Lattice-based Commit-Transferrable Signatures and Applications to Anonymous Credentials

Qiqi Lai[1,2], Feng-Hao Liu[3], Anna Lysyanskaya[4], Zhedong Wang[5,2]

[1] School of Computer Science, Shaanxi Normal University, Xi'an, China
laiqq@snnu.edu.cn.
[2] State Key Laboratory of Cryptology, P. O. Box 5159, Beijing ,100878,China
[3] Florida Atlantic University, Boca Raton, FL, USA
fenghao.liu@fau.edu.
[4] Brown University, Providence, RI, USA
anna@cs.brown.edu.
[5] School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
wzdstill@sjtu.edu.cn.

Abstract. Anonymous Credentials are an important tool to protect user's privacy for proving possession of certain credentials. Although various efficient constructions have been proposed based on pre-quantum assumptions, there have been limited accomplishments in the post-quantum and especially practical settings. This research aims to derive new methods that enhance the current state of the art.

To achieve this, we make the following contributions. By distilling prior design insights, we propose a new primitive to instantiate *signature with protocols*, called commit-transferrable signature (CTS). When combined with a multi-theorem straight-line extractable non-interactive zero-knowledge proof of knowledge (NIZKPoK), CTS gives a modular approach to construct anonymous credentials. We then show efficient instantiations of CTS and the required NIZKPoK from lattices, which are believed to be post-quantum hard. Finally, we propose concrete parameters for the CTS, NIZKPoK, and the overall Anonymous Credentials, based on Module-SIS and Ring-LWE. This would serve as an important guidance for future deployment in practice.

Keywords: Anonymous Credentials; Commit-Transferrable Signature; Lattice-Based Cryptography; Post-Quantum Security

1 Introduction

In an Anonymous Credential system [15, 17, 40, 41], users interact with organizations, obtain digital credentials from them, and prove possession of these credentials anonymously and unlinkably. Anonymous Credentials are increasingly important in practice: they have been implemented by industry leaders such as IBM and Microsoft, have found their way into industrial standards (such as the

TCG standard), and underlined the policies that both the United States government and the EU government have towards balancing privacy and legitimate identification and authentication needs.

The advent of quantum computing threatens the security of all the prior anonymous credential constructions whose efficiency was suitable for use in practice, since all of them require either the RSA or the discrete logarithm assumption to hold (in fact, they need even stronger assumptions). The goal of this paper is to give more efficient Anonymous Credentials based on standard lattice assumptions, as they provide a plausible foundation against quantum attacks. Moreover, we propose a modular approach so that each building block might be improved individually for future work.

Anonymous Credentials from general-purpose crypto tools. The well-known approach [16, 42] to giving Anonymous Credentials is to provide: (1) a commitment scheme for committing to $\boldsymbol{x}$ representing a user's private input, e.g., her secret key; (2) a digital signature scheme for signing $\boldsymbol{x}$ (under the commitment); (3) an efficient and secure two-party protocol between a user and a signer to prove that the user's (private) input $\boldsymbol{x}$ is consistent with the commitment and then the protocol generates a signature of $\boldsymbol{x}$; and finally (4) a suite of efficient zero-knowledge proof systems that allow the user to prove (i) knowledge of the commitment opening and (ii) knowledge of a signature from the signer on the commitment opening.

Even though each of these general building blocks can be achieved under post-quantum assumptions, however, realizing them efficiently is still a significant and on-going research direction. Therefore, it is interesting and important to determine a new approach for more efficient constructions.

Relevant Research. We notice that the research of Anonymous Credentials has deep connections with the following two cryptographic objects: (1) Group Signatures and (2) Blind Signatures. Conceptually for all these objects, there are three roles – User, CA (certificate authority), Verifier, in the system, yet they post different privacy requirements. Particularly, Group Signatures requires privacy for User against Verifier, i.e., Verifier only knows the signature is output by some one in the group, but does not know the concreted signer. Yet group signature does not require privacy for User when getting a credential from CA. On the other hand, Blind Signatures requires privacy for User against the CA, i.e., CA signs a hidden message as the credential, but the verification step reveals the message to Verifier. Finally, Anonymous Credentials requires privacy on both sides – the User's private input (e.g., her ID or messages) is hidden from both CA and Verifier.

Our Approach. Our goal is to achieve an efficient lattice-based anonymous credential, avoiding any heavy cryptographic machinery such as general secure two-party computation and general zero-knowledge proofs in the above paradigm. To achieve this goal, we first distill prior design insights, such as *signature with protocols* in [16], from prior work that realized the above general diagram,

and then propose a primitive named as commit-transferable signature (CTS), with special properties that are friendly for efficient instantiating from lattices.

Briefly, the high level idea of the prior framework of *signature with protocols* is the following – for an efficient signature scheme, we would identify a commitment scheme and an efficient zero-knowledge proof of knowledge system, such that they can be elegantly combined, yielding a protocol for signing a committed value, and a zero-knowledge proof of knowledge of the signature. This work takes another perspective – by blending a signature scheme with a proper commitment scheme as one object, we are able to see better lattice insights, leading to more efficient lattice-based instantiation and thus more practical anonymous credentials.

Particularly, CTS encompasses both a non-interactive commitment algorithm Commit and a signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$, with the following properties. Our first key property is that, in a CTS scheme, it is possible to compute a signature directly on the commitment value comm, where $\mathsf{comm} \leftarrow \mathsf{Commit}(\boldsymbol{x})$. This way, instead of designing a secure two-party protocol as described in (3) above, it is sufficient to simply require that the user performs a zero-knowledge proof of knowledge of the opening of comm. Once the signer verifies this proof, it can compute the signature σ on input comm. Moreover, it is possible to verify a signature σ through inputting the commitment comm rather than the value $\boldsymbol{x}$; therefore, to prove possession of a signature on the opening of comm, it is sufficient to reveal σ and prove knowledge of the opening of comm.

Our second key property is to require that, from the signature σ of the commitment $\mathsf{comm} \leftarrow \mathsf{Commit}(\boldsymbol{x})$, the user will be able to compute a new signature σ' for a new commitment $\mathsf{comm}' \leftarrow \mathsf{Commit}(\boldsymbol{x})$. That requires two additional algorithms: $\mathsf{Randomize}$ to randomize comm into comm', and $\mathsf{Transfer}$ to transform the signature σ into a new signature σ' with respect to the new commitment comm'. It is important that the resulting pair $(\mathsf{comm}', \sigma')$ is unlinkable to the original pair (comm, σ). Thus, in order to prove that the contents of comm' were signed, it is sufficient to just reveal σ'. More technical details and the formulation on CTS are deferred to Section 1.3.

From CTS to Anonymous Credentials and More. Using a CTS scheme and an appropriate (non-interactive) zero-knowledge proof (NIZK), we can construct Anonymous Credentials and the other related object, namely, Group Signatures and Blind Signatures. We first elaborate on the case of Anonymous Credentials.

Suppose User whose secret key is $\boldsymbol{x}$ needs to obtain a credential from some CA. First, it forms a commitment $\mathsf{comm} \leftarrow \mathsf{Commit}(\boldsymbol{x})$ and proves to CA that he knows the opening to the commitment. Next, CA runs the Sign algorithm on input comm, obtains the signature σ, and returns it to the user. After the signature is obtained, suppose that the user wants to prove possession of this credential, he uses the $\mathsf{Randomize}(\cdot)$ and $\mathsf{Transfer}(\cdot)$ algorithms to obtain a new commitment comm' to $\boldsymbol{x}$ and the issuer's signature σ' on comm'. Now, the user sends the resulting $(\mathsf{comm}', \sigma')$ to the verifier as a credential. As $(\mathsf{comm}', \sigma')$ is unlinkable to the original (comm, σ), we achieve the important property of unlinkability. We can further prove that it is computationally infeasible to forge

a σ^* with respect to $\mathsf{comm}^* = \mathsf{Commit}(\boldsymbol{x}^*)$ that a signature of commitment of $\boldsymbol{x}^*$ has never been issued.

By instantiating a half-fledged CTS (which might allow more efficient instantiations), we can achieve *Group Signatures* and *Blind Signatures*. Particularly, for Group Signatures, User only needs to send $\boldsymbol{x}$ in the clear at the first stage, i.e., viewing $\boldsymbol{x}$ as the trivial commitment, as the privacy of User is not required in this phase. After obtaining σ, User runs $\mathsf{Randomize}(\cdot)$ and $\mathsf{Transfer}(\cdot)$ to produce a hiding comm' and a corresponding signature σ'.[6] On the other hand, for Blind Signatures, User follows the first half of the Anonymous Credential construction, yet later modify the randomized algorithm as: comm' just reveals $\boldsymbol{x}$. Again this is not an issue as the privacy of Blind Signatures is not required against Verifier. In fact, the constructions of [22,23] can be viewed as realizing the half-fledged of our notion of CTS. Besides, as CTS is essentially a non-interactive version of "signature with protocols", this notion may be useful for other privacy-preserving applications related to "signature with protocols".

For the NIZK system, the recent work [22] identified a necessary property, i.e., the system needs to be multi-theorem straight-line extractable, as otherwise, the security proof of the whole system (Blind Signatures or Anonymous Credentials) would incur an exponential loss. It is important to determine efficient multi-theorem straight-line extractable proof systems from lattices for the particular commitment relation in the CTS construction.

Focus of This Work. Our main goal is to construct an efficient Anonymous Credential based on some standard lattice assumptions. As discussed before, this can be achieved by determining the following questions.

> (**Main Questions**) Can we design an efficient full-fledged CTS from standard lattice assumptions? Can we construct an efficient straight-line extractable NIZK for the commitment relation of the CTS?

1.1 Our Contributions

To address the main questions, we make the following contributions.

- We formalize the notion of CTS and its security requirements. Together with a straight-line extractable NIZK, CTS gives a simple way to construct Anonymous Credentials and other useful privacy preserving tools, such as Group Signatures and Blind Signatures.
  Moreover, the CTS-based Anonymous Credentials can be extended to the attribute-based setting, by further embedding attributes to the committed message and designing proper NIZK to prove the message relation satisfying a certain policy.
- We show how to instantiate CTS from some well-studied lattices assumptions, i.e., the module learning with errors (M-LWE) and module short integer solutions (M-SIS).

[6] To be able to open the group signature scheme, we still need to add a verifiable encryption to the signature.

– We construct an efficient lattice-based straight-line extractable NIZKPoK for our CTS commitment relation in the classical random oracle model. To achieve this, we employ the encrypt-and-prove approach similar to the work [2]. To further enhance the overall efficiency, we have adopted various optimizations specifically tailored to our setting and parameters.
– We determine parameters for all the required components for evaluating concrete efficiency. Below we present our concrete parameters and findings.

In the following tables, we show concrete parameters of Anonymous Credentials of various security levels. Particularly, our simple yet selectively secure CTS (in Section 4) can derive selectively secure Anonymous Credentials, whose concrete parameters are presented in Table 1. By scaling up the security parameter and applying the complexity leveraging argument, we can derive adaptively secure Anonymous Credentials with concrete parameters[7] in Table 2. Alternatively, we also directly construct an adaptively secure CTS as in Section E, implying asymptotically efficient adaptively secure Anonymous Credentials with concrete parameters in Table 3. For the currently used security levels (say 131 bit-security) however, the scheme via the complexity leveraging (as Table 2) is much more efficient. We leave it as an interesting open problem to optimize such directly adaptive CTS and the derived Anonymous Credential.

	PP	PK	SK	Pseudonym	Signature	Credential	Bit-security
Params 1	1.926MB	238.5KB	8KB	1.659MB	153.88KB	232.53KB	172

Table 1. Our selective Anonymous Credentials from Ring-LWE and Modulus-SIS. Here, we denote PP as public parameter, PK as public key, SK as secret key. All values in this table are computed from the example parameters of Params 1 in the Tables 12 and 13.

	PP	PK	SK	Pseudonym	Signature	Credential	Bit-security
Params 1	4.09MB	513KB	16KB	3.53MB	325.249KB	499.15KB	131

Table 2. Our adaptively secure Anonymous Credentials by applying the complexity leveraging argument to the selectively secure scheme. All values in this table are computed from the example parameters of Params 2 in the Tables 12 and 13.

	PP	PK	SK	Pseudonym	Signature	Credential	Bit-security
Params	91.89MB	40.32MB	64KB	5839.324MB	2.117MB	3.003MB	134

Table 3. Our adaptive secure Anonymous Credentials from a direct construction of adaptively secure CTS. All values in this table are computed from the example parameters of the Table 15.

1.2 Comparison with Recent Progress

Here we present a comparison between our contributions and relevant recent works, for a clear identification of our advancements over the state of the art.

[7] Here we consider 224 bits for the User ID length, and scale up the selectively secure scheme to roughly 355-bit security. This implies an adaptively secure scheme of 131 bit-security after applying the complexity leveraging argument.

Anonymous Credentials. Several earlier works [18, 28, 39, 56] have made attempts to construct lattice-based anonymous credentials, yet their approaches have various drawbacks and thus unsatisfactory. Particularly, the work [18] only achieved a weaker notion called anonymous attribute token system, where user anonymity is protected only against verifiers, but not the CA. The schemes [18,39] are not concretely efficient, and the schemes [28,56] do not achieve the important property – unlinkability. Thus, all these approaches are not suitable for scenarios that require the full-fledged anonymous credentials.

Concurrent Works. Very recently, two independent and concurrent works [12, 36] have constructed efficient lattice-based anonymous credentials. Here we undertake a comparative analysis of the findings, highlighting the unique merits of our approach despite the existence of these concurrent works.

First, the work [36] instantiates the necessary building blocks following the approach of "signature with protocols", and then derives an anonymous credential system based on the M-LWE and M-SIS assumptions. In efficiency, the credential size of their protocol is about 639 KB for 128 bit-security. However, their efficient instantiations operate in an interactive setting, requiring multiple rounds of interaction between involved parties for all signing and verification protocols. It is *important* to note that a direct application of the Fiat-Shamir transform is not sufficient to achieve a non-interactive solution in this specific context. The reason is that the NIZKPoK (essential for ensuring the commitments' well-formness) must be straight-line extractable, as discussed earlier in this section. Consequently, their non-interactive variant necessitates the re-determination of concrete parameters with a large overhead expected.

The other work [12] takes a different approach to construct non-interactive lattice-based solutions in the random oracle model. Their scheme exhibits a highly competitive level of concrete efficiency. E.g., the size of credentials is about 122 - 133 KB for 128 bit-security, or 26 - 29 KB under another new assumption. However, there are two important caveats to consider. First, their efficient scheme only achieves a very basic anonymous credential system without incorporating pseudonyms, which can be desirable for enabling some useful features, e.g., selective tracking of holders [14,42]. Second, security of all their schemes [12] depends on some new variations of ideal lattice problems.

Even though these two issues can be handled in theory, it remains challenging to derive a system with comparable efficiency under their paradigm. In particular, resolving the first issue would require additional commitments and proofs on top of their basic schemes, e.g., proving equality between committed and signed values, yet the concrete blowup needs to be re-evaluated. The second issue presents a tougher challenge, as adapting their approach to rely on more well-studied assumptions (e.g., RLWE) appears to necessitate proving knowledge of pre-images for random oracles. Unfortunately, there is currently no efficient lattice-based proof technique available to fulfill this need.

Considering the insights gained from the current post-quantum standardization process [9,20,47,54], a cautious and conservative approach would always be necessary and valuable. By building schemes under more well-studied hardness

foundations, we can mitigate the risks associated with unforeseen weaknesses in the new assumptions, ensuring better confidences in the overall security.

Summary. Our work, along with the two concurrent works, possesses unique merits in different aspects. In summary, both our work and the work [36] achieve anonymous credential systems with pseudonyms under the more extensively studied assumptions (i.e., M-SIS and M-LWE). However, our work offers advantages over [36] in terms of smaller credential size and non-interactive protocols.

When comparing our work to [12], we observe that their concrete parameters are smaller, yet their efficient instantiation is for a basic anonymous credential system without pseudonyms, and moreover their security relies on new and less-studied assumptions. Thus, we believe that these two works have incomparable advantages and both deserve attentions. Below we present Table 4 for comparisons between our work and these concurrent works.

	PP	PK	SK	Pseudonym	Signature	Credential	Bit-security	ZK	Assumption
[36]	–	7.8MB	8.9MB	–	273KB	639KB	128	Inter.	M-LWE, M-SIS
[12]	–	–	–	$\perp$	–	122KB	128	Non.	ISIS_f
Ours	4.09MB	513KB	16KB	3.53MB	325.249KB	499.15KB	131	Non.	M-LWE, M-SIS

Table 4. Comparison of efficiency estimates of Anonymous Credentials Systems between ours, [36] and [12]. In the column of ZK, we use Inter. and Non. to denote "interactive" and "non-interactive", respectively. In [12, 36], some of concrete values are not explicitly listed, so we just use the symbol "-" for these columns. Besides, as the current construction of [12] does not support pseudonym application immediately, we just use the symbol $\perp$ to represent its size. Moreover, here we focus on the non-interactive version of the underlying assumptions, so we do not list the efficiency of [12] based on the interactive ISIS_f assumption.

Straight-line Extractable Lattice-based NIZK. Next we present relevant works of straight-line extraction for lattice proofs. Generally, there are two main approaches to achieve this notion for lattice proofs:

1. The technique of extractable linear homomorphic commitments, e.g., [12,22].
2. The instantiation of the well-known encrypt-and-prove paradigm from lattices, e.g., [2, 10].

For practical parameters, recent works [2, 10] have focused on optimizing proof sizes in the classical random oracle model (ROM), and currently, the second approach achieves smaller proof sizes [22].

This work follows the second, i.e., encrypt-and-prove approach, yet makes notable optimizations in our instantiation, distinguishing it from the state of the art as presented in [2]. Particularly, we note that the design of [2] is not optimized for scenarios involving larger witnesses. Consequently, when integrating our anonymous credentials, which require languages with larger witnesses compared to the simpler setting of blind signatures, their scheme [2] (as is) would result in significantly larger proof size, roughly 1512 KB to prove consistency of one BDLOP commitment for our parameter setting Params 2 in Table 12.[8]

[8] We note that each pseudonym in our design requires several BDLOP commitments and thus proofs.

To address this challenge, we have introduced different optimizations to further reduce the proof size to roughly 604 KB, based on which the efficiency of Anonymous Credentials system in Table 2 is calculated. More comprehensive details regarding these optimizations can be found in the technical overview provided in Section 1.3.

It is worth noting that the first approach has an advantage in terms of extendability of security analysis to the quantum random oracle model (QROM) [22], though a significant efficiency overhead of add-ons is required under current techniques. An interesting open problem is whether we can enhance the efficiency and analysis in the QROM settings for either the first or second approach. As further research is needed to explore potential improvements, we believe that all of the aforementioned works, including our own contributions, would provide valuable guidance and serve as stepping stones towards achieving this goal.

1.3 Technical Overview

We present an overview of our techniques of how to efficiently construct the required CTS and straight-line extractable NIZKPoK from lattices. First we informally describe the notion of CTS and then present our technical insights. Next, we present the intuition of our efficient instantiation of mult-theorem straight-line extractable NIZKPoK. These two pieces naturally give Anonymous Credentials as we discussed above.

Commit-transferrable Signatures. Informally, a CTS is a combination of a re-randomizable commitment and a signature, with the following algorithms $(\mathsf{Commit}, \mathsf{Randomize}, \mathsf{Sign}, \mathsf{Transfer}, \mathsf{Verify})$. Intuitively, a user can send $\mathsf{comm} \leftarrow \mathsf{Commit}(x)$ to the signer, who will run the algorithm Sign to produce a signature σ on the commitment comm. Later on, the user can re-randomize the commitment $\mathsf{comm}' \leftarrow \mathsf{Randomize}(\mathsf{comm})$ and then derive a transferred signature $\sigma' \leftarrow \mathsf{Transfer}(\mathsf{comm}, \mathsf{comm}', \sigma)$ with respect to the randomized commitment comm'. For security, the CTS requires input privacy, signature unlinkability, and unforgeability. These properties can be roughly captured by – (1) the signer does not learn any information of x, (2) one cannot learn information about the original commitment-signature pair (comm, σ) from the re-randomized-transferred pair $(\mathsf{comm}', \sigma')$, and (3) an adversary cannot forge a valid σ' with respect to $\mathsf{comm}' \leftarrow \mathsf{Commit}(x^*)$, if any commitment of x^* has not been signed by the signer. Below we explain how to construct such a primitive from lattices.

Warm Up. To achieve CTS, intuitively, the first step is to obtain a scheme that allows to sign on commitments, i.e., blending a commitment scheme and signature scheme in an appropriate way. This can be achieved by using ABB signature [1] and GSW commitment [35], as observed by the work [22,23]. Briefly, the ABB scheme has public key of the form $(\mathbf{A}_0, \mathbf{B}_0, \boldsymbol{u})$ (i.e., two matrices and one vector), and the secret key is the trapdoor, i.e., $\mathbf{T}_{\mathbf{A}_0}$, of the matrix $\mathbf{A}_0$. The signature of m is a short vector $\boldsymbol{s}$, satisfying $[\mathbf{A}_0|\mathbf{B}_0 + m\mathbf{G}] \cdot \boldsymbol{s} = \boldsymbol{u}$, where $\mathbf{G}$ is the gadget matrix of [49]. The GSW commitment uses a public matrix $\mathbf{A}$. To commit to a message m, it outputs $\mathbf{A} \cdot \mathbf{R} + m\mathbf{G}$, where $\mathbf{R}$ is a short random

matrix. To open, one just reveals the message and randomness. Next, we describe the idea of [23] to blend these two together.

To sign on commitment $\mathbf{C} = \mathbf{A} \cdot \mathbf{R} + m\mathbf{G}$, the signer first generates the matrix $\mathbf{F} = [\mathbf{A}_0 | \mathbf{B}_0 + \mathbf{C}]$, and then generates a short vector $\sigma := \boldsymbol{s}$ that satisfies $\mathbf{F} \cdot \boldsymbol{s} = \boldsymbol{u}$. This can be achieved by using the trapdoor sampling technique of [49]. Suppose the commitment $\mathbf{C}$ does not need to be re-randomized, then the user can simply generate a transferred signature σ' by a ZK proof of knowledge that she holds a short vector with respect to the lattice $\Lambda_{\boldsymbol{u}}^{\perp}(\mathbf{F}) = \{\boldsymbol{z} : \mathbf{F} \cdot \boldsymbol{z} = \boldsymbol{u}$ and $\boldsymbol{z}$ is short$\}$. Intuitively, the zero-knowledge property guarantees that one cannot learn information about the original signature σ from the transferred one, i.e., σ'. The unforgeability follows from the SIS using the ABB analysis of [1].

Handle Re-randomized Commitments. The above technique achieves a half of the goal, which means just transferring the original signature $\boldsymbol{s}$ to one (i.e., ZK proof π) with respect to the same commitment $\mathbf{C}$. To achieve the full-fledge of our goal, we need to handle how to transfer signatures with respect to a re-randomized commitment $\mathbf{C}'$.

We observe that GSW commitment can be easily re-randomized, i.e., just setting $\mathbf{C}' = \mathbf{C} + \mathbf{A} \cdot \mathbf{R}'$ for some short random matrix $\mathbf{R}'$. It is easy to show that given $(\mathbf{C}, \mathbf{C}')$, one cannot determine whether the underlying messages are related or not. Given this, we define another matrix for verification with respect to $\mathbf{C}'$ as $\mathbf{F}' = [\mathbf{A}_0 | \mathbf{B}_0 + \mathbf{C}' | \mathbf{A}]$. So now our goal is to generate a short vector $\boldsymbol{s}'$ such that $\mathbf{F}' \cdot \mathbf{s}' = \boldsymbol{u}$, and then set σ' to be a ZK proof of knowing a short vector in $\Lambda_{\boldsymbol{u}}^{\perp}(\mathbf{F}')$. By the security of GSW and ZK proof of knowledge, it is easy to argue that one cannot learn information about (comm, σ) from the $(\mathsf{comm}', \sigma')$.

To achieve this, we first express $\mathbf{F}' = [\mathbf{F}|\mathbf{0}] + [\mathbf{0}|\mathbf{A} \cdot \mathbf{R}'|\mathbf{A}] = [\mathbf{A}_0|\mathbf{B}_0 + \mathbf{C}|\mathbf{0}] + [\mathbf{0}|\mathbf{A} \cdot \mathbf{R}'|\mathbf{A}]$. Then through denoting $\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix}$, we observe, $\mathbf{F}' \cdot \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \mathbf{0} \end{bmatrix} = [\mathbf{A}_0|\mathbf{B}_0+\mathbf{C}] \cdot \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix} + \mathbf{A} \cdot \mathbf{R}' \cdot \boldsymbol{s}_2 = \boldsymbol{u} + \mathbf{A} \cdot \mathbf{R}' \cdot \boldsymbol{s}_2$, so if we can find a short $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \boldsymbol{z}_3 \end{bmatrix}$ such that $\mathbf{F}' \cdot \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \boldsymbol{z}_3 \end{bmatrix} = -\mathbf{A} \cdot \mathbf{R}' \cdot \boldsymbol{s}_2$, then $\boldsymbol{s}'$ can be simply set to $\begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \boldsymbol{z}_3 \end{bmatrix}$, fulfilling our goal. By the special structure of $\mathbf{F}'$, we can just set $\boldsymbol{z} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{R}' \cdot \boldsymbol{s}_2 \end{bmatrix}$. Thus the overall $\boldsymbol{s}' = \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ -\mathbf{R}' \cdot \boldsymbol{s}_2 \end{bmatrix}$. It is not hard to verify all the prior steps, implying that $\mathbf{F}' \cdot \boldsymbol{s}' = \boldsymbol{u}$.

Conceptually, the user can massage the randomness $\mathbf{R}'$ (for the re-randomization of the commitment) and the signature $\boldsymbol{s}$ obtained from the signer, to derive a related witness, i.e., $\boldsymbol{s}'$ for the related lattice $\Lambda_{\boldsymbol{u}}^{\perp}(\mathbf{F}')$. Thus, a ZK proof of knowledge can serve as the transferred signature σ' for the re-randomized $\mathbf{C}'$. We notice that lattice-based ZK proofs for general NP languages exist in the

standard model [51] albeit poor efficiency. On the other hand, the particular proof system we need can be instantiated efficiently in the random oracle model [31]. The whole approach can be further optimized by using ideal lattices, i.e., Ring-SIS/LWE, as identified by the work [5, 23, 46].

We notice that we can further improve efficiency of the construction idea above by using multiple BDLOP commitments on related messages [6], similar to the work [22,23]. Thus in our main construction, we will present in the BDLOP form, and our parameters are set with respect to this more efficient version.

Straight-line Extractable Proofs. The next important piece is to construct an efficient multi-theorem straight-line extractable NIZKPoK, proving the well-formness of the commitment in CTS. Informally, for a multi-theorem straight-line extractable proof, there exists an extractor who can extract multiple witnesses from an adversary who generates multiple valid proofs, and moreover the extraction does not need rewinding. As pointed out by [2, 10, 22], this is an important feature for non-interactive blind signatures and anonymous credentials. For our CTS, specifically we need to prove knowledge of BDLOP commitments, which we recall below. A BDLOP commitment of message m has the structure:

$$\mathsf{Commit}(m; \boldsymbol{r}) = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \cdot \boldsymbol{r} + \begin{bmatrix} \mathbf{0} \\ m \end{bmatrix} \quad \begin{matrix} \bmod q_1 \\ \bmod q_2 \end{matrix} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix},$$

where $\boldsymbol{r}$ is the randomness (ring elements with small coefficients) and m is the message. There are two different moduli with some flexibility in the design. For efficiency optimizations, we can set $q_1 \ll q_2$, and in our application, we additionally require q_2 to be a large prime of a special form, e.g., congruent to 3 or 5 modulo 8.

There are various efficient lattice proofs of knowledge about $m, \boldsymbol{r}$ in the literature [5,6,13,23,27] in the random oracle model. However, the knowledge extraction of these constructions requires to rewind the random oracle, and as pointed out by [22, 37], this would incur an exponential security loss in the application of blind signatures and anonymous credentials. To achieve efficient straight-line extractable proof, as we discussed in the prior section, we take the approach of encrypt-and-prove, which is currently better optimized than the other one using extractable linear homomorphic commitments.

The general paradigm is to encrypt the witness and then prove well-formness o the encryption and consistency of the encrypted witness (with the BDLOP commitment). In our specific case, we can just encrypt the randomness $\boldsymbol{r}$ of the above BDLOP commitment, i.e., $\mathsf{Enc}(\boldsymbol{r})$ and then prove well-formness of the encryption, upper bound of ℓ_2 norm for $\boldsymbol{r}$, and $\mathbf{A}_1 \cdot \boldsymbol{r} = t_1$. The $\boldsymbol{r}$ can be extracted easily in a straight-line manner, by decrypting the ciphertext given the secret key of Enc. Then one can derive $m := t_2 - \mathbf{A}_2 \boldsymbol{r}$, which would be consistent with what was originally committed to by the binding property of the commitment.

To instantiate this idea, one could consider the currently most optimized lattice proof (in the classical random oracle model) [2], which takes the following high-level step. First they instantiate a RLWE-type encryption scheme $\mathsf{Enc}(\cdot)$ and then use the ABDLOP commitment to commit to $\boldsymbol{r}$ and the randomness to gen-

erate the encryption $\mathsf{Enc}(\boldsymbol{r})$, say ρ. Then they use the LNP proof technique [44] to prove (1) the randomness ρ and $\boldsymbol{r}$ are small; (2) ρ and r satisfy the linear equation as in this particular encryption algorithm, implying that the ciphertext is well-formed; and (3) $\mathbf{A}_1 \cdot \boldsymbol{r} = t_1 \pmod{q_1}$.

In our specific setting, the lengths of $\boldsymbol{r}$ and ρ are larger due to the CTS design and analysis. As a consequence, the LNP proof especially for (2) (the exact relation of the encryption) would be particularly large, resulting in the proof size roughly 1512 KB under a conservative estimation. To further improve efficiency, we observe that we can make BDLOP commitment decryptable by adding a trapdoor of [49] to the public matrices. This variant of BDLOP can serve as our $\mathsf{Enc}(\cdot)$, which enjoys the rather efficient *relaxed proof* for the relations (1) and (2) in the above. Finally, we use LNP proof [44] just for the last part (3). Even though the relations for (1) and (2) are in the relaxed form, we can still show consistency between the extracted witness and what was really encrypted under the hardness of $\mathsf{M\text{-}SIS}$. Combining these ideas, we can reduce the proof size to roughly 604 KB. We present the details in Section 5.

One More Subtlety. We identify a technical subtlety – for our anonymous construction, there still remains a gap towards the full overall security, even if one proves well-formness of BDLOP commitments using a straight-line extractable proof, due to a possible mix-and-match attack. To tackle this, we identify a stronger form of well-formness, where it is computationally infeasible to generate tuples (t_1, t_1', t_2) such that both (t_1, t_2) and (t_1', t_2) can be proved well-formed. This stronger property suffices for deriving secure anonymous credentials and can be realized in a simple and efficient way. We present more details in Section 2.3.

2 Preliminaries

Notations. $\mathbb{Z}$ and $\mathbb{R}$ denote the sets of integers and real numbers. Throughout this paper, we use λ to denote the security parameter, which is the implicit input for all algorithms. A function $f(\lambda) > 0$ is negligible and denoted by $\mathsf{negl}(\lambda)$ if for any $c > 0$ and sufficiently large λ, $f(\lambda) < 1/\lambda^c$. A probability is called to be overwhelming if it is $1 - \mathsf{negl}(\lambda)$. A column vector is denoted by a bold lower case letter (e.g., $\boldsymbol{x}$). A matrix is denoted by a bold upper case letter (e.g., $\mathbf{A}$). For a vector $\boldsymbol{x}$, its Euclidean norm (also known as the ℓ_2 norm) is defined to be $\|\boldsymbol{x}\| = (\sum_i x_i^2)^{1/2}$, and its infinity norm is defined to be $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$. For a matrix $\mathbf{A}$, its ith column vector is denoted by $\boldsymbol{a}_i$ and its transposition is denoted by $\mathbf{A}^\top$. The Euclidean norm of a matrix is the ℓ_2 norm of its longest column: $\|\mathbf{A}\| = \max_i \|\boldsymbol{a}_i\|$. For any matrix $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$, we use $\widetilde{\mathbf{B}} = (\tilde{\boldsymbol{b}}_1, \ldots, \tilde{\boldsymbol{b}}_m)$ to denote the Gram-Schmidt orthogonalization of $\mathbf{B}$. Besides, we refer to $\|\widetilde{\mathbf{B}}\|$ as the Gram-Schmidt norm of $\mathbf{B}$. Let $R = \mathbb{Z}[x]/(x^d + 1)$ be a cyclotomic ring, with d be a power of 2. And the norm of an element in R_q will be the norm of its unique representative with coefficients in $[-(q-1)/2, (q-1)/2]$. For matrix $\mathbf{A}$ in $R^{\ell \times l}$, we use $s_1(\mathbf{A}) = \max_{\|\boldsymbol{x}\|} \left(\frac{\|\mathbf{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \right)$ to denote its operator norm. For positive $\beta \in \mathbb{R}$, we use S_β to denote the set of all polynomials of infinity norm less than β, i.e., $S_\beta = \{a \in \mathcal{R} \mid \|a\|_\infty \leq \beta\}$.

For positive integers n, q, let $[n]$ denote the set $\{1, ..., n\}$ and $\mathbb{Z}_q$ denote the ring of integers modulo q. For a distribution or a set $\mathcal{X}$, we write $x \xleftarrow{\$} \mathcal{X}$ to denote the operation of sampling an uniformly random x according to $\mathcal{X}$. For two distributions $\mathcal{X}, \mathcal{Y}$, we let $\mathsf{SD}(\mathcal{X}, \mathcal{Y})$ denote their statistical distance. We write $\mathcal{X} \overset{s}{\approx} \mathcal{Y}$ to mean that they are statistically close, and $\mathcal{X} \overset{c}{\approx} \mathcal{Y}$ to say that they are computationally indistinguishable.

Due to the space limit, we defer the detailed background notations, definitions, and lemmas on lattices, rejection sampling, and algebraic structure of cyclotomic rings to Appendices A.1, A.2, and A.4, respectively.

2.1 M-LWE and M-SIS

Now we introduce the hard problems on which our schemes rely, which are denoted as M-LWE and M-SIS.

Definition 2.1 (M-SIS [38]) *The* $\mathsf{M\text{-}SIS}_{q,\ell,m,\beta}$ *problem (over an implicit ring R) is defined as follows. Given an uniformly random matrix $\mathbf{A} \in R_q^{\ell \times m}$, output vector $\boldsymbol{z} \in R^m$ such that $\mathbf{A}\boldsymbol{z} = 0$ and $0 < \|\boldsymbol{z}\| \leq \beta$.*

Definition 2.2 (M-LWE [38]) *The decision* $\mathsf{M\text{-}LWE}_{q,\ell,m}$ *problem (over an implicit ring R) is defined as follows. For $\boldsymbol{s} \xleftarrow{\$} S_1^\ell$, use $A_{q,\boldsymbol{s}}$ to denote the distribution of $(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e) \in R_q^\ell \times R_q$, where $\boldsymbol{a} \xleftarrow{\$} R_q^\ell$ and $e \xleftarrow{\$} S_1$. The goal is to distinguish m samples from either $A_{q,\boldsymbol{s}}$ or $\mathcal{U}(R_q^\ell, R_q)$.*

Notice that for $\mathsf{M\text{-}LWE}_{q,\ell,m}$, if $\ell = 1$, it can also be called as $\mathsf{RLWE}_{q,1,m}$.

2.2 Syntax of Commitment

We give a formal definition of commitment schemes, following the presentation of [6,23]. A commitment scheme consists three algorithms $(\mathsf{CKeyGen}, \mathsf{Commit}, \mathsf{Open})$, with the security parameter 1^λ as implicit input:

$\mathsf{CKeyGen}$ is a PPT algorithm that outputs the public parameters params containing the descriptions of the message space $\mathcal{M}$ and randomness space $\mathcal{R}$.
Commit is a PPT algorithm that, on input the public parameters params and a message $x \in \mathcal{M}$, outputs the commitment c and its related randomness $r \in \mathcal{R}$.
Open is a deterministic poly-time algorithm that, on input the public parameters params, a message $x \in \mathcal{M}$ and values c and $r \in \mathcal{R}$, outputs a bit $b \in \{0, 1\}$.

A secure commitment scheme requires the two properties: hiding and binding. We defer the presentation to Appendix A.3.

