigma_3, \mathbf{y_s}, \mathbf{y_x}, \mathbf{y_u}, \mathbf{y_f}, \mathbf{y_e}; \rho_1, \rho_2),$$

that satisfy all of the verification conditions when $\mathsf{Ch} = 1$, $\mathsf{Ch} = 2$, and $\mathsf{Ch} = 3$, respectively. More explicitly, we have

$$
\begin{aligned}
\mathbf{w_s} \in\ & \mathsf{B}(m, \omega), \quad \mathbf{w_x} = \delta^N_{\mathsf{B2I}(\mathbf{b}_1)}, \\
\mathbf{w_f} =\ & \mathsf{Encode}(\mathsf{B2I}(\mathbf{b}_1)), \quad \mathbf{w_e} \in \mathsf{B}(n, t), \\
c_1 =\ & \mathsf{COM}\big(\mathbf{b}_2, \pi_2, \sigma_2, \mathbf{H} \cdot \mathbf{z_s}^\top \oplus \mathbf{A} \cdot \mathbf{z_x}^\top, \\
& \quad (\mathbf{z_u} \| \mathbf{z_f}) \cdot \widehat{\mathbf{G}} \oplus \mathbf{z_e} \oplus \mathbf{c}; \rho_1\big) \\
=\ & \mathsf{COM}\big(\mathbf{b}_3, \pi_3, \sigma_3, \mathbf{H} \cdot \mathbf{y_s}^\top \oplus \mathbf{A} \cdot \mathbf{y_x}^\top, \\
& \quad (\mathbf{y_u} \| \mathbf{y_f}) \cdot \widehat{\mathbf{G}} \oplus \mathbf{y_e}; \rho_1\big), \\
c_2 =\ & \mathsf{COM}\big(\mathbf{v_s}, \mathbf{v_x}, \mathbf{v_f}, \mathbf{v_e}; \rho_2\big) \\
=\ & \mathsf{COM}\big(\pi_3(\mathbf{y_s}), T_{\mathbf{b}_3}(\mathbf{y_x}), T'_{\mathbf{b}_3}(\mathbf{y_f}), \sigma_3(\mathbf{y_e}); \rho_2\big), \\
c_3 =\ & \mathsf{COM}\big(\mathbf{v_s} \oplus \mathbf{w_s}, \mathbf{v_x} \oplus \mathbf{w_x}, \mathbf{v_f} \oplus \mathbf{w_f}, \mathbf{v_e} \oplus \mathbf{w_e};\ \rho_3\big) \\
=\ & \mathsf{COM}\big(\pi_2(\mathbf{z_s}), T_{\mathbf{b}_2}(\mathbf{z_x}), T'_{\mathbf{b}_2}(\mathbf{z_f}), \sigma_2(\mathbf{z_e});\ \rho_3\big).
\end{aligned}
$$

Based on the collision-resistance property of COM, we infer

$$
\begin{aligned}
& \mathbf{b}_2 = \mathbf{b}_3, \quad \pi_2 = \pi_3, \quad \sigma_2 = \sigma_3, \\
& \delta^N_{\mathsf{B2I}(\mathbf{b}_1)} = \mathbf{w_x} = T_{\mathbf{b}_2}(\mathbf{z_x}) \oplus T_{\mathbf{b}_3}(\mathbf{y_x}) = T_{\mathbf{b}_2}(\mathbf{z_x} \oplus \mathbf{y_x}), \\
& \mathsf{Encode}(\mathsf{B2I}(\mathbf{b}_1)) = \mathbf{w_f} = T'_{\mathbf{b}_2}(\mathbf{z_f}) \oplus T'_{\mathbf{b}_3}(\mathbf{y_f}) \\
& \hspace{4cm} = T'_{\mathbf{b}_2}(\mathbf{z_f} \oplus \mathbf{y_f}), \\
& \mathsf{B}(m, \omega) \ni \mathbf{w_s} = \pi_2(\mathbf{z_s}) \oplus \pi_3(\mathbf{y_s}) = \pi_2(\mathbf{z_s} \oplus \mathbf{y_s}), \\
& \mathsf{B}(n, t) \ni \mathbf{w_e} = \sigma_2(\mathbf{z_e}) \oplus \sigma_3(\mathbf{y_e}) = \sigma_2(\mathbf{z_e} \oplus \mathbf{y_e}), \\
& \mathbf{H} \cdot (\mathbf{z_s} \oplus \mathbf{y_s})^\top \oplus \mathbf{A} \cdot (\mathbf{z_x} \oplus \mathbf{y_x})^\top = \mathbf{0}, \text{ and} \\
& (\mathbf{z_u} \oplus \mathbf{y_u} \| \mathbf{z_f} \oplus \mathbf{y_f}) \cdot \widehat{\mathbf{G}} \oplus (\mathbf{z_e} \oplus \mathbf{y_e}) = \mathbf{c}.
\end{aligned}
$$

Letting

$$j' = \mathsf{B2I}(\mathbf{b}_1 \oplus \mathbf{b}_2) \in [0, N-1] \text{ and } \mathbf{x}' = \mathbf{z_x} \oplus \mathbf{y_x} \in \mathbb{F}_2^N,$$

we derive $\mathbf{x}' = \delta^N_{j'}$ by using (4). Thus, $\mathbf{A} \cdot \mathbf{x}'^\top = \mathbf{y}_{j'}^\top$. Let $\mathbf{f}' = \mathbf{z_f} \oplus \mathbf{y_f} \in \mathbb{F}_2^\ell$. Then, by (5), we have $\mathbf{f}' = \mathsf{Encode}(j')$. Let $\mathbf{s}' = \mathbf{z_s} \oplus \mathbf{y_s} \in \mathbb{F}_2^m$. We have $\mathbf{s}' = \pi_2^{-1}(\mathbf{w_s}) \in \mathsf{B}(m, \omega)$. Let $\mathbf{e}' = \mathbf{z_e} \oplus \mathbf{y_e} \in \mathbb{F}_2^n$. We get $\mathbf{e}' = \sigma_2^{-1}(\mathbf{w_e}) \in \mathsf{B}(n, t)$. Let $\mathbf{u}' = \mathbf{z_u} \oplus \mathbf{y_u} \in \mathbb{F}_2^{k-\ell}$. Furthermore, we obtain

$$\mathbf{H} \cdot \mathbf{s}'^\top \oplus \mathbf{A} \cdot \mathbf{x}'^\top = \mathbf{0} \text{ and } (\mathbf{u}' \| \mathsf{Encode}(j')) \cdot \widehat{\mathbf{G}} \oplus \mathbf{e}' = \mathbf{c}.$$

They imply, respectively, that

$$\mathbf{H} \cdot \mathbf{s}'^\top = \mathbf{A} \cdot \mathbf{x}'^\top = \mathbf{y}_{j'}^\top \text{ and } (\mathbf{u}' \| \mathsf{I2B}(j')) \cdot \mathbf{G} \oplus \mathbf{e}' = \mathbf{c}.$$

We have thus constructed an efficient extractor $\mathcal{E}$ that outputs

$$(j', \mathbf{s}', \mathbf{u}', \mathbf{e}') \in [0, N-1] \times \mathbb{F}_2^m \times \mathbb{F}_2^{k-\ell} \times \mathbb{F}_2^n,$$

satisfying

$$
\begin{aligned}
& \mathbf{H} \cdot \mathbf{s}'^\top = \mathbf{y}_{j'}^\top, \quad \mathbf{s}' \in \mathsf{B}(m, \omega), \\
& (\mathbf{u}' \| \mathsf{I2B}(j')) \cdot \mathbf{G} \oplus \mathbf{e}' = \mathbf{c}, \text{ and } \mathbf{e}' \in \mathsf{B}(n, t).
\end{aligned}
$$

This completes the proof. ∎

B. The Interactive Protocol Underlying the CCA-Anonymous Group Signature

The required ZK protocol for our CCA-anonymous group signature scheme of Section V is a simple extension of the one underlying the CPA-anonymous group signature that we have described in Section III-A. Here, we handle two ciphertexts $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ of $\mathsf{I2B}(j)$ by executing two instances of the techniques used for handling one ciphertext $\mathbf{c}$ from Section III-A.

Applying the same transformations as in Section III-A, we can translate the statement to be proved to proving knowledge of $\mathbf{s}, \mathbf{x}, \{\mathbf{u}^{(i)}\}_{i \in [2]}, \mathbf{f}, \{\mathbf{e}^{(i)}\}_{i \in [2]}$ such that

$$
\begin{aligned}
& \mathbf{H} \cdot \mathbf{s}^\top \oplus \mathbf{A} \cdot \mathbf{x}^\top = \mathbf{0}, \quad \mathbf{x} = \delta^N_j, \quad \mathbf{s} \in \mathsf{B}(m, \omega), \\
& \{(\mathbf{u}^{(i)} \| \mathbf{f}) \cdot \widehat{\mathbf{G}}^{(i)} \oplus \mathbf{e}^{(i)} = \mathbf{c}^{(i)}\}_{i \in [2]}, \quad \mathbf{f} = \mathsf{Encode}(j), \\
& \{\mathbf{e}^{(i)} \in \mathsf{B}(n, t)\}_{i \in [2]},
\end{aligned}
$$

for public input $(\mathbf{H}, \mathbf{A}, \widehat{\mathbf{G}}^{(1)}, \widehat{\mathbf{G}}^{(2)}, \mathbf{c}^{(1)}, \mathbf{c}^{(2)})$.

A ZK argument for the obtained equivalent statement can then be obtained in Stern's framework, using the same permuting and masking techniques of Section III-A. The resulting interactive protocol is described in **Fig.** 2, where COM is a collision-resistant hash function modelled as a random oracle. The protocol is a statistical ZKAoK. Its simulator and extractor are constructed in the same manner as for the protocol in Section III-A. The details can therefore be safely omitted here.

IV. A CPA-ANONYMOUS CODE-BASED GROUP SIGNATURE

This section discusses our basic scheme that achieves CPA-anonimity. We start with a description of the scheme and end with the evaluation of its properties.

A. Description, Efficiency, and Correctness of the Scheme

Our group signature scheme consists of the following four algorithms.

1) $\mathsf{KeyGen}(1^\lambda, 1^N)$: On input a security parameter λ and an expected number of group users $N = 2^\ell \in \mathsf{poly}(\lambda)$, for some positive integer ℓ, this algorithm first selects the following parameters and hash functions.
   a) Parameters $n = n(\lambda), k = k(\lambda), t = t(\lambda)$ for a binary $[n, k, 2t+1]$ Goppa code.