2.3 BDLOP Commitment Scheme

We use as a building block the efficient lattice-based commitment scheme in [6, 23], implicitly denoted as BDLOP Commitment. Particularly, BDLOP Commitment consists of three algorithms ($\mathsf{CKeyGen}$,Commit,Open) as follows.

– $\mathsf{CKeyGen}\ (1^\lambda)$: Given the security parameter λ as input, the algorithm first sets the parameters n, k, ℓ, q_1, q_2, and ring $R = \mathbb{Z}[x]/\langle x^N + 1\rangle$ where N is a power of 2, or other cyclotomic rings as Table 5, and then chooses random matrices $\mathbf{A}_1' \xleftarrow{\$} R_{q_1}^{n\times(k-n)}$ and $\mathbf{A}_2' \xleftarrow{\$} R_{q_2}^{\ell\times(k-n-\ell)}$. Finally, the algorithm outputs the public paramters $\mathsf{params} := \mathbf{A}_0 = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}$ with $\mathbf{A}_1 := [\mathbf{I}_n, \mathbf{A}_1'] \in R_{q_1}^{n\times k}$, $\mathbf{A}_2 := [\mathbf{0}^{\ell\times n}, \mathbf{I}_\ell, \mathbf{A}_2'] \in R_{q_2}^{\ell\times k}$.
– $\mathsf{Commit}(\mathsf{params}, \boldsymbol{m}; \boldsymbol{r})$: In order to commit to a message $\boldsymbol{m} \in R_{q_2}^\ell$, the algorithm first samples a random short vector $\boldsymbol{r} \xleftarrow{\$} S_\beta^k$, and then outputs $\mathsf{comm} := \begin{bmatrix}\boldsymbol{t}_1\\ \boldsymbol{t}_2\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}\boldsymbol{r} + \begin{bmatrix}\mathbf{0}\\ \boldsymbol{m}\end{bmatrix}$.
– $\mathsf{Open}(\mathsf{params}, \mathsf{comm})$: For $\mathsf{comm} := (\boldsymbol{t}_1^\top, \boldsymbol{t}_2^\top)^\top \in R_{q_1}^n \times R_{q_2}^\ell$, there are two types of openings for slightly different commitment relations in the literature: relaxed one and exact one. Here we just focus on the latter one.
Particularly, The valid opening is with respect to the following exact relation

$$\hat{L} =: \Big\{\mathsf{comm} : \exists(\boldsymbol{m}, \boldsymbol{r}) \text{ such that } \mathsf{comm} = \mathsf{Commit}(\mathsf{params}, \boldsymbol{m}, \boldsymbol{r})\Big\}.$$

A valid opening of $\mathsf{comm} := (\boldsymbol{t}_1^\top, \boldsymbol{t}_2^\top)^\top \in R_{q_1}^n \times R_{q_2}^\ell$ consists of a message $\boldsymbol{m} \in R_{q_2}^\ell$, and a short vector $\boldsymbol{r} = (r_1, \ldots, r_k)^\top \in R^k$, such that $\begin{bmatrix}\boldsymbol{t}_1\\ \boldsymbol{t}_2\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}\boldsymbol{r} + \begin{bmatrix}\mathbf{0}\\ \boldsymbol{m}\end{bmatrix}$, where for all i, $\|r_i\|_\infty \le \beta$.

Besides, there are two additional algorithms for the randomness vector in the valid commitment.

– $\mathsf{Combine}(\boldsymbol{r}, \boldsymbol{r}')$: Given two vectors $\boldsymbol{r} \in S_\beta^k$ and $\boldsymbol{r}' \in S_\beta^k$, output $\hat{\boldsymbol{r}} = \boldsymbol{r} + \boldsymbol{r}' \in S_{2\beta}^k$.
– $\mathsf{Randomize}(\mathsf{params}, \mathsf{comm}, \boldsymbol{r}')$: Taking as input params, $\boldsymbol{r}' \in S_\beta^k$, and a commitment comm, output $\mathsf{comm}' = \mathsf{comm} + \mathbf{A}_0 \cdot \boldsymbol{r}'$.[9]

According to [6,23], we know that BDLOP Commitment satisfies binding and hiding properties, following from $\mathsf{M\text{-}SIS}_{q_1,n,k,8\sqrt{2}\cdot\eta\cdot\kappa\cdot\beta\cdot k\cdot N}$ and $\mathsf{M\text{-}LWE}_{q_2,k-n-\ell,n+\ell}$, respectively. Here, η is the parameter for rejection sampling as in Lemma A.9, κ is the parameter for the challenge set of NIZKPoK system as in Table 5.

Well-formness. For our application, we need to prove the well-formness of BDLOP commitments along with the commitment generation. This task has been studied in the original BDLOP scheme and several follow up works, e.g., [6,23]. Particularly, given the public matrices $\mathbf{A}_1, \mathbf{A}_2$ and commitment $\mathsf{comm} := (\boldsymbol{t}_1^\top, \boldsymbol{t}_2^\top)^\top$, the relation can be described as: there exists vector $\boldsymbol{r}, \boldsymbol{m}$ such that

$$\begin{bmatrix}\mathbf{A}_1 & \mathbf{0}\\ \mathbf{A}_2 & \mathbf{I}\end{bmatrix} \cdot \begin{bmatrix}\boldsymbol{r}\\ \boldsymbol{m}\end{bmatrix} = \begin{bmatrix}\boldsymbol{t}_1\\ \boldsymbol{t}_2\end{bmatrix},$$

[9] Notice that, if comm is a valid commitment of m with randomness $\boldsymbol{r}$, then comm' is still a valid commitment of m, but with randomness $\hat{\boldsymbol{r}} = \mathsf{Combine}(\boldsymbol{r}, \boldsymbol{r}')$.

where $\mathbf{0} \in R_{q_1}^{n \times \ell}$ and $\mathbf{I} \in R_{q_2}^{\ell \times \ell}$ denote the zero and identity matrices.

For the original BDLOP scheme where the message space is $R_{q_2}^{\ell}$, i.e., $\boldsymbol{m}$ can be any element in $R_{q_2}^{\ell}$, well-formness can be proved by showing that there exists a $\boldsymbol{r}$ such that $\mathbf{A}_1 \cdot \boldsymbol{r} = f \cdot \boldsymbol{t}_1$, i.e., with respect to the relaxed relation

$$L_{\gamma_1', q_1, q_2, \bar{\mathcal{C}}} := \Big\{ (\mathbf{A}_1, \mathbf{A}_2, \boldsymbol{t}_1, \boldsymbol{t}_2) : \exists\ (\boldsymbol{r}, \boldsymbol{m}) \text{ and } f \in \bar{\mathcal{C}} \text{ such that } 0 < \|\boldsymbol{r}\| \leq \gamma_1', \text{ and } \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \cdot \boldsymbol{r} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{m} \end{bmatrix} = f \cdot \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix} \Big\}.$$

In fact, this is what the "proof of opening" does in several prior work [6, 23].

However, our application requires a stronger form of well-formness, which is not implied by what we just described above. Particularly, our application needs a stronger commitment-proof binding property that for any $(\boldsymbol{t}_1, \boldsymbol{t}_2)$ that can be proved to be well-formed, it is computationally infeasible to find another $\boldsymbol{t}_1'$ such that one can prove well-formness for $(\boldsymbol{t}_1', \boldsymbol{t}_2)$. This is an important requirement that prevents the mix-and-match attacks for our anonymous credential systems. We formalize this in Appendix A.5.

Next we argue that the original BDLOP does not satisfy this property by the following example. One first generates $\mathsf{Commit}(\boldsymbol{m}) = (\boldsymbol{t}_1, \boldsymbol{t}_2)$ honestly for an arbitrary $\boldsymbol{m}$, and then computes $\boldsymbol{t}_1' = \mathbf{A}_1 \cdot \boldsymbol{r}'$, with $\boldsymbol{r} \neq \boldsymbol{r}'$. Then we can interpret $(\boldsymbol{t}_1', \boldsymbol{t}_2)$ as $\mathsf{Commit}(\boldsymbol{m}' = \boldsymbol{t}_2 - \mathbf{A}_2 \cdot \boldsymbol{r}')$. As the message space is the full ring vector $R_{q_2}^{\ell}$, this interpretation is valid, and thus $(\boldsymbol{t}_1', \boldsymbol{t}_2)$ can still be considered to be well-formed. Thus, it is easy to generate two proofs for these two commitments, breaking the commitment-proof binding property.

To tackle this, we identify a simple property – as long as the BDLOP message space is "short", i.e., $\|\boldsymbol{m}\|_\infty \leq \beta$ for some parameter β, then the stronger form of well-formness is implied naturally! Intuitively, if the adversary can come up with such a tuple $(\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_1')$, then there is a reduction that breaks the M-SIS problem (with proper parameters). In Appendix A.5, we present more details.

Now, we consider the following language for the relaxed relation of BDLOP:

$$L_{\gamma_1', \gamma_2', q_1, q_2, \bar{\mathcal{C}}} := \Big\{ (\mathbf{A}_1, \mathbf{A}_2, \boldsymbol{t}_1, \boldsymbol{t}_2) : \exists\ (\boldsymbol{r}, \boldsymbol{m}) \text{ and } f \in \bar{\mathcal{C}} \text{ such that } 0 < \|\boldsymbol{r}\| \leq \gamma_1', \ 0 < \|\boldsymbol{m}\| \leq \gamma_2', \text{ and } \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{m} \end{bmatrix} = f \cdot \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix} \Big\}.$$

By using the technique of [6, 23], we can construct a NIZKPoK (with rewinding-type extractions) as following. For the BDLOP scheme with this type of message space, the stronger well-formness can be achieved.

Theorem 2.3 *In the random oracle model, for a secure* BDLOP *commitment scheme, there exists a* NIZKPoK *system Π for the relaxed language $L_{\gamma_1', \gamma_2', q_1, q_2, \bar{\mathcal{C}}}$, with $\gamma_1' = 2\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot \beta \cdot kN$ and $\gamma_2' = 2\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot \beta \cdot \ell N$, where η is the parameter for rejection sampling as in Lemma A.9.*

For completeness, we present the proof and concrete protocol in Appendix A.5.

2.4 Non-interactive Zero-knowledge Proof

Let's recall the notion of non-interactive zero-knowledge (NIZK) proof system.

Definition 2.4 ([24]) *Let $\mathfrak{R}$ be a relation. A non-interactive proof system for $\mathfrak{R}$ is a tuple of PPT algorithms* (Setup, Prove, Verify, SimSetup) *having the following interfaces (where 1^λ are implicit inputs to* Prove, Verify, SimSetup*):*

- $\mathsf{Setup}(1^\lambda)$: *given a security parameter λ, outputs a string* crs.
- $\mathsf{Prove}(\mathsf{crs}, x, w)$*: given a string* crs *and a statement-witness pair $(x, w) \in \mathfrak{R}$, outputs a proof π.*
- $\mathsf{Verify}(\mathsf{crs}, x, \pi)$*: given a string* crs*, a statement x, and a proof π, either accepts or rejects.*
- $\mathsf{SimSetup}(1^\lambda)$*: given a security parameter λ, outputs a simulated string* $\widehat{\mathsf{crs}}$ *and a trapdoor* tk.

A secure NIZK should have three properties: Completeness, Soundness, and Zero-knowledge. Due to space limitation, we defer the definitions in Appendix A.6. As argued by [5, 22, 27, 30], Fiat-Shamir based proof systems in the random oracle model satisfy these properties. Many recent lattice-based efficient NIZKs are Fiat-Shamir based, so they also enjoy this property. Notice that, even crs is explicitly outputted by the algorithm Setup, the above definition still cover the case of Random Oracle based NIZK, just as used in [2, 12, 22, 36]

2.5 Algebraic Structure of Cyclotomic Rings

In this paper, we need to prove that the elements in the message space $\mathcal{M}$ of CTS are invertible, i.e., $\mathcal{M}$ is a subfield. In this section, we just present the core corollary for our constructions, based on the assumption that the readers are familiar to necessary algebraic background, which are deferred to Section A.4, due to space limit.

Particularly, for power-of-two cyclotomic rings, we have the following.

Corollary 1 ([23]). *Let $d \geq 1$ be powers of 2, q be a prime that is congruent to 3 or 5 modulo 8. Let K denote the power-of-two cyclotomic field $\mathbb{Q}[X]/\langle X^d + 1\rangle$ with the ring of integers $R = \mathbb{Z}[X]/\langle X^d + 1\rangle$. For any integer k satisfying $k|d$, there exists a subring $\mathcal{S}$ of R, such that $\mathcal{S}_q$ is a subfield of K consisting of q^k elements.*

Particularly, let σ_{-1}, σ_5 are automorphisms of K, which map any $X \in K$ to X^{-1} and X^5, respectively. Let $G = Gal(K/\mathbb{Q})$ denote the Galois group of K, which consists of all automorphisms of K. Let $H = \langle \sigma_{-1}, \sigma_5^k \rangle$ be a subgroup of G with index k. And let L denote the fixed field of H, where its ring of integers is denoted as $\mathcal{S}$. Then, $\mathcal{S}_q$ is a subfield of K consisting of q^k elements. Moreover, for any μ in R_q, $\mu \in \mathcal{S}_q$ iff μ is fixed by σ_{-1} and σ_5^k.

3 Commit-Transferable Signatures

Following prior work [7], our goal is to obtain a signature scheme that can be combined with an appropriate commitment scheme and zero-knowledge proof-of-knowledge protocols to obtain an Anonymous Credential scheme.

We will first describe the key *novel* building block we need: a signature scheme whose message space consists of commitments. Our starting point is a non-interactive commitment algorithm Commit parameterized by params chosen according to the Setup algorithm, i.e., $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$. The commitment scheme should admit additional algorithms that allow for randomizing commitments. Particularly, given a commitment $\mathsf{comm} = \mathsf{Commit}(\mathsf{params}, m; \mathsf{Rand})$ and randomness Rand', there is an algorithm $\mathsf{Randomize}$ that outputs another commitment $\mathsf{comm}' = \mathsf{Commit}(\mathsf{params}, m; \mathsf{Rand}'')$ to the same message m. An additional $\mathsf{Combine}$ operation is for combining Rand' with the randomness Rand of the commitment comm, i.e., $\mathsf{Rand}'' = \mathsf{Combine}(\mathsf{Rand}, \mathsf{Rand}')$.

The novel property of a *commit-transferable* signature is that, given a signature σ on a commitment $\mathsf{comm} = \mathsf{Commit}(\mathsf{params}, m; \mathsf{Rand})$, it is possible to obtain a signature σ' on a different commitment to the same message, $\mathsf{comm}' = \mathsf{Commit}(\mathsf{params}, m; \mathsf{Combine}(\mathsf{Rand}, \mathsf{Rand}'))$. The unforgeability property is defined that an adversary querying for signatures on commitments whose openings are known $m_1, \ldots, m_n$ will not be able to produce a signature on a commitment that opens to a new message $m' \neq m_i$, for $\forall\, i \in [n]$. We notice that the requirement of commitments whose openings are known can be achieved by requiring the adversary to provide an additional (non-interactive) zero-knowledge proof of knowledge in the applications, and thus our simpler form of unforgeability for CTS suffices. As discussed in the introduction, our applications need an additional property called straight-line extraction for the $\mathsf{NIZKPoK}$. We discuss more details in Remark 3.5 and Section 5.

More formally: let $(\mathsf{Setup}, \mathsf{Commit})$ be a non-interactive randomizable commitment scheme that admits $(\mathsf{Randomize}, \mathsf{Combine})$ for randomizing commitments; let $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be a signature scheme, and let $\mathsf{Transfer}$ be an additional algorithm with the following input-output behavior:

Setup Let λ be the security parameter. $\mathsf{Setup}(1^\lambda)$ outputs params, the parameters for the commitment scheme and the signature scheme; these parameters also define the message space $\mathcal{M}$, randomness space $\mathcal{R}$ for the commitment scheme, the randomness space $\mathcal{R}'$ for the $\mathsf{Randomize}$ algorithm, and the output space $\mathcal{R}''$ of the $\mathsf{Combine}$ algorithm.

Commit Let $m \in \mathcal{M}$, $\mathsf{Rand} \in \mathcal{R}$. $\mathsf{Commit}(\mathsf{params}, m; \mathsf{Rand})$ outputs comm, a commitment to m using randomness Rand. There is no separate opening algorithm: opening can be achieved by revealing m and Rand.

$\mathsf{Randomize}$ **and** $\mathsf{Combine}$ Let $\mathsf{comm} = \mathsf{Commit}(\mathsf{params}, m; \mathsf{Rand})$, with $\mathsf{Rand} \in \mathcal{R}$. $\mathsf{Randomize}(\mathsf{params}, \mathsf{comm}, \mathsf{Rand}, \mathsf{Rand}')$ returns the commitment $\mathsf{comm}' = \mathsf{Commit}(\mathsf{params}, m; \mathsf{Combine}(\mathsf{Rand}, \mathsf{Rand}'))$, where $\mathsf{Combine} : \mathcal{R} \times \mathcal{R}' \mapsto \mathcal{R}''$ is an efficiently computable operation on elements of $\mathcal{R}$ and $\mathcal{R}'$.

KeyGen Given params, $\mathsf{KeyGen}(\mathsf{params})$ outputs a secret key sk and the corresponding public key pk for commit-transferrable signature system.

Sign Let $\mathsf{comm} = \mathsf{Commit}(\mathsf{params}, m; \mathsf{Rand})$. $\mathsf{Sign}(\mathsf{params}, \mathsf{pk}, \mathsf{sk}, \mathsf{comm})$ outputs a signature σ with respect to comm.

Transfer Let $\mathsf{comm} = \mathsf{Commit}(\mathsf{params}, m; \mathsf{Rand})$, $\mathsf{comm}' = \mathsf{Randomize}(\mathsf{params}, \mathsf{comm}, \mathsf{Rand}')$, $\sigma = \mathsf{Sign}(\mathsf{params}, \mathsf{pk}, \mathsf{sk}, \mathsf{comm})$. On input $(\mathsf{params}, \mathsf{pk}, \sigma, m, (\mathsf{Rand}, \mathsf{Rand}'))$, the algorithm Transfer outputs a signature σ' with respect to the randomized commitment comm'.

Verify On input $(\mathsf{params}, \mathsf{pk}, \mathsf{comm}, \sigma)$, the algorithm Verify either accepts or rejects. For simplicity, our syntax does not distinguish whether the signature σ is an original or a transferred one. In the construction, we need to specify two different procedures when verifying different types of the signatures.

Definition 3.1 (Correctness) *Let* Setup, Commit, Randomize, Combine, KeyGen, Sign, Verify *and* Transfer *be efficient algorithms with input-output behavior as above. They define a* correct *randomizable commitment scheme if for all* params *that are output by* Setup, *for all* $m \in \mathcal{M}$, $\mathsf{Rand} \in \mathcal{R}$, $\mathsf{Rand}' \in \mathcal{R}'$, $\mathsf{Randomize}(\mathsf{Commit}(\mathsf{params}, m; \mathsf{Rand}), \mathsf{Rand}') = \mathsf{Commit}(\mathsf{params}, m; \mathsf{Combine}(\mathsf{Rand}, \mathsf{Rand}'))$.

Moreover, they define a correct *commit-transferable signature scheme if for all* params *that are output by* Setup, *for all* $m \in \mathcal{M}$, $\mathsf{Rand} \in \mathcal{R}$, $\mathsf{Rand}' \in \mathcal{R}'$, $\sigma \leftarrow \mathsf{Sign}(\mathsf{params}, \mathsf{pk}, \mathsf{sk}, \mathsf{Commit}(\mathsf{params}, m, \mathsf{Rand}))$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\mathsf{params})$, $\sigma' \leftarrow \mathsf{Transfer}(\mathsf{params}, \mathsf{pk}, \sigma, m, (\mathsf{Rand}, \mathsf{Rand}'))$, *both* $\mathsf{Verify}(\mathsf{params}, \mathsf{pk}, \mathsf{Commit}(\mathsf{params}, m, \mathsf{Rand}), \sigma)$ *and* $\mathsf{Verify}(\mathsf{params}, \mathsf{pk}, \mathsf{Commit}(\mathsf{params}, m, \mathsf{Combine}(\mathsf{Rand}, \mathsf{Rand}')), \sigma')$ *accept.*

Additionally, we require that commit-transferrable signature schemes satisfy several properties: unlinkablity/simulatability and unforgeability. Intuitively, unlinkablity means that for any two messages m_0, m_1, it is infeasible to distinguish their honest transferred signatures σ'_0 and σ'_1 (output by the algorithm Transfer). Simulatability means that the transferred signature σ' itself does not leak information about the input x (and also the randomness). Clearly, simulatability is much stronger property, and implies unlinkability. Thus, it is sufficient for us to just focus on simulatability.

Below we formulate the property of simulatability by the zero-knowledge paradigm, requiring that a simulator without knowing the input x and randomness can generate an indistinguishable σ' for an arbitrary number of queries.

Definition 3.2 (Simulatability) *We say that the* Transfer *algorithm can be* simulatable *if there exists a two-stage probabilistic polynomial time simulator* $\mathcal{S}$ *which can simulate the transfer algorithm in an indistinguishable way, without knowing the input* x *and randomness to the commitment* comm. *More formally, we define the syntax of the two-stage simulation process as follow.*

- *First,* $\mathcal{S}$ *generates* params, *together with some trapdoor information* Trap.
- *Second,* $\mathcal{S}$ *is given input* params *with the trapdoor* Trap, *and any arbitrary* pk, comm. *Then* $\mathcal{S}$ *can generate a simulated transferred signature* $\widetilde{\sigma}'$.

Then the simulatability requires that for $t = \mathsf{poly}(\lambda)$, *any* $\{m_i\}_{i\in[t]} \in \mathcal{M}$, *randomness* $\{\mathsf{Rand}_i, \mathsf{Rand}'_i\}_{i\in[t]}$, *no probabilistic polynomial time distinguisher* $\mathcal{D}$ *can distinguish* $(\mathsf{params}, \mathsf{pk}, \{\mathsf{comm}'_i\}_{i\in[t]}, \{\sigma'_i\}_{i\in[t]})$ *from* $(\mathsf{params}, \mathsf{pk}, \{\mathsf{comm}'_i\}_{i\in[t]}, \{\widetilde{\sigma}'_i\}_{i\in[t]})$ *with better than a negligible advantage, where*

– *in the former, the* params *and* pk *are sampled honestly, each* $\mathsf{comm}_i = \mathsf{Commit}(\mathsf{params}, m_i; \mathsf{Rand}_i)$, $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{params}, \mathsf{pk}, \mathsf{sk}, \mathsf{comm}_i)$, $\mathsf{comm}'_i = \mathsf{Randomize}(\mathsf{params}, \mathsf{comm}_i, \mathsf{Rand}'_i)$, *and* $\sigma'_i \leftarrow \mathsf{Transfer}(\mathsf{params}, \mathsf{pk}, \sigma_i, m_i, (\mathsf{Rand}_i, \mathsf{Rand}'_i))$*;*
– *in the latter,* params *is generated by the simulator,* pk *is sampled honestly,* comm'_i *is generated as above, and* $\widetilde{\sigma}'_i$ *is generated by the simulator.*

Definition 3.3 (Unforgeability for Commitment Relation) *We say that the algorithms as above define an* unforgeable *commit-transferable signature if for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, the probability that* $\mathcal{A}$ *wins the following game is negligible:*

Input generation phase: *On input* 1^λ*, the challenger generates* $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\mathsf{params})$.
Query phase: *Given* $(\mathsf{params}, \mathsf{pk})$ *as input, the adversary* $\mathcal{A}$ *can access to the following oracle:* $\mathcal{A}$ *makes queries with the form of* $(\mathsf{comm}_i, m_i, \mathsf{Rand}_i)$*, and gets as responses* $\sigma_i = \mathsf{Sign}(\mathsf{params}, \mathsf{sk}, \mathsf{comm}_i)$ *if* $\mathsf{comm}_i = \mathsf{Commit}(\mathsf{params}, m_i; \mathsf{Rand}_i)$*, or* $\perp$ *else.*
Challenge phase: *Finally, the adversary* $\mathcal{A}$ *outputs* $(m^*, \mathsf{Rand}, \sigma)$*. Let* $\mathsf{comm}^* = \mathsf{Commit}(\mathsf{params}, m^*; \mathsf{Rand})$*, and* $\mathcal{A}$ *wins the game if* $\mathsf{Verify}(\mathsf{params}, \mathsf{pk}, \mathsf{comm}^*, \sigma)$ *accepts, and* m^* *has never been queried in the query phase.*

The scheme is selectively secure if the adversary needs to commit to the challenge message m^* *before the input generation phase, and is adaptively secure if this condition is not required.*

Remark 3.4 *Our notion of unforgeability requires the adversary to make queries of the form* $(\mathsf{comm}, m, \mathsf{Rand})$ *such that* $\mathsf{comm} = \mathsf{Commit}(\mathsf{params}, m, \mathsf{Rand})$*. In practical applications such as anonymous credentials and blind signature, this form can be enforced by requiring the adversary to provide a zero-knowledge proof of knowledge* π*, i.e., knowing a witness* (m, Rand) *such that* $\mathsf{comm} = \mathsf{Commit}(\mathsf{params}, m, \mathsf{Rand})$*. In this way, an adversary who makes queries of* (comm, π) *can be made equivalent to an adversary who makes queries of* $(\mathsf{comm}, m, \mathsf{Rand})$*.*

Remark 3.5 *As pointed out by [22], there is a subtlety about proving knowledge of the commitment in the applications to anonymous credentials and blind signatures – the knowledge extraction needs to be straight-line, as the rewinding extraction would incur an exponential security loss (in the number of queries). In Section 5, we show how to instantiate a competitively efficient straight-line extractable* NIZKPoK *required by our anonymous credential construction.*

Remark 3.6 *A weaker notion of selective security can be considered where in the above unforgeability game, the adversary needs to commit to both* (m^*, Rand) *before the input generation phase. However, this weaker notion suffers from a drawback – the upgrade to the adaptive security via the complexity leveraging would incur* $|m^*|+|\mathsf{Rand}|$ *bits of security loss, whereas the above stronger selective notion only incurs* $|m^*|$ *bits security loss. As our construction (in Section 4) can directly achieve the stronger notion as Definition 3.3, we do not consider this weaker variant in this work.*

Finally the overall security of the CTS can be defined as follow.

Definition 3.7 (Secure commit-transferable signature) *The algorithms* Setup, Commit, KeyGen, Sign, Verify *and* Transfer *constitute a secure commit-transferable signature scheme if they constitute correct, simulatable and unforgeable (for exact commitment relation) commit-secure signature scheme, i.e. satisfy Definitions 3.1, 3.2, 3.3; and the commitment scheme* (Setup, Commit) *is hiding and binding, satisfying Definitions A.10, A.11.*

4 Efficient Construction for CTS

In this section, we first present a lattice-based commit-transferrable signature scheme, and then show that it satisfies the properties of correctness, simulatability, and unforgeability as defined in Section 3. Our construction uses the following building blocks: (1) the BDLOP commitment scheme $\Gamma = \Gamma.\{\mathsf{CKeyGen}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Combine}, \mathsf{Randomize}\}$, and (2) a NIZKPoK system $\Pi = \Pi.\{\mathsf{Setup}, \mathsf{Prove}, \mathsf{VerifyProve}, \mathsf{SimSetup}\}$ for the following language (parameterized by $\gamma', q \in \mathbb{N}$)[10]

$$L_{\gamma',q,\bar{\mathcal{C}}} = \Big\{(\mathbf{B}, \boldsymbol{u}) \in R_q^{1\times 12} \times R_q : \exists\ \boldsymbol{x} \in R_q^{12} \text{ and } f \in \bar{\mathcal{C}} \text{ such that } 0 < \|\boldsymbol{x}\| \le \gamma' \text{ and } \mathbf{B}\cdot\boldsymbol{x} = f\cdot\boldsymbol{u}\Big\}.$$

4.1 Construction

We first describe the required parameters in Table 5. Notice that in this work, we consider the cyclotomic ring $R = \mathbb{Z}[X]/(X^N + 1)$ with N a power of 2. This type of ring is commonly used in many constructions, as it is easy to analyze to the norm bounds under ring operations, convenient to implement, and has efficient zero-knowledge proof system.

In our Construction 4.1, we directly set $n = 1, \ell = 1, k = 3$ as the row and column parameters of the underlying BDLOP commitment.

Construction 4.1 (Commit-Transferrable Signature) *Our* CTS *is constructed as follow.*

– $\mathsf{Setup}(1^\lambda)$*: On input the security parameter* 1^{λ_1}*, the algorithm does:*

1. *Run* $\Gamma.\mathsf{CKeyGen}$ *to get* $\mathbf{A} := \begin{bmatrix} 1, \boldsymbol{a}_1^\top \\ 0, \boldsymbol{a}_2^\top \end{bmatrix} \leftarrow \Gamma.\mathsf{CKeyGen}(1^\lambda)$*, where* $[1, \boldsymbol{a}_1^\top] \in R_{q_1}^{1\times 3}$ *and* $[0, \boldsymbol{a}_2^\top] \in R_{q_2}^{1\times 3}$*, with* $\boldsymbol{a}_1 \in R_{q_1}^2$ *and* $\boldsymbol{a}_2^\top = (1, a_2')^\top \in R_{q_2}^2$*. Note that the commitment scheme sets message space* $\mathcal{M} \subseteq R_{q_2}$ *with randomness space* $\mathcal{R} = S_1^3$*, where* $\mathcal{M}$ *is a subset of a subfield of the ring* R_{q_2} *consisting of* q_2^τ *elements. And the* ℓ_∞ *norm of all elements in* $\mathcal{M}$ *is set be at most 1.*
2. *Sample a random vector* $\boldsymbol{d} \xleftarrow{\$} R_{q_2}^2$*.*

[10] Under current state of art, such a system Π can be efficiently instantiated from lattice-based assumptions, just as stated in Section 4.2.

Param.	Description
λ	The security parameter
R, N	Cyclotomic Ring for CTS and its dimension
q_1, q_2	Moduli used for BDLOP commitment scheme
$\mathcal{M}, \tau$	Message space, which is a subset of a subfield of R_{q_2} of order q_2^τ
S_β	Set of all elements in R with ℓ_∞ norm at most β
α	Parameter used in $\mathsf{SamplePre}$
η, M	Parameters for rejection sampling algorithm
γ	ℓ_2 norm parameter used in Verify algorithm for original signature
$\mathcal{C}, \kappa$	Challenge set of the $\mathsf{NIZKPoK}$ system Π $\mathcal{C} = \{c \in R : \|c\|_1 = \kappa, \|c\|_\infty = 1\}$
$\bar{\mathcal{C}}$	The set of differences $\mathcal{C} - \mathcal{C}$ except 0
γ'	ℓ_2 norm parameter for "short" vectors in the language of Π
δ_0	Root-Hermite Factor
$\mathsf{Bit\text{-}sec}$	Bit-security in time for our construction

Table 5. Parameters of Commit-Transferrable Signature Scheme

3. *Set parameters* κ, γ, γ'*, and a gaussian parameter* α*;*
4. *Run* $\Pi.\mathsf{Setup}(1^\lambda)$ *to get a common reference string* crs*;*
5. *Output* $\mathsf{params} := (\mathbf{A}, \boldsymbol{d}, q_1, q_2, N, \kappa, \gamma, \gamma', \alpha, \mathcal{M}, \mathcal{R}, \mathsf{crs})$.

– $\mathsf{Commit}(\mathsf{params}, m; \mathsf{Rand})$*: On input* params*,* *message* $m \in \mathcal{M}$*, and randomness* $\mathsf{Rand} \in \mathcal{R}^4$*, the algorithm does the following.*
1. *Parse* Rand *as vectors* $(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4)$*, where* $\boldsymbol{r}_i \in \mathcal{R} = S_1^3$ *for* $i \in [4]$.
2. *Run* $\mathsf{comm}_1 = \Gamma.\mathsf{Commit}(\mathbf{A}, m; \boldsymbol{r}_1)$, $\mathsf{comm}_2 = \Gamma.\mathsf{Commit}(\mathbf{A}, m\delta; \boldsymbol{r}_2)$, $\mathsf{comm}_3 = \Gamma.\mathsf{Commit}(\mathbf{A}, m\delta^2; \boldsymbol{r}_3)$*, and* $\mathsf{comm}_4 = \Gamma.\mathsf{Commit}(\mathbf{A}, m\delta^3; \boldsymbol{r}_4)$*, with* $\delta = q_2^{\frac{1}{4}}$.
3. *Output* $\mathsf{comm} = (\mathsf{comm}_1, \mathsf{comm}_2, \mathsf{comm}_3, \mathsf{comm}_4)$ *as the commitment of* m.

– $\mathsf{Randomize}(\mathsf{params}, \mathsf{comm}, \mathsf{Rand}')$*: On input* params*,* $\mathsf{Rand}' \in \mathcal{R}^4$*, and* $\mathsf{comm} = (\mathsf{comm}_1, \mathsf{comm}_2, \mathsf{comm}_3, \mathsf{comm}_4)$*, the algorithm does the following.*
1. *Parse* Rand' *as vectors* $(\tilde{\boldsymbol{r}}_1, \tilde{\boldsymbol{r}}_2, \tilde{\boldsymbol{r}}_3, \tilde{\boldsymbol{r}}_4)$*, where* $\tilde{\boldsymbol{r}}_i \in \mathcal{R} = S_1^3$ *for* $i \in [4]$.
2. *Run* $\mathsf{comm}'_i = \Gamma.\mathsf{Randomize}(\mathbf{A}, \mathsf{comm}_i, \tilde{\boldsymbol{r}}_i)$ *for* $i \in [4]$.
3. *Output* $\mathsf{comm}' = (\mathsf{comm}'_1, \mathsf{comm}'_2, \mathsf{comm}'_3, \mathsf{comm}'_4)$.[11]

– $\mathsf{Combine}(\mathsf{Rand}, \mathsf{Rand}')$*: Taking as input two randomness* $\mathsf{Rand} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4) \in S_1^{3\times 4}$*, and* $\mathsf{Rand}' = (\tilde{\boldsymbol{r}}_1, \tilde{\boldsymbol{r}}_2, \tilde{\boldsymbol{r}}_3, \tilde{\boldsymbol{r}}_4) \in S_1^{3\times 4}$*, the algorithm computes and outputs* $(\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2, \hat{\boldsymbol{r}}_3, \hat{\boldsymbol{r}}_4) \in S_2^{3\times 4}$*, where* $\hat{\boldsymbol{r}}_i = \boldsymbol{r}_i + \tilde{\boldsymbol{r}}_i$ *for* $i \in [4]$.

– $\mathsf{KeyGen}(\mathsf{params})$*: On input* params*, the algorithm does:*
1. *Sample* $\mathbf{T} \xleftarrow{\$} S_1^{2\times 4}$*, and set* $\boldsymbol{a}^\top = \boldsymbol{d}^\top \cdot \mathbf{T} + \mathbf{G} \in R_{q_2}^{1\times 4}$*, where* $\mathbf{G} = (1, \delta, \delta^2, \delta^3) = (1, q_2^{\frac{1}{4}}, q_2^{\frac{2}{4}}, q_2^{\frac{3}{4}}) \in R_{q_2}^{1\times 4}$.
2. *Sample* $\boldsymbol{b} \xleftarrow{\$} R_{q_2}^4$ *and a non-zero* $u \xleftarrow{\$} R_{q_2}$.
3. *Output* $\mathsf{pk} := (\boldsymbol{a}, \boldsymbol{b}, u)$*, and* $\mathsf{sk} := \mathbf{T}$.

– $\mathsf{Sign}(\mathsf{params}, \mathsf{pk}, \mathsf{sk}, \mathsf{comm}, \pi_1)$*: On input* params*,* pk*,* sk*, and* comm*, the algorithm does the following:*

[11] Notice that, if comm is a valid commitment of m with randomness $(\boldsymbol{r}_i)_{i\in[4]}$, then comm' is still a valid commitment of m, but with randomness $(\hat{\boldsymbol{r}}_i)_{i\in[4]} = (\mathcal{C}.\mathsf{Combine}(\boldsymbol{r}_i, \tilde{\boldsymbol{r}}_i))_{i\in[4]}$.