1) **Commitment:** $\mathcal{P}$ samples the following uniformly random objects.

$$\mathbf{b} \xleftarrow{\$} \{0,1\}^\ell, \quad \pi \xleftarrow{\$} \mathsf{S}_m, \quad \sigma^{(1)}, \sigma^{(2)} \xleftarrow{\$} \mathsf{S}_n, \quad \rho_1, \rho_2, \rho_3 \xleftarrow{\$} \{0,1\}^\lambda,$$

$$\mathbf{r_s} \xleftarrow{\$} \mathbb{F}_2^m, \quad \mathbf{r_x} \xleftarrow{\$} \mathbb{F}_2^N, \quad \mathbf{r_u^{(1)}}, \mathbf{r_u^{(2)}} \xleftarrow{\$} \mathbb{F}_2^{k-\ell}, \quad \mathbf{r_f} \xleftarrow{\$} \mathbb{F}_2^{2\ell}, \quad \mathbf{r_e^{(1)}}, \mathbf{r_e^{(2)}} \xleftarrow{\$} \mathbb{F}_2^n.$$

It then sends the commitment $\mathrm{CMT} := (c_1, c_2, c_3)$ to $\mathcal{V}$, where

$$c_1 = \mathrm{COM}\Big(\mathbf{b}, \pi, \{\sigma^{(i)}\}_{i\in[2]}, \mathbf{H} \cdot \mathbf{r_s}^\top \oplus \mathbf{A} \cdot \mathbf{r_x}^\top, \{(\mathbf{r_u^{(i)}} \| \mathbf{r_f}) \cdot \widehat{\mathbf{G}}^{(i)} \oplus \mathbf{r_e^{(i)}}\}_{i\in[2]}; \ \rho_1\Big),$$

$$c_2 = \mathrm{COM}\Big(\pi(\mathbf{r_s}), T_\mathbf{b}(\mathbf{r_x}), T'_\mathbf{b}(\mathbf{r_f}), \{\sigma^{(i)}(\mathbf{r_e^{(i)}})\}_{i\in[2]}; \ \rho_2\Big),$$

$$c_3 = \mathrm{COM}\Big(\pi(\mathbf{s} \oplus \mathbf{r_s}), T_\mathbf{b}(\mathbf{x} \oplus \mathbf{r_x}), T'_\mathbf{b}(\mathbf{f} \oplus \mathbf{r_f}), \{\sigma^{(i)}(\mathbf{e}^{(i)} \oplus \mathbf{r_e^{(i)}})\}_{i\in[2]}; \ \rho_3\Big).$$

2) **Challenge:** Upon receiving CMT, $\mathcal{V}$ sends a challenge $\mathrm{Ch} \xleftarrow{\$} \{1, 2, 3\}$ to $\mathcal{P}$.

3) **Response:** $\mathcal{P}$ responds accordingly.

  a) If $\mathrm{Ch} = 1$: Reveal c_2 and c_3. Let $\mathbf{b}_1 = \mathsf{I2B}(j) \oplus \mathbf{b}$,

$$\mathbf{v_s} = \pi(\mathbf{r_s}), \quad \mathbf{w_s} = \pi(\mathbf{s}), \quad \mathbf{v_x} = T_\mathbf{b}(\mathbf{r_x}), \quad \mathbf{v_f} = T'_\mathbf{b}(\mathbf{r_f}), \quad \{\mathbf{v_e^{(i)}} = \sigma^{(i)}(\mathbf{r_e^{(i)}})\}_{i\in[2]}, \quad \{\mathbf{w_e^{(i)}} = \sigma^{(i)}(\mathbf{e}^{(i)})\}_{i\in[2]}.$$

  Send $\mathrm{RSP} := \Big(\mathbf{b}_1, \mathbf{v_s}, \mathbf{w_s}, \mathbf{v_x}, \mathbf{v_f}, \{\mathbf{v_e^{(i)}}, \mathbf{w_e^{(i)}}\}_{i\in[2]}; \rho_2, \rho_3\Big)$ to $\mathcal{V}$.

  b) If $\mathrm{Ch} = 2$: Reveal c_1 and c_3. Let

$$\mathbf{b}_2 = \mathbf{b}, \quad \pi_2 = \pi, \quad \{\sigma_2^{(i)} = \sigma^{(i)}\}_{i\in[2]}, \quad \mathbf{z_s} = \mathbf{s} \oplus \mathbf{r_s}, \quad \mathbf{z_x} = \mathbf{x} \oplus \mathbf{r_x},$$

$$\{\mathbf{z_u^{(i)}} = \mathbf{u}^{(i)} \oplus \mathbf{r_u^{(i)}}\}_{i\in[2]}, \quad \mathbf{z_f} = \mathbf{f} \oplus \mathbf{r_f}, \quad \{\mathbf{z_e^{(i)}} = \mathbf{e}^{(i)} \oplus \mathbf{r_e^{(i)}}\}_{i\in[2]}.$$

  Send $\mathrm{RSP} := \Big(\mathbf{b}_2, \pi_2, \{\sigma_2^{(i)}\}_{i\in[2]}, \mathbf{z_s}, \mathbf{z_x}, \{\mathbf{z_u^{(i)}}\}_{i\in[2]}, \mathbf{z_f}, \{\mathbf{z_e^{(i)}}\}_{i\in[2]}; \rho_1, \rho_3\Big)$ to $\mathcal{V}$.

  c) If $\mathrm{Ch} = 3$: Reveal c_1 and c_2. Let $\mathbf{b}_3 = \mathbf{b}$,

$$\pi_3 = \pi, \quad \{\sigma_3^{(i)} = \sigma^{(i)}\}_{i\in[2]}, \quad \mathbf{y_s} = \mathbf{r_s}, \quad \mathbf{y_x} = \mathbf{r_x}, \quad \{\mathbf{y_u^{(i)}} = \mathbf{r_u^{(i)}}\}_{i\in[2]}, \quad \mathbf{y_f} = \mathbf{r_f}, \quad \{\mathbf{y_e^{(i)}} = \mathbf{r_e^{(i)}}\}_{i\in[2]}.$$

  Send $\mathrm{RSP} := \Big(\mathbf{b}_3, \pi_3, \{\sigma_3^{(i)}\}_{i\in[2]}, \mathbf{y_s}, \mathbf{y_x}, \{\mathbf{y_u^{(i)}}\}_{i\in[2]}, \mathbf{y_f}, \{\mathbf{y_e^{(i)}}\}_{i\in[2]}; \rho_1, \rho_2\Big)$ to $\mathcal{V}$.

4) **Verification:** Upon receiving RSP, $\mathcal{V}$ proceeds as follows.

  a) If $\mathrm{Ch} = 1$: Let $\mathbf{w_x} = \delta_{\mathsf{B2I}(\mathbf{b}_1)}^N \in \mathbb{F}_2^N$ and $\mathbf{w_f} = \mathsf{Encode}(\mathsf{B2I}(\mathbf{b}_1)) \in \mathbb{F}_2^{2\ell}$.

  Check that $\mathbf{w_s} \in \mathsf{B}(m, \omega)$ and $\left\{\mathbf{w_e^{(i)}} \in \mathsf{B}(n, t)\right\}_{i\in[2]}$, and that

$$c_2 = \mathrm{COM}\Big(\mathbf{v_s}, \ \mathbf{v_x}, \ \mathbf{v_f}, \ \{\mathbf{v_e^{(i)}}\}_{i\in[2]}; \ \rho_2\Big) \ \text{and}$$

$$c_3 = \mathrm{COM}\Big(\mathbf{v_s} \oplus \mathbf{w_s}, \ \mathbf{v_x} \oplus \mathbf{w_x}, \ \mathbf{v_f} \oplus \mathbf{w_f}, \ \{\mathbf{v_e^{(i)}} \oplus \mathbf{w_e^{(i)}}\}_{i\in[2]}; \ \rho_3\Big).$$

  b) If $\mathrm{Ch} = 2$: Check that

$$c_1 = \mathrm{COM}\Big(\mathbf{b}_2, \pi_2, \{\sigma_2^{(i)}\}_{i\in[2]}, \mathbf{H} \cdot \mathbf{z_s}^\top \oplus \mathbf{A} \cdot \mathbf{z_x}^\top, \{(\mathbf{z_u^{(i)}} \| \mathbf{z_f}) \cdot \widehat{\mathbf{G}}^{(i)} \oplus \mathbf{z_e^{(i)}} \oplus \mathbf{c}^{(i)}\}_{i\in[2]}; \rho_1\Big),$$

$$c_3 = \mathrm{COM}\Big(\pi_2(\mathbf{z_s}), \ T_{\mathbf{b}_2}(\mathbf{z_x}), \ T'_{\mathbf{b}_2}(\mathbf{z_f}), \ \{\sigma_2^{(i)}(\mathbf{z_e^{(i)}})\}_{i\in[2]}; \ \rho_3\Big).$$

  c) If $\mathrm{Ch} = 3$: Check that

$$c_1 = \mathrm{COM}\Big(\mathbf{b}_3, \pi_3, \{\sigma_3^{(i)}\}_{i\in[2]}, \mathbf{H} \cdot \mathbf{y_s}^\top \oplus \mathbf{A} \cdot \mathbf{y_x}^\top, \{(\mathbf{y_u^{(i)}} \| \mathbf{y_f}) \cdot \widehat{\mathbf{G}}^{(i)} \oplus \mathbf{y_e^{(i)}}\}_{i\in[2]}; \rho_1\Big),$$

$$c_2 = \mathrm{COM}\Big(\pi_3(\mathbf{y_s}), \ T_{\mathbf{b}_3}(\mathbf{y_x}), \ T'_{\mathbf{b}_3}(\mathbf{y_f}), \ \{\sigma_3^{(i)}(\mathbf{y_e^{(i)}})\}_{i\in[2]}; \ \rho_2\Big).$$

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

Fig. 2. The underlying ZK protocol of the CCA-anonymous group signature.

  b) Parameters $m = m(\lambda), r = r(\lambda), \omega = \omega(\lambda)$ for the Syndrome Decoding problem, such that

$$r \leq \log\binom{m}{w} - 2\lambda - \mathcal{O}(1). \qquad (10)$$

  c) Two collision-resistant hash functions, to be modelled as random oracles:

  i) $\mathrm{COM} : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ to generate zero-knowledge arguments.

ii) $\mathcal{H} : \{0,1\}^* \to \{1,2,3\}^\kappa$, where $\kappa = \omega \cdot \log \lambda$, for the Fiat-Shamir transformation.

The algorithm then performs the following steps.

a) Run $\mathsf{ME.KeyGen}(n,k,t)$ to obtain a key pair $\big(\mathsf{pk}_{\mathsf{ME}} = \mathbf{G} \in \mathbb{F}_2^{k \times n} \; ; \; \mathsf{sk}_{\mathsf{ME}}\big)$ for the randomized McEliece encryption scheme with respect to a binary $[n,k,2t+1]$ Goppa code. The plaintext space is $\mathbb{F}_2^\ell$.

b) Choose a matrix $\mathbf{H} \xleftarrow{\$} \mathbb{F}_2^{r \times m}$.

c) For each $j \in [0, N-1]$, pick $\mathbf{s}_j \xleftarrow{\$} \mathsf{B}(m,\omega)$ and let $\mathbf{y}_j \in \mathbb{F}_2^r$ be its syndrome, *i.e.*, $\mathbf{y}_j^\top = \mathbf{H} \cdot \mathbf{s}_j^\top$.

d) Output
$$\big(\mathsf{gpk} = (\mathbf{G}, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}), \; \mathsf{gmsk} = \mathsf{sk}_{\mathsf{ME}},$$
$$\mathsf{gsk} = (\mathbf{s}_0, \dots, \mathbf{s}_{N-1})\big). \quad (11)$$

2) $\mathsf{Sign}(\mathsf{gsk}[j], M)$: To sign a message $M \in \{0,1\}^*$ under gpk, the group user of index j, who possesses secret key $\mathbf{s} = \mathsf{gsk}[j]$, performs the following steps.

a) Encrypt the binary representation of j, *i.e.*, $\mathsf{I2B}(j) \in \mathbb{F}_2^\ell$, under the randomized McEliece encrypting key $\mathbf{G}$. This is done by sampling $(\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^{k-\ell}, \mathbf{e} \xleftarrow{\$} \mathsf{B}(n,t))$ and outputting the ciphertext
$$\mathbf{c} = \big(\mathbf{u} \,\|\, \mathsf{I2B}(j)\big) \cdot \mathbf{G} \oplus \mathbf{e} \in \mathbb{F}_2^n.$$

b) Generate a NIZKAoK Π to simultaneously prove, in zero-knowledge, the possession of an $\mathbf{s} \in \mathsf{B}(m,\omega)$ corresponding to a certain syndrome $\mathbf{y}_j \in \{\mathbf{y}_0, \dots, \mathbf{y}_{N-1}\}$ with hidden index j, and that $\mathbf{c}$ is a correct McEliece encryption of $\mathsf{I2B}(j)$. This is done by employing the interactive argument system in Section III-A with public input $(\mathbf{G}, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}, \mathbf{c})$, and prover's witness $(j, \mathbf{s}, \mathbf{u}, \mathbf{e})$ that satisfies
$$\mathbf{H} \cdot \mathbf{s}^\top = \mathbf{y}_j^\top, \quad \mathbf{s} \in \mathsf{B}(m,\omega),$$
$$\big(\mathbf{u} \,\|\, \mathsf{I2B}(j)\big) \cdot \mathbf{G} \oplus \mathbf{e} = \mathbf{c}, \quad \mathbf{e} \in \mathsf{B}(n,t). \quad (12)$$

The protocol is repeated $\kappa = \omega \cdot \log \lambda$ times to achieve negligible soundness error, before being made non-interactive by using the Fiat-Shamir heuristic. We have
$$\Pi = \big(\mathsf{CMT}^{(1)}, \dots, \mathsf{CMT}^{(\kappa)}; (\mathsf{Ch}^{(1)}, \dots, \mathsf{Ch}^{(\kappa)});$$
$$\mathsf{RSP}^{(1)}, \dots, \mathsf{RSP}^{(\kappa)}\big), \quad (13)$$
where $\big(\mathsf{Ch}^{(1)}, \dots, \mathsf{Ch}^{(\kappa)}\big)$ is given by
$$\mathcal{H}\big(M; \mathsf{CMT}^{(1)}, \dots, \mathsf{CMT}^{(\kappa)}; \mathsf{gpk}, \mathbf{c}\big) \in \{1,2,3\}^\kappa.$$

c) Output the group signature $\Sigma = (\mathbf{c}, \Pi)$.

3) $\mathsf{Verify}(\mathsf{gpk}, M, \Sigma)$: Parse Σ as $(\mathbf{c}, \Pi)$, parse Π as in (13), and then proceed as follows.

a) Return 0 if
$$\big(\mathsf{Ch}^{(1)}, \dots, \mathsf{Ch}^{(\kappa)}\big) \neq$$
$$\mathcal{H}\big(M; \mathsf{CMT}^{(1)}, \dots, \mathsf{CMT}^{(\kappa)}; \mathsf{gpk}, \mathbf{c}\big).$$

b) For $i = 1$ to κ, run the verification step of the interactive protocol in Section III-A on the public input $(\mathbf{G}, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}, \mathbf{c})$ to check if $\mathsf{RSP}^{(i)}$ is valid with respect to $\mathsf{CMT}^{(i)}$ and $\mathsf{Ch}^{(i)}$. Return 0 if any of the verification conditions fails to hold.

c) Return 1.

4) $\mathsf{Open}(\mathsf{gmsk}, M, \Sigma)$: Parse Σ as $(\mathbf{c}, \Pi)$ and decrypt $\mathbf{c}$ by running $\mathsf{ME.Dec}(\mathsf{gmsk}, \mathbf{c})$. Return $\perp$ if decryption fails. Return $j = \mathsf{B2I}(\mathbf{g}) \in [0, N-1]$ if decryption outputs $\mathbf{g} \in \mathbb{F}_2^\ell$.

Remark 1: Lemma 1 assures us that, for the parameters m, r, ω that satisfy the inequality condition in (10), the distribution of syndrome $\mathbf{y}_j$, for all $j \in [0, N-1]$, is statistically close to the uniform distribution over $\mathbb{F}_2^r$.

The efficiency, correctness, and security aspects of the above group signature scheme can now be summarized into the following theorem.

Theorem 5: The given group signature scheme is correct. The public key has size $nk + (m+N)r$ bits. The bit-size of the signatures is bounded above by
$$\big((N + 3\log N) + m(\log m + 1) +$$
$$n(\log n + 1) + k + 5\lambda\big)\kappa + n.$$

In the random oracle model we can make two further assertions. First, if the Decisional McEliece problem $\mathsf{DMcE}(n,k,t)$ and the Decisional Learning Parity with fixed-weight Noise problem $\mathsf{DLPN}(k-\ell, n, \mathsf{B}(n,t))$ are hard, then the scheme is CPA-anonymous. Second, if the Syndrome Decoding problem $\mathsf{SD}(m,r,\omega)$ is hard, then the scheme is traceable.

In terms of efficiency, it is clear from (11) that gpk has bit-size $nk + (m+N)r$. The length of the NIZKAoK Π is κ times the communication cost of the underlying interactive protocol. Thus, by Theorem 2, $\Sigma = (\mathbf{c}, \Pi)$ has bit-size bounded above by
$$\big((N + 3\log N) + m(\log m + 1) +$$
$$n(\log n + 1) + k + 5\lambda\big)\kappa + n.$$

To see that the given group signature scheme is correct, observe that the honest user with index j, for any $j \in [0, N-1]$, can always obtain a tuple $(j, \mathbf{s}, \mathbf{u}, \mathbf{e})$ satisfying (12). Then, since the underlying interactive protocol is perfectly complete, Π is a valid NIZKAoK and algorithm $\mathsf{Verify}(\mathsf{gpk}, M, \Sigma)$ always outputs 1, for any message $M \in \{0,1\}^*$. On the correctness of Open, it suffices to note that, if the ciphertext $\mathbf{c}$ is of the form $\mathbf{c} = \big(\mathbf{u} \,\|\, \mathsf{I2B}(j)\big) \cdot \mathbf{G} \oplus \mathbf{e}$, where $\mathbf{e} \in \mathsf{B}(n,t)$, then, by the correctness of the randomized McEliece encryption scheme, $\mathsf{ME.Dec}(\mathsf{gmsk}, \mathbf{c})$ outputs $\mathsf{I2B}(j)$.

B. Anonymity

Let $\mathcal{A}$ be any PPT adversary attacking the CPA-anonymity of the scheme with advantage ϵ. We will prove $\epsilon = \mathsf{negl}(\lambda)$ based on the ZK property of the underlying argument system. To do so, we retain the assumed hardness of the $\mathsf{DMcE}(n,k,t)$ and the $\mathsf{DLPN}(k-\ell, n, \mathsf{B}(n,t))$ problems. Specifically, we consider the following sequence of hybrid experiments $G_0^{(b)}, G_1^{(b)}, G_2^{(b)}, G_3^{(b)}$, and G_4.

Experiment $G_0^{(b)}$. This is the real CPA-anonymity game. The challenger runs $\mathsf{KeyGen}(1^\lambda, 1^N)$ to obtain

$$\big(\,\mathsf{gpk} = (\mathbf{G}, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}), \ \ \mathsf{gmsk} = \mathsf{sk}_{\mathsf{ME}},$$
$$\mathsf{gsk} = (\mathsf{gsk}[0], \dots, \mathsf{gsk}[N-1])\,\big),$$

and sends gpk and $\{\mathsf{gsk}[j]\}_{j \in [0, N-1]}$ to $\mathcal{A}$. In the challenge phase, $\mathcal{A}$ outputs a message M^* together with two indices $j_0, j_1 \in [0, N-1]$. The challenger sends back a challenge signature

$$\Sigma^* = (\mathbf{c}^*, \Pi^*) \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j_b]),$$

where

$$\mathbf{c}^* = \big(\,\mathbf{u}\,\|\,\mathsf{I2B}(j_b)\,\big) \cdot \mathbf{G} \oplus \mathbf{e},$$

with $\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^{k-\ell}$ and $\mathbf{e} \xleftarrow{\$} \mathsf{B}(n, t)$. The adversary then outputs b with probability $1/2 + \epsilon$.

Experiment $G_1^{(b)}$. This experiment introduces a modification in the challenge phase. Instead of faithfully generating the NIZKAoK Π^*, the challenger simulates it as follows.

1) Compute $\mathbf{c}^* \in \mathbb{F}_2^n$ as in Experiment $G_0^{(b)}$.
2) Run the simulator of the underlying interactive protocol $\kappa = \omega(\log \lambda)$ times on input $(\mathbf{G}, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}, \mathbf{c}^*)$. Program the random oracle $\mathcal{H}$ accordingly.
3) Output the simulated NIZKAoK Π^*.

Since the underlying argument system is statistically zero-knowledge, Π^* is statistically close to the real NIZKAoK. As a result, the simulated signature $\Sigma^* = (\mathbf{c}^*, \Pi^*)$ is statistically close to the one in experiment $G_0^{(b)}$. It then follows that $G_0^{(b)}$ and $G_1^{(b)}$ are indistinguishable from $\mathcal{A}$'s view.

Experiment $G_2^{(b)}$. This experiment makes the following change with respect to $G_1^{(b)}$. The encrypting key $\mathbf{G}$ obtained from $\mathsf{ME.KeyGen}(n, k, t)$ is replaced by a uniformly random matrix $\mathbf{G} \xleftarrow{\$} \mathbb{F}_2^{k \times n}$. Lemma 6 will demonstrate that Experiments $G_1^{(b)}$ and $G_2^{(b)}$ are computationally indistinguishable based on the assumed hardness of the $\mathsf{DMcE}(n, k, t)$ problem.

Lemma 6: If $\mathcal{A}$ can distinguish Experiments $G_1^{(b)}$ and $G_2^{(b)}$ with probability non-negligibly larger than $1/2$, then there exists an efficient distinguisher $\mathcal{D}_1$ that solves the $\mathsf{DMcE}(n, k, t)$ problem with the same probability.

Proof: An instance of the $\mathsf{DMcE}(n, k, t)$ problem is $\mathbf{G}^* \in \mathbb{F}_2^{k \times n}$, which can either be uniformly random or be generated by $\mathsf{ME.KeyGen}(n, k, t)$. The distinguisher $\mathcal{D}_1$ receives a challenge instance $\mathbf{G}^*$ and uses $\mathcal{A}$ to distinguish between the two. It interacts with $\mathcal{A}$ by performing the following steps.

1) Setup. Generate

$$(\mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}) \text{ and } (\mathsf{gsk}[0], \dots, \mathsf{gsk}[N-1])$$

as in the real scheme. Send $\mathcal{A}$ the pair
$$\big(\,\mathsf{gpk}^* = (\mathbf{G}^*, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}),$$
$$\mathsf{gsk} = (\mathsf{gsk}[0], \dots, \mathsf{gsk}[N-1])\,\big).$$

2) Challenge. Upon receiving the challenge (M^*, j_0, j_1), $\mathcal{D}_1$ proceeds as follows.
   a) Pick $b \xleftarrow{\$} \{0,1\}$, and compute
   $$\mathbf{c}^* = \big(\,\mathbf{u}\,\|\,\mathsf{I2B}(j_b)\,\big) \cdot \mathbf{G}^* \oplus \mathbf{e},$$
   where $\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^{k-\ell}$ and $\mathbf{e} \xleftarrow{\$} \mathsf{B}(n, t)$.

b) On input $(\mathbf{G}^*, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}, \mathbf{c}^*)$, simulate the NIZKAoK Π^* and output $\Sigma^* = (\mathbf{c}^*, \Pi^*)$.