1. *Parse* $\mathsf{comm} = (\mathsf{comm}_1, \mathsf{comm}_2, \mathsf{comm}_3, \mathsf{comm}_4)$ *as* $\mathsf{comm}_1 = \begin{bmatrix} t_{1,1} \\ t_{2,1} \end{bmatrix}$, $\mathsf{comm}_2 = \begin{bmatrix} t_{1,2} \\ t_{2,2} \end{bmatrix}$, $\mathsf{comm}_3 = \begin{bmatrix} t_{1,3} \\ t_{2,3} \end{bmatrix}$ *and* $\mathsf{comm}_4 = \begin{bmatrix} t_{1,4} \\ t_{2,4} \end{bmatrix}$;
2. *Set* $\mathbf{F}_{\mathsf{comm}} = \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{b}_{\mathsf{comm}}^\top \middle| \boldsymbol{a}_2^\top \right] = \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| [\boldsymbol{b}^\top + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4})] \middle| \boldsymbol{a}_2^\top \right]$, *and sample* $\mathsf{Sig}_{\mathsf{comm}} := \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 \end{bmatrix} \leftarrow \mathsf{SamplePre}(\left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{b}_{\mathsf{comm}}^\top \middle| \boldsymbol{a}_2^\top \right], \mathbf{T}, u, \alpha)$,[12] *and output* $\mathsf{Sig}_{\mathsf{comm}}$ *as the signature of* comm, *where* $\boldsymbol{s}_1 = \begin{bmatrix} \boldsymbol{s}_{1,1} \\ \boldsymbol{s}_{1,2} \end{bmatrix}$, *and* $\boldsymbol{s}_{1,1} \in R^2, \boldsymbol{s}_{1,2} \in R^4, \boldsymbol{s}_2 \in R^4, \boldsymbol{s}_3 \in R^2$.

– $\mathsf{Transfer}(\mathsf{params}, \mathsf{pk}, \mathsf{Sig}_{\mathsf{comm}}, m, (\mathsf{Rand}, \mathsf{Rand}'))$*: On input* $\mathsf{params}, \mathsf{pk}$, *a signature* $\mathsf{Sig}_{\mathsf{comm}}$, *message* m, *randomness* Rand *for generating the commitment* comm *for* m, *the additional randomness* Rand' *for the rerandomization of* comm, *the algorithm does the followings:*

1. *Parse* $\mathsf{Sig}_{\mathsf{comm}}$ *as vector* $\begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 \end{bmatrix}$, *where* $\boldsymbol{s}_1 \in R^6$, $\boldsymbol{s}_2 \in R^4$, $\boldsymbol{s}_3 \in R^2$.
2. *Parse* Rand *as vectors* $(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4)$, *where* $\boldsymbol{r}_i \in \mathcal{R} = S_1^3$.
3. *Parse* Rand' *as vectors* $(\tilde{\boldsymbol{r}}_1, \tilde{\boldsymbol{r}}_2, \tilde{\boldsymbol{r}}_3, \tilde{\boldsymbol{r}}_4)$, *where* $\tilde{\boldsymbol{r}}_i \in \mathcal{R} = S_1^3$.
4. *Run* $\mathsf{Commit}(\mathsf{params}, m; (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4))$ *and obtain:* $\mathsf{comm} = (\mathsf{comm}_i)_{i \in [4]}$, *where* $\mathsf{comm}_i = \begin{bmatrix} t_{1,i} \\ t_{2,i} \end{bmatrix}$.
5. *Run* $\mathsf{Randomize}$ $(\mathsf{params}, \mathsf{comm}, (\tilde{\boldsymbol{r}}_1, \tilde{\boldsymbol{r}}_2, \tilde{\boldsymbol{r}}_3, \tilde{\boldsymbol{r}}_4))$ *and obtain* $\mathsf{comm}' = (\mathsf{comm}'_i)_{i \in [4]}$, *where* $\mathsf{comm}'_i = \begin{bmatrix} \hat{t}_{1,i} \\ \hat{t}_{2,i} \end{bmatrix}$.
6. *Compute a (temporary) signature* $\mathsf{Sig}_{\mathsf{comm}'}$ *as*

$$\mathsf{Sig}_{\mathsf{comm}'} := \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 - \tilde{\mathbf{R}}_2 \cdot \boldsymbol{s}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{1,1} \\ \boldsymbol{s}_{1,2} \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 - \tilde{\mathbf{R}}_2 \cdot \boldsymbol{s}_2 \end{bmatrix} \in R^{12},$$

   where we denote $\tilde{\mathbf{R}} = \begin{bmatrix} \tilde{\mathbf{R}}_1 \\ \tilde{\mathbf{R}}_2 \end{bmatrix} = \left[\tilde{\boldsymbol{r}}_1, \tilde{\boldsymbol{r}}_2, \tilde{\boldsymbol{r}}_3, \tilde{\boldsymbol{r}}_4 \right] \in R^{3 \times 4}$, *with* $\tilde{\mathbf{R}}_1 \in R^{1 \times 4}$ *and* $\tilde{\mathbf{R}}_2 \in R^{2 \times 4}$.
7. *Compute* $\mathbf{F}_{\mathsf{comm}'} := \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{b}_{\mathsf{comm}'}^\top \middle| \boldsymbol{a}_2^\top \right] = \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{b}^\top + (\hat{t}_{2,1}, \hat{t}_{2,2}, \hat{t}_{2,3}, \hat{t}_{2,4}) \middle| \boldsymbol{a}_2^\top \right]$.
8. *Run the prove algorithm and output* $\mathsf{Sig}'_{\mathsf{comm}'} := \pi_2 \leftarrow \Pi_2.\mathsf{Prove}(\mathsf{crs}_2, (\mathbf{F}_{\mathsf{comm}'}, u), \mathsf{Sig}_{\mathsf{comm}'})$, *proving that* $\mathsf{Sig}_{\mathsf{comm}'}$ *is a short* ℓ_2 *norm vector and satisfies* $\mathbf{F}_{\mathsf{comm}'} \cdot \mathsf{Sig}_{\mathsf{comm}'} = u$, *through using the* $\mathsf{NIZKPoK}$ *system* Π *with the relaxed language* $L_{\gamma', q_2, \bar{\mathcal{C}}}$.

– $\mathsf{Verify}(\mathsf{params}, \mathsf{pk}, \mathsf{comm}, \mathsf{Sig})$*: On input* $\mathsf{params}, \mathsf{pk}, \mathsf{comm}, \mathsf{Sig}$, *the algorithm does the following.*

[12] Here, we implicitly use $\mathbf{T}$ as the $\mathbf{G}$-trapdoor of the matrix $[\boldsymbol{d}^\top | \boldsymbol{a}^\top]$, which can be easily extended to get the corresponding $\mathbf{G}$-trapdoor for $\left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{b}_{\mathsf{comm}}^\top \middle| \boldsymbol{a}_2^\top \right]$.

1. *Parse* $\mathsf{comm} = (\mathsf{comm}_1, \mathsf{comm}_2, \mathsf{comm}_3, \mathsf{comm}_4)$ *as* $\mathsf{comm}_1 = \begin{bmatrix} t_{1,1} \\ t_{2,1} \end{bmatrix}$, $\mathsf{comm}_2 = \begin{bmatrix} t_{1,2} \\ t_{2,2} \end{bmatrix}$, $\mathsf{comm}_3 = \begin{bmatrix} t_{1,3} \\ t_{2,3} \end{bmatrix}$, *and* $\mathsf{comm}_4 = \begin{bmatrix} t_{1,4} \\ t_{2,4} \end{bmatrix}$;
2. *Based on the type of* Sig*, the verification works as follow.*
   - *If* Sig *is a non-zero short vector within* ℓ_2 *norm* γ*, then the algorithm does*
     (a) *Set matrix* $\mathbf{F}_{\mathsf{comm}} := \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| [\boldsymbol{b}^\top + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4})] \middle| \boldsymbol{a}_2^\top\right]$.
     (b) *Check whether* Sig *satisfies* $\mathbf{F}_{\mathsf{comm}} \cdot \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 \end{bmatrix} = u \in \mathcal{R}_{q_2}$.
   - *If* Sig *is a proof of the* NIZKPoK *system* Π_2*,*
     (a) *Set matrix* $\mathbf{F}_{\mathsf{comm}} := \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| [\boldsymbol{b}^\top + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4})] \middle| \boldsymbol{a}_2^\top\right]$.
     (b) *Run the verify algorithm (with respect to language* $L_{\gamma', q_2, \bar{\mathcal{C}}}$*)* $\Pi_2.\mathsf{VerifyProve}(\mathsf{crs}, (\mathbf{F}_{\mathsf{comm}}, u), \mathsf{Sig})$ *and output its result.*

Lemma 4.2 (Correctness) *For parameters* $N, q_2, \alpha, \gamma = \alpha\sqrt{2 \cdot 12 \cdot N}$*, the* NIZKPoK *system* Π *for the relaxed language* $L_{\gamma', q_2, \bar{\mathcal{C}}}$ *with* $\gamma' \geq (3.5\alpha N \cdot 2\sqrt{2} + \alpha\sqrt{2 \cdot 12 \cdot N})$*, Construction 4.1 satisfies the correctness property as defined in Definition 3.1.*

The correctness directly follows the correctness of BDLOP commitment, the completeness of the NIZKPoK system Π and our parameter settings. Due to space limitation, we defer the proof in Appendix B.1.

4.2 Instantiation of NIZKPoK system Π in CTS

Before presenting the NIZKPoK system Π, we first specify the concrete language $L_{\gamma', q_2, \bar{\mathcal{C}}}$ in the algorithms Transfer and Verify,

$$L_{\gamma', q_2, \bar{\mathcal{C}}} = \Big\{ (\mathbf{F}_{\mathsf{comm}'}, u) \in R_{q_2}^{1 \times 12} \times R_{q_2} : \exists\, \boldsymbol{x} \in R^{12} \text{ and } f \in \bar{\mathcal{C}} \\ \text{such that } 0 < \|\boldsymbol{x}\| \leq \gamma' \text{ and } \mathbf{F}_{\mathsf{comm}'} \cdot \boldsymbol{x} = f \cdot u \Big\}.$$

Then, according to [6, 23], there exists such an efficient Π for $L_{\gamma', q_2, \bar{\mathcal{C}}}$. The formal theorem is presented as follows.

Theorem 4.3 ([6, 23]) *In the random oracle model, there exists a* NIZKPoK *system* Π *for the relaxed language* $L_{\gamma', q_2, \bar{\mathcal{C}}}$*, with* $\gamma' = 2\sqrt{2 \cdot 12N} \cdot \eta \cdot \sqrt{\kappa} \cdot (3.5\alpha N \cdot 2\sqrt{2} + \alpha\sqrt{2 \cdot 12N})$.

Moreover, assuming a t*-time adversary* $\mathcal{A}$ *forging a proof with probability* ε*, there exists a* $O(t/\varepsilon)$*-time extractor, who can successfully extract the witness* $\boldsymbol{x}$ *and* $c \in \bar{\mathcal{C}}$ *with probability* $\frac{1}{2}$.

Remark 4.4 *Notice that the concrete instantiation of* NIZKPoK *system* Π *in Theorem 4.3 is essentially a Fiat-Shamir signature, which is quite practical.*

4.3 Security of CTS

In this section, we establish the simulatability and unforgeability of the above Construction 4.1.

Lemma 4.5 (Simulatability) *The algorithm* Transfer *in Construction 4.1 is simulatable.*

Proof. (Sketch) We show the simulatability of our construction by first constructing a two-stage PPT simulator $\mathcal{S}$, and then proving that after running any polynomial $t = \mathsf{poly}(\lambda)$ times, the distribution of $\{\widetilde{\mathsf{Sig}}'_{\mathsf{comm}'_i}\}_{i\in[t]}$ output by $\mathcal{S}$ are statistically close to that of $\{\mathsf{Sig}'_{\mathsf{comm}'_i}\}_{i\in[t]}$ output by Transfer. Due to space limitation, we defer the full proof in Appendix B.2. □

Below, we analyse the unforgeability of Construction 4.1. Before this, we first specify the corresponding commitment relation $\hat{L}_{q_1,q_2}$ as follows.

$$\hat{L}_{q_1,q_2} := \Big\{\mathsf{comm} = (\mathsf{comm}_i)_{i\in[4]} : \exists(m, q_1, q_2, \boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4) \text{ such that } \|m\|_\infty \leq 1,$$
$$\boldsymbol{r}_i \in S_1^3 \text{ and } \mathsf{comm}_i = \mathsf{Commit}(\mathsf{params}, m \cdot q_2^{\frac{i-1}{4}}; \boldsymbol{r}_i) \text{ for } i \in [4]\Big\}.$$

Lemma 4.6 (Unforgeability) *Assume that* M-SIS$_{q_2,1,9,\nu}$ *problem and* M-SIS$_{q_2,1,9,\nu'}$ *problem are hard with* $\nu = 22\alpha \cdot N$ *and* $\nu' = \frac{22\gamma'\sqrt{N}}{\sqrt{2\cdot 12}}$*, then our above lattice-based commitment-transferrable signature scheme is partially selectively unforgeable for the commitment relation* $\hat{L}_{q_1,q_2}$*, i.e., the advantage of any* PPT *adversary* $\mathcal{A}$ *against the partially selective unforgeability game of* CTS *is at most*

$$\mathsf{Adv}^{unforge}_{\mathcal{A}}(\lambda) \leq 2\mathsf{Adv}^{\mathsf{RLWE}}_{\mathcal{A}} + \mathsf{Adv}^{unforge^*}_{\mathcal{A}}(\lambda)$$

Due to space limitation, we defer the detailed proof in Appendix B.3.

5 Efficient Straight-Line Extractable NIZKPoK System

In this section, we present a multi-theorem straight-line extractable NIZKPoK system Π to prove the well-formness of commitment comm output by CTS.Commit.

For clarity of presentation, we describe in a modular way: (i) first we present the generic construction of the encrypt-and-proof paradigm, and then (ii) our concrete instantiation. Particularly, for part (i), we first introduce three building blocks in Section 5.1, and then formally present the generic construction in Section 5.2. For part (ii), we describe the instantiations of three building blocks in Section C and the concrete parameters in Section D.

5.1 Building Blocks of the Encrypt-and-prove Generic Paradigm

We present the generic construction, proving the exact commitment relation $\hat{L}_{q_1,q_2}$ (implicitly including the commitment public parameter in the crs):

$$\hat{L}_{q_1,q_2} := \Big\{\mathsf{comm} = (\mathsf{comm}_i)_{i\in[4]} : \exists(m, q_1, q_2, \boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4) \text{ such that } \|m\|_\infty \leq 1,$$
$$\boldsymbol{r}_i \in S_1^3 \text{ and } \mathsf{comm}_i = \mathsf{Commit}(\mathsf{params}, m \cdot q_2^{\frac{i-1}{4}}; \boldsymbol{r}_i) \text{ for } i \in [4]\Big\}.$$

Following the ideas of [2, 24], this can be achieved by encrypting the witness and proving that the message under the ciphertext satisfies the relation. For our specific setting, we need these three building blocks:

(i) a NIZK system $\Pi^{(1)}$ for the relaxed commitment relation, i.e., the language

$$\hat{L}_{\Pi^{(1)}} := \Big\{\mathsf{comm} = (\mathsf{comm}_i)_{i\in[4]} : \exists m, q_1, q_2, \{\boldsymbol{r}_i\}_{i\in[4]}, f \in \bar{\mathcal{C}}, \text{ such that } 0 < \|\boldsymbol{r}_i\| \leq \gamma_1',$$
$$0 < \|m\| \leq \gamma_2', \text{ and } \mathsf{comm}_i = \mathsf{Commit}(\mathsf{params}, m \cdot q_2^{\frac{i-1}{4}}, \boldsymbol{r}_i/f) \text{ for } i \in [4]\Big\},$$

together with the fact that the committed message m is included in the message space, a particular subfield of R_{q_2}. Besides, we use the particular algebraic structure of this subfield to ensure $\|m\|_\infty \leq 1$.

(ii) a CPA-secure encryption scheme E with pseudorandom public-keys, together with the NIZK system $\Pi^{(2)}$ for its validness, i.e., the relaxed language

$$\hat{L}_{\Pi^{(2)}} := \Big\{\mathsf{ct} = (\mathsf{ct}_i)_{i\in[4]} : \exists(\boldsymbol{r}_i, \mathsf{Rand}_i) \text{ and } f \in \bar{\mathcal{C}}, \text{ such that } \boldsymbol{r}_i \in S_1^3,$$
$$0 < \|\mathsf{Rand}_i\| \leq \gamma_3', \text{ and } \mathsf{ct}_i = \mathsf{E.Enc}(\mathsf{pk}, \boldsymbol{r}_i; \mathsf{Rand}_i/f), \text{ for } i \in [4]\Big\},$$

where pk is the public-key of E;

(iii) a NIZK proof system $\Pi^{(3)}$ for the consistency of the witness in $\Pi_1^{(1)}$ and the encrypted message of Enc, i.e., for the relation

$$\hat{L}_{\Pi^{(3)}} = \Big\{(\mathsf{comm} = (\mathsf{comm}_i)_{i\in[4]}, \mathsf{ct} = (\mathsf{ct}_i)_{i\in[4]}) : \exists(m, \boldsymbol{r}_i, \mathsf{Rand}_i) \text{ and } f \in \bar{\mathcal{C}}, \text{ such}$$
$$\text{that } \boldsymbol{r}_i \in S_1^3, 0 < \|\mathsf{Rand}_i\| \leq \gamma_3', \mathsf{comm}_i = \mathsf{Commit}(\mathsf{params}, m \cdot q_2^{\frac{i-1}{4}}; \boldsymbol{r}_i)$$
$$\text{and } \mathsf{ct}_i = \mathsf{E.Enc}(\mathsf{pk}, \boldsymbol{r}_i; \mathsf{Rand}_i/f) \text{ for } i \in [4]\Big\}.$$

Using these building blocks, we can derive a multiple-theorem straight-line extractable $\mathsf{NIZKPoK}$ in a generic way as in Section 5.2. In Section C, we present how to efficiently instantiate all the components. We notice that the idea of the generic construction has appeared in the literature e.g., [2, 24], and thus not surprising. Our merit is to efficiently instantiate a lattice proof for our need.

5.2 General Construction of Straight-line Extractable NIZKPoK

Construction 5.1 (Straight-line extractable NIZKPoK Π) *Given building blocks* $\Pi^{(1)} = \Pi^{(1)}.\{\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{SimSetup}\}$, $\mathsf{E} = \mathsf{E}.\{\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}\}$, $\Pi^{(2)} = \Pi^{(2)}.\{\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{SimSetup}\}$, *and* $\Pi^{(3)} = \Pi^{(3)}.\{\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{SimSetup}\}$, *where the message space of* E *and the randomness space of* $\Pi^{(1)}$ *are compatible, then a straight-line extractable* $\mathsf{NIZKPoK}$ *system* Π *can be builded as follows.*

– $\mathsf{Setup}(1^\lambda)$: *given a security parameter* λ*, the algorithm does:*
  1. *Run* $\Pi^{(1)}.\mathsf{Setup}$*,* $\Pi^{(2)}.\mathsf{Setup}$*, and* $\Pi^{(3)}.\mathsf{Setup}$ *to generate* crs_1*,* crs_2*, and* crs_3*, respectively;*
  2. *Choose a random public key* pk*, which is computational indistinguishable from a real public key output by* $\mathsf{E.KeyGen}$*;*
  3. *Output* $\mathsf{crs} = (\mathsf{crs}_1, \mathsf{crs}_2, \mathsf{crs}_3, \mathsf{pk})$.
– $\mathsf{Prove}(\mathsf{crs}, x, w)$*: given* crs*,* $x := \mathsf{comm}$*, and* $w := (m, \mathsf{Rand})$*, the algorithm conducts the following steps:*
  1. *Parse* crs *as* $(\mathsf{crs}_1, \mathsf{crs}_2, \mathsf{crs}_3, \mathsf{pk})$*;*
  2. *Run* $\Pi^{(1)}.\mathsf{Prove}(\mathsf{crs}_1, \mathsf{comm}, m, \mathsf{Rand})$ *to get* π_1*, which proves the validness of* comm *and* m*;*
  3. *Run* $\mathsf{E.Enc}(\mathsf{pk}, \mathsf{Rand})$ *to get* ct*, which encrypts the randomness* Rand *in* comm *under the random public key* pk*;*
  4. *Run* $\Pi^{(2)}.\mathsf{Prove}(\mathsf{crs}_2, \mathsf{ct}, \mathsf{Rand})$ *to get* π_2*, which proves that the validness of the ciphertext* ct*;*
  5. *Run* $\Pi^{(3)}.\mathsf{Prove}(\mathsf{crs}_3, \mathsf{ct}, \mathsf{comm}, \mathsf{Rand})$ *to get* π_3*, which proves the consistency of the encrypted message in* ct *and the used randomness in the commitment* comm.
  6. *Output* $\pi := (\pi_1, \mathsf{ct}, \pi_2, \pi_3)$.
– $\mathsf{Verify}(\mathsf{crs}, x, \pi)$*: given a string* crs*, a statement* $x := \mathsf{comm}$*, and a proof* π*, either accepts or rejects.*
  1. *Parse* crs *as* $(\mathsf{crs}_1, \mathsf{crs}_2, \mathsf{crs}_3, \mathsf{pk})$*;*
  2. *Parse* π *as* $(\pi_1, \mathsf{ct}, \pi_2, \pi_3)$*;*
  3. *Run* $\Pi^{(1)}.\mathsf{Verify}(\mathsf{crs}_1, \mathsf{comm}, \pi_1)$ *to verify the validness of* comm *and* m*;*
  4. *Run* $\Pi^{(2)}.\mathsf{Verify}(\mathsf{crs}_2, \mathsf{ct}, \pi_2)$ *to verify the validness of* ct*;*
  5. *Run* $\Pi^{(3)}.\mathsf{Verify}(\mathsf{crs}_2, \mathsf{ct}, \mathsf{comm}, \pi_3)$ *to verify the consistency of* ct *and* comm*;*
  6. *Accept* π*, if all the above three proof are verified successfully; Otherwise, reject* π.
– $\mathsf{SimSetup}(1^{\lambda_2})$*: given a security parameter* λ_2*, the algorithm conducts the following steps:*
  1. *Run* $\Pi^{(1)}.\mathsf{SimSetup}$*,* $\Pi^{(2)}.\mathsf{SimSetup}$*, and* $\Pi^{(3)}.\mathsf{SimSetup}$ *to generate* $(\widehat{\mathsf{crs}}_1, \widehat{\mathsf{tk}}_1)$*,* $(\widehat{\mathsf{crs}}_2, \widehat{\mathsf{tk}}_2)$*, and* $(\widehat{\mathsf{crs}}_3, \widehat{\mathsf{tk}}_3)$*, respectively;*
  2. *Run* $\mathsf{E.KeyGen}(1^\lambda)$ *to generate a pair of* $(\widehat{\mathsf{pk}}, \widehat{\mathsf{sk}})$*;*
  3. *Output* $\widehat{\mathsf{crs}} = (\widehat{\mathsf{crs}}_1, \widehat{\mathsf{crs}}_2, \widehat{\mathsf{crs}}_3, \widehat{\mathsf{pk}})$,$\mathsf{tk} = (\widehat{\mathsf{tk}}_1, \widehat{\mathsf{tk}}_2, \widehat{\mathsf{tk}}_3, \widehat{\mathsf{sk}})$.
– $\mathsf{Ext}(\mathsf{crs}, \mathsf{tk}, x, \pi)$*: given a string* crs*, a related trapdoor* tk*, a statement* x*, and a proof* π*, the algorithm conducts the following steps:*
  1. *Parse* tk *as* $(\mathsf{tk}_1, \mathsf{tk}_2, \mathsf{tk}_3, \mathsf{sk})$*;*
  2. *Run* $\mathsf{Verify}(\mathsf{crs}, x, \pi)$ *to verify the valid of* π*. If the verification is accepted, then abort. Otherwise, continue the following steps.*
  3. *Run* $\mathsf{E.Dec}(\mathsf{sk}, \mathsf{ct})$ *to get* Rand.

Then, we have the following theorem.

Theorem 5.2 ([6, 23, 55]) *The* Π *is a multi-theorem straight-line extractable* $\mathsf{NIZKPoK}$ *system for* $\hat{L}_{q_1,q_2}$*, if the underlying commitment* comm *is binding and hiding, the encryption scheme* E *is* CPA*-secure and its public-key is indistinguishable from uniform, and* $\Pi^{(1)}$ *for* $\hat{L}_{\Pi^{(1)}}$*,* $\Pi^{(2)}$ *for* $\hat{L}_{\Pi^{(2)}}$*, and* $\Pi^{(3)}$ *for* $\hat{L}_{\Pi^{(3)}}$ *are three multi-theorem* NIZK*s.*

Proof (Sketch). According to the definition of the multi-theorem straight-line extractable NIZKPoK, we need to prove three properties: completeness, zero-knowledge, and multi-theorem extractability. Generally, the proofs of these properties follow from the related properties of $\Pi^{(1)}$, $\Pi^{(2)}$, $\Pi^{(3)}$, and E, which has been similarly described in lecture note of Dodis in [24] and Agrawal et al. [2]. Here, we omit the details of them for clarity.

Besides, one interesting point in our construction is that, even the used NIZKs $\Pi^{(1)}$ and $\Pi^{(2)}$ are just with respect to the relaxed (commitment/encryption) relation, the resulting NIZKPoK Π is with respect to the exact relation, i.e., the extracted output of $\Pi.\mathsf{Ext}$ are the exact randomness of comm. This is due to the soundness of $\Pi^{(3)}$, the binding of comm, and the correctness of E.

Particularly, suppose there is an adversary $\mathcal{A}$ outputting a valid pair (x, π) (i.e., $\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1$), such that the extracted randomness $\mathsf{Rand}' = \Pi.\mathsf{Ext}(\mathsf{crs}, \mathsf{tk}, x, \pi)$ is different from the original Rand in comm. Then we can induce the contradiction as follows. According to the correctness of E, the encrypted message in ct should be Rand'. Then, if $\mathcal{A}$ can not break the soundness of $\Pi^{(3)}$, then the randomness in comm should be equivalent to Rand'. Furthermore, according to the binding property of comm, it should hold $\mathsf{Rand} = \mathsf{Rand}'$. However, this is contradict to the assumption that Rand and Rand' are different. This means there is no such an adversary $\mathcal{A}$. □

5.3 Efficient Instantiations

In this section, we present the high level ideas for how to instantiate the building blocks efficiently. More details and concrete parameters are described in Sections C and D.

As we discussed in the introduction, the encryption scheme E we use is a trapdoor-version of the BDLOP, whose relaxed proof of well-formness is rather efficient. The detailed scheme is presented in Construction C.2 and the relaxed proof of well-formness is described in Theorem C.3. Then we use the LNP proof for the linear relation, i.e., $\mathbf{A}_1 \cdot \boldsymbol{r} = t_1$ as summarized in Theorem C.4. The overall proof size (for one BDLOP commitment) is roughly 604 KB. More details about the concrete numbers can be found in Section D.

6 Application to Anonymous Credentials

In this section, we present how to construct Anonymous Credentials from CTS and NIZKPoK. Particularly, we first recall the definition and security requirement of the basic Anonymous Credentials in [40], and then describe the construction. Then we describe how to extend the basic scheme into one that supports some attribute settings.

6.1 Definition and Security of Anonymous Credentials

We use the formulation of Anonymous Credentials by Lysyanskaya [40]. A basic credential system has *users*, *organizations*, and *verifiers* as types of players.

Users are entities that receive credentials. Organizations are entities that issue the credentials of the users. Finally, verifiers are entities that verify credentials of the users. Specifically, the system is defined as follows:

- $\mathsf{AC.Setup}$: System parameters params are generated, users generates his secret key usk, and organizations generate their public and secret keys $(\mathsf{pk}_O, \mathsf{sk}_O)$;
- $\mathsf{AC.Registration}$: A user generates a pseudonym nym, and sends it to an organization. The user's private input is usk. the organization does not have any private input.
- $\mathsf{AC.Issue}$: As a result of this protocol, a user obtains a credential from an organization without revealing his private input, just based on his pseudonym nym. The user's private input to the protocol is his usk. The organization's private input is its secret key sk_O. And the user's private output is the credential Cred;
- $\mathsf{AC.Prove}$: The user who is known to one organization O_1 under nym_1, and to a verifier under nym_2, and a credential Cred from O_1, proves to the verifier that he has a credential from O_1. The user's private input to this protocol consists of $(\mathsf{usk}, \mathsf{nym}_1, \mathsf{Cred})$, while the values nym_2 and pk_{O_1} are public;
- $\mathsf{AC.Verify}$: The verifier verifies if the user possesses a credential from O_1 or not.

We follow the security formulation of [7] – an anonymous credential should satisfy unforgeability, anonymity, and unlikability. Intuitively, unforgeability requires that an adversary cannot provide a valid proof of credential Cred^* with respect to a pseudonym nym^* of some usk^* that he has never received a credential from an organization.

Anonymity, informally, requires two different privacy properties: (1) privacy against an organization: the organization cannot distinguish any two different users with two different private inputs in the registration process, and (2) privacy against a verifier: the proof of credential leaks no information other than the validity of owning a credential with respect to the pseudonym.

Unlinkability requires that the adversary cannot distinguish whether (nym_1, π_1) and (nym_2, π_2) are from the same user or not, where π_1, π_2 are two proofs of credentials with respect to nym_1 and nym_2, respectively.

6.2 Anonymous Credentials from CTS

Now we show how to construct an anonymous credential system from a secure CTS and a zero-knowledge proof of knowledge of commitment opening.

Building blocks. Suppose we are given a secure commit-transferable signature scheme $(\mathsf{CTS.Setup}, \mathsf{CTS.Commit}, \mathsf{CTS.Randomize}, \mathsf{CTS.KeyGen}, \mathsf{CTS.Sign}, \mathsf{CTS.Transfer}, \mathsf{CTS.Verify})$ as in Construction 4.1, and an efficient multi-theorem straight-line extractable $\mathsf{NIZKPoK}$ $\Pi = (\mathsf{NIZKSetup}(\mathsf{params}), \mathsf{NIZKProve}, \mathsf{NIZKVerify}, \mathsf{SimSetup})$ for the following commitment relation $\hat{L}$ as Definition 3.3.

$$\hat{L} =: \Big\{ \mathsf{comm} : \exists (m, \mathsf{Rand}) \text{ such that } \mathsf{comm} = \mathsf{Commit}(\mathsf{params}, x, \mathsf{Rand}) \Big\}.$$

Then we can construct an anonymous credential system as follows:

Construction 6.1 (Anonymous Credential) *The anonymous credential scheme can be constructed in the following way.*

- AC.Setup*: System runs* CTS.Setup *to obtain* CTS.params*, and runs* NIZKSetup(params) *to obtain* NIZKpara*. An honest user U generates her secret key* usk *by sampling $\mathcal{M}_{\mathsf{params}}$. An honest organization O generates its keys as follows:* $(\mathsf{sk}_O, \mathsf{pk}_O) \leftarrow \mathsf{CTS.KeyGen}(\mathsf{params})$*;*
- AC.Registration*: The user U first samples* $\mathsf{Rand} \leftarrow \mathcal{R}_{\mathsf{params}}$ *and generates a commitment* $\mathsf{comm} = \mathsf{CTS.Commit}(\mathsf{params}, \mathsf{usk}, \mathsf{Rand})$*. Then U generates an* NIZK *proof π by running* $\mathsf{NIZKProve}(\mathsf{params}, \mathsf{comm}, \mathsf{usk}, \mathsf{Rand})$*. Furthermore, U sends* $\mathsf{nym} = (\mathsf{comm}, \pi)$ *as the pseudonym to the organization O. Finally, O would run* NIZKVerify *to check whether the pseudonym (commitment) is properly formed;*
- AC.Issue*: Suppose that a user U is known to organization O under pseudonym* $\mathsf{nym} = (\mathsf{comm}, \pi)$*. O computes* $\sigma \leftarrow \mathsf{CTS.Sign}(\mathsf{params}, \mathsf{sk}_O, \mathsf{comm})$ *and gives* $\mathsf{Cred} := \sigma$ *to U;*
- AC.Prove*: User U samples* Rand'*, and runs* $\mathsf{comm}' \leftarrow \mathsf{CTS.Randomize}(\mathsf{comm}, \mathsf{Rand}, \mathsf{Rand}')$*. Then she computes* $\sigma' = \mathsf{Transfer}(\mathsf{params}, \mathsf{pk}_{O_1}, \mathsf{usk}, \mathsf{Rand}, \mathsf{Rand}', \sigma)$*, which (by correctness of the* CTS*) is a signature under* pk_{O_1} *on the commitment* comm'*. Next, she gives the verifier the values σ' and* $\mathsf{nym}' = (\mathsf{comm}', \pi')$*, where π' is an NIZK proof that* comm' *is properly formed as well;*
- AC.Verify*: The verifier runs* $\mathsf{Verify}(\mathsf{params}, \mathsf{pk}_{O_1}, \mathsf{comm}', \sigma')$ *and the NIZK verifier of π' on input* $(\sigma', \mathsf{nym}' = (\mathsf{comm}', \pi'))$ *to verify U's credential on the new pseudonym* nym'*.*

Security of the anonymous credential system follows from the security of CTS and Π and NIZKPoK.

Theorem 6.2 *Assuming that* CTS *is secure for the exact commit relation, and Π is a secure multi-theorem straight-line extractable* NIZKPoK *system for $\hat{L}$, Construction 4.1 is a secure anonymous credential system.*

Proof. (Sktech) Intuitively, the anonymity against the organization follows from the security of NIZKPoK and hiding of the commitment scheme, and that against the verifier follows from the simulatability of the CTS, as the transferred signature does not leak information beyond the validity. The unlinkability follows by the hiding property of the re-randomized commitments and the simulatability of the CTS, so that any user cannot relate two pairs of pseudonym-proofs.

To prove unforgeability, we rely on the NIZKPoK extractor (of the commitment relation) and the unforgeability of CTS. Assuming that there exists an adversary $\mathcal{A}$ that forges a valid proof of the anonymous credential, then we can construct a reduction $\mathcal{B}$ that breaks CTS unforgeability in the following way. $\mathcal{B}$ first simulates the NIZKPoK and extracts $\mathcal{A}$'s (m, Rand) in the commitment of the registration queries, from the ZKPoK proof he provides. Then when $\mathcal{A}$ makes an issue query, $\mathcal{B}$ makes a signing query to the CTS challenger. As $\mathcal{B}$ has extracted the witness from the commitment, $\mathcal{B}$ can make a valid CTS signing query. It is easy to verify that as long as $\mathcal{A}$ can forge a valid proof, $\mathcal{B}$ can break

the CTS unforgeability. We note that if the NIZKPoK is with respect to the exact commitment relation, then $\mathcal{B}$ breaks CTS unforgeability with respect to the exact relation. If the proof system is with respect to the relaxed commitment relation, then $\mathcal{B}$ breaks CTS unforgeability with respect to the relaxed relation. □

6.3 Extension to Attribute-based Settings

In the above basic anonymous credential system, the user's secret value usk is her id or some secret key. In a more general setting of attribute-based credentials, the user's secret value can include additional attributes, denoted as $\mathbf{att} = (\mathsf{att}_1, \ldots, \mathsf{att}_\ell)$ where each att_i is some small integer (or a short bit string). The user might wish to reveal some subset of the attributes to any party, e.g., an organization or a verifier, while keep the other attributes and the secret key/id private. This property of *chosen disclosure of attributes* has been identified useful in the literature [12, 18, 32, 36]. We observe that our system can easily be extended to support such an extension. Below we elaborate.

In the basic scheme, the user sets the message as the secret value, i.e., $m = \mathsf{usk}$, and generates a BDLOP commitment $\mathsf{comm} = \mathsf{Commit}(m)$ for the CTS as a pseudonym. Then the user proves well-formness of the commitment and then the organizations would sign on the commitment. To generalize to the attribute setting, we can use m to encode usk and the attributes $\mathbf{att}$, simultaneously. For example, for $m = \sum_{i=0}^{N-1} m_i X^i \in \mathbb{Z}_{q_2}[X]/\langle X^N + 1\rangle$, we can use the coefficients to encode $(\mathsf{usk}\|\mathbf{att})$ (for simplicity we assume N to be the bit-length of $(\mathsf{usk}\|\mathbf{att})$, i.e., $N = |\mathsf{usk}| + |\mathbf{att}|$). Then the user generates $\mathsf{comm} = \mathsf{Commit}(m)$ as before, yet with m under such an encoding.

To disclose some subset of attribute, say $\mathbf{att}_\mathcal{I} = \{\mathsf{att}_i\}_{i\in\mathcal{I}}$ for $\mathcal{I} \subseteq [N]$, the user can prove well-formness of the commitment and additionally that the coefficients of m corresponding to these attributes are consistent with $\mathbf{att}_\mathcal{I}$. To achieve this, we observe that it suffices to use the following protocol $\Pi_{\mathsf{Disclosure}}$ in Table 6, which proves well-formness of a BDLOP commitment $\mathsf{Commit}(m)$ and as well consistency that a certain subset of coefficients in m are the same as those were disclosed. To achieve this, we present the following interactive protocol adapted from the ENS and LNP proof [27, 44], which can be made non-interactive easily using the Fiat-Shamir Transform.