We observe that if $\mathbf{G}^*$ is generated by $\mathsf{ME.KeyGen}(n, k, t)$ then the view of $\mathcal{A}$ in the interaction with $\mathcal{D}_1$ is statistically close to its view in Experiment $G_1^{(b)}$ with the challenger. On the other hand, if $\mathbf{G}^*$ is uniformly random, then $\mathcal{A}$'s view is statistically close to its view in Experiment $G_2^{(b)}$. Therefore, if $\mathcal{A}$ can guess whether it is interacting with the challenger in $G_1^{(b)}$ or $G_2^{(b)}$ with probability non-negligibly larger than $1/2$, then $\mathcal{D}_1$ can use $\mathcal{A}$'s guess to solve the challenge instance $\mathbf{G}^*$ of the $\mathsf{DMcE}(n, k, t)$ problem, with the same probability. ∎

Experiment $G_3^{(b)}$. In Experiment $G_2^{(b)}$ we have

$$\mathbf{c}^* = \big(\,\mathbf{u}\,\|\,\mathsf{I2B}(j_b)\,\big) \cdot \mathbf{G} \oplus \mathbf{e} = (\mathbf{u} \cdot \mathbf{G}_1 \oplus \mathbf{e}) \oplus \mathsf{I2B}(j_b) \cdot \mathbf{G}_2,$$

where

$$\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^{k-\ell}, \ \ \mathbf{e} \xleftarrow{\$} \mathsf{B}(n, t), \ \ \mathbf{G}_1 \in \mathbb{F}_2^{(k-\ell) \times n}, \text{ and } \mathbf{G}_2 \in \mathbb{F}_2^{\ell \times n}$$

such that $\begin{bmatrix} \mathbf{G}_1 \\ \hline \mathbf{G}_2 \end{bmatrix} = \mathbf{G}$. Now, Experiment $G_3^{(b)}$ modifies the generation of $\mathbf{c}^*$ by replacing the original one with

$$\mathbf{c}^* = \mathbf{v} \oplus \mathsf{I2B}(j_b) \cdot \mathbf{G}_2, \text{ where } \mathbf{v} \xleftarrow{\$} \mathbb{F}_2^n.$$

As shown in Lemma 7, Experiments $G_2^{(b)}$ and $G_3^{(b)}$ are computationally indistinguishable based on the assumed hardness of the $\mathsf{DLPN}(k - \ell, n, \mathsf{B}(n, t))$ problem.

Lemma 7: If $\mathcal{A}$ can distinguish Experiments $G_2^{(b)}$ and $G_3^{(b)}$ with probability non-negligibly larger than $1/2$, then there exists an efficient distinguisher $\mathcal{D}_2$ that can solve the $\mathsf{DLPN}(k - \ell, n, \mathsf{B}(n, t))$ problem with the same probability.

Proof: An instance of the $\mathsf{DLPN}(k-\ell, n, \mathsf{B}(n, t))$ problem is a pair $(\mathbf{B}, \mathbf{v}) \in \mathbb{F}_2^{(k-\ell) \times n} \times \mathbb{F}_2^n$, where $\mathbf{B}$ is uniformly random, and $\mathbf{v}$ is either uniformly random or of the form

$$\mathbf{v} = \mathbf{u} \cdot \mathbf{B} \oplus \mathbf{e}, \text{ for } \mathbf{u} \xleftarrow{\$} \mathbb{F}_2^{k-\ell} \text{ and } \mathbf{e} \xleftarrow{\$} \mathsf{B}(n, t).$$

The distinguisher $\mathcal{D}_2$ receives a challenge instance $(\mathbf{B}, \mathbf{v})$ and uses $\mathcal{A}$ to distinguish between the two. It interacts with $\mathcal{A}$ in the following manners.

1) Setup. Pick $\mathbf{G}_2 \xleftarrow{\$} \mathbb{F}_2^{\ell \times n}$ and let $\mathbf{G}^* = \begin{bmatrix} \mathbf{B} \\ \hline \mathbf{G}_2 \end{bmatrix}$. Generate $(\mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1})$ and $(\mathsf{gsk}[0], \dots, \mathsf{gsk}[N-1])$ as in the real scheme, and send $\mathcal{A}$ the pair
$$\big(\,\mathsf{gpk}^* = (\mathbf{G}^*, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}),$$
$$\mathsf{gsk} = (\mathsf{gsk}[0], \dots, \mathsf{gsk}[N-1])\,\big).$$

2) Challenge. Upon receiving the challenge (M^*, j_0, j_1), $\mathcal{D}_2$ first picks $b \xleftarrow{\$} \{0, 1\}$ and lets

$$\mathbf{c}^* = \mathbf{v} \oplus \mathsf{I2B}(j_b) \cdot \mathbf{G}_2,$$

where $\mathbf{v}$ comes from the challenge DLPN instance. It then simulates the NIZKAoK Π^*, on input $(\mathbf{G}^*, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}, \mathbf{c}^*)$, and outputs $\Sigma^* = (\mathbf{c}^*, \Pi^*)$. If $\mathcal{D}_2$'s input pair $(\mathbf{B}, \mathbf{v})$ is of the form

$$(\mathbf{B}, \mathbf{v} = \mathbf{u} \cdot \mathbf{B} \oplus \mathbf{e}), \text{ where } \mathbf{u} \xleftarrow{\$} \mathbb{F}_2^{k-\ell} \text{ and } \mathbf{e} \xleftarrow{\$} \mathsf{B}(n, t),$$

then the view of $\mathcal{A}$ in the interaction with $\mathcal{D}_2$ is statistically close to its view in Experiment $G_2^{(b)}$ with the challenger.

On the other hand, if the pair $(\mathbf{B}, \mathbf{v})$ is uniformly random, then $\mathcal{A}$'s view is statistically close to its view in Experiment $G_3^{(b)}$. Therefore, if $\mathcal{A}$ can guess whether it is interacting with the challenger in $G_2^{(b)}$ or $G_3^{(b)}$ with probability non-negligibly larger than $1/2$, then $\mathcal{D}_2$ can use $\mathcal{A}$'s guess to solve the challenge instance of the $\mathsf{DLPN}(k - \ell, \mathsf{B}(n,t))$ problem with the same probability. ∎

Experiment G_4. This is a modification of Experiment $G_3^{(b)}$. The ciphertext $\mathbf{c}^*$ is now set as $\mathbf{c}^* = \mathbf{r} \xleftarrow{\$} \mathbb{F}_2^n$. Clearly, the distributions of $\mathbf{c}^*$ in Experiments $G_3^{(b)}$ and G_4 are identical. As a result, G_4 and $G_3^{(b)}$ are statistically indistinguishable. We note that G_4 no longer depends on the challenger's bit b, and thus, $\mathcal{A}$'s advantage in this experiment is 0.

The above discussion shows that Experiments $G_0^{(b)}, G_1^{(b)}, G_2^{(b)}, G_3^{(b)}, G_4$ are indistinguishable and that $\mathbf{Adv}_{\mathcal{A}}(G_4) = 0$. It then follows that the advantage of $\mathcal{A}$ in attacking the CPA-anonymity of the scheme, *i.e.*, in Experiment $G_0^{(b)}$, is negligible. The CPA-anonymity property is, thus, confirmed.

C. Traceability

Let $\mathcal{A}$ be a PPT traceability adversary against our group signature scheme with success probability ϵ. We construct a PPT algorithm $\mathcal{F}$ that solves the $\mathsf{SD}(m, r, \omega)$ problem with success probability polynomially related to ϵ.

Algorithm $\mathcal{F}$ receives a challenge $\mathsf{SD}(m, r, \omega)$ instance, *i.e.*, a uniformly random matrix-syndrome pair $(\widetilde{\mathbf{H}}, \tilde{\mathbf{y}}) \in \mathbb{F}_2^{r \times m} \times \mathbb{F}_2^r$. The goal of $\mathcal{F}$ is to find a vector $\mathbf{s} \in \mathsf{B}(m, \omega)$ such that $\widetilde{\mathbf{H}} \cdot \mathbf{s}^\top = \tilde{\mathbf{y}}^\top$. It then carry out the following tasks.
1) Pick a guess $j^* \xleftarrow{\$} [0, N-1]$ and set $\mathbf{y}_{j^*} = \tilde{\mathbf{y}}$.
2) Set $\mathbf{H} = \widetilde{\mathbf{H}}$. For each $j \in [0, N-1]$ such that $j \neq j^*$, sample $\mathbf{s}_j \xleftarrow{\$} \mathsf{B}(m, \omega)$ and set $\mathbf{y}_j \in \mathbb{F}_2^r$ to be its syndrome, *i.e.*, $\mathbf{y}_j^\top = \mathbf{H} \cdot \mathbf{s}_j^\top$.
3) Run $\mathsf{ME.KeyGen}(n, k, t)$ to obtain a key pair
$$\left(\mathsf{pk}_{\mathsf{ME}} = \mathbf{G} \in \mathbb{F}_2^{k \times n} \; ; \; \mathsf{sk}_{\mathsf{ME}}\right).$$
4) Send $\mathsf{gpk} = (\mathbf{G}, \mathbf{H}, \mathbf{y}_0, \ldots, \mathbf{y}_{N-1})$ and $\mathsf{gmsk} = \mathsf{sk}_{\mathsf{ME}}$ to $\mathcal{A}$.

Since the parameters m, r, ω were chosen such that $r \leq \log\binom{m}{w} - 2\lambda - \mathcal{O}(1)$, Lemma 1 says that the distribution of the syndrome $\mathbf{y}_j$, for all $j \neq j^*$, is statistically close to the uniform distribution over $\mathbb{F}_2^r$. In addition, the syndrome $\mathbf{y}_{j^*} = \tilde{\mathbf{y}}$ is truly uniform over $\mathbb{F}_2^r$. It then follows that the distribution of $(\mathbf{y}_0, \ldots, \mathbf{y}_{N-1})$ is statistically close to that of the real scheme as noted in Remark 1. As a result, the distribution of $(\mathsf{gpk}, \mathsf{gmsk})$ is statistically close to the distribution expected by $\mathcal{A}$.

The forger $\mathcal{F}$ then initializes a set $CU = \emptyset$ and handles the queries from $\mathcal{A}$ according to the following procedure.
1) Queries to the random oracle $\mathcal{H}$ are handled by consistently returning uniformly random values in $\{1, 2, 3\}^\kappa$. Suppose that $\mathcal{A}$ makes $Q_{\mathcal{H}}$ queries, then for each $\eta \leq Q_{\mathcal{H}}$, we let r_η denote the answer to the η-th query.
2) $\mathcal{O}^{\mathsf{Corrupt}}(j)$, for any $j \in [0, N-1]$, depends on how j and j^* are related. If $j \neq j^*$, then $\mathcal{F}$ sets $CU := CU \cup \{j\}$ and gives $\mathbf{s}_j$ to $\mathcal{A}$. If $j = j^*$, then $\mathcal{F}$ aborts.
3) $\mathcal{O}^{\mathsf{Sign}}(j, M)$, for any $j \in [0, N-1]$ and any message M, also depends on j and j^*. If $j \neq j^*$, then $\mathcal{F}$ honestly

computes a signature, since it has the secret key $\mathbf{s}_j$. If $j = j^*$, then $\mathcal{F}$ returns a simulated signature Σ^* computed as in Section IV-B. Please consult, specifically, Experiment $G_1^{(b)}$ in the proof of anonymity.

At some point, $\mathcal{A}$ outputs a forged group signature Σ^* on some message M^*, where
$$\Sigma^* = \Big(\mathbf{c}^*, \big(\mathsf{CMT}^{(1)}, \ldots, \mathsf{CMT}^{(\kappa)};$$
$$\mathsf{Ch}^{(1)}, \ldots, \mathsf{Ch}^{(\kappa)}; \mathsf{RSP}^{(1)}, \ldots, \mathsf{RSP}^{(\kappa)}\big)\Big).$$

By the requirements of the traceability experiment, one has $\mathsf{Verify}(\mathsf{gpk}, M^*, \Sigma^*) = 1$ and, for all $j \in CU$, signatures of user j on M^* were never queried. Now $\mathcal{F}$ uses $\mathsf{sk}_{\mathsf{ME}}$ to open Σ^*, and aborts if the opening algorithm does not output j^*. It can be checked that $\mathcal{F}$ aborts with probability at most $(N-1)/N + (2/3)^\kappa$. This is because the choice of $j^* \in [0, N-1]$ is completely hidden from $\mathcal{A}$'s view and $\mathcal{A}$ can only violate the soundness of the argument system with probability at most $(2/3)^\kappa$. Thus, with probability at least $1/N - (2/3)^\kappa$,

$$\mathsf{Verify}(\mathsf{gpk}, M^*, \Sigma^*) = 1 \; \wedge \; \mathsf{Open}(\mathsf{sk}_{\mathsf{ME}}, M^*, \Sigma^*) = j^*.$$
$$(14)$$

Suppose that (14) holds. Algorithm $\mathcal{F}$ then exploits the forgery as follows. Denote by Δ the tuple
$$\left(M^*; \mathsf{CMT}^{(1)}, \ldots, \mathsf{CMT}^{(\kappa)}; \mathbf{G}, \mathbf{H}, \mathbf{y}_0, \ldots, \mathbf{y}_{N-1}, \mathbf{c}^*\right).$$

Observe that if $\mathcal{A}$ has never queried the random oracle $\mathcal{H}$ on input Δ, then $\Pr\big[(\mathsf{Ch}^{(1)}, \ldots, \mathsf{Ch}^{(\kappa)}) = \mathcal{H}(\Delta)\big] \leq 3^{-\kappa}$. Thus, with probability at least $\epsilon - 3^{-\kappa}$, there exists certain $\eta^* \leq Q_{\mathcal{H}}$ such that Δ was the input of the η^*-th query. Next, $\mathcal{F}$ picks η^* as the target forking point and replays $\mathcal{A}$ many times with the same random tape and input as in the original run. In each rerun, for the first $\eta^* - 1$ queries, $\mathcal{A}$ is given the same answers $r_1, \ldots, r_{\eta^*-1}$ as in the initial run. From the η^*-th query onwards, however, $\mathcal{F}$ replies with fresh random values $r'_{\eta^*}, \ldots, r'_{q_{\mathcal{H}}} \xleftarrow{\$} \{1, 2, 3\}^\kappa$. The Improved Forking Lemma of Pointcheval and Vaudenay [66, Lemma 7] implies that, with probability larger than $1/2$ and within less than $32 \cdot Q_{\mathcal{H}}/(\epsilon - 3^{-\kappa})$ executions of $\mathcal{A}$, algorithm $\mathcal{F}$ can obtain a 3-fork involving the tuple Δ. Now, let the answers of $\mathcal{F}$ with respect to the 3-fork branches be
$$r_{1,\eta^*} = \left(\mathsf{Ch}_1^{(1)}, \ldots, \mathsf{Ch}_1^{(\kappa)}\right);$$
$$r_{2,\eta^*} = \left(\mathsf{Ch}_2^{(1)}, \ldots, \mathsf{Ch}_2^{(\kappa)}\right);$$
$$r_{3,\eta^*} = \left(\mathsf{Ch}_3^{(1)}, \ldots, \mathsf{Ch}_3^{(\kappa)}\right).$$

Then, by a simple calculation, one has
$$\Pr\big[\exists i \in \{1, \ldots, \kappa\} : \{\mathsf{Ch}_1^{(i)}, \mathsf{Ch}_2^{(i)}, \mathsf{Ch}_3^{(i)}\} = \{1, 2, 3\}\big]$$
$$= 1 - (7/9)^\kappa.$$

Conditioned on the existence of such index i, one parses the three forgeries corresponding to the fork branches to obtain $\left(\mathsf{RSP}_1^{(i)}, \mathsf{RSP}_2^{(i)}, \mathsf{RSP}_3^{(i)}\right)$. They turn out to be three *valid* responses with respect to 3 different challenges for the same commitment $\mathsf{CMT}^{(i)}$. Then, by using the knowledge extractor

of the underlying interactive argument system (see Lemma 4), one can efficiently extract a tuple

$$(j', \mathbf{s}', \mathbf{u}', \mathbf{e}') \in [0, N-1] \times \mathbb{F}_2^m \times \mathbb{F}_2^{k-\ell} \times \mathbb{F}_2^n$$

such that

$$\mathbf{H} \cdot \mathbf{s}'^\top = \mathbf{y}_{j'}^\top, \quad \mathbf{s}' \in \mathsf{B}(m, \omega),$$
$$\left(\mathbf{u}' \,\|\, \mathsf{I2B}(j') \right) \cdot \mathbf{G} \oplus \mathbf{e}' = \mathbf{c}^*, \quad \mathbf{e}' \in \mathsf{B}(n, t).$$

Since the given group signature scheme is correct, the equation $\left(\mathbf{u}' \,\|\, \mathsf{I2B}(j') \right) \cdot \mathbf{G} \oplus \mathbf{e}' = \mathbf{c}^*$ implies that

$$\mathsf{Open}(\mathsf{sk}_{\mathsf{ME}}, M^*, \Sigma^*) = j'.$$

On the other hand, we have

$$\mathsf{Open}(\mathsf{sk}_{\mathsf{ME}}, M^*, \Sigma^*) = j^*,$$

which leads to $j' = j^*$. Therefore, it holds that

$$\widetilde{\mathbf{H}} \cdot \mathbf{s}'^\top = \mathbf{H} \cdot \mathbf{s}'^\top = \mathbf{y}_{j^*}^\top = \tilde{\mathbf{y}}^\top \text{ and } \mathbf{s}' \in \mathsf{B}(m, \omega).$$

In other words, $\mathbf{s}'$ is a valid solution to the challenge $\mathsf{SD}(m, r, \omega)$ instance $(\widetilde{\mathbf{H}}, \tilde{\mathbf{y}})$.

Finally, the above analysis shows that, if $\mathcal{A}$ has success probability ϵ and running time T in attacking the traceability of our group signature scheme, then $\mathcal{F}$ has success probability at least $1/2\left(1/N - (2/3)^\kappa\right)\left(1 - (7/9)^\kappa\right)$ and running time at most $32 \cdot T \cdot Q_{\mathcal{H}}/(\epsilon - 3^{-\kappa}) + \mathsf{poly}(\lambda, N)$. This concludes the proof of the traceability property.

V. ACHIEVING CCA-ANONYMITY

In this section, we propose and analyse a code-based group signature that achieves the strong notion of CCA-anonymity. The scheme is an extension of the CPA-anonymous scheme described in Section IV-A. To achieve CCA-security for the underlying encryption layer via the Naor-Yung transformation [51], the binary representation of the signer's index j is now verifiably encrypted twice under two different randomized McEliece public keys $\mathsf{pk}_{\mathsf{ME}}^{(1)}$ and $\mathsf{pk}_{\mathsf{ME}}^{(2)}$. This enables CCA-anonymity for the resulting group signature scheme. In describing the scheme, the focus is on presenting the modifications that we must make with respect to the earlier scheme from Section IV-A.

A. Description of the Scheme

We start with the four algorithms that constitute the scheme.
1) $\mathsf{KeyGen}(1^\lambda, 1^N)$: The algorithm proceeds as the key generation algorithm of Section IV-A, with the following alteration. $\mathsf{ME.KeyGen}(n, k, t)$ is run twice, producing two key pairs

$$\left(\mathsf{pk}_{\mathsf{ME}}^{(1)} = \mathbf{G}^{(1)}, \mathsf{sk}_{\mathsf{ME}}^{(1)} \right) \text{ and}$$
$$\left(\mathsf{pk}_{\mathsf{ME}}^{(2)} = \mathbf{G}^{(2)}, \mathsf{sk}_{\mathsf{ME}}^{(2)} \right)$$

for the randomized McEliece encryption. Then $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$ are included in the group public key gpk, the opening secret key gmsk is defined to be $\mathsf{sk}_{\mathsf{ME}}^{(1)}$, while $\mathsf{sk}_{\mathsf{ME}}^{(2)}$ is discarded.
2) $\mathsf{Sign}(\mathsf{gsk}[j], M)$: The binary representation $\mathsf{I2B}(j) \in \mathbb{F}_2^\ell$ of the user's index j is now encrypted twice, under the

keys $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$. The resulting ciphertexts $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ have the form

$$\mathbf{c}^{(1)} = \left(\mathbf{u}^{(1)} \,\|\, \mathsf{I2B}(j) \right) \cdot \mathbf{G}^{(1)} \oplus \mathbf{e}^{(1)} \in \mathbb{F}_2^n \text{ and}$$
$$\mathbf{c}^{(2)} = \left(\mathbf{u}^{(2)} \,\|\, \mathsf{I2B}(j) \right) \cdot \mathbf{G}^{(2)} \oplus \mathbf{e}^{(2)} \in \mathbb{F}_2^n,$$

where $\mathbf{u}^{(1)}, \mathbf{u}^{(2)} \xleftarrow{\$} \mathbb{F}_2^{k-\ell}$ and $\mathbf{e}^{(1)}, \mathbf{e}^{(2)} \xleftarrow{\$} \mathsf{B}(n, t)$. The zero-knowledge protocol of the scheme from Section IV-A is then developed to enable the prover, possessing witness $(j, \mathbf{s}, \mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{e}^{(1)}, \mathbf{e}^{(2)})$, to convince the verifier, with public input

$$(\mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}, \mathbf{c}^{(1)}, \mathbf{c}^{(2)}),$$

that

$$\begin{aligned}
\mathbf{H} \cdot \mathbf{s}^\top &= \mathbf{y}_j^\top, \quad \mathbf{s} \in \mathsf{B}(m, \omega), \\
\mathbf{e}^{(1)} &\in \mathsf{B}(n, t), \quad \mathbf{e}^{(2)} \in \mathsf{B}(n, t), \\
\left(\mathbf{u}^{(1)} \,\|\, \mathsf{I2B}(j) \right) \cdot \mathbf{G}^{(1)} \oplus \mathbf{e}^{(1)} &= \mathbf{c}^{(1)}, \\
\left(\mathbf{u}^{(2)} \,\|\, \mathsf{I2B}(j) \right) \cdot \mathbf{G}^{(2)} \oplus \mathbf{e}^{(2)} &= \mathbf{c}^{(2)}. \quad (15)
\end{aligned}$$

The protocol described in Section III-B can be employed for the above zero-knowledge protocol.
Let Π be the NIZKAoK obtained by repeating the protocol $\kappa = \omega \cdot \log \lambda$ times and making it non-interactive via the Fiat-Shamir heuristic. The group signature is set to be $\Sigma = (\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \Pi)$, where

$$\begin{aligned}
\Pi = \big(& \mathsf{CMT}^{(1)}, \dots, \mathsf{CMT}^{(\kappa)}; \\
& (\mathsf{Ch}^{(1)}, \dots, \mathsf{Ch}^{(\kappa)}); \mathsf{RSP}^{(1)}, \dots, \mathsf{RSP}^{(\kappa)} \big) \quad (16)
\end{aligned}$$

and $\left(\mathsf{Ch}^{(1)}, \dots, \mathsf{Ch}^{(\kappa)} \right) \in \{1, 2, 3\}^\kappa$ is given by

$$\mathcal{H} \left(M; \mathsf{CMT}^{(1)}, \dots, \mathsf{CMT}^{(\kappa)}; \mathsf{gpk}, \mathbf{c}^{(1)}, \mathbf{c}^{(2)} \right).$$

3) $\mathsf{Verify}(\mathsf{gpk}, M, \Sigma)$: This algorithm proceeds as the verification algorithm of the scheme from Section IV-A, with $(\mathbf{c}^{(1)}, \mathbf{c}^{(2)})$ taking the place of $\mathbf{c}$.
4) $\mathsf{Open}(\mathsf{gmsk}, M, \Sigma)$: Parse Σ as $(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \Pi)$ and run $\mathsf{ME.Dec}\left(\mathsf{gmsk}, \mathbf{c}^{(1)}\right)$ to decrypt $\mathbf{c}^{(1)}$. If decryption fails, then return $\perp$. If decryption outputs $\mathbf{g} \in \mathbb{F}_2^\ell$, then return $j = \mathsf{B2I}(\mathbf{g}) \in [0, N-1]$.

The efficiency, correctness, and security aspects of the above group signature scheme are summarized in the following theorem.