We notice that the above approach supports the case when we can embed the usk and the attribute into one single ring element. A noticeable advantage is that the proof size is essentially independent of the cardinality of $\mathcal{I}$, i.e., the number of disclosed attributes. In our particular parameter selection, the ring dimension is 4096/8192, which can be already sufficient for many applications. The whole approach can handle even longer attributes by extending the current CTS instantiation to handle more commitments, e.g., using more matrices in the $\mathbf{F}_{\mathsf{comm}}$ of Construction 4.1. The concrete efficiency of the extension needs to be further determined, and we leave it as an interesting future work.

Interactive proof system $\Pi_{\mathsf{Disclosure}}$

Public Parameter for Commitment Scheme:

$\mathbf{A} = \begin{bmatrix} \boldsymbol{a}_1^\top \\ \boldsymbol{a}_2^\top \end{bmatrix} = \begin{bmatrix} 1, & \boldsymbol{a}_1'^\top \\ 0, 1, & a_2' \end{bmatrix}$ as in Construction 4.1, $B = \xi \cdot \sqrt{6N}$, $\mathbf{B}^\top = (\boldsymbol{b}_i) \in R_{q_2}^{3 \times \hat{k}}$

Commitment: $\mathsf{comm}_1 := \begin{bmatrix} t_{1,1} \\ t_{2,1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1^\top \\ \boldsymbol{a}_2^\top \end{bmatrix} \cdot \boldsymbol{r}_1 + \begin{bmatrix} 0 \\ m \end{bmatrix}$, with $m = \sum_{i \in [N]} m_{i-1} X^{i-1}$.

Let $\mathcal{I} \subseteq [N]$ denote the subset of indices where the prover wants to disclose $m_{\mathcal{I}} = \{m_i\}_{i \in \mathcal{I}}$. Set $m_{\mathsf{att}} = \sum_{i \in \mathcal{I}} m_i X^{i-1}$, $m' = \sum_{i \in [N] \setminus \mathcal{I}} m_{i-1} X^{i-1}$, where any coefficient of m' with respect to X^i is 0 for $i \in \mathcal{I}$. The prover discloses m_{att} publicly.

Prover		Verifier
$\boldsymbol{g} := (g_1, \ldots, g_{\hat{k}})^\top \xleftarrow{\$} \{f \in R_q : f_0 = \cdots = f_{\lvert\mathcal{I}\rvert - 1} = 0\}^{\hat{k}}$, $\boldsymbol{t}_g = (t_{g,i}) = \mathbf{B} \cdot \boldsymbol{r}_1 + \boldsymbol{g}$		
	$\xrightarrow{\boldsymbol{t}_g}$	
	$\xleftarrow{(\gamma_i)_{i \in [\hat{k}]}}$	$(\gamma_i) \xleftarrow{\$} \mathbb{Z}_{q_2}^{\hat{k}}$
$\forall i \in [\hat{k}], h_i = g_i + \gamma_i \cdot m'$ $\boldsymbol{y} \leftarrow \mathcal{D}_\xi^3$, $w = \boldsymbol{a}_1^\top \cdot \boldsymbol{y}$ $\forall i \in [\hat{k}]$, $w_i = (\gamma_i \cdot \boldsymbol{a}_2^\top + \boldsymbol{b}_i^\top) \cdot \boldsymbol{y}$		
	$\xrightarrow{w, h_i, w_i}$	
		$d \xleftarrow{\$} \mathcal{C}$
	$\xleftarrow{d}$	
$\boldsymbol{z} = \boldsymbol{y} + d \cdot \boldsymbol{r}_1$ $\mathsf{Rej}(\boldsymbol{z}_{i,1}, d \cdot \boldsymbol{r}_i, \xi)$		
	$\xrightarrow{\boldsymbol{z}}$	
	Check:	
	1. $\lVert\boldsymbol{z}\rVert \overset{?}{\leq} B$, $\boldsymbol{a}_1^\top \cdot \boldsymbol{z} \overset{?}{=} w + d \cdot t_{1,1}$	
	2. $\forall i \in [\hat{k}]$, whether the coefficients with respect to $\mathcal{I}$ in h_i are zero, and $(\gamma_i \cdot \boldsymbol{a}_2^\top + \boldsymbol{b}_i^\top) \cdot \boldsymbol{z} \overset{?}{=} w + d \cdot (\gamma_i \cdot (t_{2,1} - m_{\mathsf{att}}) + t_{g,i} - h_i)$	
	Accept if all the above conditions is sufficient.	

Table 6. The interactive version of $\Pi_{\mathsf{Disclosure}}$: Disclosure of certain coefficients in the committed message m.

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Heidelberg, May / June 2010.
2. S. Agrawal, E. Kirshanova, D. Stehlé, and A. Yadav. Practical, round-optimal lattice-based blind signatures. In H. Yin, A. Stavrou, C. Cremers, and E. Shi, editors, *ACM CCS 2022*, pages 39–53. ACM Press, Nov. 2022.
3. M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, and T. Wunderer. Estimate all the LWE, NTRU schemes! In Catalano and De Prisco [21], pages 351–367.
4. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange - A new hope. In T. Holz and S. Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, Aug. 2016.
5. T. Attema, V. Lyubashevsky, and G. Seiler. Practical product proofs for lattice commitments. In D. Micciancio and T. Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 470–499. Springer, Heidelberg, Aug. 2020.
6. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In Catalano and De Prisco [21], pages 368–385.
7. M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. P-signatures and noninteractive anonymous credentials. In R. Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 356–374. Springer, Heidelberg, Mar. 2008.
8. M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In A. Juels, R. N. Wright, and S. De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399. ACM Press, Oct. / Nov. 2006.
9. W. Beullens. Breaking rainbow takes a weekend on a laptop. In Dodis and Shrimpton [25], pages 464–479.
10. W. Beullens, V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Lattice-based blind signatures: Short, efficient, and round-optimal. Cryptology ePrint Archive, Paper 2023/077, 2023. `https://eprint.iacr.org/2023/077`.
11. A. Boldyreva and D. Micciancio, editors. *CRYPTO 2019, Part I*, volume 11692 of *LNCS*. Springer, Heidelberg, Aug. 2019.
12. J. Bootle, V. Lyubashevsky, N. K. Nguyen, and A. Sorniotti. A framework for practical anonymous credentials from lattices. Cryptology ePrint Archive, Paper 2023/560, 2023. `https://eprint.iacr.org/2023/560`.
13. J. Bootle, V. Lyubashevsky, and G. Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In Boldyreva and Micciancio [11], pages 176–202.
14. J. Camenisch, A. Lehmann, G. Neven, and A. Rial. Privacy-preserving auditing for attribute-based credentials. In M. Kutylowski and J. Vaidya, editors, *ESORICS 2014, Part II*, volume 8713 of *LNCS*, pages 109–127. Springer, Heidelberg, Sept. 2014.
15. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In B. Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, Heidelberg, May 2001.
16. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In S. Cimato, C. Galdi, and G. Persiano, editors, *SCN 02*, volume 2576 of *LNCS*, pages 268–289. Springer, Heidelberg, Sept. 2003.

17. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, Heidelberg, Aug. 2004.
18. J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In I. Visconti and R. D. Prisco, editors, *SCN 12*, volume 7485 of *LNCS*, pages 57–75. Springer, Heidelberg, Sept. 2012.
19. R. Canetti and J. A. Garay, editors. *CRYPTO 2013, Part I*, volume 8042 of *LNCS*. Springer, Heidelberg, Aug. 2013.
20. W. Castryck and T. Decru. An efficient key recovery attack on sidh. Springer-Verlag, 2023.
21. D. Catalano and R. De Prisco, editors. *SCN 18*, volume 11035 of *LNCS*. Springer, Heidelberg, Sept. 2018.
22. R. del Pino and S. Katsumata. A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In Dodis and Shrimpton [25], pages 306–336.
23. R. del Pino, V. Lyubashevsky, and G. Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In D. Lie, M. Mannan, M. Backes, and X. Wang, editors, *ACM CCS 2018*, pages 574–591. ACM Press, Oct. 2018.
24. Y. Dodis. Graduate course - advanced cryptography: Lecture 13. 2009.
25. Y. Dodis and T. Shrimpton, editors. *CRYPTO 2022, Part II*, volume 13508 of *LNCS*. Springer, Heidelberg, Aug. 2022.
26. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal Gaussians. In Canetti and Garay [19], pages 40–56.
27. M. F. Esgin, N. K. Nguyen, and G. Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In S. Moriai and H. Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 259–288. Springer, Heidelberg, Dec. 2020.
28. M. F. Esgin, R. Steinfeld, J. K. Liu, and D. Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In Boldyreva and Micciancio [11], pages 115–146.
29. M. F. Esgin, R. K. Zhao, R. Steinfeld, J. K. Liu, and D. Liu. MatRiCT: Efficient, scalable and post-quantum blockchain confidential transactions protocol. In L. Cavallaro, J. Kinder, X. Wang, and J. Katz, editors, *ACM CCS 2019*, pages 567–584. ACM Press, Nov. 2019.
30. S. Faust, M. Kohlweiss, G. A. Marson, and D. Venturi. On the non-malleability of the Fiat-Shamir transform. In S. D. Galbraith and M. Nandi, editors, *INDOCRYPT 2012*, volume 7668 of *LNCS*, pages 60–79. Springer, Heidelberg, Dec. 2012.
31. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, Aug. 1987.
32. G. Fuchsbauer, C. Hanser, and D. Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *Journal of Cryptology*, 32(2):498–546, Apr. 2019.
33. N. Gama and P. Q. Nguyen. Predicting lattice reduction. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 31–51. Springer, Heidelberg, Apr. 2008.
34. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

35. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Canetti and Garay [19], pages 75–92.
36. C. Jeudy, A. Roux-Langlois, and O. Sanders. Lattice signature with efficient protocols, application to anonymous credentials. Cryptology ePrint Archive, Paper 2022/509, 2022. `https://eprint.iacr.org/2022/509`.
37. S. Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to QROM secure NIZKs. In T. Malkin and C. Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 580–610, Virtual Event, Aug. 2021. Springer, Heidelberg.
38. A. Langlois and D. Stehle. Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography, 2015.
39. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 373–403. Springer, Heidelberg, Dec. 2016.
40. A. Lysyanskaya. *Signature schemes and applications to cryptographic protocol design*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, Sept. 2002.
41. A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In H. Heys and C. Adams, editors, *Selected Areas in Cryptography*, volume 1758, 1999.
42. A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In H. M. Heys and C. M. Adams, editors, *SAC 1999*, volume 1758 of *LNCS*, pages 184–199. Springer, Heidelberg, Aug. 1999.
43. V. Lyubashevsky. Lattice signatures without trapdoors. In Pointcheval and Johansson [52], pages 738–755.
44. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Dodis and Shrimpton [25], pages 71–101.
45. V. Lyubashevsky, N. K. Nguyen, M. Plançon, and G. Seiler. Shorter lattice-based group signatures via "almost free" encryption and other optimizations. In M. Tibouchi and H. Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 218–248. Springer, Heidelberg, Dec. 2021.
46. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In J. Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 215–241. Springer, Heidelberg, May 2021.
47. L. Maino, C. Martindale, L. Panny, G. Pope, and B. Wesolowski. A direct key recovery attack on sidh. Springer-Verlag, 2023.
48. D. Micciancio. On the hardness of learning with errors with binary secrets. *Theory of Computing*, 14(1):1–17, 2018.
49. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In Pointcheval and Johansson [52], pages 700–718.
50. D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381. IEEE Computer Society Press, Oct. 2004.
51. C. Peikert and S. Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Boldyreva and Micciancio [11], pages 89–114.
52. D. Pointcheval and T. Johansson, editors. *EUROCRYPT 2012*, volume 7237 of *LNCS*. Springer, Heidelberg, Apr. 2012.
53. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.

54. D. Robert. Breaking sidh in polynomial time. Springer-Verlag, 2023.
55. Y. Tao, X. Wang, and R. Zhang. Short zero-knowledge proof of knowledge for lattice-based commitment. In J. Ding and J.-P. Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 268–283. Springer, Heidelberg, 2020.
56. R. Yang, M. H. Au, J. Lai, Q. Xu, and Z. Yu. Lattice-based techniques for accountable anonymity: Composition of abstract stern's protocols and weak PRF with efficient protocols from LWR. Cryptology ePrint Archive, Report 2017/781, 2017. `https://eprint.iacr.org/2017/781`.

Roadmap of the Appendix

Here we present a roadmap so that the readers can find relevant texts more easily. In Section A, we present additional preliminaries. In Section B, we present the security analysis of our selectively secure CTS. In Section C, we show the efficient instantiations of the three building blocks in Section 5 from lattices. Going a step further, Section D provides parameter setting details for CTS construction in Section 4.1 and $\mathsf{NIZKPoK}$ system in Section 5.

Notice that the CTS in Section 4 is just proven to be selectively secure. This means we need to use the approach of complexity leverage to achieve adaptive security, which will induce security loss related to the size of message space. As a technical supplement, in Section E, we present how to construct an adaptively secure CTS without the complexity leveraging argument, and determine concrete parameters.

A Supplementary Material for Section 2

A.1 Lattices with Algebra Structure

Below, we use R to denote a polynomial ring of the form $\mathbb{Z}[X]/(\Phi_m(X))$, where $\Phi_m(X)$ is the m^{th} cyclotomic polynomial, and denote $N = \varphi(m)$. For an integer $q \in \mathbb{Z}$, we also consider the quotient ring $R_q = R/qR$. Any element in R can be considered as a vector of its coefficients. Namely, an element $a = \sum_{i\in[N]} a_i x^i \in R$ can be seen as the vector $\boldsymbol{a} = (a_0, ..., a_{N-1})$. We call this map as coefficient embedding (denoted as $\mathsf{Coeffs}(\cdot)$). Furthermore, we can also represent a ring element $a \in R$ as a matrix in $\mathbb{Z}^{N\times N}$ by the following map $\mathsf{Rot} : R \to \mathbb{Z}^{N\times N}$:

$$\mathsf{Rot}(a) := \begin{bmatrix} \mathsf{Coeffs}(a)^\top \\ \mathsf{Coeffs}(xa \bmod \Phi(x))^\top \\ \vdots \\ \mathsf{Coeffs}(x^{N-1}a \bmod \Phi(x)\ )^\top \end{bmatrix}.$$

Furthermore, we extend this map to ring vectors and matrices by applying it entry-wise, i.e., for a vector $\boldsymbol{a}^\top = (a_1, \ldots, a_\ell) \in R^\ell$, we define $\mathsf{Rot}(\boldsymbol{a}^\top) = [\mathsf{Rot}(a_1)|\ldots|\mathsf{Rot}(a_\ell)] \in \mathbb{Z}^{n\times n\ell}$, and the map for matrices can be defined similarly. In the case of power of 2 cyclotomic rings, i.e., $\Phi(x) = x^N + 1$ for n being some power of 2, the above rotation matrix $\mathsf{Rot}(a)$ is the anti-cyclic matrix.

If $\mathcal{I}$ is an ideal in the polynomial ring R, then it is also an additive subgroup of $\mathbb{Z}^N$, and therefore a N-dimensional lattice. Such lattices are therefore sometimes referred to as *ideal lattices*. Similarly, we can also define the *module lattices* $M \subseteq (\mathbb{Q}[X]/(\Phi_m(X)))^\ell$ as a ℓN-dimensional lattice. We simply denote *ideal lattices* or *module lattices* as Λ.

Discrete Gaussian distribution. We now define the Gaussian distribution used in our schemes.

Definition A.1 *The discrete Gaussian distribution on $\Lambda \subseteq R^\ell$ centered around $\boldsymbol{v} \in R^\ell$ with standard deviation $s > 0$ is given by* $D_{\Lambda,\boldsymbol{v},s}(\boldsymbol{x}) = \frac{e^{-\|\boldsymbol{x}-\boldsymbol{v}\|^2/2s^2}}{\sum_{\boldsymbol{z}\in\Lambda} e^{-\|\boldsymbol{z}-\boldsymbol{v}\|^2/2s^2}}$. *When it is centered around $\mathbf{0}$, we denote $D_{\Lambda,s}$ for short.*

Specifically, for ring vector $\boldsymbol{x}$, we write $\boldsymbol{x} \leftarrow D_{\Lambda,s}$ to mean that $\boldsymbol{x} \in \Lambda \subseteq R^\ell$ and every coefficient of each component $x_i \in R$ is distributed according to $D_{\mathbb{Z},s}$. Then, we have the following properties.

Lemma A.2 ([43]) *Let D_s is a discrete Gaussian distribution over the ring R. Then for $\boldsymbol{x} \leftarrow D_s^\ell$, it holds* $\Pr\left[\|\boldsymbol{x}\| > t \cdot s\sqrt{\ell N}\right] \leq \left(te^{\frac{1-t^2}{2}}\right)^{\ell N}$

For positive integers δ and $k = \lceil \log_\delta(q) \rceil$, let $\boldsymbol{g}_\delta^\top = [1|\delta|\delta^2|...|\delta^{k-1}] \in R^k$ be the gadget matrix. Then we have the following lemmas.

Lemma A.3 ([49]) *There exists an efficient algorithm that on input ring vector $\boldsymbol{a} \in R_q^\ell$ such that $\mathsf{Rot}(\boldsymbol{a}^\top) \in \mathbb{Z}^{N\times N\ell}$ is full-rank, elements $x \in R_q^*, u \in R_q$ and matrix $\mathbf{R} \in R_q^{\ell\times k}$, outputs a random sample $\boldsymbol{r} \in R^{\ell+k}$ from a distribution that is statistically close to $D_{\Lambda_q^u[\boldsymbol{a}^\top|\boldsymbol{a}^\top\mathbf{R}+x\cdot\boldsymbol{g}_\delta^\top],\sigma}(\boldsymbol{x})$, where $\sigma \geq 2\sqrt{\delta^2+1}(s_1(\mathbf{R})+1)$.*

Lemma A.4 ([49]) *For $\boldsymbol{g}_\delta^\top = [1|\delta|\delta^2|...|\delta^{k-1}] \in R^k$, there exists a deterministic polynomial time algorithm $\mathbf{G}^{-1}$ which takes input $\boldsymbol{u} \in R_q^k$, and outputs $\mathbf{R} \leftarrow \mathbf{G}^{-1}(\boldsymbol{u}^\top)$ such that $\boldsymbol{g}_\delta \cdot \mathbf{R} = \boldsymbol{u}^\top$, such that $s_1(\mathbf{R}) \leq kN\delta$.*

We here recall the definition of smoothing parameter of a lattice and its upper bound as follow.

Definition A.5 ([50]) *For any n-dimensional lattice Λ and positive real $\epsilon_s > 0$, the smoothing parameter $\eta_{\epsilon_s}(\Lambda)$ is the smallest real $s > 0$ such that $\rho_{1/s}(\Lambda^*\backslash\{0\}) \leq \epsilon_s$, where Λ^* is the dual lattice of Λ.*

Lemma A.6 (Generalization of Lemma 2.6 in [48] to ring setting) *For any primitive matrix $\mathbf{P} \in R^{\ell\times k}$, positive reals $\alpha, \sigma > 0$, and negligible ε, if $\mathbf{P}\cdot\mathbf{P}^\top = \alpha^2 \cdot \mathbf{I}$ and $\eta_\varepsilon(\ker(\mathbf{P})) \leq \sigma$, then $\mathbf{P} \cdot D_\sigma^{kN} \overset{s}{\approx} D_{\alpha\sigma}^{\ell N}$.*

From Lemmas A.3 and A.6, we have the following lemma.

Lemma A.7 *There exists an efficient algorithm that on input ring vectors $\boldsymbol{a}_1 \in R_q^{\ell_1}, \boldsymbol{a}_2 \in R_q^{\ell_2}$ such that $\mathsf{Rot}([\boldsymbol{a}_1^\top|\boldsymbol{a}_2^\top]) \in \mathbb{Z}^{N\times N(\ell_1+\ell_2)}$ is full-rank, elements $x, c \in R_q^*, u \in R_q$ with $\|c\|_2 \leq \tau$ and matrices $\mathbf{R}_1 \in R_q^{\ell_1\times k}, \mathbf{R}_2 \in R_q^{\ell_2\times k}$, outputs a random sample $\boldsymbol{r} \in R^{\ell_1+\ell_2+k}$ from a distribution that is statistically close to $D_\sigma(\Lambda_q^u[\boldsymbol{a}_1^\top|\boldsymbol{a}_1^\top\mathbf{R}_1 + \boldsymbol{a}_2^\top\mathbf{R}_2 + x\cdot\boldsymbol{g}_\delta^\top|\boldsymbol{a}_2^\top])$, where $\sigma \geq 2\sqrt{\delta^2+1}(s_1(\begin{bmatrix}\mathbf{R}_1\\ \mathbf{R}_2\end{bmatrix})+1)$.*

Proof. Given the vector $[\boldsymbol{a}_1^\top|\boldsymbol{a}_1^\top\mathbf{R}_1 + \boldsymbol{a}_2^\top\mathbf{R}_2 + x\cdot\boldsymbol{g}_\delta^\top|\boldsymbol{a}_2^\top] \in R_q^{\ell_1+\ell_2+k}$, consider matrix

$$\mathbf{P} = \begin{pmatrix} \mathbf{I}_{\ell_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_k \\ \mathbf{0} & \mathbf{I}_{\ell_2} & \mathbf{0} \end{pmatrix} \in R_q^{(\ell_1+\ell_2+k)\times(\ell_1+\ell_2+k)},$$

we have $[\boldsymbol{a}_1^\top|\boldsymbol{a}_2^\top|\boldsymbol{a}_1^\top\mathbf{R}_1+\boldsymbol{a}_2^\top\mathbf{R}_2+x\cdot\boldsymbol{g}_\delta^\top]=[\boldsymbol{a}_1^\top|\boldsymbol{a}_1^\top\mathbf{R}_1+\boldsymbol{a}_2^\top\mathbf{R}_2+x\cdot\boldsymbol{g}_\delta^\top|\boldsymbol{a}_2^\top]\cdot\mathbf{P}$.

Let $\ell=\ell_1+\ell_2$, $\boldsymbol{a}^\top=[\boldsymbol{a}_1^\top|\boldsymbol{a}_2^\top]\in R_q^\ell$, and $\mathbf{R}=\begin{bmatrix}-\mathbf{R}_1\\-\mathbf{R}_2\end{bmatrix}$. Clearly, we have $[\boldsymbol{a}^\top|\boldsymbol{a}_1^\top\mathbf{R}_1+\boldsymbol{a}_2^\top\mathbf{R}_2+x\cdot\boldsymbol{g}_\delta^\top]\cdot\begin{bmatrix}\mathbf{R}\\1\end{bmatrix}=x\cdot\boldsymbol{g}_\delta^\top$, where x and 1 are invertible ring elements. Hence, we can view this matrix $\mathbf{R}$ as the $\mathbf{G}$-trapdoor.

Therefore, by Lemma A.3, we can sample vector $\boldsymbol{r}\in R^{\ell+k}$ such that $[\boldsymbol{a}^\top|\boldsymbol{a}^\top\mathbf{R}+x\cdot\boldsymbol{g}_\delta^\top]\cdot\boldsymbol{r}=u\ (\bmod q)$, and the distribution of $\boldsymbol{r}$ is statistically close to $\mathcal{D}_\sigma(\Lambda_q^u[\boldsymbol{a}^\top|\boldsymbol{a}^\top\mathbf{R}+x\cdot\boldsymbol{g}_\delta^\top])$, where $\sigma\geq 2\sqrt{\delta^2+1}(s_1(\mathbf{R})+1)$. As a result, $[\boldsymbol{a}_1^\top|\boldsymbol{a}_1^\top\mathbf{R}_1+\boldsymbol{a}_2^\top\mathbf{R}_2+x\cdot\boldsymbol{g}_\delta^\top|\boldsymbol{a}_2^\top]\cdot\mathbf{P}\cdot\boldsymbol{r}=u\ (\mathrm{mod}q)$. Furthermore, by Lemma A.6, the distribution of $\mathbf{P}\cdot\boldsymbol{r}$ is statistically close to $D_\sigma(\Lambda_q^u[\boldsymbol{a}_1^\top|\boldsymbol{a}_1^\top\mathbf{R}_1+\boldsymbol{a}_2^\top\mathbf{R}_2+x\cdot\boldsymbol{g}_\delta^\top|\boldsymbol{a}_2^\top])$ (it's easy to see $\mathbf{P}\cdot\mathbf{P}^\top=\mathbf{I}$ and $\eta_\varepsilon\leq\sigma$). This completes the proof. □

In this paper, we use the following sampling algorithm. The following lemma have been established in a sequence of works.

Lemma A.8 ([1, 34]) *Given integers $n\geq 1, q\geq 2$ there exists some $m=m(n,q)=O(n\log q)$, there exists a sampling algorithm $\mathsf{SamplePre}(\mathbf{A},\mathbf{T_A},\boldsymbol{u},s)$, that takes as input: (1) a rank-n matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$, (2) a "short" basis $\mathbf{T_A}$ for lattice $\Lambda_q^\perp(\mathbf{A})$, a vector $\boldsymbol{u}\in\mathbb{Z}_q^n$, (3) a Gaussian parameter $s>\|\widetilde{\mathbf{T_A}}\|\cdot\omega(\sqrt{\log m})$; then outputs a vector $\boldsymbol{r}\in\mathbb{Z}^m$ distributed statistically close to $D_{\Lambda_q^{\boldsymbol{u}}(\mathbf{A}),s}$.*

We note that when $\mathbf{A}\in R_q^{\ell\times k}$ is a ring matrix, and $\mathbf{T_A}$ is the trapdoor for $\mathbf{A}$, the $\mathsf{SamplePre}$ algorithm also works by taking $\mathbf{A}$ as a matrix in $\mathbb{Z}_q^{\ell N\times kN}$, which is the coefficient embedding of $\mathbf{A}$.

A.2 Rejection Sampling

Lemma A.9 (Rejection Sampling) *Let V be a subset of $\mathbb{R}^m$ in which all elements have norms less than T, and $h:V\to[0,1]$ be a probability distribution. Let $\sigma=\eta T$ for $\eta=O(\sqrt{\lambda})$ and*

$$M=\exp\left(\sqrt{\frac{2(\lambda+1)}{\log e}}\cdot\frac{1}{\eta}+\frac{1}{2\eta^2}\right)=O(1).$$

Now, sample $\boldsymbol{v}\xleftarrow{\$}h$ and $\boldsymbol{y}\xleftarrow{\$}D_\sigma^m$, set $\boldsymbol{z}=\boldsymbol{y}+\boldsymbol{v}$, and run $b\leftarrow\mathsf{Rej}(\boldsymbol{z},\boldsymbol{v},\sigma)$ in Table 7. Then, the probability that $b=0$ is at least $\frac{1-2^{-\lambda}}{M}$. And conditioned on $b=0$, the distribution of $(\boldsymbol{v},\boldsymbol{z})$ is within statistical distance of $\frac{2^{-\lambda}}{M}$ of the product distribution $h\times D_\sigma^m$.

A.3 Security of Commitment

A secure commitment scheme requires the two properties: hiding and binding.

$\underline{\mathsf{Rej}(\boldsymbol{z}, \boldsymbol{v}, \sigma)}$
01 $u \xleftarrow{\$} [0,1)$
02 If $u > \frac{1}{M} \cdot \exp(\frac{-2\langle \boldsymbol{z}, \boldsymbol{v} \rangle + \|\boldsymbol{v}\|^2}{2\sigma^2})$
03 return 0 (i.e. abort)
04 Else
05 return 1 (i.e. non-abort)

Table 7. Rejection Sampling.

Definition A.10 (Hiding, [6]) *We say that a commitment scheme* (CKeyGen, Commit, Open) *with message space* $\mathcal{M}$ *and randomness space* $\mathcal{R}$ *is hiding, if for all adversaries* $\mathcal{A}$*, the probability (over the randomness of* CKeyGen,Commit*, and* $\mathcal{A}$*) that* $b' = b$ *in the following experiment is negligible:*

Parameter setup *The challenger sets up* $\mathsf{params} \leftarrow \mathsf{CKeyGen}(1^\lambda)$*, and send* params *to* $\mathcal{A}$*.*

Message selection $\mathcal{A}(\mathsf{params})$ *selects two messages* $m_0, m_1 \in \mathcal{M}$*, and then sends them to* $\mathcal{C}$*.*

Commitments *The challenger computes* $\mathsf{comm}_b = \mathsf{Commit}(\mathsf{params}, m_b; r)$*, where* $b \xleftarrow{\$} \{0,1\}$*,* $r \xleftarrow{\$} \mathcal{R}$*, and sends* comm_b *to* $\mathcal{A}$*.*

Output $\mathcal{A}$ *outputs a bit* b'*.*

If $\mathcal{A}$ *are restricted to polynomial-time algorithms, then the scheme is called computationally hiding. If there is no restriction on the running time of such algorithms, then the scheme is statistically hiding.*

Definition A.11 (Binding, [6]) *We say that a commitment scheme* (CKeyGen, Commit, Open) *with message space* $\mathcal{M}$ *and randomness space* $\mathcal{R}$ *is binding, if for all adversaries* $\mathcal{A}$*, the probability*

$$\Pr\left[\begin{array}{l} \mathsf{params} \leftarrow \mathsf{CKeyGen}(1^\lambda), \\ (m, m', r, r', \mathsf{comm}) \leftarrow \mathcal{A}(\mathsf{params}) \\ s.t.\ m \neq m' \wedge \mathsf{Open}(\mathsf{params}, m, \mathsf{comm}, r) = \\ \mathsf{Open}(\mathsf{params}, m', \mathsf{comm}, r') = 1 \end{array}\right] \leq \mathsf{negl}(\lambda),$$

where the probability is taken over the randomness of CKeyGen *and* $\mathcal{A}$*.*

Similarly, if $\mathcal{A}$ *are restricted to polynomial-time algorithms, then the scheme is called computationally binding. If there is no restriction on the running time of such algorithms, then the scheme is statistically binding.*

A.4 Algebraic Structure of Cyclotomic Rings

In this section, we first recall some necessary algebraic background, and then introduce the related and necessary lemmas for our constructions.

We focus mainly on the algebraic structure of m-th cyclotimic filed $K = \mathbb{Q}[X]/\langle \Phi_m(X) \rangle$ of degree $d = \phi(m)$ with the ring of integers $R = \mathbb{Z}[X]/\langle \Phi_m(X) \rangle$.

Here, K is a d-degree Galois extension of $\mathbb{Q}$. Then, we use $G = \mathrm{Gal}(K/\mathbb{Q})$ to denote the Galois group of K, which consists of all automorphisms of K and is computed under composition. Clearly, all these automorphisms fix the rational numbers $\mathbb{Q}$, i.e., for any $\sigma \in G$ and any $x \in \mathbb{Q}$, it holds $\sigma(x) = x$. Conversely, cyclotomic field are Galois over $\mathbb{Q}$ meaning that only the elements of $\mathbb{Q}$ are fixed by all automorphisms in G.

Moreover, the Galois group G of K is isomorphic to $\mathbb{Z}_m^\times$, where the isomorphism $j \mapsto \sigma_j : \mathbb{Z}_m^\times \mapsto \mathrm{Gal}(K/\mathbb{Q})$ is defined by $\sigma_j(X) = X^j$. In general, the degree of a Galois extension of a field is always equal to the order of its Galois group. The main theorem of Galois theory says that there is one-by-one correspondence between the subgroups of G and the subfields of K. For example, let H to be a subgroup of G, i.e., $H < G$. Then H is corresponded to a subfield L of K, i.e., $L < K$. And H is the Galois group of K over L, i.e., $H = \mathrm{Gal}(K/L)$ consists of the automorphisms of K that fix the elements in L. Conversely, as the subfield of K, L consists precisely of all the elements that are fixed by all automorphisms in H, and thus L is called as the fixed field of H. This implies that the extension K/L is again Galois.

Furthermore, by restricting the automorphisms of K to the cyclotomic ring $R \subset K$, we get ring automorphisms of R. And the property that certain subset $S \subset R$ is fixed under automorphisms is still set up. More formally, we have the following lemma from [23].

Lemma A.12 (Theorem 3.1 in [23]) *Let K be a cyclotomic number field with the ring of integers R, and let L be a subfield of K with the ring of integers $\mathcal{S}$. Let G denote the Galois group of K, and H denote a subgroup that consists of all these automorphisms fixing L. Let q is a prime number that is inert in the subfield L, $\mu \in R_q$ be an element that is fixed modulo q by all Galois automorphisms $\sigma \in H$; that is, $\sigma(\mu) \equiv \mu (\mathrm{mod} qR)$ for all $\sigma \in H$. Then, μ is contained in the subfield $\mathcal{S}_q$ of R_q.*

Moreover, for the special case of power-of-two cyclotomic rings, given a power of 2 integer n, we denote $R = \mathbb{Z}[X]/\langle X^d+1\rangle$ as the related cyclotomic ring, since X^d+1 is the $2d$-th cyclotomic polynomial. Similarly, we denote $K = \mathbb{Q}[X]/\langle X^d+1\rangle$ as the related cyclotomic field, which is a d-degree Galois extension of $\mathbb{Q}$. Here, the Galois group G of K is isomorphic to $\mathbb{Z}_{2d}^\times$, which has the structure $\mathbb{Z}_2 \times \mathbb{Z}_{d/2}$. Notice that the cyclic subgroup $\mathbb{Z}_2$ and $\mathbb{Z}_{d/2}$ are generated by σ_{-1} and σ_5, respectively.

Given a prime q and the integer ring $R = \mathbb{Z}[X]/\langle X^d + 1\rangle$, we need to ensure the message space $\mathcal{M} \subseteq R_q$ is a subfield of $K = \mathbb{Q}[X]/\langle X^d + 1\rangle$. According to the above mentioned Galois group structure of general cyclotomic rings, the necessary and sufficient conditions for $\mathcal{M}$ to be a subfield is:

1. Its elements are fixed by a subgroup of G. This means that the message is contained in $\mathcal{S}_q = \mathcal{S}/q\mathcal{S}$ where $\mathcal{S} \subseteq R$ is the ring of integers of a subfield of K.
2. Prime number q stay inert in $\mathcal{S}$ such that $\mathcal{S}_q$ is a field.

With respect to the above two conditions, we have the following two formal lemmas from [23].

Lemma A.13 (Theorem 3.2 in [23]) *Let $d > k \geq 1$ be powers of 2. The subgroup $H = \langle \sigma_{-1}, \sigma_5^k \rangle$ of the Galois group $G = Gal(K/\mathbb{Q})$ has index k. Its fixed field L is generated by $\alpha = X^{d-\frac{d}{2k}} - X^{\frac{d}{2k}}$ over Q inside K, $L = \mathbb{Q}[\alpha] \subset K$.*

Lemma A.14 (Theorem 3.3 in [23]) *The prime numbers that are inert in the fixed field L of $\langle \sigma_{-1}, \sigma_5^k \rangle$ with $1 < k < d$ be power of two, are precisely the primes that are congruent to 3 or 5 modulo 8. They split into two prime ideals in K.*

A.5 Proof of Theorem 2.3

In this section, we first present the concrete protocol for the well-formness of BDLOP commitment, and then prove Theorem 2.3. Finally, we analyze the efficiency of this concrete protocol, and compare it with that of the previous opening proof in [6, 23].

Interactive Proof Protocol

Here, we first present the interactive version in Table 8.

Proof of Theorem 2.3

Theorem A.15 (Restatement of Theorem 2.3) *In the random oracle model, for a secure BDLOP commitment, there exists a NIZKPoK system Π for the relaxed language $L_{\gamma_1', \gamma_2', q_1, q_2, \bar{c}}$, with $\gamma_1' = 2\sqrt{2}\eta \cdot \kappa \cdot \beta \cdot kN$ and $\gamma_2' = 2\sqrt{2}\eta \cdot \kappa \cdot \beta \cdot \ell N$, where η is the parameter for rejection sampling as in Lemma A.9.*

Proof. Essentially, this proof consists of two steps: the first is that of proving the protocol in Table 8 is complete, statistical honest verifier zero-knowledge and computational sound under the M-SIS assumption; the second is that of making it non-interactive with the help of the standard Fiat-Shamir technique. As the second one is natural, it suffices for us to just focus on the first one. Details are given as follows.