Theorem 8: The given group signature scheme is correct. The public key has size $2nk + (m + N)r$ bits while the signatures have bit-size bounded above by

$$\begin{aligned}
\big((N + 3\log N) + m(\log m + 1) + \\
2n(\log n + 1) + 2k + 5\lambda\big)\kappa + n.
\end{aligned}$$

In the random oracle model we can make two further assertions. First, if the Decisional McEliece problem $\mathsf{DMcE}(n, k, t)$ and the Decisional Learning Parity with fixed-weight Noise problem $\mathsf{DLPN}(k - \ell, n, \mathsf{B}(n, t))$ are hard, and if the underlying NIZKAoK system is simulation-sound, then the scheme is CCA-anonymous. Second, if the Syndrome Decoding problem $\mathsf{SD}(m, r, \omega)$ is hard, then the scheme is traceable.

Compared with the basic scheme in Section IV, the present scheme introduces one more McEliece encrypting matrix of size nk bits in the group public key, one more n-bit ciphertext and its supporting ZK sub-argument of well-formedness contained in Π in the group signature. Overall, the upgrade from CPA-anonymity to CCA-anonymity incurs only a small and reasonable overhead in terms of efficiency. Since the correctness and traceability analyses of the scheme are almost identical to those of the basis scheme in Section IV, the details are omitted here. In the next subsection, we will prove the CCA-anonymity property, which is the distinguished feature that we aim to accomplish.

B. A Scheme Achieving CCA-Anonymity

Let $\mathcal{A}$ be any PPT adversary attacking the CCA-anonymity of the scheme with advantage ϵ. We will prove that $\epsilon = \mathsf{negl}(\lambda)$ based on the ZK property and simulation-soundness of the underlying argument system. We keep the assumed hardness of the $\mathsf{DMcE}(n, k, t)$ and the $\mathsf{DLPN}(k-\ell, n, \mathsf{B}(n,t))$ problems. Specifically, we consider the following sequence of hybrid experiments $G_0^{(b)}, G_1^{(b)}, \ldots, G_{10}^{(b)}, G_{11}$, where b is the bit chosen by the challenger when generating the challenge signature.

Experiment $G_0^{(b)}$. This is the real CCA-anonymity experiment. The challenger runs $\mathsf{KeyGen}(1^\lambda, 1^N)$ to obtain

$$\big(\, \mathsf{gpk} = (\mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \mathbf{H}, \mathbf{y}_0, \ldots, \mathbf{y}_{N-1}),$$
$$\mathsf{gmsk} = \mathsf{sk}_{\mathsf{ME}}^{(1)}, \;\; \mathsf{gsk} = (\mathsf{gsk}[0], \ldots, \mathsf{gsk}[N-1]) \big),$$

and then gives gpk and $\{\mathsf{gsk}[j]\}_{j \in [0, N-1]}$ to $\mathcal{A}$. Queries to the opening oracle are answered using the opening secret key $\mathsf{sk}_{\mathsf{ME}}^{(1)}$. In the challenge phase, $\mathcal{A}$ outputs a message M^* together with two indices $j_0, j_1 \in [0, N-1]$. The challenger sends back a challenge signature

$$\Sigma^* = \big(\mathbf{c}^{(1),*}, \mathbf{c}^{(2),*}, \Pi^*\big) \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j_b]).$$

The adversary outputs b with probability $1/2 + \epsilon$.

Experiment $G_1^{(b)}$. The only difference between this experiment and Experiment $G_0^{(b)}$ is that the challenger, when running $\mathsf{KeyGen}(1^\lambda, 1^N)$, retains the second decryption key $\mathsf{sk}_{\mathsf{ME}}^{(2)}$ instead of discarding it. The views of $\mathcal{A}$ in the two experiments are identical.

Experiment $G_2^{(b)}$. This experiment is like Experiment $G_1^{(b)}$ with one modification in the signature opening oracle. Instead of using $\mathsf{sk}_{\mathsf{ME}}^{(1)}$ to open signatures, the challenger uses $\mathsf{sk}_{\mathsf{ME}}^{(2)}$. It is easy to see that $\mathcal{A}$'s view will be the same as in Experiment $G_1^{(b)}$ until an event F_2 when $\mathcal{A}$ queries the opening of a signature $\Sigma = (\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \Pi)$ for which $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ encrypt distinct elements of $\mathbb{F}_2^\ell$. Since such an event F_2 could break the soundness of the zero-knowledge protocol used to generate Π, it could happen only with negligible probability. Therefore, the probability that $\mathcal{A}$ outputs b in this experiment is negligibly close to $1/2 + \epsilon$.

Experiment $G_3^{(b)}$. This experiment is identical to Experiment $G_2^{(b)}$, except on one modification. Instead of faithfully computing the NIZKAoK Π^* using witness $(j, \mathbf{s}, \mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \mathbf{e}^{(1)}, \mathbf{e}^{(2)})$, the challenger simulates it by running the simulator of the underlying zero-knowledge protocol and programming the random oracle $\mathcal{H}$. Note that Π^* is a simulated argument for a true statement, since $\mathbf{c}^{(1),*}$ and $\mathbf{c}^{(2),*}$ are honestly computed. Thanks to the statistical zero-knowledge property of the underlying protocol, Experiment $G_3^{(b)}$ is statistically close to Experiment $G_2^{(b)}$.

Experiment $G_4^{(b)}$. This experiment is similar to Experiment $G_2^{(b)}$ in the proof of CPA-anonymity in Section IV-B. The only modification, with respect to the encrypting key $\mathbf{G}^{(1)}$, is that instead of generating it using $\mathsf{ME.KeyGen}(n, k, t)$, we sample it uniformly at random over $\mathbb{F}_2^{k \times n}$. By the assumed hardness of the $\mathsf{DMcE}(n, k, t)$ problem, this experiment is computationally indistinguishable from the previous experiment.

Experiment $G_5^{(b)}$. This experiment is similar to Experiment $G_3^{(b)}$ in the proof of CPA-anonymity in Section IV-B. Instead of computing $\mathbf{c}^{(1),*}$ as

$$\mathbf{c}^{(1),*} = \big(\mathbf{u}^{(1)} \,\|\, \mathsf{I2B}(j_b)\big) \cdot \mathbf{G}^{(1)} \oplus \mathbf{e}^{(1)}$$
$$= (\mathbf{u}^{(1)} \cdot \mathbf{G}_1^{(1)} \oplus \mathbf{e}^{(1)}) \oplus \mathsf{I2B}(j_b) \cdot \mathbf{G}_2^{(1)},$$

we let

$$\mathbf{c}^{(1),*} = \mathbf{v}^{(1)} \oplus \mathsf{I2B}(j_b) \cdot \mathbf{G}_2^{(1)}, \;\; \text{where } \mathbf{v}^{(1)} \xleftarrow{\$} \mathbb{F}_2^n.$$

The simulated NIZKAoK Π^* now corresponds to a false statement, since $\mathbf{c}^{(1),*}$ is not a well-formed ciphertext. Nevertheless, assuming the hardness of the $\mathsf{DLPN}(k - \ell, n, \mathsf{B}(n,t))$ problem, in the challenge phase the adversary $\mathcal{A}$ can only observe the modification of $\mathbf{c}^{(1),*}$ with probability at most negligible in λ. The view of $\mathcal{A}$ is thus computationally close to that in Experiment $G_4^{(b)}$ above until an event F_5 which could happen after the challenge phase when $\mathcal{A}$ queries the opening of a signature $\Sigma = \big(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \Pi\big)$ for which $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ encrypt distinct elements of $\mathbb{F}_2^\ell$. Since such an event F_5 could break the simulation-soundness of the underlying NIZKAoK, it could happen only with negligible probability. Therefore, the success probability of $\mathcal{A}$ in this experiment is negligibly close to that in Experiment $G_4^{(b)}$.

Experiment $G_6^{(b)}$. This experiment modifies Experiment $G_5^{(b)}$ slightly. The ciphertext $\mathbf{c}^{(1),*}$ is now $\mathbf{c}^{(1),*} = \mathbf{r}^{(1)} \xleftarrow{\$} \mathbb{F}_2^n$. Clearly, the distributions of $\mathbf{c}^{(1),*}$ in $G_5^{(b)}$ and $G_6^{(b)}$ are identical. Hence, the two experiments are statistically indistinguishable.

Experiment $G_7^{(b)}$. This experiment switches the encrypting key $\mathbf{G}^{(1)}$ back to an honest key generated by $\mathsf{ME.KeyGen}(n, k, t)$, and stores the corresponding decryption key $\mathsf{sk}_{\mathsf{ME}}^{(1)}$. By the assumed hardness of the $\mathsf{DMcE}(n, k, t)$ problem, this experiment is computationally indistinguishable from Experiment $G_6^{(b)}$.

Experiment $G_8^{(b)}$. In this experiment, we use $\mathsf{sk}_{\mathsf{ME}}^{(1)}$, instead of $\mathsf{sk}_{\mathsf{ME}}^{(2)}$, to answer signature opening queries. The view of $\mathcal{A}$ is identical to that in Experiment $G_7^{(b)}$ above until an event F_8 when $\mathcal{A}$ queries the opening of a signature $\Sigma = \big(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \Pi\big)$ for which $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ encrypt distinct elements of $\mathbb{F}_2^\ell$. Since such an event F_8 could break the

TABLE II
IMPLEMENTATION RESULTS AND SIZES OF OUR CPA-ANONYMOUS SCHEME

N	Public Key Size	Average Signature Size	Message		KeyGen	Sign	Verify	Open
$2^4 = 16$	625 KB	111 KB	1 B		5.448	0.044	0.031	0.112
			1 GB			5.372	5.355	
$2^8 = 256$	642 KB	114 KB	1 B		5.407	0.045	0.032	0.111
			1 GB			5.363	5.351	
$2^{12} = 4,096$	906 KB	159 KB	1 B		5.536	0.058	0.040	0.112
			1 GB			5.366	5.347	
$2^{16} = 65,536$	5.13 MB	876 KB	1 B		7.278	0.282	0.186	0.111
			1 GB			5.591	5.497	
$2^{20} = 1,048,576$	72.8 MB	12.4 MB	1 B		33.947	3.874	2.498	0.111
			1 GB			9.173	7.795	
$2^{24} = 16,777,216$	1.16 GB	196 MB	1 B		481.079	61.164	39.218	0.111
			1 GB			66.613	44.575	

The unit for time is second. All implementation results are the averages from 100 tests.

simulation-soundness of the underlying NIZKAoK, it could happen only with negligible probability.

Experiment $G_9^{(b)}$. This experiment is similar to Experiment $G_4^{(b)}$ above. We replace the randomized McEliece encrypting key $\mathbf{G}^{(2)}$ by a uniformly random matrix in $\mathbb{F}_2^{k \times n}$. By the assumed hardness of the $\mathsf{DMcE}(n, k, t)$ problem, this experiment is computationally indistinguishable from Experiment $G_8^{(b)}$.

Experiment $G_{10}^{(b)}$. This experiment is akin to Experiment $G_5^{(b)}$. The second ciphertext $\mathbf{c}^{(2),*}$ is now computed as

$$\mathbf{c}^{(2),*} = \mathbf{v}^{(2)} \oplus \mathsf{I2B}(j_b) \cdot \mathbf{G}_2^{(2)}, \text{ where } \mathbf{v}^{(2)} \xleftarrow{\$} \mathbb{F}_2^n.$$

Assuming the hardness of the $\mathsf{DLPN}(k - \ell, n, \mathsf{B}(n, t))$ problem, this experiment is computationally indistinguishable from Experiment $G_9^{(b)}$.

Experiment G_{11}. This experiment resembles Experiment $G_6^{(b)}$. The second ciphertext is now $\mathbf{c}^{(2),*} = \mathbf{r}^{(2)} \xleftarrow{\$} \mathbb{F}_2^n$. Clearly, the distributions of $\mathbf{c}^{(2),*}$ in $G_{10}^{(b)}$ and G_{11} are identical. Hence, the two experiments are statistically indistinguishable. Moreover, since G_{11} no longer depends on the challenger's bit b, the advantage of $\mathcal{A}$ is 0 here.

The above discussion establishes that Experiments $G_0^{(b)}$, $G_1^{(b)}, \ldots, G_{10}^{(b)}$, G_{11} are indistinguishable and that $\mathcal{A}$ has no advantage in G_{11}. It then follows that the advantage of $\mathcal{A}$ in attacking the CCA-anonymity of the scheme, *i.e.*, in Experiment $G_0^{(b)}$, is negligible. This concludes the justification for the CCA-anonymity property.

VI. IMPLEMENTATION RESULTS

This section presents basic implementation results of our proposed group signature schemes to certify their feasibility.

A. Test Environment

The testing platform was a modern PC with a 3.4 GHz Intel Core i5 CPU and 32 GB of RAM. We employed the NTL [67] and gf2x [68] libraries for efficient polynomial operations

over fields of characteristic 2. The Paterson decoding algorithm [69] was used on the binary Goppa codes to implement the McEliece encryption. We employed SHA-3 with various output sizes to realize several hash functions.

To achieve 80-bit security, we decided on the following parameters. The McEliece parameters were set to $(n, k, t) = (2^{11}, 1696, 32)$, as in [70]. The parameters for Syndrome Decoding were set to $(m, r, \omega) = (2756, 550, 121)$ so that the distribution of $\mathbf{y}_0, \ldots, \mathbf{y}_{N-1}$ is 2^{-80}-close to the uniform distribution over $\mathbb{F}_2^r$, by Lemma 1, and that the $\mathsf{SD}(m, r, \omega)$ problem is intractable with respect to the best known attacks. In particular, these parameters ensure the following work factor evaluations. First, the Information Set Decoding algorithm proposed in [43] has work factor more than 2^{80}. For an evaluation formula, one can also refer to [71, Slide 3]. Second, the birthday attacks presented in [42] have work factors more than 2^{80}. The number of protocol repetitions κ was set to 140 to obtain soundness $1 - 2^{-80}$.

B. Experimental Results

Table II shows the implementation results of our CPA-anonymous group signature scheme, together with its public key and signature sizes, with respect to various numbers of group users and different message sizes. To reduce the signature size, in the underlying zero-knowledge protocol, we sent a random seed instead of permutations when $\mathsf{Ch} = 2$. Similarly, we sent a random seed instead of the whole response RSP when $\mathsf{Ch} = 3$. Using this technique, the average signature sizes were reduced to about 159 KB for $4,096 = 2^{12}$ users and to 876 KB for $65,536 = 2^{16}$ users, respectively. Our public key and signature sizes are linear in the number of group users N, but it does not come to the fore while N is less than 2^{12} due to the respective sizes of $\mathbf{G}$ and $\mathbf{H}$.

Our implementation took 0.282 and 0.186 seconds for a 1 B message and 5.591 and 5.497 seconds for a 1 GB message, respectively, to sign a message and to verify a generated signature for a group of $65,536$ users. In our experiments, it took about 5.30 seconds to hash a 1 GB message, leading to the differences of signing and verifying times between a

TABLE III
IMPLEMENTATION RESULTS AND SIZES OF OUR CCA-ANONYMOUS SCHEME

N	Public Key Size	Average Signature Size	Message		KeyGen	Sign	Verify	Open
$2^4 = 16$	1.06 MB	157 KB	1	B	10.660	0.065	0.046	0.111
			1	GB		5.366	5.351	
$2^8 = 256$	1.08 MB	160 KB	1	B	10.605	0.066	0.046	0.112
			1	GB		5.382	5.369	
$2^{12} = 4,096$	1.34 MB	205 KB	1	B	10.731	0.080	0.056	0.112
			1	GB		5.381	5.362	
$2^{16} = 65,536$	5.56 MB	922 KB	1	B	12.438	0.309	0.202	0.111
			1	GB		5.629	5.519	
$2^{20} = 1,048,576$	73.2 MB	12.5 MB	1	B	39.099	3.998	2.504	0.111
			1	GB		9.322	7.829	
$2^{24} = 16,777,216$	1.16 GB	196 MB	1	B	490.219	62.878	39.358	0.111
			1	GB		68.177	44.648	

The unit for time is second. All implementation results are the averages from 100 tests.

1 B and a 1 GB messages. One may naturally expect that running times should increase as N grows. But, as recorded in Table II, the increases were negligible and, on occasions, the running time even decreased slightly as N grew up to 2^{12}. This could also be due to the effect that the time required to perform other basic operations with parameters **G** and **H** had on the overall running time.

Table III contains the implementation results of our CCA-anonymous group signature scheme, along with public key and average signature sizes for various number of group users and different message sizes. The public key size of our CCA-anonymous scheme was 434 KB larger than that of our CPA-anonymous version because of the additionally required matrix $\mathbf{G}^{(2)}$. The average signature size was also about 46 KB larger since the response RSP additionally included $\mathbf{v}_e^{(2)}$, $\mathbf{w}_e^{(2)}$ when Ch = 1 and $\mathbf{z}_u^{(2)}$, $\mathbf{z}_e^{(2)}$ when Ch = 2. We remark that there was no additional element to be sent for Ch = 3 since we just sent a random seed instead of the whole response, as in the implementation of our CPA-anonymous scheme. The results in Table III show that the CCA-anonymous version requires only a small overhead for key generation, signing, and verification. For example, the key generation algorithm took about 5 seconds more, which corresponds to the key generation time for the McEliece encryption. When the number of group users was less than 2^{20}, it also took about 0.020 seconds and 0.015 seconds more to generate a signature and to verify it, respectively. The overheads for signing were increased slightly once the number of group users reached 2^{24}, but they account for about 2.28% and 2.73% of the total signing times for 1 B and 1 GB messages, respectively.

In conclusion, to our best knowledge, the implementation results presented here are the first ones for group signatures from quantum-resistant assumptions. We have thus demonstrated that our schemes, while not yet truly practical, are bringing this class of group signatures closer to practice.

VII. CONCLUSION

We put forward two provably secure code-based group signature schemes in the random oracle model. The first scheme satisfies the CPA-anonymity and traceability requirements for group signatures under the assumed hardness of three well-known problems in code-based cryptography. These are the McEliece problem, the Learning Parity with Noise problem, and a variant of the Syndrome Decoding problem. We extend the basic scheme to achieve CCA-anonymity by exploiting the Naor-Yung transformation. The feasibility of the proposed schemes is backed by implementation results.

The work we presented here inaugurates a foundational step in code-based group signatures. The natural continuation is to work towards either one of the following goals: constructing practically efficient schemes whose signature sizes are sub-linear in the number of group users and obtaining provably secure schemes in either the standard model or in the quantum random oracle model discussed in, *e.g.*, [72].

REFERENCES

[1] D. Chaum and E. van Heyst, "Group signatures," in *EUROCRYPT* (Lecture Notes in Computer Science), vol. 547. Berlin, Germany: Springer, 1991, pp. 257–265.

[2] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *CRYPTO* (Lecture Notes in Computer Science), vol. 1294. Berlin, Germany: Springer, 1997, pp. 410–424.

[3] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *CRYPTO* (Lecture Notes in Computer Science), vol. 1880. Berlin, Germany: Springer, 2000, pp. 255–270.

[4] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *CRYPTO* (Lecture Notes in Computer Science), vol. 3152. Berlin, Germany: Springer, 2004, pp. 41–55.

[5] X. Boyen and B. Waters, "Compact group signatures without random oracles," in *EUROCRYPT* (Lecture Notes in Computer Science), vol. 4004. Berlin, Germany: Springer, 2006, pp. 427–444.

[6] B. Libert, T. Peters, and M. Yung, "Scalable group signatures with revocation," in *EUROCRYPT* (Lecture Notes in Computer Science), vol. 7237. Berlin, Germany: Springer, 2012, pp. 609–627.

[7] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, Jan. 1999.

[8] S. D. Gordon, J. Katz, and V. Vaikuntanathan, "A group signature scheme from lattice assumptions," in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 6477. Berlin, Germany: Springer, 2010, pp. 395–412.

[9] J. Camenisch, G. Neven, and M. Rückert, "Fully anonymous attribute tokens from lattices," in *Security and Cryptography for Networks* (Lecture Notes in Computer Science), vol. 7485. Berlin, Germany: Springer, 2012, pp. 57–75.

[10] F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé, "Lattice-based group signatures with logarithmic signature size," in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 8270. Berlin, Germany: Springer, 2013, pp. 41–61.

[11] A. Langlois, S. Ling, K. Nguyen, and H. Wang, "Lattice-based group signature scheme with verifier-local revocation," in *Public-Key Cryptography* (Lecture Notes in Computer Science), vol. 8383. Berlin, Germany: Springer, 2014, pp. 345–361.

[12] S. Ling, K. Nguyen, and H. Wang, "Group signatures from lattices: Simpler, tighter, shorter, ring-based," in *Public-Key Cryptography* (Lecture Notes in Computer Science), vol. 9020. Berlin, Germany: Springer, 2015, pp. 427–449.

[13] P. Nguyen, J. Zhang, and Z. Zhang, "Simpler efficient group signatures from lattices," in *Public-Key Cryptography* (Lecture Notes in Computer Science), vol. 9020. Berlin, Germany: Springer, 2015, pp. 401–426.

[14] M. Bellare, D. Micciancio, and B. Warinschi, "Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions," in *EUROCRYPT* (Lecture Notes in Computer Science), vol. 2656. Berlin, Germany: Springer, 2003, pp. 614–629.

[15] N. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme," in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 2248. Berlin, Germany: Springer, 2001, pp. 157–174.

[16] L. Dallot, "Towards a concrete security proof of Courtois, Finiasz and Sendrier signature scheme," in *WEWoRC* (Lecture Notes in Computer Science), vol. 4945. Berlin, Germany: Springer, 2008, pp. 65–77.

[17] M. Finiasz, "Parallel-CFS: Strengthening the CFS McEliece-based signature scheme," in *Selected Areas in Cryptography* (Lecture Notes in Computer Science), vol. 6544. Berlin, Germany: Springer, 2010, pp. 159–170.

[18] K. P. Mathew, S. Vasant, and C. P. Rangan, "On provably secure code-based signature and signcryption scheme," IACR Cryptol. ePrint Arch., Tech. Rep. 2012/585, 2012. [Online]. Available: http://eprint.iacr.org/2012/585

[19] J.-C. Faugere, V. Gauthier-Umana, A. Otmani, L. Perret, and J.-P. Tillich, "A distinguisher for high-rate McEliece cryptosystems," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6830–6844, Oct. 2013.

[20] J. Stern, "A new paradigm for public key identification," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1757–1768, Nov. 1996.

[21] P. Véron, "Improved identification schemes based on error-correcting codes," *Applicable Algebra Eng., Commun. Comput.*, vol. 8, no. 1, pp. 57–69, Jan. 1997.

[22] P. Cayrel, P. Véron, and S/ M. El Yousfi Alaoui, "A zero-knowledge identification scheme based on the *q*-ary syndrome decoding problem," in *Selected Areas in Cryptography*. Berlin, Germany: Springer, 2010, pp. 171–186.

[23] C. Aguilar, P. Gaborit, and J. Schrek, "A new zero-knowledge code based identification scheme with reduced communication," in *Proc. IEEE Inf. Theory Workshop*, Oct. 2011, pp. 648–652.

[24] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *CRYPTO* (Lecture Notes in Computer Science), vol. 263. Berlin, Germany: Springer, 1986, pp. 186–194.

[25] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *Deep Space Netw. Progr. Rep.*, vol. 44, pp. 114–116, Oct. 1978.