Completeness. The vectors $\boldsymbol{z}_1, \boldsymbol{z}_2$ sent by $\mathcal{P}$ are independent and their distributions have statistical distance at most $2^{-\lambda}$ from $\mathcal{D}_{\sigma_1}^n$ and $\mathcal{D}_{\sigma_2}^\ell$ respectively, by Lemma A.9 on rejection sampling. Furthermore, Lemma A.2 implies that the bounds $\|\boldsymbol{z}_i\|_2 \leq B_i$ holds with overwhelming probability. Besides, it is easy to verify that all of the other verification equations are always true for the messages sent by $\mathcal{P}$.

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
Inputs:		
$\mathbf{A}_1 \in R_{q_1}^{n\times k}, \mathbf{A}_2 \in R_{q_2}^{\ell\times k}$		$\mathbf{A}_1, \mathbf{A}_2$
$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{I} \end{bmatrix}$		$\mathbf{A}, \boldsymbol{t}_1, \boldsymbol{t}_2$
$\boldsymbol{t}_1 \in R_{q_1}^n, \boldsymbol{t}_2 \in R_{q_2}^k$		$B_1 \geq \sigma_1 \cdot \sqrt{2N \cdot k}$
$\boldsymbol{r} \in S_\beta^n, \boldsymbol{m} \in S_{\beta'}^\ell$		$B_2 \geq \sigma_2 \cdot \sqrt{2N \cdot \ell}$
$\boldsymbol{y}_1 \leftarrow \mathcal{D}_{\sigma_1}^k$		
$\boldsymbol{y}_2 \leftarrow \mathcal{D}_{\sigma_2}^\ell$		
$\boldsymbol{w}_1 = \mathbf{A}_1 \cdot \boldsymbol{y}_1$		
$\boldsymbol{w}_2 = \mathbf{A}_2 \cdot \boldsymbol{y}_1 + \boldsymbol{y}_2$		
	$\xrightarrow{\boldsymbol{w}_1, \boldsymbol{w}_2}$	
	$\xleftarrow{c}$	$c \xleftarrow{\$} \mathcal{C}$
$\boldsymbol{z}_1 = \boldsymbol{y}_1 + c \cdot \boldsymbol{r}$		
$\boldsymbol{z}_2 = \boldsymbol{y}_2 + c \cdot \boldsymbol{m}$		
If $\mathsf{Rej}(\boldsymbol{z}_1, c \cdot \boldsymbol{r}, \sigma_1) = 1$		
or $\mathsf{Rej}(\boldsymbol{z}_2, c \cdot \boldsymbol{m}, \sigma_2) = 1$,	$\xrightarrow{\boldsymbol{z}_1, \boldsymbol{z}_2}$	
abort		
		Check:
		$\|\boldsymbol{z}_1\| \leq B_1$, $\|\boldsymbol{z}_2\| \leq B_2$
		$\mathbf{A}_1 \cdot \boldsymbol{z}_1 = \boldsymbol{w}_1 + c \cdot \boldsymbol{t}_1$
		$\mathbf{A}_2 \cdot \boldsymbol{z}_1 + \boldsymbol{z}_2 = \boldsymbol{w}_2 + c \cdot \boldsymbol{t}_2$

Table 8. Well-formness proof of BDLOP commitment.

Statistical honest verifier zero-knowledge. Here, we just need to prove that the protocol is zero-knowledge when $\mathcal{P}$ does not abort prior to sending $\boldsymbol{z}_i$. This is because after converting into non-interactive proofs via Fiat-Shamir transform, $\mathcal{V}$ never sees the aborting transcripts. We can prove this zero-knowledge properties by designing a PPT simulator $\mathcal{S}$ whose outputs are statistically close to the transcript of real protocol. Particularly, given matrices $\mathbf{A}_1 \in R_{q_1}^{n\times k}$, $\mathbf{A}_2 \in R_{q_2}^{\ell\times k}$, and commitment vectors $\boldsymbol{t}_1 \in R_{q_1}^n$, $\boldsymbol{t}_2 \in R_{q_2}^\ell$, $\mathcal{S}$ conducts the followings

- Sample $c \xleftarrow{\$} \mathcal{C}$;
- Sample $\boldsymbol{z}_1 \leftarrow \mathcal{D}_{\sigma_1}^k$, and $\boldsymbol{z}_2 \leftarrow \mathcal{D}_{\sigma_2}^\ell$;
- Set $\boldsymbol{w}_1 = \mathbf{A}_1 \cdot \boldsymbol{z}_1 - c \cdot \boldsymbol{t}_1$, $\boldsymbol{w}_2 = \mathbf{A}_2 \cdot \boldsymbol{z}_1 + \boldsymbol{z}_2 - c \cdot \boldsymbol{t}_2$;
- Output $(\boldsymbol{w}_1, \boldsymbol{w}_2, c, \boldsymbol{z}_1, \boldsymbol{z}_2)$.

Clearly, the vectors $\boldsymbol{z}_1, \boldsymbol{z}_2$ output by $\mathcal{S}$ will be accepted with overwhelming probability. Besides, the distribution of $\boldsymbol{z}_i$ output in the real protocol is within a negligible statistical distance of $\mathcal{D}_{\sigma_1}^k$ or $\mathcal{D}_{\sigma_2}^\ell$. Since $\boldsymbol{w}_1, \boldsymbol{w}_2$ are completely determined by $\mathbf{A}_1, \mathbf{A}_2$, $\boldsymbol{t}_1, \boldsymbol{t}_2, c$, $\boldsymbol{z}_1, \boldsymbol{z}_2$, the output distribution of $\mathcal{S}$ is within a negligible statistical distance of these random variables in the actual protocol.

Special soundness. Suppose there exists an adversary $\mathcal{A}$ who can produce a valid proof $\pi := (\boldsymbol{w}_1, \boldsymbol{w}_2, c, \boldsymbol{z}_1, \boldsymbol{z}_2)$ for two vectors $(\boldsymbol{t}_1, \boldsymbol{t}_2) \notin L_{\gamma_1', \gamma_2', q_1, q_2, \bar{c}}$. Then,

we can rewind $\mathcal{A}$ to obtain another adversary $\pi := (\boldsymbol{w}_1, \boldsymbol{w}_2, c', \boldsymbol{z}'_1, \boldsymbol{z}'_2)$ with $c \neq c'$ and $\bar{c} = c - c' \in \bar{C}$ is invertible. Then, we can compute $f = (c - c') \in \bar{\mathcal{C}}$, and set $\bar{\boldsymbol{r}} = \boldsymbol{z}_1 - \boldsymbol{z}'_1$, $\bar{\boldsymbol{m}} = \boldsymbol{z}_2 - \boldsymbol{z}'_2$ such that $\mathbf{A}_1 \cdot \bar{\boldsymbol{r}} = f \cdot \boldsymbol{t}_1$, and $\mathbf{A}_2 \cdot \bar{\boldsymbol{r}} + \bar{\boldsymbol{m}} = f \cdot \boldsymbol{t}_2$.

Below, we compute the ℓ_2-norm of the extracted vectors $\bar{\boldsymbol{r}}, \bar{\boldsymbol{m}}$. According to the rejection sampling in Lemma A.9 and $\|\boldsymbol{r}\|_\infty \leq \beta$, $\|\boldsymbol{m}\|_\infty \leq \beta'$, we need to set $\sigma_1 = \eta \cdot \sqrt{\kappa} \cdot \beta \cdot \sqrt{kN}$ and $\sigma_2 = \eta \cdot \sqrt{\kappa} \cdot \beta' \cdot \sqrt{\ell N}$. And thus, we get $\|\bar{\boldsymbol{r}}\|_2 \leq 2\sqrt{2} \cdot \sigma_1 \cdot \sqrt{kN} = 2\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot \beta \cdot kN = \gamma'_1$ and $\|\bar{\boldsymbol{m}}\|_2 \leq 2\sqrt{2} \cdot \sigma_2 \cdot \sqrt{\ell N} = 2\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot \beta' \cdot \ell N = \gamma'_2$, where η is the parameter for rejection sampling as in Lemma A.9. This implies we can view $(\bar{\boldsymbol{r}}, \bar{\boldsymbol{m}})$ as a witness for $(\boldsymbol{t}_1, \boldsymbol{t}_2) \in L_{\gamma'_1, \gamma'_2, q_1, q_2, \bar{\mathcal{C}}}$. However, this is clearly contradictive with the previous assumption that $(\boldsymbol{t}_1, \boldsymbol{t}_2)$ is not in the language $L_{\gamma'_1, \gamma'_2, q_1, q_2, \bar{\mathcal{C}}}$, which implies the protocol is computationally special soundness. Notice that this special soundness implicitly implies the properties of computational soundness. □

Proof Size of Non-Interactive Protocol

In this section, we analyze the efficiency of the non-interactive protocol of Table 8. This means that the challenge $c \in \mathcal{C}$ is computed by $\mathcal{P}$ via hashing all previous messages and public information. And the hash function is modeled as a random oracle. In order to shorten the length of the proof, we can adopt a standard technique that is not to directly send the input to the hash function, but rather send its output (i.e. the challenge). In this case, given the transmitted vector $\boldsymbol{z}_i$, the verifier can recompute the input, through using the verification equation, and then check that the hash of these computed input terms is indeed the transmitted challenge c. As a result, the proof size of the non-interactive protocol consists of that of vectors $\boldsymbol{z}_i$ and the challenge c, i.e.,

$$k \cdot N \cdot \lceil \log(12\sigma_1) \rceil + \ell \cdot N \cdot \lceil \log(12\sigma_2) \rceil + 256,$$

where the output size of random oracle is supposed to be 256 bits.

Notice that for our parameter setting on Construction 4.1 (i.e., $n = \ell = 1, k = 3$), if we choose message polynomial $\boldsymbol{m}$ with coefficients in $\{-1, 0, 1\}$, this proof of well-formness is just larger than the previous opening proof in [6, 23] by one third times. Clearly, this overhead is mild.

Additional Properties of the Protocol in Table 8

In this section, we first present a proof of knowledge on linear relationship as in [23]. Particularly, given a set of commitments $\boldsymbol{t}_i = \begin{bmatrix} \boldsymbol{t}_{i,1} \\ \boldsymbol{t}_{i,2} \end{bmatrix}$, we prove that their openings $\boldsymbol{m}_i$ satisfying $\sum \mathbf{B}_i \boldsymbol{m}_i = 0$ for any fixed $\mathbf{B}_i$. The detailed protocol is presented in Table 9. Here, due to the similarity with [23], we omit the detailed proof of completeness, honest-verifier zero-knowledge, and special soundness for simplity.

Moreover, just as mentioned in Section 2.3, our new well-formeness proof in Table 8 can prevent the mix-and-match attacks for our anonymous credential systems. In order to specify this more clearly, below we first introduce what the mix-and-match attack is, and then argue this will induce a solution for M-SIS problem.

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
Inputs:		
For $i \in [\tau]$:		
$\mathbf{A}_{1,i} \in R_{q_1}^{n_i \times k_i}$		$\mathbf{A}_{i,1}, \mathbf{A}_{i,2}$
$\mathbf{A}_{2,i} \in R_{q_2}^{\ell_i \times k_i}$		$\mathbf{B}_i$
$\mathbf{B}_i \in R_{q_2}^{x \times \ell_i}$		
$\mathbf{A}_i = \begin{bmatrix} \mathbf{A}_{i,1} & \mathbf{0}_i \\ \mathbf{A}_{i,2} & \mathbf{I}_i \end{bmatrix}$		$\mathbf{A}_i, \boldsymbol{t}_{i,1}, \boldsymbol{t}_{i,2}$
with zero matrix:		
$\mathbf{0}_i \in R^{n_i \times \ell_i}$		
identity matrix:		
$\mathbf{I}_i \in R^{\ell_i \times \ell_i}$		
$\boldsymbol{t}_{i,1} \in R_{q_1}^{n_i}, \boldsymbol{t}_{i,2} \in R_{q_2}^{k_i}$		$B_{i,1} \geq \sigma_{i,1} \cdot \sqrt{N \cdot k_i}$
$\boldsymbol{r}_i \in S_{\beta}^{n_i}, \boldsymbol{m}_i \in S_{\beta'}^{\ell_i}$		$B_{i,2} \geq \sigma_{i,2} \cdot \sqrt{N \cdot \ell_i}$
s.t.		
$\begin{bmatrix} \boldsymbol{t}_{i,1} \\ \boldsymbol{t}_{i,2} \end{bmatrix} = \mathbf{A}_i \begin{bmatrix} \boldsymbol{r}_{i,1} \\ \boldsymbol{m}_{i,2} \end{bmatrix}$		
$\sum \mathbf{B}_i \boldsymbol{m}_i = \mathbf{0} \in R_{q_2}^x$		
For $\forall i \in [\tau]$		
$\boldsymbol{y}_{i,1} \leftarrow \mathcal{D}_{\sigma_{i,1}}^{k_i}$		
$\boldsymbol{y}_{i,2} \leftarrow \mathcal{D}_{\sigma_{i,2}}^{\ell_i}$		
$\boldsymbol{w}_{i,1} = \mathbf{A}_{i,1} \cdot \boldsymbol{y}_{i,1}$		
$\boldsymbol{w}_{i,2} = \mathbf{A}_{i,2} \cdot \boldsymbol{y}_{i,1} + \boldsymbol{y}_{i,2}$		
$\boldsymbol{w}_2 = \sum_i \mathbf{B}_i \mathbf{A}_{i,2} \boldsymbol{y}_{i,1}$		
	$\boldsymbol{w}_2$ $\boldsymbol{w}_{i,1}$ $\boldsymbol{w}_{i,2}$ $\longrightarrow$	
	$\longleftarrow c$	$c \xleftarrow{\$} \mathcal{C}$
$\boldsymbol{z}_{i,1} = \boldsymbol{y}_{i,1} + c \cdot \boldsymbol{r}_i$		
$\boldsymbol{z}_{i,2} = \boldsymbol{y}_{i,2} + c \cdot \boldsymbol{m}_i$		
For $\forall i \in [\tau]$, if $\mathsf{Rej}(\boldsymbol{z}_{i,1}, c \cdot \boldsymbol{r}_i, \sigma_{i,1}) = 1$ or $\mathsf{Rej}(\boldsymbol{z}_{i,2}, c \cdot \boldsymbol{m}_i, \sigma_{i,2}) = 1$, abort	$\boldsymbol{z}_1, \boldsymbol{z}_2 \longrightarrow$	
		Check: for $\forall i \in [\tau]$
		$\lVert \boldsymbol{z}_{i,1} \rVert \leq B_{i,1}$
		$\lVert \boldsymbol{z}_{i,2} \rVert \leq B_{i,2}$
		$\mathbf{A}_{i,1} \cdot \boldsymbol{z}_{i,1} = \boldsymbol{w}_{i,1} + c \cdot \boldsymbol{t}_{i,1}$
		$\mathbf{A}_{i,2} \cdot \boldsymbol{z}_{i,1} + \boldsymbol{z}_{i,2} = \boldsymbol{w}_{i,2} + c \cdot \boldsymbol{t}_{i,2}$
		$\sum \mathbf{B}_i \mathbf{A}_{i,2} \boldsymbol{z}_{i,1} = c \sum \mathbf{B}_i \boldsymbol{t}_{i,2} + \boldsymbol{w}_2$

Table 9. Linear-relationship Proof of BDLOP commitment.

Definition A.16 (Mix-and-Match attack) *Given a pair of* BDLOP *public matrices* $\mathbf{A}_1, \mathbf{A}_2$ *and two vectors* $\boldsymbol{t}_{1,1}, \boldsymbol{t}_{1,2}$*, together with an opening* NIZKPoK *proof* π *showing that* $\begin{bmatrix} \boldsymbol{t}_{1,1} \\ \boldsymbol{t}_{1,2} \end{bmatrix}$ *is a valid commitment with respect to* $\mathbf{A}_1, \mathbf{A}_2$*, if the adversary can find out a new vector* $\boldsymbol{t}_{2,1}$ *together with a new opening* NIZKPoK *proof* π' *showing that* $\begin{bmatrix} \boldsymbol{t}_{2,1} \\ \boldsymbol{t}_{1,2} \end{bmatrix}$ *is a valid commitment with respect to* $\mathbf{A}_1, \mathbf{A}_2$*.*

Definition A.17 (Commitment-Proof Binding Property) *We say the* BDLOP *commitment scheme and its* NIZKPoK *proof system* Π *satisfy the commitment-proof binding property, if they can resist mix-and-match attacks, i.e., it is negligible for any* PPT *adversary to find a vector* $\boldsymbol{t}_{2,1}$ *and a proof* π' *to conduct a successful mix-and-match attack.*

It is easy to verify that for BDLOP commitment scheme, the previous opening proof systems as in [6,23] can not satisfy the commitment-proof binding property. Particularly, suppose $\boldsymbol{t}_1$ is a valid commitment of $\boldsymbol{m}$, with respect to the public matrices $\mathbf{A}_1$, $\mathbf{A}_2$. This means $\boldsymbol{t}_1 = \begin{bmatrix} \boldsymbol{t}_{1,1} \\ \boldsymbol{t}_{1,2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \boldsymbol{r}_1 + \begin{bmatrix} 0 \\ \boldsymbol{m} \end{bmatrix}$. According to the opening proof for BDLOP commitment in [6, 23], given their opening proofs π_1, the corresponding extracted openings are $(\boldsymbol{m}, \bar{\boldsymbol{r}}_1, f_1)$ such that $\mathbf{A}_1 \cdot \bar{\boldsymbol{r}}_1 = f_1 \cdot \boldsymbol{t}_{1,1}$ and $\boldsymbol{m} = \boldsymbol{t}_{1,2} - f_1^{-1} \cdot \mathbf{A}_2 \bar{\boldsymbol{r}}_1$.

In this case, through computing $\boldsymbol{t}_{2,1} = \mathbf{A}_1 \boldsymbol{r}_2$, the adversary can obtain a modified commitment $\boldsymbol{t}_1' = \begin{bmatrix} \boldsymbol{t}_{2,1} \\ \boldsymbol{t}_{1,2} \end{bmatrix}$. Furthermore, the adversary can directly use $\boldsymbol{r}_2$ to generate opening proof π' for $\boldsymbol{t}_1'$, and the corresponding extracted openings are $(\boldsymbol{m}', \bar{\boldsymbol{r}}_2, f_2)$, such that $\mathbf{A}_1 \cdot \bar{\boldsymbol{r}}_2 = f_2 \cdot \boldsymbol{t}_{2,1}$ and $\boldsymbol{m}' = \boldsymbol{t}_{1,2} - f_2^{-1} \cdot \mathbf{A}_2 \bar{\boldsymbol{r}}_2$. Clearly, this is a successful mix-and-match attack.

Fortunately, for our new proof in Table 8, this attack can be prevented.

Claim A.18 *When using the protocol in Table 8 as the opening proof, the* BDLOP *commitment satisfies the commitment-proof binding property.*

Proof. Generally, we give a reduction that if the adversary can conduct the mix-and-match attacks successfully, then we can construct a new algorithm to solve the M-SIS problem.

Particularly, suppose $\boldsymbol{t}_1 = \begin{bmatrix} \boldsymbol{t}_{1,1} \\ \boldsymbol{t}_{1,2} \end{bmatrix}$ is a valid commitment of $\boldsymbol{m}$, with respect to the public matrices $\mathbf{A}_1$, $\mathbf{A}_2$. This means $\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{m} \end{bmatrix} = \begin{bmatrix} \boldsymbol{t}_{1,1} \\ \boldsymbol{t}_{1,2} \end{bmatrix}$. Through using the new proof in Table 8, one can extract witness $(\bar{\boldsymbol{r}}, \bar{\boldsymbol{m}}, f)$ such that $\mathbf{A}_1 \cdot \bar{\boldsymbol{r}} = f \cdot \boldsymbol{t}_{1,1}$ and $\mathbf{A}_2 \cdot \bar{\boldsymbol{r}} + \bar{\boldsymbol{m}} = f \cdot \boldsymbol{t}_{1,2}$, which implies

$$[\mathbf{A}_2, \mathbf{I}] \begin{bmatrix} \bar{\boldsymbol{r}} \\ \bar{\boldsymbol{m}} \end{bmatrix} = f \cdot \boldsymbol{t}_{1,2}. \tag{1}$$

Here, assume the adversary can compute a vector $\boldsymbol{t}_{2,1} \neq \boldsymbol{t}_{1,1}$ such that the modified commitment $\boldsymbol{t}_1' = \begin{bmatrix} \boldsymbol{t}_{2,1} \\ \boldsymbol{t}_{1,2} \end{bmatrix}$ is valid. This means $\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{r}' \\ \boldsymbol{m}' \end{bmatrix} = \begin{bmatrix} \boldsymbol{t}_{2,1} \\ \boldsymbol{t}_{1,2} \end{bmatrix}$

for certain vectors $\boldsymbol{r}', \boldsymbol{m}'$. Furthermore, through using the new proof in Table 8, the extracted witness is $(\bar{\boldsymbol{r}}', \bar{\boldsymbol{m}}', f')$ such that $\mathbf{A}_1 \cdot \bar{\boldsymbol{r}}' = f' \cdot \boldsymbol{t}_{2,1}$ and $\mathbf{A}_2 \cdot \bar{\boldsymbol{r}}' + \bar{\boldsymbol{m}}' = f' \cdot \boldsymbol{t}_{1,2}$, which implies

$$[\mathbf{A}_2, \mathbf{I}] \begin{bmatrix} \bar{\boldsymbol{r}}' \\ \bar{\boldsymbol{m}}' \end{bmatrix} = f' \cdot \boldsymbol{t}_{1,2}. \tag{2}$$

Through multiplying f' and f into Equations (1) and (2), we can get

$$[\mathbf{A}_2, \mathbf{I}] \begin{bmatrix} f' \cdot \bar{\boldsymbol{r}} \\ f' \cdot \bar{\boldsymbol{m}} \end{bmatrix} = f' \cdot f \cdot \boldsymbol{t}_{1,2} \tag{3}$$

and

$$[\mathbf{A}_2, \mathbf{I}] \begin{bmatrix} f \cdot \bar{\boldsymbol{r}}' \\ f \cdot \bar{\boldsymbol{m}}' \end{bmatrix} = f \cdot f' \cdot \boldsymbol{t}_{1,2}. \tag{4}$$

And through subtracting (4) from (3), we can get

$$[\mathbf{A}_2, \mathbf{I}] \begin{bmatrix} f' \cdot \bar{\boldsymbol{r}} - f \cdot \bar{\boldsymbol{r}}' \\ f' \cdot \bar{\boldsymbol{m}} - f \cdot \bar{\boldsymbol{m}}' \end{bmatrix} = 0. \tag{5}$$

Below, we just need to prove that $\begin{bmatrix} f' \cdot \bar{\boldsymbol{r}} - f \cdot \bar{\boldsymbol{r}}' \\ f' \cdot \bar{\boldsymbol{m}} - f \cdot \bar{\boldsymbol{m}}' \end{bmatrix}$ is a non-zero short vector. First, as both f, f' are small, and the ℓ_2 norms of all vectors $\bar{\boldsymbol{r}}, \bar{\boldsymbol{r}}', \bar{\boldsymbol{m}}, \bar{\boldsymbol{m}}'$ are small, the ℓ_2 norm of $\begin{bmatrix} f' \cdot \bar{\boldsymbol{r}} - f \cdot \bar{\boldsymbol{r}}' \\ f' \cdot \bar{\boldsymbol{m}} - f \cdot \bar{\boldsymbol{m}}' \end{bmatrix}$ should be bounded by a small value too.

Second, from $\mathbf{A}_1 \cdot \bar{\boldsymbol{r}} = f \cdot \boldsymbol{t}_{1,1}$ and $\mathbf{A}_1 \cdot \bar{\boldsymbol{r}}' = f \cdot \boldsymbol{t}_{2,1}$, we know that

$$\mathbf{A}_1 \cdot (f' \cdot \bar{\boldsymbol{r}} - f \cdot \bar{\boldsymbol{r}}') = f' f \cdot \boldsymbol{t}_{1,1} - f \cdot f' \cdot \boldsymbol{t}_{2,1}.$$

Then, by the assumption that $\boldsymbol{t}_{2,1} \neq \boldsymbol{t}_{1,1}$, we know the above equation is non-zero, which implies $(f' \cdot \bar{\boldsymbol{r}} - f \cdot \bar{\boldsymbol{r}}') \neq 0$. Finally, this implies the vector $\begin{bmatrix} f' \cdot \bar{\boldsymbol{r}} - f \cdot \bar{\boldsymbol{r}}' \\ f' \cdot \bar{\boldsymbol{m}} - f \cdot \bar{\boldsymbol{m}}' \end{bmatrix}$ is non-zero.

Overall, this implies if there exists the adversary successfully conducting mix-and-match attacks, then we can construct another reduction algorithm to solve M-SIS problem with respect to $[\mathbf{A}_2, \mathbf{I}]$. □

A.6 Security for NIZK

Let's recall the notion of non-interactive zero-knowledge (NIZK) proof system.

Definition A.19 ([24]) *Let $\mathfrak{R}$ be a relation. A non-interactive proof system Π for $\mathfrak{R}$ is a tuple of PPT algorithms* (Setup, Prove, Verify, SimSetup) *having the following interfaces (where 1^λ are implicit inputs to* Prove, Verify, SimSetup*):*

- $\mathsf{Setup}(1^\lambda)$: *given a security parameter λ, outputs a string* crs.
- $\mathsf{Prove}(\mathsf{crs}, x, w)$*: given a string* crs *and a statement-witness pair $(x, w) \in \mathfrak{R}$, outputs a proof π.*

– $\mathsf{Verify}(\mathsf{crs}, x, \pi)$*: given a string* crs*, a statement* x*, and a proof* π*, either accepts or rejects.*
– $\mathsf{SimSetup}(1^\lambda)$*: given a security parameter* λ*, outputs a simulated string* $\widehat{\mathsf{crs}}$ *and a trapdoor* tk.

A secure NIZK system Π should have three properties: Completeness, Soundness, and Zero-knowledge. As argued by [5, 22, 27, 30], Fiat-Shamir based proof systems in the random oracle model satisfy these properties. Many recent lattice-based efficient NIZKs are Fiat-Shamir based, so they also enjoy this property. We require that the following three properties hold:

– *Completeness*: for every $(x, w) \in \mathfrak{R}$ and every λ, $\mathsf{Verify}(\mathsf{crs}, x, \pi)$ accepts with probability 1, over the choice of $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ and $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w)$.
– *Soundness* : let $L_{\mathfrak{R}}$ be the language defined by relation $\mathfrak{R}$. For any PPT adversary $\mathcal{A}$,

$$\mathsf{Pr}_{\mathsf{crs}\leftarrow\mathsf{Setup}(1^\lambda)}\left[\exists x \; s.t. \pi^* \leftarrow \mathcal{A}(\mathsf{crs}, x) : \mathsf{Verify}(\mathsf{crs}, x, \pi^*) \text{ accepts} \wedge x \notin L_{\mathfrak{R}}\right] \leq \mathsf{negl}(\lambda).$$

– *Zero-Knowledge* : There exists one PPT algorithm $\mathsf{SimProve}$, such that, for any PPT adversary $\mathcal{A}$ we have $|\mathsf{Pr}[\mathcal{A} \; wins] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$ in the following game:
  1. The challenger samples $(\widehat{\mathsf{crs}}, \mathsf{tk}) \leftarrow \mathsf{SimSetup}(1^\lambda)$ such that $\widehat{\mathsf{crs}}$ is indistinguishable from crs output by Setup, and gives the simulated $\widehat{\mathsf{crs}}$ to $\mathcal{A}$.
  2. The adversary $\mathcal{A}$ chooses $(x, w) \in \mathfrak{R}$ and gives these to the challenger.
  3. The challenger samples $\pi_0 \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w), \pi_1 \leftarrow \mathsf{SimProve}(\widehat{\mathsf{crs}}, x, \mathsf{tk}), b \leftarrow \{0, 1\}$ and gives π_b to $\mathcal{A}$.
  4. The adversary $\mathcal{A}$ outputs a bit b' and wins if $b' = b$.

Notice that in the above zero-knowledge game, if we allow the adversary $\mathcal{A}$ to choose any polynomial numbers of (x_i, w_i), and all the resulting $\{\pi_{i,0}\}$ and $\{\pi_{i,1}\}$ are still indistinguishable, we say that Π is a multi-theorem NIZK system.

We define proof of knowledge which is a stronger property than soundness. Generally, a NIZK system is called NIZKPoK if we can efficiently recover the witness w from the valid proof output by the adversary. More formally, we say a non-interactive system is a proof of knowledge, if there exists a pair of PPT algorithms $(\mathsf{SimSetup}, \mathsf{Ext})$, such that $\mathsf{SimSetup}$ outputs a correctly generated $\widehat{\mathsf{crs}}$ together with an extraction key tk, and Ext can use tk to extract a valid witness from a proof.

Moreover, we consider two flavors for proof of knowledge: single-proof extractability and multi-theorem straight-line extractability.

Definition A.20 (Single-Theorem Extractability in [22]) *An* NIZK *proof system is single-proof extractable if there exists a* PPT *extractor* Ext*, constant* c_1*,* c_2*,* e *and a non-negligible polynomial* $p(\lambda)$ *such that for any* crs*, any* $x \in L_{\mathfrak{R}}$*, any* $Q = \mathsf{poly}(\lambda)$*, and* PPT *adversary* $\mathcal{A}$ *that makes at most* Q *random oracle queries with*

$$\Pr\left[\pi \xleftarrow{\$} \mathcal{A}(\mathsf{crs}, x) : \mathsf{Verify}(\mathsf{crs}, x, \pi) = 1\right] \geq \mu(\lambda),$$

then we have,

$$\Pr\left[w \xleftarrow{\$} \mathsf{Ext}^{\mathcal{A}}(\mathsf{crs}, x) : (x, w) \in \mathfrak{R}\right] \geq \frac{1}{p(\lambda \cdot Q^e)} \cdot \mu(\lambda)^{c_1} - \mathsf{negl}(\lambda),$$

where the runtime of Ext *is upper bounded by* $c_2 \cdot \mathsf{Time}(\mathcal{A})$ *and we assume one oracle access to* $\mathcal{A}$ *takes* $\mathsf{Time}(\mathcal{A})$*.*

Particularly, if we compile a sigma protocol with the Fiat-Shamir transform, then we have $(c_1, c_2, e) = (2, 2, 1)$ and $p(\lambda) = 1$ via rewinding the prover and the forking lemma [8, 53]. Additionally, we need to use a stronger extractability, i.e., multi-theorem straight-line extractability, where we can directly extract witnesses from multiple pairs of statement and proof output by the adversary. Moreover, for such multiple-theorem extractability, we allow the adversary to choose the queried statements adaptively.

Definition A.21 (Multi-Theorem Extractability in [22]) *An* NIZK *system is multi-theorem straight-line extractable, if there exists a* PPT *oracle simulator* $\mathsf{SimSetup}$ *and a* PPT *extractor* Ext *with the following properties:*

CRS indistinguishability. *For any* PPT *adversary* $\mathcal{A}$*, we have*

$$\begin{aligned}\mathsf{Adv}(\mathcal{A}) := \Big| &\Pr[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) : \mathcal{A}(\mathsf{crs}) = 1] \\ &- \Pr[(\widehat{\mathsf{crs}}, \mathsf{tk}) \leftarrow \mathsf{SimSetup}(1^\lambda) : \mathcal{A}(\widehat{\mathsf{crs}}) = 1]\Big| \leq \mathsf{negl}(\lambda).\end{aligned}$$

Straight-Line Extractability. *There exists constants* c, e_1, e_2 *and polynomial* $p(\lambda)$ *such that for any* $Q = \mathsf{poly}(\lambda)$ *and* PPT *adversary* $\mathcal{A}$ *that makes at most* Q *random oracle queries with*

$$\begin{aligned}\Pr\Big[&(\widehat{\mathsf{crs}}, \mathsf{tk}) \leftarrow \mathsf{SimSetup}(1^\lambda), \{(x_i, \pi_i)\}_{i \in [Q_s]} \leftarrow \mathcal{A}(\widehat{\mathsf{crs}}) : \\ &\forall i \in [Q_s], \mathsf{Verify}(\widehat{\mathsf{crs}}, x_i, \pi_i) = 1\Big] \geq \mu(\lambda),\end{aligned}$$

we have

$$\begin{aligned}\Pr\Big[&(\widehat{\mathsf{crs}}, \mathsf{tk}) \leftarrow \mathsf{SimSetup}(1^\lambda), \{(x_i, \pi_i)\}_{i \in [Q_s]} \leftarrow \mathcal{A}(\widehat{\mathsf{crs}}), \\ &\{w_i \leftarrow \mathsf{Ext}(1^\lambda, Q_H, Q_s, 1/\mu, \mathsf{tk}, x_i, \pi_i)\}_{i \in [Q_s]} : \\ &\forall i \in [Q_s], (x_i, \pi_i) \in \mathfrak{R} \wedge \mathsf{Verify}(\widehat{\mathsf{crs}}, x_i, \pi_i) = 1\Big] \\ &\geq \frac{1}{2} \cdot \mu(\lambda) - \mathsf{negl}(\lambda).\end{aligned}$$

Moreover, the running time of Ext *is upper bounded by* $Q_H^{e_1} \cdot Q_s^{e_2} \cdot \frac{1}{\mu^c} \cdot p(\lambda)$*.*

B Supplementary Material for Section 4

B.1 Correctness Proof for Construction 4.1

Lemma B.1 (Restatement of Lemma 4.2) *For parameters $N, q_2, \alpha, \gamma = \alpha\sqrt{2 \cdot 12 \cdot N}$, the* NIZKPoK *system Π for the relaxed language $L_{\gamma', q_2, \bar{\mathcal{C}}}$ with $\gamma' \geq (3.5\alpha N \cdot 2\sqrt{2} + \alpha\sqrt{2 \cdot 12 \cdot N})$, Construction 4.1 satisfies the correctness property as defined in Definition 3.1.*

Proof. The correctness according to Definition 3.1 requires to prove the following three statements: (1) four algorithms (Setup,Commit,Randomize, Combine) define a correct randomizable commitment scheme; (2) the signature by algorithm Sign passes the verification algorithm, i.e., Verify; and (3) the transferred signature (with respect to the randomized commitment) from Transfer also passes Verify.

Notice that, statement (1) follows naturally from the used BDLOP commitment scheme Γ. And statement (2) simply follows from the fact that SamplePre outputs a short vector of lattice $\Lambda_u^{\perp}(\mathbf{F}_{\mathsf{comm}})$ with an overwhelming probability, and thus the verification would pass. To show statement (3), it suffices to show that $\mathbf{F}_{\mathsf{comm}'} \cdot \mathsf{Sig}_{\mathsf{comm}'} = u$ (as defined in the algorithm Transfer) $\mathsf{Sig}_{\mathsf{comm}'}$ is within ℓ_2 norm $(3.5\alpha N \cdot 2\sqrt{2} + \alpha\sqrt{2 \cdot 12N})$, as the rest of the proof simply follows from the completeness of the NIZKPoK systems Π.