[26] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Problems Control Inf. Theory*, vol. 15, no. 2, pp. 159–166, 1986.

[27] R. Nojima, H. Imai, K. Kobara, and K. Morozov, "Semantic security for the McEliece cryptosystem without random oracles," *Des., Codes Cryptogr.*, vol. 49, nos. 1–3, pp. 289–305, Mar. 2008.

[28] N. Dottling, R. Dowsley, J. Muller-Quade, and A. C. A. Nascimento, "A CCA2 secure variant of the McEliece cryptosystem," *IEEE Trans. Inf. Theory*, vol. 58, no. 10, pp. 6672–6680, Oct. 2012.

[29] E. Persichetti, "On a CCA2-secure variant of McEliece in the standard model," IACR Cryptol. ePrint Arch., Tech. Rep. 2012/168, 2012. [Online]. Available: http://eprint.iacr.org/2012/268

[30] K. P. Mathew, S. Vasant, S. Venkatesan, and C. P. Rangan, "An efficient IND-CCA2 secure variant of the Niederreiter encryption scheme in the standard model," in *Proc. Australas. Conf. Inf. Secur. Privacy*. Berlin, Germany: Springer, 2012, pp. 166–179.

[31] P. Cayrel, P. Gaborit, and M. Girault, "Identity-based identification and signature schemes using correcting codes," in *Proc. WCC*, 2007, pp. 69–78.

[32] S. M. E. Y. Alaoui, P. Cayrel, and M. Meziani, "Improved identity-based identification and signature schemes using quasi-dyadic Goppa codes," in *Proc. Int. Conf. Inf. Secur. Assurance*, 2011, pp. 146–155.

[33] G. Yang, C. H. Tan, Y. Mu, W. Susilo, and D. S. Wong, "Identity based identification from algebraic coding theory," *Theor. Comput. Sci.*, vol. 520, pp. 51–61, Feb. 2014.

[34] C. A. Melchor, P. -L. Cayrel, P. Gaborit, and F. Laguillaumie, "A new efficient threshold ring signature scheme based on coding theory," in *PQCrypto* (Lecture Notes in Computer Science), vol. 5299. Berlin, Germany: Springer, 2008, pp. 1–16.

[35] C. A. Melchor, P.-L. Cayrel, P. Gaborit, and F. Laguillaumie, "A new efficient threshold ring signature scheme based on coding theory," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4833–4842, Jul. 2011.

[36] R. Hu, K. Morozov, and T. Takagi, "Proof of plaintext knowledge for code-based public-key encryption revisited," in *Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur. (ASIA CCS)*, 2013, pp. 535–540.

[37] P.-L. Cayrel and M. Meziani, "Post-quantum cryptography: Code-based signatures," in *Advances in Computer Science and Information Technology* (Lecture Notes in Computer Science), vol. 6059. Berlin, Germany: Springer, 2010, pp. 82–99.

[38] J. Camenisch and V. Shoup, "Practical verifiable encryption and decryption of discrete logarithms," in *CRYPTO* (Lecture Notes in Computer Science), vol. 2729. Berlin, Germany: Springer, 2003, pp. 126–144.

[39] J. Fischer and J. Stern, "An efficient pseudo-random generator provably as secure as syndrome decoding," in *EUROCRYPT* (Lecture Notes in Computer Science), vol. 1070. Berlin, Germany: Springer, 1996, pp. 245–255.

[40] S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan, "Robustness of the learning with errors assumption," in *Proc. ICS*. Beijing, China: Tsinghua Univ. Press, 2010, pp. 230–240.

[41] A. Meurer, "A coding-theoretic approach to cryptanalysis," Ph.D. dissertation, Dept. Math., Horst Görtz Inst. IT-Secur., Ruhr-Univ. Bochum, Bochum, Germany, 2013. [Online]. Available: http://www.cits.rub.de/imperia/md/content/diss.pdf

[42] M. Finiasz and N. Sendrier, "Security bounds for the design of code-based cryptosystems," in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 5912. Berlin, Germany: Springer, 2009, pp. 88–105.

[43] A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in $2^{n/20}$: How $1+1 = 0$ improves information set decoding," in *EUROCRYPT* (Lecture Notes in Computer Science), vol. 7237. Berlin, Germany: Springer, 2012, pp. 520–536.

[44] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 14th Annu. ACM Symp. Theory Comput. (STOC)*, 2008, pp. 197–206.

[45] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 2248. Berlin, Germany: Springer, 2001, pp. 552–565.

[46] S. Bettaieb and J. Schrek, "Improved lattice-based threshold ring signature scheme," in *PQCrypto* (Lecture Notes in Computer Science), vol. 7932. Berlin, Germany: Springer, 2013, pp. 34–51.

[47] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," in *EUROCRYPT* (Lecture Notes in Computer Science), vol. 6110. Berlin, Germany: Springer, 2010, pp. 523–552.

[48] 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* (Lecture Notes in Computer Science), vol. 8873. Berlin, Germany: Springer, 2014, pp. 551–572.

[49] 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* (Lecture Notes in Computer Science), vol. 9666. Berlin, Germany: Springer, 2016, pp. 1–31.

[50] S. Ling, K. Nguyen, H. Wang, and Y. Xu, "Constant-size group signatures from lattices," in *Public-Key Cryptography* (Lecture Notes in Computer Science), vol. 10770. Cham, Switzerland: Springer, 2018, pp. 58–88.

[51] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," in *Proc. 22nd Annu. ACM Symp. Theory Comput. (STOC)*, 1990, pp. 427–437.

[52] A. Sahai, "Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security," in *Proc. 40th Annu. Symp. Found. Comput. Sci.*, Oct. 1999, pp. 543–553.

[53] 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* (Lecture Notes in Computer Science), vol. 9452. Berlin, Germany: Springer, 2015, pp. 260–285.

[54] Q. Alameélou, O. Blazy, S. Cauchie, and P. Gaborit, "A code-based group signature scheme," in *Proc. 9th Int. Workshop Coding Cryptogr. (WCC)*, P. Charpin N. Sendrier, and J.-P. Tillich, Eds. France, Apr. 2015.

[55] Q. Alamélou, O. Blazy, S. Cauchie, and P. Gaborit, "A code-based group signature scheme," *Des., Codes Cryptogr.*, vol. 82, nos. 1–2, pp. 469–493, Sep. 2016.

[56] S. Ling, K. Nguyen, A. Roux-Langlois, and H. Wang, "A lattice-based group signature scheme with verifier-local revocation," *Theor. Comput. Sci.*, vol. 730, pp. 1–20, Jun. 2018.

[57] K. Nguyen, H. Tang, H. Wang, and N. Zeng, "New code-based privacy-preserving cryptographic constructions," IACR, Cryptol. ePrint Arch., Tech. Rep. 2019/513, 2019. [Online]. Available: https://eprint.iacr.org/2019/513

[58] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang, "Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions," in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 10032. Berlin, Germany: Springer, 2016, pp. 373–403.

[59] C. Boschini, J. Camenisch, and G. Neven, "Floppy-sized group signatures from lattices," in *Applied Cryptography and Network Security* (Lecture Notes in Computer Science), vol. 10892. Cham, Switzerland: Springer, 2018, pp. 163–182.

[60] R. del Pino, V. Lyubashevsky, and G. Seiler, "Lattice-based group signatures and zero-knowledge proofs of automorphism stability," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 574–591.

[61] E. Berlekamp, R. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 384–386, 1978.

[62] N. Döttling, "Cryptography based on the hardness of decoding," Ph.D. dissertation, Dept. Inform., Inst. Cryptogr. Secur., Karlsruhe Inst. Technol., Karlsruhe, Germany, 2014. [Online]. Available: https://crypto.iti.kit.edu/fileadmin/User/Doettling/thesis.pdf

[63] A. Kawachi, K. Tanaka, and K. Xagawa, "Concurrently secure identification schemes based on the worst-case hardness of lattice problems," in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 5350. Berlin, Germany: Springer, 2008, pp. 372–389.

[64] S. Ling, K. Nguyen, D. Stehlé, and H. Wang, "Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications," in *Public-Key Cryptography* (Lecture Notes in Computer Science), vol. 7778. Berlin, Germany: Springer, 2013, pp. 107–124.

[65] J. Groth, "Evaluating security of voting schemes in the universal composability framework," in *Applied Cryptography and Network Security* (Lecture Notes in Computer Science), vol. 3089. Berlin, Germany: Springer, 2004, pp. 46–60.

[66] D. Pointcheval and S. Vaudenay, "On provable security for digital signature algorithms," Lab. d'Informatique de Ecole Normale Superieure, Paris, France, Tech. Rep. LIENS-96-17, 1997.

[67] *NTL: A Library for Doing Number Theory Version 11.3.0.* Accessed: Oct. 1, 2018. [Online]. Available: http://www.shoup.net/ntl/

[68] *Gf2x Library, Ver. 1.2.* Accessed: Oct. 1, 2018. [Online]. Available: https://gforge.inria.fr/projects/gf2x/

[69] N. Patterson, "The algebraic decoding of Goppa codes," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 203–207, Mar. 1975.

[70] B. Biswas and N. Sendrier, "McEliece cryptosystem implementation: Theory and practice," in *PQCrypto* (Lecture Notes in Computer Science), vol. 5299. Berlin, Germany: Springer, 2008, pp. 47–62.

[71] N. Sendrier, "QC-MDPC-McEliece: A public-key code-based encryption scheme based on quasi-cyclic moderate density parity check codes," in *Proc. Workshop Post-Quantum Cryptogr., Recent Results Trends*, Fukuoka, Japan, Nov. 2014, pp. 1–19. [Online]. Available: https://hal.inria.fr/hal-01095935

[72] D. Boneh, O. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 7073. Berlin, Germany: Springer, 2011, pp. 41–69.

Martianus Frederic Ezerman was born in East Java, Indonesia. He received the B.A. degree in philosophy, the B.Sc. degree in mathematics, and the M.Sc. degree in mathematics from Ateneo de Manila University, Philippines, in 2005 and 2007, respectively, and the Ph.D. degree in mathematics from Nanyang Technological University (NTU), Singapore, in 2011.

He was a Research Fellow at the Laboratoire d'Information Quantique, Université Libre de Bruxelles, Belgium, and at the Centre for Quantum Technologies (CQT), National University of Singapore. After that, he returned to NTU in March 2014, where he is currently a Senior Research Fellow. He is interested in coding theory, cryptography, and quantum information processing.

Hyung Tae Lee received the B.Sc., M.Sc., and Ph.D. degrees in mathematics from Seoul National University, South Korea, in 2006, 2008, and 2013, respectively.

He is currently an Assistant Professor with the Division of Computer Science and Engineering, College of Engineering, Jeonbuk National University, South Korea. Prior to that, he was a Research Fellow with Nanyang Technological University, Singapore. His research interests include computational number theory, cryptography, and information security.

San Ling received the B.A. degree in mathematics from the University of Cambridge and the Ph.D. degree in mathematics from the University of California at Berkeley.

He is currently the President's Chair in mathematical sciences at the Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, where he joined in April 2005. Prior to that, he was with the Department of Mathematics, National University of Singapore. His research fields include arithmetic of modular curves and application of number theory to combinatorial designs, coding theory, cryptography, and sequences.

Khoa Nguyen received the M.Sc. degree from Moscow State University in 2008 and the Ph.D. degree from Nanyang Technological University in 2014.

He is currently a Gopalakrishnan—NTU Presidential Postdoctoral Fellow at Nanyang Technological University. His research interests are in the areas of cryptography and information security.

Huaxiong Wang received the Ph.D. degree in mathematics from the University of Haifa, Israel, in 1996, and the Ph.D. degree in computer science from the University of Wollongong, Australia, in 2001.

Since 2006, he has been an Associate Professor with the Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. His research interests include cryptography, information security, coding theory, and theoretical computer science.