Particularly, for all $m \in \mathcal{M} \subseteq R_{q_2}$, $(\mathsf{sk}, \mathsf{pk})$ output by KeyGen, and signature $\mathsf{Sig}_{\mathsf{comm}} = (\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, \boldsymbol{s}_3^T) = ((\boldsymbol{s}_{1,1}^T, \boldsymbol{s}_{1,2}^T), \boldsymbol{s}_2^T, \boldsymbol{s}_3^T)$ output by Sign, it holds

$$\mathbf{F}_{\mathsf{comm}} \cdot \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 \end{bmatrix} = u \in \mathcal{R}_{q_2},$$

where $\mathbf{F}_{\mathsf{comm}} = \Big[[\boldsymbol{d}^T | \boldsymbol{a}^T] \big| [\boldsymbol{b}^T + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4})] \big| \boldsymbol{a}_2^T \Big]$. And the ℓ_2 norm of the vector $(\boldsymbol{s}_{1,1}^T, \boldsymbol{s}_{1,2}^T, \boldsymbol{s}_2^T, \boldsymbol{s}_3^T)$ is less than $\alpha\sqrt{2 \cdot 12N}$. This implies

$$\langle \boldsymbol{d}, \boldsymbol{s}_{1,1} \rangle + \langle \boldsymbol{a}, \boldsymbol{s}_{1,2} \rangle + \langle \boldsymbol{b} + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4})^T, \boldsymbol{s}_2 \rangle + \langle \boldsymbol{a}_2, \boldsymbol{s}_3 \rangle = u \in \mathcal{R}_{q_2}.$$

We notice that the above equation is equivalent to

$$\begin{aligned} u =& \langle \boldsymbol{d}, \boldsymbol{s}_{1,1} \rangle + \langle \boldsymbol{a}, \boldsymbol{s}_{1,2} \rangle + \langle \boldsymbol{b} + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4})^T, \boldsymbol{s}_2 \rangle \\ &+ \langle \boldsymbol{a}_2, \tilde{\mathbf{R}}_2 \cdot \boldsymbol{s}_2 \rangle - \langle \boldsymbol{a}_2, \tilde{\mathbf{R}}_2 \cdot \boldsymbol{s}_2 \rangle + \langle \boldsymbol{a}_2, \boldsymbol{s}_3 \rangle \\ =& \langle \boldsymbol{d}, \boldsymbol{s}_{1,1} \rangle + \langle \boldsymbol{a}, \boldsymbol{s}_{1,2} \rangle + \langle \boldsymbol{b} + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4})^T + \tilde{\mathbf{R}}_2 \cdot \boldsymbol{a}_2, \boldsymbol{s}_2 \rangle \\ &+ \langle \boldsymbol{a}_2, -\tilde{\mathbf{R}}_2 \cdot \boldsymbol{s}_2 \rangle + \langle \boldsymbol{a}_2, \boldsymbol{s}_3 \rangle, \end{aligned}$$

which can be rewritten as

$$\Big[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \big| \boldsymbol{b}^\top + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4}) + \boldsymbol{a}_2^\top \cdot \tilde{\mathbf{R}}_2 | \boldsymbol{a}_2^\top \Big] \cdot \begin{bmatrix} \boldsymbol{s}_{1,1} \\ \boldsymbol{s}_{1,2} \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 - \tilde{\mathbf{R}}_2 \cdot \boldsymbol{s}_2 \end{bmatrix} = u.$$

Here we denote $\tilde{\mathbf{R}} = \begin{bmatrix} \widetilde{\mathbf{R}}_1 \\ \widetilde{\mathbf{R}}_2 \end{bmatrix} = \left[\tilde{\boldsymbol{r}}_1, \tilde{\boldsymbol{r}}_2, \tilde{\boldsymbol{r}}_3, \tilde{\boldsymbol{r}}_4\right] \in R^{3\times 4}$, with $\widetilde{\mathbf{R}}_1 \in R^{1\times 4}$ and $\widetilde{\mathbf{R}}_2 \in R^{2\times 4}$.

Then we observe that

$$\begin{aligned}\mathbf{F}_{\mathsf{comm}'} :&= \left[[\boldsymbol{d}^\top|\boldsymbol{a}^\top]\middle|\boldsymbol{b}^\top + (\hat{t}_{2,1}, \hat{t}_{2,2}, \hat{t}_{2,3}, \hat{t}_{2,4})\middle|\boldsymbol{a}_2^\top\right] \\ &= \left[[\boldsymbol{d}^\top|\boldsymbol{a}^\top]\middle|\boldsymbol{b}^\top + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4}) + \boldsymbol{a}_2^\top \cdot \tilde{\mathbf{R}}_2|\boldsymbol{a}_2^\top\right],\end{aligned}$$

and $\mathsf{Sig}_{\mathsf{comm}'} := \begin{bmatrix} \boldsymbol{s}_{1,1} \\ \boldsymbol{s}_{1,2} \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 - \tilde{\mathbf{R}}_2 \cdot \boldsymbol{s}_2 \end{bmatrix}$. Now, it is easy to verify that the ℓ_2 norm of $\mathsf{Sig}_{\mathsf{comm}'}$ is within $(3.5\alpha N \cdot 2\sqrt{2} + \alpha\sqrt{2 \cdot 12N})$ and $\mathbf{F}_{\mathsf{comm}'} \cdot \mathsf{Sig}_{\mathsf{comm}'} = 0$, since for such a matrix $\widetilde{\mathbf{R}}_2 \in S_1^{2\times 4}$, its singular value $s_1(\widetilde{\mathbf{R}}_2)$ is bounded by $3.5\sqrt{N}$. This completes the proof. □

B.2 Simulatability Proof for Construction 4.1

Lemma B.2 (Restatement of Lemma 4.5) *The algorithm* Transfer *in Construction 4.1 is simulatable.*

Proof. According to Definition 3.2, we need to first construct a two-stage PPT simulator $\mathcal{S}$, and then prove that after running any polynomial $t = \mathsf{poly}(\lambda)$ times, the distribution of $\{\widetilde{\mathsf{Sig}}'_{\mathsf{comm}'_i}\}_{i\in[t]}$ output by $\mathcal{S}$ are statistically close to that of $\{\mathsf{Sig}'_{\mathsf{comm}'_i}\}_{i\in[t]}$ output by Transfer.

Particularly, the two-stage PPT simulator $\mathcal{S}$ can be constructed in the following way:

- First Stage: $\mathcal{S}$ conducts the following steps:
  1. Generate and output $\mathsf{params} := (\mathbf{A}, \boldsymbol{d}, \mathcal{M}, \mathcal{R}, \mathsf{crs})$.
- Second Stage: given params, and valid pk, comm', $\mathcal{S}$ conducts the following steps:
  1. Recognize pk as $(\boldsymbol{a}, \boldsymbol{b}, u)$.
  2. Parse $\mathsf{comm}' = (\mathsf{comm}'_i)_{i\in[4]}$ with $\mathsf{comm}'_i = \begin{bmatrix} \hat{t}_{1,i} \\ \hat{t}_{2,i} \end{bmatrix}$;
  3. Set matrix $\mathbf{F}'_{\mathsf{comm}'} := \left[[\boldsymbol{d}^\top|\boldsymbol{a}^\top]\middle|[\boldsymbol{b}^\top + (\hat{t}_{2,1}, \hat{t}_{2,2}, \hat{t}_{2,3}, \hat{t}_{2,4})]\middle|\boldsymbol{a}_2^\top\right]$.
  4. With respect to the NIZKPoK system Π for the relaxed language $L_{\gamma',q_2,\bar{\mathcal{C}}}$,

$$\begin{aligned}L_{\gamma',q_2,\bar{\mathcal{C}}} = \Big\{&(\mathbf{F}'_{\mathsf{comm}'}, u) \in R_q^{1\times 12} \times R_q : \exists\ \boldsymbol{x} \in R_q^{12} \text{ and} \\ &f \in \bar{\mathcal{C}} \text{ such that } \|\boldsymbol{x}\| \le \gamma' \text{ and } F'_{\mathsf{comm}'} \cdot \boldsymbol{x} = f \cdot u\Big\},\end{aligned}$$

  we can run the corresponding simulation algorithm to generate a simulated proof π', whose distribution is statistically indistinguishable from that of the real proof π.

5. Output $\widetilde{\mathsf{Sig}}'_{\mathsf{comm}'} := \pi'$.

According to the zero knowledge property of the used $\mathsf{NIZKPoK}$ system Π, it is clear that after running any polynomial $t = \mathsf{poly}(\lambda)$ times, the distribution of $\{\widetilde{\mathsf{Sig}}'_{\mathsf{comm}'_i}\}_{i\in[t]}$ output by $\mathcal{S}$ are statistically close to that of $\{\mathsf{Sig}'_{\mathsf{comm}'_i}\}_{i\in[t]}$ output by $\mathsf{Transfer}$. □

B.3 Unforgeability Proof for Construction 4.1

Lemma B.3 (Restatement of Lemma 4.6) *Assume that* $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu}$ *problem and* $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu'}$ *problem are hard with* $\nu = 22\alpha \cdot N$ *and* $\nu' = \frac{22\gamma'\sqrt{N}}{\sqrt{2\cdot 12}}$*, then our above lattice-based commitment-transferrable signature scheme is partially selectively unforgeable for the exact commitment relation* $\hat{L}_{q_1,q_2,\beta}$*, i.e., the advantage of any* PPT *adversary* $\mathcal{A}$ *against the partially selective unforgeability game of* CTS *is at most*

$$\mathsf{Adv}^{unforge}_{\mathcal{A}}(\lambda) \leq 2\mathsf{Adv}^{\mathsf{RLWE}}_{\mathcal{A}} + \mathsf{Adv}^{unforge^*}_{\mathcal{A}}(\lambda)$$

Proof. We argue the unforgeability using the series of hybrids.

$\mathsf{H_0}$: The challenger $\mathcal{B}$ runs the CTS honestly. He gives to the adversary $\mathcal{A}$ the public key pk and signatures with respect to the queried commitments comm_i. In this hybrid, we say $\mathcal{A}$ has advantage $\varepsilon = \mathsf{Adv}^{\mathrm{unforge}}_{\mathcal{A}}(\lambda)$ in the unforgeability game. Then, it holds

$$\mathsf{Adv}^{\mathsf{H}_0}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathrm{unforge}}_{\mathcal{A}}(\lambda).$$

$\mathsf{H_1}$: The challenger $\mathcal{B}$ runs the identical procedures as H_0, except that he samples $\mathbf{R}_0 \xleftarrow{\$} S_1^{2\times 4}$, and set $\boldsymbol{b}^\top = \boldsymbol{d}^\top \cdot \mathbf{R}_0 - (m^*, m^*\delta, m^*\delta^2, m^*\delta^3) \in R_{q_2}^{1\times 4}$. Here, we use m^* to denote the committed message in the challenge commitment comm^*. According to the RLWE assumption, we know that H_0 and H_1 are computational indistinguishability. Then, it holds

$$|\mathsf{Adv}^{\mathsf{H}_0}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{H}_1}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{RLWE}}_{\mathcal{A}}(\lambda).$$

$\mathsf{H_2}$: The challenger $\mathcal{B}$ runs the identical procedures as H_1, except that he samples $\boldsymbol{a} \xleftarrow{\$} R^4$, and $\mathcal{B}$ answers the signature queries through using Lemma A.7, rather than Lemma A.8. According to the RLWE assumption, we know that H_1 and H_2 are computational indistinguishability. Then, it holds

$$|\mathsf{Adv}^{\mathsf{H}_1}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{H}_2}_{\mathcal{A}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{RLWE}}_{\mathcal{A}}(\lambda).$$

Besides, we denote the challenger in H_2 as $\mathcal{B}^*$. Thus, we have

$$\mathsf{Adv}^{\mathsf{H}_2}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathrm{unforge}^*}_{\mathcal{A}}(\lambda).$$

Lemma B.4 *Let $\mathcal{A}$ be a* PPT *adversary with advantage ε in the selective unforgeability game with respect to $\mathcal{B}^*$ for the exact commitment relation $\hat{L}_{q_1,q_2,\beta}$, i.e., $\mathsf{Adv}_{\mathcal{A}}^{unforge^*}(\lambda) = \varepsilon$. Let h be a bound on the number of random oracle queries made by $\mathcal{A}$. Let $\nu = 22\alpha \cdot N$ and $\nu' = \frac{22\gamma'\sqrt{N}}{\sqrt{2 \cdot 12}}$. Then there exists a reduction algorithm $\mathcal{R}$ for $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu}$ or $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu'}$ such that*

$$\mathsf{Adv}_{\mathcal{R}}^{\mathsf{M\text{-}SIS}}(\lambda) \geq \varepsilon(\frac{\varepsilon}{h} - 2^{-\lambda}).$$

Proof. According to our construction, the verifier need to consider two cases: original signature and transferred signature. Thus, we need to prove the unforgeability for both cases. Overall, both of them have the similar proof process, and are based on the hardness of $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu}$ and $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu'}$ problems, respectively. Below, we present the details for both cases in an unified form, and just separate in their different points.

Particularly, we prove that if the adversary $\mathcal{A}$ can forge a valid original/transferred signature in the selective way, then we can construct an efficient reduction algorithm $\mathcal{B}$ to solve the $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu}$/$\mathsf{M\text{-}SIS}_{q_2,1,9,\nu'}$ problem. In particular, $\mathcal{B}$ is given an uniformly random matrix $\boldsymbol{x}^\top = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9] \in R_{q_2}^9$, and need to output a vector $\boldsymbol{y}$ such that $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0 \bmod q_2$ and $\|\boldsymbol{y}\| \leq \nu = 22\alpha \cdot N$ or $\|\boldsymbol{y}\| \leq \nu' = \frac{22\gamma'\sqrt{N}}{\sqrt{2 \cdot 12}}$. Similar to the consideration in [23], we choose to use $\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, 1, x_8]$, since one of x_i will have an inverse with high probability.

In this case, $\mathcal{B}$ conducts the following steps:

1. Choose $\boldsymbol{x}_1' \xleftarrow{\$} R_{q_1}^2$ and set $\boldsymbol{a}_1^T = (1, \boldsymbol{x}_1'^\top) \in R_{q_1}^3$.
2. Set $\boldsymbol{a}_2^\top = (1, x_8) \in R_{q_2}^2$.
3. Set $\mathbf{A} = \begin{bmatrix} \boldsymbol{a}_1^\top \\ 0, \boldsymbol{a}_2^\top \end{bmatrix}$ and send it to $\mathcal{A}$.

Clearly, $\mathbf{A}$ is a valid public parameter output by $\Gamma.\mathsf{CKeyGen}$.

Next, we need to argue that $\mathcal{B}$ can simulate the environment of $\mathcal{A}$ successfully for the exact commitment relation $\hat{L}_{q_1,q_2}$. In particular, we use the following Claim B.5 to specify the case.

Claim B.5 *$\mathcal{B}$ can simulate the environment of $\mathcal{A}$ successfully in the unforgeability game with respect to the exact commitment relation $\hat{L}_{q_1,q_2}$.*

Proof. With this $\mathbf{A}$, according to Remark 3.6 of Definition 3.3, $\mathcal{A}$ can commit to the challenge message m^* at the beginning of unforgeability game.

Then $\mathcal{B}$ can set the public parameters in the following way:

1. Set $\boldsymbol{d}^\top = (x_1, x_2) \in R_{q_2}^2$, $\boldsymbol{a}^\top = (x_3, x_4, x_5, x_6) \in R_{q_2}^4$, $u = x_7 \in R_{x_2}$.
2. Sample $\mathbf{R}_0 \xleftarrow{\$} S_1^{2\times 4}$, and set

$$\boldsymbol{b}^\top = \boldsymbol{d}^\top \cdot \mathbf{R}_0 - (m^*, m^*\delta, m^*\delta^2, m^*\delta^3) \in R_{q_2}^{1\times 4};$$

3. Send $\mathsf{pk} := (\boldsymbol{a}, \boldsymbol{b}, u)$ to $\mathcal{A}$.

According to the uniformity of x_3, x_4, x_5, x_6 and the distribution of $\mathbf{R}_0$, pk is a valid public key of our commit-transferrable signature, which follows from the Ring-LWE assumption.

Then, the $\mathcal{A}$ can conduct signature queries and get responds from $\mathcal{B}$. In particular, after receiving the signature query $(\mathsf{comm}, m, (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4))$ from $\mathcal{A}$, where $\mathsf{comm} = (\mathsf{comm}_1, \mathsf{comm}_2, \mathsf{comm}_3, \mathsf{comm}_4)$ and

$$\mathsf{comm}_1 := \begin{bmatrix} t_{1,1} \\ t_{2,1} \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_1 + \begin{bmatrix} 0 \\ m \end{bmatrix},$$

$$\mathsf{comm}_2 := \begin{bmatrix} t_{1,2} \\ t_{2,2} \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_2 + \begin{bmatrix} 0 \\ m\delta \end{bmatrix},$$

$$\mathsf{comm}_3 := \begin{bmatrix} t_{1,3} \\ t_{2,3} \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_3 + \begin{bmatrix} 0 \\ m\delta^2 \end{bmatrix},$$

$$\mathsf{comm}_4 := \begin{bmatrix} t_{1,4} \\ t_{2,4} \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_4 + \begin{bmatrix} 0 \\ m\delta^3 \end{bmatrix}.$$

$\mathcal{B}$ can compute

$$\begin{aligned}
\mathbf{F}_{\mathsf{comm}} &= \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{b} + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4}) \middle| \boldsymbol{a}_2 \right] \\
&= \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{d}^\top \cdot \mathbf{R}_0 - (m^*, m^*\delta, m^*\delta^2, m^*\delta^3) + (t_{2,1}, t_{2,2}, t_{2,3}, t_{2,4}) \middle| \boldsymbol{a}_2 \right] \\
&= \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{d}^\top \cdot \mathbf{R}_0 + \boldsymbol{a}_2^\top \cdot \mathbf{R}_2 + (m - m^*)(1, \delta, \delta^2, \delta^3) \middle| \boldsymbol{a}_2 \right],
\end{aligned}$$

where we denote $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} = \left[\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4 \right] \in R^{3\times 4}$ with $\mathbf{R}_2 \in R^{2\times 4}$.

For any $m \neq m^*$, we know that $m - m^*$ is invertible over the subfield S_{q_2} of ring R_{q_2}. According to the algorithm in Lemma A.7, the challenger can get a short vector $\boldsymbol{z} \in R^{12}$ such that $\mathbf{F}_{\mathsf{comm}} \cdot \boldsymbol{z} = u$. Notice that for $\mathbf{R} \in S_1^{3\times 4}$, it holds $s_1(\mathbf{R}) \leq 3.8\sqrt{N}$, except with at most a negligible probability. □

From above Claim B.5, we know that $\mathcal{B}$ can simulate the environment of $\mathcal{A}$ successfully.

Next, for the challenge query, the adversary sends randomness $(\boldsymbol{r}_1^*, \boldsymbol{r}_2^*, \boldsymbol{r}_3^*, \boldsymbol{r}_4^*)$ to the challenger, such that the final challenge query is of the form $(\mathsf{comm}^*, m^*, (\boldsymbol{r}_1^*, \boldsymbol{r}_2^*, \boldsymbol{r}_3^*, \boldsymbol{r}_4^*))$, where $\mathsf{comm}^* = (\mathsf{comm}_1^*, \mathsf{comm}_2^*, \mathsf{comm}_3^*, \mathsf{comm}_4^*)$ and

$$\mathsf{comm}_1^* := \begin{bmatrix} t_{1,1}^* \\ t_{2,1}^* \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_1^* + \begin{bmatrix} 0 \\ m^* \end{bmatrix},$$

$$\mathsf{comm}_2^* := \begin{bmatrix} t_{1,2}^* \\ t_{2,2}^* \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_2^* + \begin{bmatrix} 0 \\ m^*\delta \end{bmatrix}.$$

$$\mathsf{comm}_3^* := \begin{bmatrix} t_{1,3}^* \\ t_{2,3}^* \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_3^* + \begin{bmatrix} 0 \\ m^*\delta^2 \end{bmatrix},$$

$$\mathsf{comm}_4^* := \begin{bmatrix} t_{1,4}^* \\ t_{2,4}^* \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_4^* + \begin{bmatrix} 0 \\ m^* \delta^3 \end{bmatrix}.$$

In this case, we have

$$\mathbf{F}_{\mathsf{comm}^*} = \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{d}^\top \cdot \mathbf{R}_0 + \boldsymbol{a}_2^\top \cdot \mathbf{R}_2^* \middle| \boldsymbol{a}_2^\top \right].$$

Below, according to the fact that the adversary's forgery is for original signature or transferred one, we need to separate the following proof into two cases.

For the case of original one. If the adversary can forge a valid signature $\mathsf{Sig}_{\mathsf{comm}^*} := \begin{bmatrix} \boldsymbol{s}_{1,1}^* \\ \boldsymbol{s}_{1,2}^* \\ \boldsymbol{s}_2^* \\ \boldsymbol{s}_3^* \end{bmatrix}$ with $\boldsymbol{s}_3^* = (s_{3,1}^*, s_{3,2}^*)^T \in R^2$, such that

$$\begin{aligned} \mathbf{F}_{\mathsf{comm}^*} \cdot \mathsf{Sig}_{\mathsf{comm}^*} &= \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{d}^\top \cdot \mathbf{R}_0 + \boldsymbol{a}_2^\top \cdot \mathbf{R}_2^* \middle| \boldsymbol{a}_2 \right] \cdot \begin{bmatrix} \boldsymbol{s}_{1,1}^* \\ \boldsymbol{s}_{1,2}^* \\ \boldsymbol{s}_2^* \\ \boldsymbol{s}_3^* \end{bmatrix} \\ &= \langle \boldsymbol{d}, \boldsymbol{s}_{1,1}^* \rangle + \langle \boldsymbol{a}, \boldsymbol{s}_{1,2}^* \rangle + \langle \boldsymbol{d}^\top \cdot \mathbf{R}_0 + \boldsymbol{a}_2^\top \cdot \mathbf{R}_2^*, \boldsymbol{s}_2^* \rangle + \langle \boldsymbol{a}_2, \boldsymbol{s}_3^* \rangle \\ &= u, \end{aligned}$$

then $\mathcal{B}$ can compute $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{s}_{1,1}^* + \mathbf{R}_0 \cdot \boldsymbol{s}_2^* \\ \boldsymbol{s}_{1,2}^* \\ -1 \\ \mathbf{R}_2^* \cdot \boldsymbol{s}_2^* + \boldsymbol{s}_3^* \end{bmatrix}$ as a solution to the $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu}$ problem defined by $[x_1, x_2, x_3, x_4, x_5, x_6, x_7, 1, x_8]$. And the ℓ_2 norm of this solution is less than $\|\boldsymbol{y}\| \leq \alpha\sqrt{2 \cdot 8N} + 14\sqrt{2}\alpha \cdot N + 1 \leq \alpha \cdot (4\sqrt{N} + 14\sqrt{2}N) + 1 \leq 22\alpha \cdot N$.

For the case of transferred one. If the adversary can forge a valid proof for the language $L_{\gamma',q_2,\bar{\mathcal{C}}}$, then the reduction algorithm $\mathcal{B}$ can run the extractor of the $\mathsf{NIZKPoK}$ system Π_2, and get a ℓ_2 norm short vector $\mathsf{Sig}'_{\mathsf{comm}^*} := \begin{bmatrix} \boldsymbol{s}_{1,1}^* \\ \boldsymbol{s}_{1,2}^* \\ \boldsymbol{s}_2^* \\ \boldsymbol{s}_3^* \end{bmatrix}$ with $\boldsymbol{s}_3^* = (s_{3,1}^*, s_{3,2}^*)^\top \in R^2$, such that

$$\begin{aligned} \mathbf{F}_{\mathsf{comm}^*} \cdot \mathsf{Sig}_{\mathsf{comm}^*} &= \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{d}^\top \cdot \mathbf{R}_0 + \boldsymbol{a}_2^\top \cdot \mathbf{R}_2^* \middle| \boldsymbol{a}_2 \right] \cdot \begin{bmatrix} \boldsymbol{s}_{1,1}^* \\ \boldsymbol{s}_{1,2}^* \\ \boldsymbol{s}_2^* \\ \boldsymbol{s}_3^* \end{bmatrix} \\ &= \langle \boldsymbol{d}, \boldsymbol{s}_{1,1}^* \rangle + \langle \boldsymbol{a}, \boldsymbol{s}_{1,2}^* \rangle + \langle \boldsymbol{d}^\top \cdot \mathbf{R}_0 + \boldsymbol{a}_2^\top \cdot \mathbf{R}_2^*, \boldsymbol{s}_2^* \rangle \\ &\quad + \langle \boldsymbol{a}_2, \boldsymbol{s}_3^* \rangle \\ &= \bar{c} \cdot u, \end{aligned}$$

then $\mathcal{B}$ can compute $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{s}^*_{1,1} + \mathbf{R}_0 \cdot \boldsymbol{s}^*_2 \\ \boldsymbol{s}^*_{1,2} \\ -\bar{c} \\ \mathbf{R}^*_2 \cdot \boldsymbol{s}^*_2 + \boldsymbol{s}^*_3 \end{bmatrix}$ as a solution to the $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu'}$ problem defined by $[x_1, x_2, x_3, x_4, x_5, x_6, x_7, 1, x_8]$. And the ℓ_2 norm of this solution is less than $\|\boldsymbol{y}\| \leq \alpha'\sqrt{2 \cdot 8N} + 14\sqrt{2}\alpha' \cdot N + 2\sqrt{\kappa} \leq \alpha' \cdot (4\sqrt{N} + 14\sqrt{2}N) + 2\sqrt{\kappa} \leq 22\alpha' \cdot N$, with $\alpha' = \gamma'/\sqrt{2 \cdot 12N}$.

Furthermore, according to the forking lemma of [8, 53], $\mathcal{R}$ can complete the above reduction with probability at least $\varepsilon(\frac{\varepsilon}{h} - 2^{-\lambda})$.

Summing up all above arguments, we conclude that our commit transferrable signature satisfies unforgeable in the selective way. □

C Supplementary Materials for Section 5 □

In this section, we present how to instantiate the three building blocks in Section 5 efficiently from lattices. Before this, we need to introduce the additional parameters for the straight-line extractable $\mathsf{NIZKPoK}$ in Table 10, as other common parameters also used for CTS have been introduced in Table 5.

Param.	Description
q_{zk}	Moduli used for BDLOP commitment scheme in $\Pi^{(1)}$
τ	Parameters to describe the committed value space in $\Pi^{(1)}$
$n, k, \hat{\lambda}$	Parameters for the encryption scheme E
$\hat{k}$	Parameter for the repetition times in $\Pi_{\mathsf{Disclosure}}$
η, M	Parameters for rejection sampling algorithm
γ'_1, γ'_2	ℓ_2 norm parameters for "short" vectors in the language of $\hat{L}_{\Pi^{(1)}}$
γ'_3	ℓ_2 norm parameters for "short" vectors in the language of $\hat{L}_{\Pi^{(2)}}$

Table 10. Additional Parameters of Multi-theorem Straight-line extractable $\mathsf{NIZKPoK}$

Concrete Construction of NIZK $\Pi^{(1)}$

According to the BDLOP commitment scheme and our definition experiment of unforgeability for CTS, the relaxed language $\hat{L}_{\Pi^{(1)}}$ can be concretely written through using $L_{\gamma'_1,\gamma'_2,q_1,q_2,\bar{\mathcal{C}}}$ and $L_{\gamma'_1,q_1,q_2,\bar{\mathcal{C}}}$ in the following way:

$$
\begin{aligned}
\hat{L}_{\Pi^{(1)}} :=& \Big\{ \mathsf{comm} = (\mathsf{comm}_i)_{i\in[4]} : \exists m, q_1, q_2, \{\boldsymbol{r}_i\}_{i\in[4]}, f \in \bar{\mathcal{C}}, \text{ such that } 0 < \|\boldsymbol{r}_i\| \leq \gamma'_1, \\
& 0 < \|m\| \leq \gamma'_2, \text{ and } \mathsf{comm}_i = \mathsf{Commit}(\mathsf{params}, m \cdot q_2^{\frac{i-1}{4}}, \boldsymbol{r}_i/f) \text{ for } i \in [4] \Big\} \\
=& \Big\{ (\mathbf{A}_1, \mathbf{A}_2, \mathsf{comm}_i = (\boldsymbol{t}_{i,1}, \boldsymbol{t}_{i,2})) : \exists\ (\boldsymbol{r}_i, m) \text{ and } f \in \bar{\mathcal{C}} \text{ such that } 0 < \|\boldsymbol{r}_i\| \leq \gamma'_1, \\
& 0 < \|m\| \leq \gamma'_2, \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{r}_1 \\ m \end{bmatrix} = f \cdot \begin{bmatrix} \boldsymbol{t}_{1,1} \\ \boldsymbol{t}_{1,2} \end{bmatrix} \text{ and } \\
& \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \cdot \boldsymbol{r}_i + f \cdot \begin{bmatrix} \mathbf{0} \\ m \cdot q_2^{\frac{i-1}{4}} \end{bmatrix} = f \cdot \begin{bmatrix} \boldsymbol{t}_{i,1} \\ \boldsymbol{t}_{i,2} \end{bmatrix} \text{ for } i \in \{2,3,4\} \Big\},
\end{aligned}
$$

where $\mathbf{A}_1 = (1, \boldsymbol{a}_1^\top)$ and $\mathbf{A}_2 = (0, \boldsymbol{a}_2^\top)$.

Interactive proof system $\Pi^{(1)}$

Public Parameter for Commitment Scheme:

$\mathbf{A} = \begin{bmatrix} \boldsymbol{a}_1^\top \\ \boldsymbol{a}_2^\top \end{bmatrix} = \begin{bmatrix} 1, & \boldsymbol{a}_1'^\top \\ 0, 1, & a_2' \end{bmatrix}$ as in Construction 4.1, $\delta = \lfloor q_2^{\frac{1}{4}} \rceil, \sigma_{-1}, \sigma_5, \tau | N$,
$B_1 = \xi \cdot \sqrt{6N}$, $B_2 = \xi' \cdot \sqrt{2N}$.
Prover's Witness: $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4 \in S_1^3$, $m = \sum_{i \in [N]} m_{i-1} X^{i-1}$ is in a subfield of R_{q_2},
Commitment: $\mathsf{comm} = (\mathsf{comm}_i)_{i \in [4]}$, $\mathsf{comm}_i := \begin{bmatrix} t_{1,i} \\ t_{2,i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1^\top \\ \boldsymbol{a}_2^\top \end{bmatrix} \cdot \boldsymbol{r}_i + \begin{bmatrix} 0 \\ m \cdot \delta^{i-1} \end{bmatrix}$.

Prover		Verifier
$\forall i \in [4], \boldsymbol{y}_i \leftarrow \mathcal{D}_\xi^3$, $y_{1,2} \leftarrow \mathcal{D}_{\xi'}$, $\boldsymbol{y}_{-1}, \boldsymbol{y}_5 \leftarrow \mathcal{D}_\xi^3$		
$\forall i \in [4], w_{i,1} = \boldsymbol{a}_1^\top \cdot \boldsymbol{y}_i$, $w_{1,2} = \boldsymbol{a}_1^\top \cdot \boldsymbol{y}_1 + y_{1,2}$		
$\forall i \in [3], w'_{i,2} = \delta \cdot \boldsymbol{a}_2^\top \cdot \boldsymbol{y}_i - \boldsymbol{a}_2^\top \cdot \boldsymbol{y}_{i+1}$		
$w_{1,-1} = \sigma_{-1}(\boldsymbol{a}_1^\top) \cdot \boldsymbol{y}_{-1}$, $w_{1,5} = \sigma_5^\tau(\boldsymbol{a}_1^\top) \cdot \boldsymbol{y}_5$		
$w_{2,-1} = \boldsymbol{a}_2^\top \cdot \boldsymbol{y}_1 - \sigma_{-1}(\boldsymbol{a}_2^\top) \cdot \boldsymbol{y}_{-1}$		
$w_{2,5} = \boldsymbol{a}_2^\top \cdot \boldsymbol{y}_1 - \sigma_5^\tau(\boldsymbol{a}_2^\top) \cdot \boldsymbol{y}_5$		
	$\xrightarrow{w_{i,1}, w_{1,2}, w'_{i,2}, w_{1,-1}, w_{1,5}, w_{2,-1}, w_{2,5}}$	
		$d \xleftarrow{\$} \mathcal{C}$
	$\xleftarrow{d}$	
$\forall i \in [4], \boldsymbol{z}_{i,1} = \boldsymbol{y}_i + d \cdot \boldsymbol{r}_i$		
$\boldsymbol{z}_{1,2} = \boldsymbol{y}_{1,2} + d \cdot m$		
$\boldsymbol{z}_{-1} = \boldsymbol{y}_{-1} + d \cdot \sigma_{-1}(\boldsymbol{r}_1)$		
$\boldsymbol{z}_5 = \boldsymbol{y}_5 + d \cdot \sigma_5^\tau(\boldsymbol{r}_1)$		
$\mathsf{Rej}(\boldsymbol{z}_{i,1}, d \cdot \boldsymbol{r}_i, \xi)$		
$\mathsf{Rej}(\boldsymbol{z}_{1,2}, d \cdot m, \xi')$		
$\mathsf{Rej}(\boldsymbol{z}_{-1}, d \cdot \sigma_{-1}(\boldsymbol{r}_1), \xi)$		
$\mathsf{Rej}(\boldsymbol{z}_5, d \cdot \sigma_5^\tau(\boldsymbol{r}_1), \xi)$	$\xrightarrow{\boldsymbol{z}_{i,1}, \boldsymbol{z}_{1,2}, \boldsymbol{z}_{-1}, \boldsymbol{z}_5}$	
		Check:
		1. for $i \in [4], \lVert \boldsymbol{z}_{i,1} \rVert \overset{?}{\leq} B_1, \lVert \boldsymbol{z}_{1,2} \rVert \overset{?}{\leq} B_2$, $\lVert z_{-1} \rVert \overset{?}{\leq} B_1, \lVert z_5 \rVert \overset{?}{\leq} B_1$
		2. for $i \in [4], \boldsymbol{a}_1^\top \cdot \boldsymbol{z}_{i,1} \overset{?}{=} w_{i,1} + d \cdot t_{1,i}$
		3. $\boldsymbol{a}_1^\top \cdot \boldsymbol{z}_{1,1} + \boldsymbol{z}_{1,2} \overset{?}{=} w_{1,2} + d \cdot t_{2,1}$
		4. $\sigma_{-1}(\boldsymbol{a}_1^\top) \cdot \boldsymbol{z}_{-1} \overset{?}{=} w_{1,-1} + d \cdot \sigma_{-1}(t_{1,1})$
		5. $\sigma_5^\tau(\boldsymbol{a}_1^\top) \cdot \boldsymbol{z}_5 \overset{?}{=} w_{1,5} + d \cdot \sigma_5^\tau(t_{1,1})$
		6. for $i \in [3]$: $\boldsymbol{a}_2^\top \cdot (\delta \cdot \boldsymbol{z}_{i,1} - \boldsymbol{z}_{i+1,1}) \overset{?}{=} w'_{i,2} + d \cdot (\delta \cdot t_{2,i} - t_{2,i+1})$
		7. $\boldsymbol{a}_2^\top \cdot \boldsymbol{z}_{1,1} - \sigma_{-1}(\boldsymbol{a}_2^\top) \cdot \boldsymbol{z}_{-1} \overset{?}{=} w_{2,-1} + d \cdot (t_{2,1} - \sigma_{-1}(t_2))$
		8. $\boldsymbol{a}_2^\top \cdot \boldsymbol{z}_{1,1} - \sigma_5^\tau(\boldsymbol{a}_2^\top) \cdot \boldsymbol{z}_5 \overset{?}{=} w_{2,5} + d \cdot (t_{2,1} - \sigma_5^\tau(t_2))$
		Accept if all the above conditions hold.

Table 11. The interactive version of $\Pi^{(1)}$.

For $\hat{L}_{\Pi^{(1)}}$ and the requirement that the committed message $\boldsymbol{m}$ is included in a subfield of R_{q_2} consisting of q_2^{τ} elements with $\tau | N$, the interactive zero knowledge proof system is presented in Table 11. Based on this, the corresponding non-interactive one can be obtained through standard Fiat-Shamir transformation in the random oracle model.

More formally, we have the following Theorem C.1.

Theorem C.1 *In the random oracle model, let N to be a power of 2, prime $q_1 = 5 \bmod 8$, prime $q_2 = 3$ or $5 \bmod 8$, there exists a multi-theorem* NIZK *system $\Pi^{(1)}$ for $\hat{L}_{\Pi^{(1)}}$ (i.e., (1) comm_1 are valid in the relation $L_{\gamma_1', \gamma_2', q_1, q_2, \bar{\mathcal{C}}}$, (2) $\mathsf{comm}_2, \mathsf{comm}_3, \mathsf{comm}_4$ are valid in the relation $L_{\gamma_1', q_1, q_2, \bar{\mathcal{C}}}$, (3) all comm_i satisfy the corresponding linear relationship), and (4) the committed message m in comm is included in a subfield of R_{q_2}, where $\gamma_1' = 6\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot N$ and $\gamma_2' = 2\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot N$, where η is the parameter for rejection sampling as in Lemma A.9.*

The proof of Theorem C.1 is similar to that of [6, 23], and thus we omit it for simplicity.

Concrete Construction of E and NIZK $\Pi^{(2)}$ for Ciphertext Validity

Second, for the encryption scheme, we choose to use the following variant of BDLOP commitment scheme.

Construction C.2 (Encryption Scheme E) *The scheme is as follows.*

- $\mathsf{KeyGen}(\lambda)$*: Given a security parameter λ, the algorithm conducts the following steps:*
  1. *Choose two integers N, q_1, where N is a power of 2, and q_1 is a prime with $q_1 = 5 \mod 8$;*
  2. *Set $n, k, \hat{\lambda}$ be integers satisfying $k = n + 6 + \hat{\lambda}$.*
  3. *For the ring $R = \mathbb{Z}[X]/(X^N + 1)$, and let $R_{q_1} = \mathbb{Z}_{q_1}[X]/(X^N + 1)$, χ be an error distribution over R.*
  4. *Sample $\mathbf{A} \xleftarrow{\$} R_{q_1}^{n \times k}$, $\boldsymbol{s}_i \leftarrow S_1^n, \boldsymbol{e}_i \leftarrow S_1^k$ for $i \in [6]$, where $\boldsymbol{s}_i, \boldsymbol{e}_i$ are vectors over R_{q_1}.*
  5. *Compute $\boldsymbol{b}_i = \mathbf{A}^{\top} \cdot \boldsymbol{s}_i + \boldsymbol{e}_i (\mathrm{mod} q_1)$.*
  6. *Output $\mathsf{pk} := (\mathbf{A}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_6)$, $\mathsf{sk} := \{\boldsymbol{s}_i\}_{i \in [6]}$.*
- $\mathsf{Enc}(\mathsf{pk}, \boldsymbol{r})$*: Given public key pk and the message vector $\boldsymbol{r} = (r_1, r_2, r_3)^{\top} \in R_{q_1}^3$, where each coefficient of r_i is from $\{-1, 0, 1\}$, the algorithm conducts the following steps:*
  1. *Sample $\hat{\boldsymbol{r}} \xleftarrow{\$} S_1^k$.*
  2. *Compute $\boldsymbol{t}_0 = \mathbf{A} \cdot \hat{\boldsymbol{r}} \in R_{q_1}^n$, $t_i = \langle \boldsymbol{b}_i, \hat{\boldsymbol{r}} \rangle + r_i \in R_{q_1}$, $t_{3+i} = \langle \boldsymbol{b}_{3+i}, \hat{\boldsymbol{r}} \rangle + r_i \cdot \lfloor \sqrt{q_1} \rceil \in \hat{R}_{q_1}$, for $i \in [3]$.*
  3. *Output $\mathsf{ct} = (\boldsymbol{t}_0, t_1, \ldots, t_6)$.*
- $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, \mathsf{ct})$*: Given public key pk, secret key sk and the ciphertext $\mathsf{ct} = (\boldsymbol{t}_0, t_1, \ldots, t_6)$, where $\boldsymbol{t}_0 \in R_{q_1}^n$, $t_i \in R_{q_1}$, the algorithm conducts the following steps:*

1. *Compute* $u_i = t_i - \langle \boldsymbol{t}_0, \boldsymbol{s}_i \rangle$ *and* $u_{i+3} = t_{i+3} - \langle \boldsymbol{t}_0, \boldsymbol{s}_{i+3} \rangle$, *for* $i \in [3]$.
2. $\Delta_{2,i} = u_{i+3} - u_i \cdot \lfloor \sqrt{q_1} \rceil (\bmod \lfloor \sqrt{q_1} \rceil)$, *for* $i \in [3]$.
3. $r_i = \frac{u_{i+3} - \Delta_{2,i}}{\lfloor \sqrt{q_1} \rceil}$, *for* $i \in [3]$.
4. *Compute* $\boldsymbol{r} = (r_1 \| \cdots \| r_3 \in R_{q_1}^3)^\top$.

For simplicity of presentation, we defer the correctness and security proof to Section C.1. Below, we just focus on the validness of the ciphertext in Construction C.2, which is necessary for the general framework in Section 5.2.

Validity of the Ciphertext of Construction C.2. Viewing Construction C.2 as a general BDLOP commitment in R_{q_1}, the validity of ciphertext can be proven via the opening proof of BDLOP commitments for $L_{\gamma_1', q_1, q_1, \bar{\mathcal{C}}}$ as in [6, 23],[13] together with linear relations among them, and the ℓ_2-norm of the committed polynomial is bounded by $\sqrt{3N}$. Thus, the relaxed language $\hat{L}_{\Pi^{(2)}}$ can be concretely written through using $L_{\gamma_1', q_1, q_1, \bar{\mathcal{C}}}$ in the following way:

$$
\begin{aligned}
\hat{L}_{\Pi^{(2)}} := & \Big\{ \mathsf{ct} = (\mathsf{ct}_i)_{i \in [4]} : \exists (\boldsymbol{r}_i, \mathsf{Rand}_i) \text{ and } f \in \bar{\mathcal{C}}, \text{ such that } \boldsymbol{r}_i \in S_1^3, \\
& \quad 0 < \|\mathsf{Rand}_i\| \le \gamma_3', \text{ and } \mathsf{ct}_i = \mathsf{E.Enc}(\mathsf{pk}, \boldsymbol{r}_i; \mathsf{Rand}_i / f), \text{ for } i \in [4] \Big\} \\
= & \Big\{ (\mathbf{A}, \mathbf{B}^\top, \widehat{\mathsf{comm}} = (\widehat{\mathsf{comm}}_i)_{i \in [4]}) : \exists\ (\boldsymbol{r}_i, \hat{\boldsymbol{r}}) \text{ and } f \in \bar{\mathcal{C}}' \text{ such that} \\
& \quad 0 < \|\hat{\boldsymbol{r}}\| \le \gamma_3', 0 < \|\boldsymbol{r}_i\| \le \sqrt{3N}, \boldsymbol{r}_i' = \lfloor \sqrt{q_1} \rceil \cdot \boldsymbol{r}_i, \widehat{\mathsf{comm}}_i = (\hat{\boldsymbol{t}}_{1,i}, \hat{\boldsymbol{t}}_{2,i}), \\
& \quad \begin{bmatrix} \mathbf{A} \\ \mathbf{B}^\top \end{bmatrix} \cdot \hat{\boldsymbol{r}} + f \cdot \begin{bmatrix} \mathbf{0} \\ \begin{pmatrix} \boldsymbol{r}_i \\ \boldsymbol{r}_i' \end{pmatrix} \end{bmatrix} = f \cdot \begin{bmatrix} \hat{\boldsymbol{t}}_{1,i} \\ \hat{\boldsymbol{t}}_{2,i} \end{bmatrix}, \text{ for } i \in [4] \Big\},
\end{aligned}
$$

where $\mathbf{B} = [b_1, \ldots, b_6]$.

For the tight upper bound on ℓ_2-norm of $\boldsymbol{r}_i$, we further view the above encryption scheme as the BDLOP part of an ABDLOP commitment, and directly use the related techniques and protocol in [2, 44]. Here, we omit the detailed protocol for simplicity. More formally, we have the following Theorem C.3.

Theorem C.3 *In the random oracle model, let N to a power of 2 and q_1 be prime with $q_1 = 5 \mod 8$, there exists a multi-theorem* NIZK *system $\Pi^{(2)}$ for $\hat{L}_{\Pi^{(2)}}$, i.e., (1)* ct *are valid ciphertexts; (2) the ℓ_2 norm of the encrypted polynomial $\boldsymbol{r}_i$ is bounded by $\sqrt{3N}$, with $\gamma_3' = 2\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot k \cdot N$, where η is the parameter for rejection sampling as in Lemma A.9.*

Concrete construction of NIZK system $\Pi^{(3)}$

Third, in order to prove the consistency of the witness of $\boldsymbol{r} = (r_1, r_2, r_3)^\top \in S_1^3$ in $\mathsf{comm} := \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r} + \begin{bmatrix} 0 \\ m \end{bmatrix}$ with $\mathbf{A} := \begin{bmatrix} 1, \boldsymbol{a}_1^\top \\ 0, \boldsymbol{a}_2^\top \end{bmatrix}$, and the encryption of

[13] Here, this general BDLOP commitment scheme just use one modulus q_1, rather than two different moduli. So we use $L_{\gamma_1', q_1, q_1, \bar{\mathcal{C}}}$ to represent its relaxed opening relation. Below, we will use $\widehat{\mathsf{comm}}$ to denote its concrete commitment value, in order to distinguish it from that of the prior commitment in CTS.

$\boldsymbol{r}$ in $\mathsf{ct} := \begin{bmatrix} \hat{\mathbf{A}} \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \boldsymbol{b}_5 \\ \boldsymbol{b}_6 \end{bmatrix} \cdot \hat{\boldsymbol{r}} + \begin{bmatrix} 0 \\ r_1 \\ r_2 \\ r_3 \\ \lfloor\sqrt{q}\rceil \cdot r_1 \\ \lfloor\sqrt{q}\rceil \cdot r_2 \\ \lfloor\sqrt{q}\rceil \cdot r_3 \end{bmatrix}$, we just need to prove the committed value $\boldsymbol{r} = (r_1, r_2, r_3)^\top \in R_{q_1}$ in ct satisfies the exact linear relationship $[1, \boldsymbol{a}_1{}^\top] \cdot \boldsymbol{r} = t_1$.

Thus the language $\hat{L}_{\Pi^{(3)}}$ can be concretely written through using $\hat{L}_{\Pi^{(2)}}$ in the following way:

$$\begin{aligned}
\hat{L}_{\Pi^{(3)}} = \Big\{ & (\mathsf{comm} = (\mathsf{comm}_i)_{i\in[4]}, \mathsf{ct} = (\mathsf{ct}_i)_{i\in[4]}) : \exists (m, \boldsymbol{r}_i, \mathsf{Rand}_i) \text{ and } f \in \bar{\mathcal{C}}', \text{ such} \\
& \text{that } \boldsymbol{r}_i \in S_1^3, 0 < \|\mathsf{Rand}_i\| \le \gamma_3', \mathsf{comm}_i = \mathsf{Commit}(\mathsf{params}, m \cdot q_2^{\frac{i-1}{4}}; \boldsymbol{r}_i) \\
& \text{and } \mathsf{ct}_i = \mathsf{E.Enc}(\mathsf{pk}, \boldsymbol{r}_i; \mathsf{Rand}_i / f) \text{ for } i \in [4] \Big\} \\
= \Big\{ & ((1, \boldsymbol{a}_1^\top), (t_{1,i})_{i\in[4]}, \widehat{\mathsf{comm}} = (\widehat{\mathsf{comm}}_i)_{i\in[4]}) : \exists (\boldsymbol{r}_i, \hat{\boldsymbol{r}}_i) \text{ and } f \in \bar{\mathcal{C}}, \text{ such that} \\
& 0 < \|\hat{\boldsymbol{r}}_i\| \le \gamma_3', 0 < \|\boldsymbol{r}_i\| \le \sqrt{3N}, \widehat{\mathsf{comm}}_i = \mathsf{Commit}(\boldsymbol{r}_i \| \lfloor\sqrt{q_1}\rceil \cdot \boldsymbol{r}_i; f^{-1} \cdot \hat{\boldsymbol{r}}_i), \\
& \text{and } \langle (1, \boldsymbol{a}_1^\top), \boldsymbol{r}_i \rangle = t_{1,i}, \text{ for } i \in [4] \Big\}.
\end{aligned}$$

In fact, this can be easily proven through using the existing techniques for ABDLOP proving the linear relations of the committed message in [44]. Thus, we omit the detailed protocol for simplicity.

More formally, we have the following theorem.

Theorem C.4 ([2, 44]) *In the random oracle model, let N to be power of 2, prime q_1 with $q_1 = 5 \mod 8$, there exists a multi-theorem* NIZK *system $\Pi^{(3)}$ for $\hat{L}_{\Pi^{(3)}}$ with $\gamma_3' = 2\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot k \cdot N$, where η is the parameter for rejection sampling as in Lemma A.9.*

Summing up the above analysis, for NIZKPoK system Π for $\hat{L}_{q_1,q_2}$, we just need to prove the validity of ABDLOP commitments, the upper ℓ_2 bound of the committed values and several different linear relationships for them. More formally, we have the following corollary.

Corollary 2 ([2, 44]). *In the random oracle model, let N to be powers of 2, prime q_1 be prime such that $q_1 = 5 \mod 8$, prime $q_2 = 3$ or $5 \bmod 8$, there exists a multi-theorem straight-line extractable* NIZKPoK *system Π for the exact relationship $\hat{L}_{q_1,q_2}$.*

Just as mentioned above, the $\Pi^{(2)}$ and $\Pi^{(3)}$ should be put together and instantiated through using ABDLOP commitment and the related Figure 10 in [44]. Here, we omit the detailed steps of the zero-knowledge proof from [44], but instead state how to choose the parameters to match our parameter settings. With this, we can easily compute the concrete efficiency values, which is deferred to Section D.

C.1 Security and Correctness of Construction C.2

Below, we present the security and correctness of Construction C.2.

Correctness. Suppose $\mathsf{ct} = (\boldsymbol{t}_0, t_1, \ldots, t_6)$ is a valid ciphertext, then for the valid public key and secret key $\mathsf{pk} := (\mathbf{A}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_6)$, $\mathsf{sk} := \{\boldsymbol{s}_i\}_{i\in[6]}$, and $i \in [3]$, it holds

$$\begin{cases} u_i = t_i - & \langle \boldsymbol{t}_0, \boldsymbol{s}_i \rangle = \langle \boldsymbol{e}_i, \hat{\boldsymbol{r}} \rangle + m_i \ (\mathrm{mod} q_1) \\ u_{i+3} & = t_{i+3} - \langle \boldsymbol{t}_0, \boldsymbol{s}_{i+3} \rangle \\ & = \langle \boldsymbol{e}_{i+3}, \hat{\boldsymbol{r}} \rangle + m_i \cdot \lfloor \sqrt{q_1} \rceil \ (\mathrm{mod} q_1) \end{cases} \tag{6}$$

In this case, we denote $\langle \boldsymbol{e}_i, \hat{\boldsymbol{r}} \rangle$ and $\langle \boldsymbol{e}_{i+3}, \hat{\boldsymbol{r}} \rangle$ as $\Delta_{1,i}$ and $\Delta_{2,i}$, respectively. Thus, we have

$$\begin{cases} u_i = \Delta_{1,i} + m_i \ (\mathrm{mod} q_1) \\ u_{i+3} = \Delta_{2,i} + m_i \cdot \lfloor \sqrt{q_1} \rceil \ (\mathrm{mod} q_1) \end{cases} \tag{7}$$

Then after multiplying $\lfloor \sqrt{q} \rceil$ into both sides of the first equation, we can get

$$\begin{cases} u_i \cdot \lfloor \sqrt{q_1} \rceil = \Delta_{1,i} \cdot \lfloor \sqrt{q_1} \rceil + m_i \cdot \lfloor \sqrt{q_1} \rceil \ (\mathrm{mod} q_1) \\ u_{i+3} = \Delta_{2,i} + m_i \cdot \lfloor \sqrt{q_1} \rceil \ (\mathrm{mod} q_1) \end{cases} \tag{8}$$

Furthermore, we can get

$$k_i = u_{i+3} - u_i \cdot \lfloor \sqrt{q_1} \rceil = \Delta_{2,i} - \Delta_{1,i} \cdot \lfloor \sqrt{q_1} \rceil (\mathrm{mod} q_1). \tag{9}$$

Notice that each coefficient of $\langle \boldsymbol{e}_i, \hat{\boldsymbol{r}} \rangle = \sum_{j\in[k]} (e_{i,j} \cdot \hat{r}_j)$ is upper bounded by $k \cdot N$. Notice that if $\Delta_{1,i}, \Delta_{2,i}$ are small enough such that $\|\Delta_{i,j}\|_\infty \leq \lfloor \sqrt{q_1} \rceil / 4$, then no reduction modulo q_1 takes place in the Equation (9).

In this case, $\Delta_{2,i}$ can be easily recovered by further modulo $\lfloor \sqrt{q_1} \rceil$ for Equation (9), i.e., $\Delta_{2,i} = k_i (\mathrm{mod} \lfloor \sqrt{q_1} \rceil)$. Finally, we can obtain that

$$m_i = \frac{u_{i+3} - \Delta_{2,i}}{\lfloor \sqrt{q_1} \rceil} \bmod q_1.$$

Security of Construction C.2. Notice that according to the $\mathsf{M\text{-}LWE}_{q_1,n,k}$ assumption, $\boldsymbol{b}_i$ is computational indistinguishability from uniform. Conditioned on this case, the above encryption scheme can be viewed as a BDLOP commitment scheme with parameter $n, k, \ell = 6$, $R_{q_1} = \mathbb{Z}_{q_1}[X]/(X^N + 1)$, and thus the ciphertext of $(r_1, r_2, r_3) \in R_{q_1}^3$ can be viewed as BDLOP commitments. And the security of the encryption follows naturally from the hiding property of the BDLOP commitment. More formally, we have the following theorem.

Theorem C.5 (Security Proof of Construction C.2) *Assuming the hardness of* $\mathsf{M\text{-}LWE}_{q_1,3+\hat{\lambda},k}$ *over* $R = \mathbb{Z}[X]/\langle X^N + 1 \rangle$*, Construction C.2 is* CPA*-secure, with the message space to be* R_q^3*.*

Proof. This theorem can be proven by the following hybrid argument.

$\mathsf{H_0}$: In this hybrid, the adversary conducts the standard CPA-security experiment with the challenger, who runs steps according to the real encryption scheme.

Notice that in H_0, the vectors $\boldsymbol{b}_i$ in the public key are essential the standard $\mathsf{M\text{-}LWE}_{q_1,n,k}$ instances over $R = \mathbb{Z}[X]/\langle X^N + 1\rangle$.

$\mathsf{H_1}$: This hybrid is identical to H_0, except that $\boldsymbol{b}_i$ in the public key is replaced with really uniform ones.

Clearly, H_0 and H_1 are computational indistinguishability, following from the $\mathsf{M\text{-}LWE}_{q_1,n,k}$ assumption.

Moreover, with really uniform vector $\boldsymbol{b}_i$, the modified encryption scheme in H_1 can be viewed as a general BDLOP commitment scheme with parameter $n, k, \ell = 6$. And thus, H_1 can be viewed as the hiding experiment for the BDLOP commitment scheme, as defined in Definition A.10. And it is clear that this hiding property also holds according to the $\mathsf{M\text{-}LWE}_{q_1,k-n-\ell,k}$ assumption, according to [5, 29, 45].

As a result, the CPA-security of Construction C.2 holds according to the $\mathsf{M\text{-}LWE}_{q_1,3+\hat{\lambda},k}$ assumption. □

D Parameter Settings of Construction 4.1 and NIZKPoK system in Section 5

In this section, we set the concrete parameters for Construction 4.1 and the straight-line extractable $\mathsf{NIZKPoK}$ system, according to the related requirements in correctness and security. For clarity, we denote the straight-line extractable $\mathsf{NIZKPoK}$ system for $\hat{L}_{q_1,q_2}$ in Section 5 as Π_1, and denote the $\mathsf{NIZKPoK}$ system for $L_{\gamma',q_2,\bar{\mathcal{C}}}$ in Section 4.2 as Π_2.

Requirements for Correctness. We require the following:

- The $\mathsf{SamplePre}$ in the Sign step needs to work properly. According to Lemma A.3, we need to set $\alpha \geq 2\sqrt{q_2^{\frac{1}{2}} + 1} \cdot (3.5\sqrt{N} + 1)$.
- The valid original signature $\mathsf{Sig}_{\mathsf{comm}}$ can be verified successfully. According to Lemma A.2, we need to set $\gamma = \alpha\sqrt{2 \cdot 12 \cdot N}$.
- The valid transferred signature $\mathsf{Sig}'_{\mathsf{comm}'}$ can be verified successfully. According to Lemma 4.2 and the relaxed language in Theorem 4.3, we need to set $\gamma' = 2\sqrt{2 \cdot 12N} \cdot \eta \cdot \sqrt{\kappa} \cdot (3.5\alpha N \cdot 2\sqrt{2} + \alpha\sqrt{2 \cdot 12N})$.
- Ciphertexts of Construction C.2 can be decrypted correctly. According to the corresponding analysis, we need set $k \cdot N \leq \lfloor\sqrt{q_1}\rceil/4$, with $k = n + 6 + \hat{\lambda}$.

Requirements for Security. We require the following:

- The ring R is cyclotomic, i.e., $R = \mathbb{Z}[X]/(\Phi_m(X))$, where $\Phi_m(X)$ is the m^{th} cyclotomic polynomial, and denote $N = \varphi(m)$. Here, we consider the cyclotomic polynomials $\Phi_m(X) = X^N + 1$ with N to be a power of 2.

– For the fixed security parameter λ, we require that the output distribution of the rejection sampling algorithm is within statistical distance of $\frac{2^{-\lambda}}{M}$ of the related product distribution, according to Lemma A.9. Thus, we need to set η satisfying $M = \exp\left(\sqrt{\frac{2(\lambda+1)}{\log e}} \cdot \frac{1}{\eta} + \frac{1}{2\eta^2}\right) = O(1)$.[14]
– There exists a multi-theorem straight-line extractable $\mathsf{NIZKPoK}$ system Π_1 for the commitment relation $\hat{L}_{q_1,q_2}$. Hence, according to Theorem 5.2 and Corollary 2, in order to make the languages $\hat{L}_{\Pi^{(1)}}$ and $\hat{L}_{\Pi^{(2)}}$ are hard, the following problems

  1. With respect to $\hat{L}_{\Pi^{(1)}}$: $\mathsf{M\text{-}SIS}_{q_1,1,3,\gamma_1'}$ (for $L_{\gamma_1',q_1,q_2,\bar{\mathcal{C}}}$), $\mathsf{M\text{-}SIS}_{q_2,1,4,\gamma_1'+\gamma_2'}$ (for $L_{\gamma_1',\gamma_2',q_1,q_2,\bar{\mathcal{C}}}$) over $R = \mathbb{Z}[X]/\langle X^N + 1\rangle$, with $\gamma_1' = 6\sqrt{2} \cdot \eta \cdot \sqrt{\kappa} \cdot N$, and $\gamma_2' = 2\sqrt{2} \cdot \eta \cdot \sqrt{\kappa} \cdot N$, N to be a power of 2, prime $q_2 = 3$ or 5 mod 8,
  2. With respect to $\hat{L}_{\Pi^{(2)}}$: $\mathsf{M\text{-}SIS}_{q_1,n,k,\gamma_3'}$ over $R = \mathbb{Z}[X]/\langle X^N + 1\rangle$, with $\gamma_3' = 2\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot k \cdot N$, N to be power of 2, prime $q_1 = 5$ mod 8,

  need to be hard.[15]
– There exists a $\mathsf{NIZKPoK}$ system Π_2 for the language $L_{\gamma',q_2,\bar{\mathcal{C}}}$, according to Theorem 4.3. Thus, in order to make this language is hard, the problem $\mathsf{M\text{-}SIS}_{q_2,1,12,\gamma'}$ needs to be hard.
– The constructed CTS satisfies unforgeability in Definition 3.3. Particularly,
  • For Definition 3.3 with respect to the exact commitment relation $\hat{L}_{q_1,q_2}$, according to Lemma 4.6, and Claim B.5, we need to set $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu}$ problem and $\mathsf{M\text{-}SIS}_{q_2,1,9,\nu'}$ being hard with $\nu = 22\alpha \cdot N$ and $\nu' = \frac{22\gamma'\sqrt{N}}{\sqrt{2 \cdot 12}}$.
– The underlying BDLOP satisfies hiding and binding. Hence, according to Section 2.3, we need to set $\mathsf{M\text{-}LWE}_{q_2,1,1}$ and $\mathsf{M\text{-}SIS}_{q_1,1,3,24\sqrt{2}\cdot\eta\cdot\kappa\cdot N}$ being hard.
– The commitment scheme underlying Construction C.2 satisfies hiding and binding. Hence, we need to set $\mathsf{M\text{-}SIS}_{q_1,n,k,8\sqrt{2}\cdot\eta\cdot\kappa\cdot k\cdot N}$, $\mathsf{M\text{-}LWE}_{q_1,\hat{\lambda},k}$ over $R = \mathbb{Z}[X]/\langle X^N + 1\rangle$ to be hard, according to Section C.1.
– The successful simulation of the adversary in Claim B.5. Here, according to the Lemma A.7, we need to set $\alpha \geq 2\sqrt{q_2^{\frac{1}{2}} + 1} \cdot (3.8 \cdot \sqrt{N} + 1)$.

Concrete Parameter instantiations. From the above analysis, according to unforgeability for exact commitment relation $\hat{L}_{q_1,q_2}$, we give the specific parameter setting as in Table 12. Moreover, we use LNP techniques, i.e., Figure 10 in [44], to instantiate $\Pi^{(2)}$ and $\Pi^{(3)}$. Thus, we set the concrete related parameters as in Table 13, and denote the related proof size as $\mathsf{size}_{\mathsf{LNP}}$ in the final computation on the pseudonym size of our Anonymous Credentials system. Notice that for the instantiation of LNP, we follow almost the same parameter notations as [44], which should be helpful for the readers to understand.

With respect to Table 13, we also need to conduct the following explanations,

[14] When we set $M = \exp(1)$, it holds $\frac{2^{-\lambda}}{M} \approx 2^{-\lambda}$.

[15] Notice that as the language $\hat{L}_{\Pi^{(3)}}$ just describe the linear relation among the witnesses of $\hat{L}_{\Pi^{(1)}}$ and $\hat{L}_{\Pi^{(2)}}$, the hardness of $\hat{L}_{\Pi^{(3)}}$ implicitly follows from that of $\hat{L}_{\Pi^{(1)}}$ and $\hat{L}_{\Pi^{(2)}}$.

1. One main difference of our instantiation of LNP from that of [2] is that [2] puts the committed value in the Ajtai part of the ABDLOP commitment, but we put in the BDLOP part. Besides, we just need to prove the upper bound on ℓ_2-norm for one vector.
2. For the challenge set in the protocol, we directly use $\mathcal{C} = \{c \in R : \|c\|_1 = \kappa, \|c\|_\infty = 1\}$, rather than the setting according to Lemma 2.15 in [44]. This is because our ring dimension is large enough to provide enough soundness error.

	Params 1	Params 2
N	4096	8192
q_1	$\sim 2^{34}$	$\sim 2^{36}$
q_2	$\sim 2^{52.52}$	$\sim 2^{56.36}$
λ	172	355
M	5	6
η	9.6544	12.421
κ	17	35
τ	4	4
k	8	8
$\hat{k}$	4	4
$\hat{\lambda}$	1	1
α	$2^{22.062}$	$2^{23.52}$
γ	$2^{29.854}$	$2^{31.81}$
γ'	$2^{50.988}$	$2^{54.83}$
γ'_1	$2^{20.4}$	$2^{22.28}$
γ'_2	$2^{18.815}$	$2^{20.7}$
γ'_3	$2^{21.81}$	$2^{23.7}$
δ_0	1.00281661	1.00151149
Bit-sec	189.8	432.16

Table 12. Concrete Settings for the Parameters and the Related Security in the case of unforgeability with exact relation.

	Params 1	Params 2
λ	172	355
N	4096	8192
q_{zk}	$\sim 2^{34}$	$\sim 2^{36}$
l	2	$\sim 2^{35.77}$
γ_1	14.13	12.421
γ_2	1.5	1.5
γ_e	5	5
κ^*	1	1
λ^*	4	4
η^*	$\sqrt{17}$	$\sqrt{35}$
ν^*	1	1
D	19	20
n	1	1
m_1	1	1
m_2	8	8
ℓ	6	6
γ^*	2^{30}	2^{32}
v_e	1	1
$\alpha^{(e)}$	128	181
$\mathfrak{s}_1$	3166	6650.44
$\mathfrak{s}_2$	1119.54	2271.77
$\mathfrak{s}^{(e)}$	11748.84	16615.37
$\mathsf{size}_{\mathsf{LNP}}$	290.587KB	603.655KB

Table 13. Concrete Settings of the used LNP parameters.

Below, we roughly explain about the calculations of these two tables.

– According to the used rejection sampling algorithms in Lemma A.9, we need to set the parameter η to satisfy the $M = \exp\left(\sqrt{\frac{2(\lambda+1)}{\log e}} \cdot \frac{1}{\eta} + \frac{1}{2\eta^2}\right)$, for any fixed M and λ.
– Given the concrete value of N, we need to fix κ such that the size of the challenge sets are larger than 2^λ, i.e., $\binom{N}{\kappa} \times 2^\kappa \geq 2^\lambda$.

- According to the used NIZKPoK system Π_1 in Theorem 5.2 and Corollary 2, we need to set N to be power of 2, prime $q_1 = 5 \mod 8$, prime $q_2 = 3$ or $5 \mod 8$, $\gamma'_1 = 6\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot N$, $\gamma'_2 = 2\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot N$, and $\gamma'_3 = 2\sqrt{2}\eta \cdot \sqrt{\kappa} \cdot k \cdot N$.
- According to the used NIZKPoK system Π_2 in Theorem 4.3, we need to set $\gamma' = 2\sqrt{2 \cdot 12N} \cdot \eta \cdot \sqrt{\kappa} \cdot (3.5\alpha N \cdot 2\sqrt{2} + \alpha\sqrt{2 \cdot 12N})$, such that the assumption $\mathsf{M\text{-}SIS}_{q_2,1,12,\gamma'}$ is hard, and a NIZKPoK system Π_2 exists for the relaxed language $L_{\gamma',q_2,\bar{\mathcal{C}}}$.
- Given N, q_2, κ, we can calculate the values of α, γ, γ' (all these parameters need to be used in the description of our CTS in Construction 4.1), according to the above parameter analysis for correctness and security.
- We can further compute the values of ν, ν' (all these parameters need to be used to ensure the security proof of our CTS in Construction 4.1), as the requirement of security proof.
- In order to obtain much better tradeoff between efficiency and security, we first choose modulus q_2 such that both the hiding (based on $\mathsf{M\text{-}LWE}_{q_2,1,1}$) and the unforgeability (based on $\mathsf{M\text{-}SIS}_{q_2,1,12,\nu'}$) properties have the sufficient security level.
- Then, we set n, $\hat{\lambda}, k$, and q_1, such that the additional underlying assumptions for for NIZKPoK Π_1 also have sufficient hardness.

During the above calculation process, we use the Root-Hermite Factor δ_0 to estimate bit-hardness of the underlying assumptions, i.e., M-SIS and M-LWE, according to the best known attacks, and δ_0 can be determined given N, q_1, q_2, α. Generally, we can use the work [3, 4, 33] to estimate δ_0 and its corresponding hardness of the assumptions.

Our reduction from each building block is essentially tight (by calling the adversary a constant number of times), so the attained security of our construction is essentially the same as that of the underlying M-LWE and M-SIS problems.

Size Computation. Based on the above parameters on CTS and multi-theorem straight-line extractable NIZKPoK listed in Table 12 and 13, the size of public parameter of CTS is about

$$2 \cdot N\lceil \log q_1 \rceil + 3 \cdot N\lceil \log q_2 \rceil + \log \alpha + \log N + \log q_1 + \log q_2 + \log(\kappa) \text{ bits},$$

the additional size of public parameter of multiple-theorem straight-line extractable NIZKPoK is about

$$\begin{aligned} nkN\lceil \log(q_1) \rceil + (6+4) \cdot k \cdot N \cdot \lceil \log(q_1) \rceil + 3 \cdot \hat{k}N\lceil \log q_2 \rceil \\ + \log n + \log k + \log(\hat{\lambda}) + \log(\hat{k}) + \log(\kappa) \text{ bits}. \end{aligned}$$

Thus, the total size of public parameter is about

$$\begin{aligned} 2 \cdot N\lceil \log q_1 \rceil + 3 \cdot N\lceil \log q_2 \rceil + nkN\lceil \log(q_1) \rceil + (6+4) \cdot k \cdot N \cdot \lceil \log(q_1) \rceil \\ + 3 \cdot \hat{k}N\lceil \log q_2 \rceil + \log(\alpha \cdot N \cdot q_1 \cdot q_2 \cdot n \cdot k \cdot \hat{\lambda} \cdot \hat{k} \cdot \kappa \cdot \kappa) \text{ bits}. \end{aligned}$$

Besides, the sizes of public key and secret key of CTS or the final Anonymous Credentials are about

$$9 \cdot N\lceil \log q_2 \rceil \text{ bits and } 8 \cdot N\lceil \log 3 \rceil \text{ bits},$$

respectively. Furthermore, the size of signature is about

$$12 \cdot N \cdot \log(12\alpha) \text{ bits}.$$

Moreover, the pseudonym consists of two parts: commitments and the related multi-theorem straight-line extractable $\mathsf{NIZKPoK}$ system. And the size of commitment is about

$$4 \cdot N\lceil \log q_1 \rceil + 4 \cdot N\lceil \log q_2 \rceil \text{ bits},$$

the size of proof is about

$$6{\cdot}3N \cdot \log(12\eta \cdot \sqrt{3\kappa N}) + N \cdot \log(12\eta \cdot \sqrt{\kappa N}) + 2\hat{k}N\lceil \log q_2 \rceil + \lambda + \hat{k} \log q_2 + 4 \cdot \mathsf{size}_{\mathsf{LNP}} \text{ bits}.$$

Thus, the total size of pseudonym is about

$$4{\cdot}N\lceil \log q_1 \rceil + 4 \cdot N\lceil \log q_2 \rceil \text{ bits} + 6 \cdot 3N \cdot \log(12\eta \cdot \sqrt{3\kappa N}) + N \cdot \log(12\eta \cdot \sqrt{\kappa N}) + 2\hat{k}N\lceil \log q_2 \rceil + \lambda + \hat{k} \log q_2 + 4 \cdot \mathsf{size}_{\mathsf{LNP}} \text{ bits}.$$

Finally, the credential size is about

$$12 \cdot N \cdot \log(12\eta\sqrt{\kappa} \cdot \gamma) \text{ bits}.$$

E Adaptively Secure CTS

In this section, we present an adaptively secure CTS scheme. Our construction follows the partitioning approach as [1], and results some additional reduction loss compared with the selective construction in section 4. We first introduce the additional preliminaries for this adaptive construction, and then present the construction, and show the correctness and security of the scheme. Finally, we provide some parameter settings for concrete instantiations.

E.1 Pairwise Independent Hash Function

We give a lemma which shows that pairwise independent hash function family which is denoted as $\mathcal{H}$ has the isolation property as long as a conditional probability defined as below approximates $1 = |Q|$.

Lemma E.1 *Let $Q \subseteq \mathcal{M}$, A, B be integers such that $B \leq A$, $|Q| \leq \delta B$ for some $\delta \in (0, 1)$, and let $\mathcal{H} : \mathcal{M} \to \mathcal{Y}$ be an pairwise independent hash function family which has the following properties:*

– $\forall \boldsymbol{a} \in \mathcal{M}$, $\mathsf{Pr}_{H \leftarrow \mathcal{H}}[H(\boldsymbol{a}) = 0] = 1/A$;
– $\forall \boldsymbol{a} \neq \boldsymbol{b} \in \mathcal{M}$, $\mathsf{Pr}_{H \leftarrow \mathcal{H}}[H(\boldsymbol{a}) = 0 | H(\boldsymbol{b}) = 0] \leq 1/B$.

Then for any element $\boldsymbol{a} \notin Q$, we have

$$\mathsf{Pr}_{H \in \mathcal{H}}[H(\boldsymbol{a}) = 0 \bigwedge H(\boldsymbol{a}') \neq 0, \forall \boldsymbol{a}' \in Q] \in \left(\frac{1-\delta}{A}, \frac{1}{A}\right).$$

An Explicit Almost Pairwise Independent Hash Construction. Let $q \in \mathbb{N}$ be a prime, $N, \ell \in \mathbb{N}$, $R_q = \mathbb{Z}_q[X]/\langle x^N + 1\rangle$, $S_q \subset R_q$ be a subfield of R_q with order q^τ. We define the hash function family $\mathcal{H} : (S_q)^\ell \to S_q$ as follows: $\forall H \in \mathcal{H}$, H is indexed by $(\alpha, h_1, ...h_\ell) \in (S_q)^{\ell+1}$, $\forall \boldsymbol{x} = (x_1, ..., x_\ell) \in (S_q)^\ell$, $H(\boldsymbol{x}) = \alpha + \langle \boldsymbol{x}, \boldsymbol{h}\rangle \in S_q$. We have the following lemma.

Lemma E.2 ([1]) *The function family $\mathcal{H}$ defined above is an pairwise independent hash function. Moreover, we have*

– $\forall H \leftarrow \mathcal{H}$ *and* $\forall \boldsymbol{x} \in (S_q)^\ell$, $\mathsf{Pr}[H(\boldsymbol{x}) = 0] = 1/q^\tau$.
– $\forall H \leftarrow \mathcal{H}$ *and* $\forall \boldsymbol{x} \neq \boldsymbol{y} \in (S_q)^\ell$, $\mathsf{Pr}[H(\boldsymbol{y}) = 0 | H(\boldsymbol{x}) = 0] \leq 1/q^\tau$.

E.2 Adaptively Secure Construction

Our construction uses the following building blocks: (1) the BDLOP commitment scheme $\Gamma = \Gamma.\{\mathsf{CKeyGen}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Combine}, \mathsf{Randomize}\}$, and (2) a single-theorem rewinding extractable NIZKPoK system $\Pi = \Pi.\{\mathsf{Setup}, \mathsf{Prove}, \mathsf{VerifyProve}, \mathsf{SimProve}\}$ for the following language (parameterized by $\gamma', q \in \mathbb{N}$)

$$L_{\gamma',q,\bar{\mathcal{C}}} = \Big\{(\mathbf{B}, u) \in R_q^{1\times(2k+4)} \times R_q : \exists\ \boldsymbol{x} \in R_q^{(2k+4)} \text{ and } f \in \bar{\mathcal{C}} \text{ such that } \|\boldsymbol{x}\|_2 \leq \gamma' \text{ and } \mathbf{B} \cdot \boldsymbol{x} = f \cdot \boldsymbol{u}\Big\},$$

Similar to the presentation of Construction 4.1 in Section 4.1, we first describe the required parameters in Table 14. Notice that for the adaptive security, we need to set the message space $\mathcal{M}$ as the concatenation of several independent and identical spaces $\bar{\mathcal{M}}_i$ for $i \in [\ell]$, i.e., $\mathcal{M} = (\bar{\mathcal{M}}_1, \ldots, \bar{\mathcal{M}}_\ell)$. Moreover, $\bar{\mathcal{M}}_i$ should satisfy two requirements: (1) $\bar{\mathcal{M}}_i$ is a subset of a subfield of R_{q_2}; (2) the ℓ_2 norm of all elements in $\bar{\mathcal{M}}_i$ should be upper bounded by B.

Particularly, all parameters are in the following table.

Construction E.3 (Commit-Transferrable Signature) *Our adaptive* CTS *is constructed as follow.*

– $\mathsf{Setup}(1^\lambda, \ell, B)$*: On input the security parameter 1^λ, the algorithm does the following.*
  1. *Run $\Gamma.\mathsf{CKeyGen}$ to get $\mathbf{A} := \begin{bmatrix} 1, & \boldsymbol{a}_1'^\top \\ 0, 1, & a_2' \end{bmatrix} \leftarrow \Gamma.\mathsf{CKeyGen}(1^\lambda)$, where $[1, \boldsymbol{a}_1'^\top] \in R_{q_1}^{1\times 3}$ and $[0, 1, a_2'] \in R_{q_2}^{1\times 3}$, with $\boldsymbol{a}_1' \in R_{q_1}^2$, $\boldsymbol{a}_2^\top = (1, a_2') \in R_{q_2}^2$. Note that the commitment scheme sets message space $\mathcal{M} \subseteq (R_{q_2})^\ell$ with randomness space $(\mathcal{R})^{k\ell} = (S_1^3)^{k\ell}$, where $\mathcal{M} = (\bar{\mathcal{M}})^\ell$, and $\bar{\mathcal{M}}$ is a set with B bounded ℓ_2 norm and included in a subfield $\mathcal{S}_{q_2}$ of R_{q_2}.*

2. *Sample* $\boldsymbol{d} \xleftarrow{\$} R_{q_2}^2$;
3. *Run* $\Pi.\mathsf{Setup}(1^\lambda)$ *to get common references string* crs;
4. *Output* $\mathsf{params} := (\mathbf{A}, \boldsymbol{d}, \mathcal{M}, \mathcal{R}, \mathsf{crs})$.

– $\mathsf{Commit}(\mathsf{params}, \boldsymbol{m}; \mathsf{Rand})$*: On input* params*, message* $\boldsymbol{m} \in \mathcal{M}$*, and randomness* $\mathsf{Rand} \in \mathcal{R}^{k\ell}$*, the algorithm does the following.*

1. *Parse* Rand *as* ℓ *vectors* $\{(\boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,k})\}_{i\in[\ell]}$*, where* $\boldsymbol{r}_{i,j} \in \mathcal{R} = S_1^3$ *for* $i \in [\ell], j \in [k]$.
2. *Parse* $\boldsymbol{m}$ *as* $(m_1, \cdots, m_\ell)$*. For* $i \in [\ell]$*, run* $\mathsf{comm}_{i,1} = \Gamma.\mathsf{Commit}(\mathbf{A}, m_i; \boldsymbol{r}_{i,1})$, $\mathsf{comm}_{i,2} = \Gamma.\mathsf{Commit}(\mathbf{A}, m_i\delta; \boldsymbol{r}_{i,2})$, $\cdots, \mathsf{comm}_{i,k} = \Gamma.\mathsf{Commit}(\mathbf{A}, m_i\delta^k; \boldsymbol{r}_{i,k})$.
3. *Output* $\mathsf{comm} = \{(\mathsf{comm}_{i,1}, \mathsf{comm}_{i,2}, \cdots, \mathsf{comm}_{i,k})\}_{i\in[\ell]}$ *as the commitment of* $\boldsymbol{m}$.

– $\mathsf{Randomize}(\mathsf{params}, \mathsf{comm}, \boldsymbol{m}, \mathsf{Rand}, \mathsf{Rand}')$*: On input* params*,* $\mathsf{Rand}, \mathsf{Rand}' \in \mathcal{R}^{k\ell}$*, and* $\mathsf{comm} = \{(\mathsf{comm}_{i,1}, \mathsf{comm}_{i,2}, \cdots, \mathsf{comm}_{i,k})\}_{i\in[\ell]}$*, the algorithm does the following.*

1. *Parse* Rand' *as* ℓ *vectors* $\{(\tilde{\boldsymbol{r}}_{i,1}, \tilde{\boldsymbol{r}}_{i,2}, \cdots, \tilde{\boldsymbol{r}}_{i,k})\}_{i\in[\ell]}$*, where* $\tilde{\boldsymbol{r}}_{i,j} \in \mathcal{R} = S_1^3$ *for* $i \in [\ell], j \in [k]$.
2. *For* $i \in [\ell]$*, run* $\mathsf{comm}'_{i,1} = \Gamma.\mathsf{Randomize}(\mathbf{A}, \mathsf{comm}_{i,1}, \tilde{\boldsymbol{r}}_{i,1})$, $\mathsf{comm}'_{i,2} = \Gamma.\mathsf{Randomize}(\mathbf{A}, \mathsf{comm}_{i,2}, \tilde{\boldsymbol{r}}_{i,2}), \cdots, \mathsf{comm}'_{i,k} = \Gamma.\mathsf{Randomize}(\mathbf{A}, \mathsf{comm}_{i,k}, \tilde{\boldsymbol{r}}_{i,k})$. *Set* $\mathsf{comm}' = \{(\mathsf{comm}'_{i,1}, \mathsf{comm}'_{i,2}, \cdots, \mathsf{comm}'_{i,k})\}_{i\in[\ell]}$.
3. *Output* comm' *as the rerandomized commitment of* $\boldsymbol{m}$.

– $\mathsf{Combine}(\mathsf{Rand}, \mathsf{Rand}')$*: Taking as input two randomness* $\mathsf{Rand} = \{(\boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,k})\}_{i\in[\ell]} \in S_1^{3\times k\ell}$*, and* $\mathsf{Rand}' = \{(\tilde{\boldsymbol{r}}_{i,1}, \tilde{\boldsymbol{r}}_{i,2}, \cdots, \tilde{\boldsymbol{r}}_{i,k})\}_{i\in[\ell]} \in S_1^{3\times k\ell}$*, the algorithm computes and outputs* $\{(\hat{\boldsymbol{r}}_{i,1}, \hat{\boldsymbol{r}}_{i,2}, \cdots, \hat{\boldsymbol{r}}_{i,k})\}_{i\in[\ell]} \in S_2^{3\times k\ell}$*, where* $\hat{\boldsymbol{r}}_{i,j} = \boldsymbol{r}_{i,j} + \tilde{\boldsymbol{r}}_{i,j}$ *for* $i \in [\ell], j \in [k]$.

– $\mathsf{KeyGen}(\mathsf{params})$*: On input* params*, the algorithm does:*

1. *Sample* $\mathbf{T} \xleftarrow{\$} S_1^{2\times k}$*, and set* $\boldsymbol{a}^\top = \boldsymbol{d}^\top \cdot \mathbf{T} + \boldsymbol{g}_\delta^\top \in R_{q_2}^{1\times k}$*, where* $\boldsymbol{g}_\delta^\top = (1, \delta, \delta^2, \cdots, \delta^{k-1}) \in R_{q_2}^{1\times k}$.
2. *Sample* $(\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_\ell) \xleftarrow{\$} R_{q_2}^{k(\ell+1)}$;
3. *Output* $\mathsf{pk} := (\boldsymbol{a}, \boldsymbol{b}_0, \{\boldsymbol{b}_i\}_{i\in[\ell]})$*, and* $\mathsf{sk} := \mathbf{T}$.

– $\mathsf{Sign}(\mathsf{params}, \mathsf{pk}, \mathsf{sk}, \mathsf{comm})$*: On input* params*,* pk*,* sk *and* comm*, the algorithm does the following:*

1. *For* $i \in [\ell]$*, parse* $\mathsf{comm}_i = (\mathsf{comm}_{i,1}, \mathsf{comm}_{i,2}, \mathsf{comm}_{i,3}, \mathsf{comm}_{i,4})$ *as* $\mathsf{comm}_{i,1} = \begin{bmatrix} t_{1,1}^{(i)} \\ t_{2,1}^{(i)} \end{bmatrix}$, $\mathsf{comm}_{i,2} = \begin{bmatrix} t_{1,2}^{(i)} \\ t_{2,2}^{(i)} \end{bmatrix}, \cdots, \mathsf{comm}_{i,k} = \begin{bmatrix} t_{1,k}^{(i)} \\ t_{2,k}^{(i)} \end{bmatrix}$;
2. *Set* $\mathbf{F}_{\mathsf{comm}} = \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{b}_{\mathsf{comm}}^\top \middle| \boldsymbol{a}_2^\top\right] = \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| [\boldsymbol{b}_0 + \sum_{i\in[\ell]} \left((t_{2,1}^{(i)}, t_{2,2}^{(i)}, \cdots, t_{2,k}^{(i)}) \cdot \mathbf{G}^{-1}(\boldsymbol{b}_i)\right)] \middle| \boldsymbol{a}_2^\top\right]$, *and sample* $\mathsf{Sig}_{\mathsf{comm}} := \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 \end{bmatrix} \leftarrow \mathsf{SamplePre}(\left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{b}_{\mathsf{comm}}^\top \middle| \boldsymbol{a}_2^\top\right], \mathbf{T}, 0, \alpha)$*, and output* $\mathsf{Sig}_{\mathsf{comm}}$ *as the signature of* comm*, where* $\boldsymbol{s}_1 = \begin{bmatrix} \boldsymbol{s}_{1,1} \\ \boldsymbol{s}_{1,2} \end{bmatrix}$*, and* $\boldsymbol{s}_{1,1} \in R^2, \boldsymbol{s}_{1,2} \in R^k, \boldsymbol{s}_2 \in R^k$*,* $\boldsymbol{s}_3 \in R^2$.

– $\mathsf{Transfer}(\mathsf{params}, \mathsf{pk}, \mathsf{Sig}_{\mathsf{comm}}, \boldsymbol{m}, (\mathsf{Rand}, \mathsf{Rand}'))$*: On input* $\mathsf{params}, \mathsf{pk}$*, a signature* $\mathsf{Sig}_{\mathsf{comm}}$*, message* $\boldsymbol{m}$*, randomness* Rand *for generating the commitment* comm *for* $\boldsymbol{m}$*, the additional randomness* Rand' *for the rerandomization of* comm*, the algorithm does the followings:*

1. *Parse* $\mathsf{Sig}_{\mathsf{comm}}$ *as vector* $\begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 \end{bmatrix}$*, where* $\boldsymbol{s}_1 \in R^{k+2}$*,* $\boldsymbol{s}_2 \in R^k$*,* $\boldsymbol{s}_3 \in R^2$*.*
2. *Parse* Rand *as* ℓ *vectors* $\{(\boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,k})\}_{i\in[\ell]}$*, where* $\boldsymbol{r}_{i,j} \in \mathcal{R} = S_1^3$*.*
3. *Parse* Rand' *as* ℓ *vectors* $\{(\tilde{\boldsymbol{r}}_{i,1}, \tilde{\boldsymbol{r}}_{i,2}, \cdots, \tilde{\boldsymbol{r}}_{i,k})$*, where* $\tilde{\boldsymbol{r}}_{i,j} \in \mathcal{R} = S_1^3$*.*
4. *For* $i \in [\ell]$*, run* $\mathsf{Commit}(\mathsf{params}, m_i; (\boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,k}))$ *and obtain:* $\mathsf{comm}_i = (\mathsf{comm}_{i,1}, \mathsf{comm}_{i,2}, \cdots, \mathsf{comm}_{i,k})$*, where* $\mathsf{comm}_{i,1} = \begin{bmatrix} t_{1,1}^{(i)} \\ t_{2,1}^{(i)} \end{bmatrix}$, $\mathsf{comm}_{i,2} = \begin{bmatrix} t_{1,2}^{(i)} \\ t_{2,2}^{(i)} \end{bmatrix}, \cdots, \mathsf{comm}_{i,k} = \begin{bmatrix} t_{1,k}^{(i)} \\ t_{2,k}^{(i)} \end{bmatrix}$.
5. *For* $i \in [\ell]$*, run* $\mathsf{Randomize}$ $(\mathsf{params}, \mathsf{comm}_i, (\tilde{\boldsymbol{r}}_{i,1}, \tilde{\boldsymbol{r}}_{i,2}, \cdots, \tilde{\boldsymbol{r}}_{i,k}))$ *and obtain* $\mathsf{comm}'_i = (\mathsf{comm}'_{i,1}, \mathsf{comm}'_{i,2}, \cdots, \mathsf{comm}'_{i,k})$*, where* $\mathsf{comm}'_{i,1} = \begin{bmatrix} \hat{t}_{1,1}^{(i)} \\ \hat{t}_{2,1}^{(i)} \end{bmatrix}$, $\mathsf{comm}'_{i,2} = \begin{bmatrix} \hat{t}_{1,2}^{(i)} \\ \hat{t}_{2,2}^{(i)} \end{bmatrix}, \cdots,$ $\mathsf{comm}'_{i,k} = \begin{bmatrix} \hat{t}_{1,k}^{(i)} \\ \hat{t}_{2,k}^{(i)} \end{bmatrix}$.
6. *Compute a (temporary) signature* $\mathsf{Sig}_{\mathsf{comm}'}$ *as*

$$\mathsf{Sig}_{\mathsf{comm}'} := \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 - \sum_{i\in[\ell]} \left(\tilde{\mathbf{R}}_{i,2}\mathbf{G}^{-1}(\boldsymbol{b}_i)\right) \cdot \boldsymbol{s}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{1,1} \\ \boldsymbol{s}_{1,2} \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 - \sum_{i\in[\ell]} \left(\tilde{\mathbf{R}}_{i,2}\mathbf{G}^{-1}(\boldsymbol{b}_i)\right) \cdot \boldsymbol{s}_2 \end{bmatrix} \in R^{2k+4},$$

   where we denote $\tilde{\mathbf{R}}_i = \begin{bmatrix} \tilde{\mathbf{R}}_{i,1} \\ \tilde{\mathbf{R}}_{i,2} \end{bmatrix} = \left[\tilde{\boldsymbol{r}}_{i,1}, \tilde{\boldsymbol{r}}_{i,2}, \cdots, \tilde{\boldsymbol{r}}_{i,k}\right] \in R^{3\times k}$*, with* $\tilde{\mathbf{R}}_{i,1} \in R^{1\times k}$ *and* $\tilde{\mathbf{R}}_{i,2} \in R^{2\times k}$*.*
7. *Compute* $\mathbf{F}_{\mathsf{comm}'} := \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{b}_{\mathsf{comm}'}^\top \middle| \boldsymbol{a}_2^\top\right] = \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| [\boldsymbol{b}_0 + \sum_{i\in[\ell]} \left((\hat{t}_{2,1}^{(i)}, \hat{t}_{2,2}^{(i)}, \cdots, \hat{t}_{2,k}^{(i)}) \cdot \mathbf{G}^{-1}(\boldsymbol{b}_i)\right)] \middle| \boldsymbol{a}_2^\top\right]$.
8. *Run the prove algorithm, output* $\mathsf{Sig}'_{\mathsf{comm}'} := \pi \leftarrow \Pi.\mathsf{Prove}(\mathsf{crs}, (\mathbf{F}_{\mathsf{comm}'}, 0), \mathsf{Sig}_{\mathsf{comm}'})$*, proving that* $\mathsf{Sig}_{\mathsf{comm}'}$ *is a short* ℓ_2 *norm vector and satisfies* $\mathbf{F}_{\mathsf{comm}'} \cdot \mathsf{Sig}_{\mathsf{comm}'} = 0$*, through using the NIZKPoK system* Π *with the relaxed language* $L_{\gamma', q_2, \bar{\mathcal{C}}}$*.*

– $\mathsf{Verify}(\mathsf{params}, \mathsf{pk}, \mathsf{comm}, \mathsf{Sig})$*: On input* $\mathsf{params}, \mathsf{pk}, \mathsf{comm}, \mathsf{Sig}$*, the algorithm does the following.*

1. *Parse* $\mathsf{comm} = \{(\mathsf{comm}_{i,1}, \mathsf{comm}_{i,2}, \cdots, \mathsf{comm}_{i,4})\}_{i\in[\ell]}$ *as* $\mathsf{comm}_{i,1} = \begin{bmatrix} t_{1,1}^{(i)} \\ t_{2,1}^{(i)} \end{bmatrix}$, $\mathsf{comm}_{i,2} = \begin{bmatrix} t_{1,2}^{(i)} \\ t_{2,2}^{(i)} \end{bmatrix}, \cdots, \mathsf{comm}_{i,k} = \begin{bmatrix} t_{1,k}^{(i)} \\ t_{2,k}^{(i)} \end{bmatrix}$;
2. *Based on the type of* Sig*, the verification works as follow.*
   - *If* Sig *is a short vector within* ℓ_2 *norm* γ*, then the algorithm does*
     (a) Set matrix

$$\mathbf{F}_{\mathsf{comm}} := \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| [\boldsymbol{b}_0 + \sum_{i\in[\ell]} \left((t_{2,1}^{(i)}, t_{2,2}^{(i)}, \cdots, t_{2,k}^{(i)}) \cdot \mathbf{G}^{-1}(\boldsymbol{b}_i) \right)] \middle| \boldsymbol{a}_2^\top \right].$$

     (b) Check whether Sig *satisfies*

$$\mathbf{F}_{\mathsf{comm}} \cdot \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 \end{bmatrix} = 0 \in \mathcal{R}_{q_2}.$$

   - *If* Sig *is a proof of the NIZKPOK system* Π*,*
     (a) Set matrix

$$\mathbf{F}_{\mathsf{comm}} := \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| [\boldsymbol{b}_0 + \sum_{i\in[\ell]} \left((t_{2,1}^{(i)}, t_{2,2}^{(i)}, \cdots, t_{2,k}^{(i)}) \cdot \mathbf{G}^{-1}(\boldsymbol{b}_i) \right)] \middle| \boldsymbol{a}_2^\top \right].$$

     (b) Run the verify algorithm (with respect to language $L_{\gamma', q_2, \bar{\mathcal{C}}}$*)* $\Pi.\mathsf{VerifyProve}(\mathsf{crs}, (\mathbf{F}_{\mathsf{comm}}, 0), \mathsf{Sig})$*, and output its result.*

Lemma E.4 (Correctness) *For parameters* $N, q_2, \alpha, \gamma = \alpha\sqrt{(2k+4)\cdot N}$*, the* NIZKPoK *system* Π *for the relaxed language* $L_{\gamma', q_2, \bar{\mathcal{C}}}$ *with* $\gamma' \geq (\sqrt{2k}+k)k\ell\alpha N^2\delta + \alpha\sqrt{(2k+4)N}$*, Construction E.3 satisfies the correctness property as defined in Definition 3.1.*

Proof. The correctness according to Definition 3.1 requires to prove the following three statements: (1) four algorithms (Setup,Commit,Randomize, Combine) define a correct randomizable commitment scheme; (2) the signature by algorithm Sign passes the verification algorithm, i.e., Verify; and (3) the transferred signature (with respect to the randomized commitment) from Transfer also passes Verify.

The correctness of statement (1) and statement (2) are easy to verify. We just sketch the correctness of statement (3). Similar to the analysis of Lemma 4.2, it it suffices to show that $\mathbf{F}_{\mathsf{comm}'} \cdot \mathsf{Sig}_{\mathsf{comm}'} = 0$ (as defined in the algorithm Transfer) $\mathsf{Sig}_{\mathsf{comm}'}$ is within ℓ_2 norm $(\sqrt{2k}+k)k\ell\alpha N^2\delta + \alpha\sqrt{(2k+4)N}$.

Particularly, for all $\boldsymbol{m} \in \mathcal{M} = (\bar{\mathcal{M}})^\ell \subseteq (\mathcal{R}_{q_2})^\ell$, $\boldsymbol{r}_{i,j}, \tilde{\boldsymbol{r}}_{i,j} \in S_1^3, i \in [\ell], j \in [k]$, $(\mathsf{sk}, \mathsf{pk})$ output by KeyGen, and signature $\mathsf{Sig}_{\mathsf{comm}} = (\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, \boldsymbol{s}_3^T) = ((\boldsymbol{s}_{1,1}^T, \boldsymbol{s}_{1,2}^T), \boldsymbol{s}_2^T, \boldsymbol{s}_3^T)$ output by Sign, it holds

$$\mathbf{F}_{\mathsf{comm}} \cdot \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 \end{bmatrix} = u \in \mathcal{R}_{q_2},$$

where

$$\mathbf{F}_{\mathsf{comm}} = \left[[\boldsymbol{d}^T | \boldsymbol{a}^T] \big| [\boldsymbol{b}_0 + \sum_{i\in[\ell]} \left((t_{2,1}^{(i)}, t_{2,2}^{(i)}, \cdots, t_{2,k}^{(i)}) \cdot \mathbf{G}^{-1}(\boldsymbol{b}_i) \right)] \big| \boldsymbol{a}_2^T \right].$$

And the ℓ_2 norm of the vector $(\boldsymbol{s}_{1,1}^T, \boldsymbol{s}_{1,2}^T, \boldsymbol{s}_2^T, \boldsymbol{s}_3^T)$ is less than $\alpha\sqrt{(2k+4)N}$. This implies

$$\begin{aligned} \langle \boldsymbol{d}, \boldsymbol{s}_{1,1} \rangle + \langle \boldsymbol{a}, \boldsymbol{s}_{1,2} \rangle + \langle \boldsymbol{b}_0 + \sum_{i\in[\ell]} \left((t_{2,1}^{(i)}, t_{2,2}^{(i)}, \cdots, t_{2,k}^{(i)}) \cdot \mathbf{G}^{-1}(\boldsymbol{b}_i) \right), \\ \boldsymbol{s}_2 \rangle + \langle \boldsymbol{a}_2, \boldsymbol{s}_3 \rangle = u \in \mathcal{R}_{q_2}. \end{aligned}$$

We notice that the above equation is equivalent to

$$\begin{aligned} 0 =& \langle \boldsymbol{d}, \boldsymbol{s}_{1,1} \rangle + \langle \boldsymbol{a}, \boldsymbol{s}_{1,2} \rangle + \langle \boldsymbol{b}_0 + \sum_{i\in[\ell]} (t_{2,1}^{(i)}, t_{2,2}^{(i)}, \cdots, t_{2,k}^{(i)}) \cdot \mathbf{G}^{-1}(\boldsymbol{b}_i), \\ & \boldsymbol{s}_2 \rangle + \langle \boldsymbol{a}_2, \sum_{i\in[\ell]} \tilde{\mathbf{R}}_{i,2} \mathbf{G}^{-1}(\boldsymbol{b}_i) \cdot \boldsymbol{s}_2 \rangle - \langle \boldsymbol{a}_2, \sum_{i\in[\ell]} \tilde{\mathbf{R}}_{i,2} \mathbf{G}^{-1}(\boldsymbol{b}_i) \cdot \boldsymbol{s}_2 \rangle \\ & + \langle \boldsymbol{a}_2, \boldsymbol{s}_3 \rangle \\ =& \langle \boldsymbol{d}, \boldsymbol{s}_{1,1} \rangle + \langle \boldsymbol{b}_0 + \sum_{i\in[\ell]} (t_{2,1}^{(i)}, t_{2,2}^{(i)}, \cdots, t_{2,k}^{(i)}) \cdot \mathbf{G}^{-1}(\boldsymbol{b}_i) + \\ & \boldsymbol{a}_2^\top \cdot \sum_{i\in[\ell]} \tilde{\mathbf{R}}_{i,2} \mathbf{G}^{-1}(\boldsymbol{b}_i), \boldsymbol{s}_2 \rangle + \langle \boldsymbol{a}, \boldsymbol{s}_{1,2} \rangle + \langle \boldsymbol{a}_2, \\ & - \sum_{i\in[\ell]} \tilde{\mathbf{R}}_{i,2} \mathbf{G}^{-1}(\boldsymbol{b}_i) \cdot \boldsymbol{s}_2 \rangle + \langle \boldsymbol{a}_2, \boldsymbol{s}_3 \rangle, \end{aligned}$$

which can be rewritten as

$$\begin{aligned} & [[\boldsymbol{d}^\top | \boldsymbol{a}^\top] | \boldsymbol{b}_0 + \sum_{i\in[\ell]} (t_{2,1}^{(i)}, t_{2,2}^{(i)}, \cdots, t_{2,k}^{(i)}) \cdot \mathbf{G}^{-1}(\boldsymbol{b}_i) + \\ & \boldsymbol{a}_2^\top \cdot \sum_{i\in[\ell]} \tilde{\mathbf{R}}_{i,2} \mathbf{G}^{-1}(\boldsymbol{b}_i) | \boldsymbol{a}_2^\top] \cdot \begin{bmatrix} \boldsymbol{s}_{1,1} \\ \boldsymbol{s}_{1,2} \\ \boldsymbol{s}_2 \\ \boldsymbol{s}_3 - \sum_{i\in[\ell]} \tilde{\mathbf{R}}_{i,2} \mathbf{G}^{-1}(\boldsymbol{b}_i) \cdot \boldsymbol{s}_2 \end{bmatrix} = u. \end{aligned}$$

Here we denote $\tilde{\mathbf{R}}_i = \begin{bmatrix} \widetilde{\mathbf{R}}_{i,1} \\ \widetilde{\mathbf{R}}_{i,2} \end{bmatrix} = \left[\widetilde{\boldsymbol{r}}_{i,1}, \widetilde{\boldsymbol{r}}_{i,2}, \cdots, \tilde{\boldsymbol{r}}_{i,k} \right] \in R^{3\times k}$, with $\widetilde{\mathbf{R}}_{i,1} \in R^{1\times k}$ and $\widetilde{\mathbf{R}}_{i,2} \in R^{2\times k}$.

Then we observe that

$$\begin{aligned}\mathbf{F}_{\mathsf{comm}'} :=& \left[[\boldsymbol{d}^\top|\boldsymbol{a}^\top]\big|\boldsymbol{b}_0 + \sum_{i\in[\ell]}(\hat{t}^{(i)}_{2,1}, \hat{t}^{(i)}_{2,2}, \cdots, \hat{t}^{(i)}_{2,k})\cdot\mathbf{G}^{-1}(\boldsymbol{b}_i)\big|\boldsymbol{a}_2^\top\right]\\ =&\Big[[\boldsymbol{d}^\top|\boldsymbol{a}^\top]\big|\boldsymbol{b}_0 + \sum_{i\in[\ell]}(t^{(i)}_{2,1}, t^{(i)}_{2,2}, \cdots, t^{(i)}_{2,k})\cdot\mathbf{G}^{-1}(\boldsymbol{b}_i)\\ &+ \boldsymbol{a}_2^\top\cdot\sum_{i\in[\ell]}\tilde{\mathbf{R}}_{i,2}\mathbf{G}^{-1}(\boldsymbol{b}_i)|\boldsymbol{a}_2^\top\Big],\end{aligned}$$

and $\mathsf{Sig}_{\mathsf{comm}'} := \begin{bmatrix}\boldsymbol{s}_{1,1}\\ \boldsymbol{s}_{1,2}\\ \boldsymbol{s}_2\\ \boldsymbol{s}_3 - \sum_{i\in[\ell]}\tilde{\mathbf{R}}_{i,2}\mathbf{G}^{-1}(\boldsymbol{b}_i)\cdot\boldsymbol{s}_2\end{bmatrix}$. Now, it is easy to verify that the ℓ_2 norm of $\mathsf{Sig}_{\mathsf{comm}',1}$ and $\mathsf{Sig}_{\mathsf{comm}',2}$ are within $(\sqrt{2k}+k)k\ell\alpha N^2\delta+\alpha\sqrt{(2k+4)N}$ and $\mathbf{F}_{\mathsf{comm}'}\cdot\mathsf{Sig}_{\mathsf{comm}'} = 0$, since for such matrices $\widetilde{\mathbf{R}}_{i,2}\in S_1^{2\times k}$, its singular value $s_1(\widetilde{\mathbf{R}}_{i,2})$ is bounded by $(\sqrt{2}+\sqrt{k})\sqrt{N}$, and the singular value of $\mathbf{G}^{-1}(\boldsymbol{b}_i)$ is bounded by $kN\delta$ by Lemma A.4. This completes the proof. □

E.3 Security Proof

In this section, we show the simulatability and unforgeability of the above Construction E.3.

Lemma E.5 (Simulatability) *The algorithm* Transfer *in Construction E.3 is simulatable.*

Proof. Similar to the proof of Lemma 4.5, we first construct a two-stage PPT simulator $\mathcal{S}$, and then prove that after running any polynomial $t = \mathsf{poly}(\lambda)$ times, the distribution of $\{\widetilde{\mathsf{Sig}}'_{\mathsf{comm}'_i}\}_{i\in[t]}$ output by $\mathcal{S}$ are statistically close to that of $\{\mathsf{Sig}'_{\mathsf{comm}'_i}\}_{i\in[t]}$ output by Transfer.

The two-stage PPT simulator $\mathcal{S}$ can be constructed in the following way:

- First Stage: $\mathcal{S}$ conducts the following steps:
  1. Generate and output $\mathsf{params} := (\mathbf{A}, \boldsymbol{d}, \mathcal{M}, \mathcal{R}, \mathsf{crs})$.
- Second Stage: given params, and valid pk, comm', $\mathcal{S}$ conducts the following steps:
  1. Recognize pk as $(\boldsymbol{a}, \boldsymbol{b}_0, \{\boldsymbol{b}_i\}_{i\in[\ell]}, u)$.
  2. Parse $\mathsf{comm}' = (\{\mathsf{comm}'_{i,1}, \mathsf{comm}'_{i,2}, \cdots, \mathsf{comm}'_{i,k}\}_{i\in[\ell]})$ as $\mathsf{comm}_{i,1} = \begin{bmatrix}\hat{t}^{(i)}_{1,1}\\ \hat{t}^{(i)}_{2,1}\end{bmatrix}$, $\mathsf{comm}_{i,2} = \begin{bmatrix}\hat{t}^{(i)}_{1,2}\\ \hat{t}^{(i)}_{2,2}\end{bmatrix}, \cdots, \mathsf{comm}_{i,k} = \begin{bmatrix}\hat{t}^{(i)}_{1,k}\\ \hat{t}^{(i)}_{2,k}\end{bmatrix}$;
  3. Set matrix

$$\mathbf{F}'_{\mathsf{comm}'} := \left[[\boldsymbol{d}^\top|\boldsymbol{a}^\top]\big|\boldsymbol{b}_0 + \sum_{i\in[\ell]}(\hat{t}^{(i)}_{2,1}, \hat{t}^{(i)}_{2,2}, \cdots, \hat{t}^{(i)}_{2,4})\cdot\mathbf{G}^{-1}(\boldsymbol{b}_i)\big|\boldsymbol{a}_2^\top\right].$$

4. With respect to the NIZKPoK system Π for the relaxed language $L_{\gamma',q_2,\bar{\mathcal{C}}}$,

$$L_{\gamma',q_2,\bar{\mathcal{C}}} = \Big\{(\mathbf{F}'_{\mathsf{comm}'}, u) \in R_q^{1\times(2k+4)} \times R_q : \exists\ \boldsymbol{x} \in R_q^{2k+4} \text{ and } f \in \bar{\mathcal{C}} \text{ such that } \|\boldsymbol{x}\|_2 \le \gamma' \text{ and } \mathbf{F}'_{\mathsf{comm}'} \cdot \boldsymbol{x} = f \cdot u\Big\},$$

   we can run the corresponding simulation algorithm to generate a simulated proof π', whose distribution is statistically indistinguishable from that of the real proof π.
5. Output $\widetilde{\mathsf{Sig}}'_{\mathsf{comm}'} := \pi'$.

According to the zero knowledge property of the used NIZKPoK system Π, it is clear that after running any polynomial $t = \mathsf{poly}(\lambda)$ times, the distribution of $\{\widetilde{\mathsf{Sig}}'_{\mathsf{comm}'_i}\}_{i\in[t]}$ output by $\mathcal{S}$ are statistically close to that of $\{\mathsf{Sig}'_{\mathsf{comm}'_i}\}_{i\in[t]}$ output by $\mathsf{Transfer}$. □

Below, we analyse the unforgeability of Construction E.3. Before this, we first specify the exact commitment relation $\hat{L}_{q_2}$ and $\hat{L}_{\hat{\gamma},q_2,\bar{\mathcal{C}}}$. Intuitively, the relation $\hat{L}_{q_2}$ states the validity of the commitments similar to the basic construction. For the second relation, our motivation is to prove that every message m_i is with small ℓ_2 norm (or bounded norm), which can be realized by proving an exact relation that $[\boldsymbol{a}_2^\top, 1] \cdot \begin{bmatrix} \boldsymbol{r}'_{i,1} \\ m_i \end{bmatrix} = t_{2,1}^{(i)}$ for $\begin{bmatrix} \boldsymbol{r}'_{i,1} \\ m_i \end{bmatrix}$ with bounded ℓ_2 norm, where $\boldsymbol{r}'_{i,1} \in S_1^2$ contains the bottom two lines of randomness $\boldsymbol{r}_{i,1}$. For simplicity, a proof for the relaxed relation $\hat{L}_{\hat{\gamma},q_2,\bar{\mathcal{C}}}$ is sufficient. Concretely, the two relations are as follow:

$$\hat{L}_{q_2} := \Big\{\mathsf{comm} = \{\mathsf{comm}_{i,j}\}_{i\in[\ell],j\in[4]} : \exists((m_i)_{i\in[\ell]}, q_2, \{\boldsymbol{r}_{i,j}\}_{i\in[\ell],j\in[4]}) \text{ such that } \boldsymbol{r}_{i,j} \in S_1^3 \text{ with } \|\boldsymbol{r}_{i,j}\|_\infty \le 1 \text{ and } \mathsf{comm}_{i,j} = \mathsf{Commit}(\mathsf{params}, m_i \cdot q_2^{\frac{j-1}{4}}, \boldsymbol{r}_{i,j}) \text{ for } i \in [\ell], j \in [4]\Big\}$$

and

$$\hat{L}_{\hat{\gamma},q_2,\bar{\mathcal{C}}} := \Big\{\Big([\boldsymbol{a}_2^\top, 1] \in R_{q_2}^3, \{t_{2,1}^{(i)} \in R_{q_2}\}_{i\in[\ell]}\Big) : \exists\ \{\boldsymbol{r} \in S_1^2 \times \mathcal{M}, f_i \in \bar{\mathcal{C}}\}_{i\in[\ell]} \text{ such that } \|\boldsymbol{r}\|_2 \le \hat{\gamma} \text{ and } [\boldsymbol{a}_2^\top, 1] \cdot \boldsymbol{r} = f_i \cdot t_{2,1}^{(i)}\Big\}.$$

Lemma E.6 (Unforgeability) *Assume that* $\mathsf{M\text{-}SIS}_{q_2,1,8,\nu}$ *problem and* $\mathsf{M\text{-}SIS}_{q_2,1,8,\nu'}$ *problem are hard with* $\nu = 2k\ell N\alpha\sqrt{B^2 + 1k^2N^2\delta^2}$ *and* $\nu' = \frac{k\ell\gamma'\sqrt{2NB^2+2k^2N^3\delta^2}}{\sqrt{k+2}}$*, then our above lattice-based commitment-transferrable signature scheme is adaptively unforgeable for the exact commitment relation* $\hat{L}_{q_2}$ *and* $\hat{L}_{\hat{\gamma},q_2,\bar{\mathcal{C}}}$*, where* $\hat{\gamma} = 2\sqrt{(2k+4)N} \cdot 11\kappa \cdot (\sqrt{2N + B^2})$*, and* $\gamma' = 2\sqrt{(2k+4)N} \cdot 11\kappa \cdot ((\sqrt{2k}+k)k\ell\alpha N^2\delta + \alpha\sqrt{(2k+4)N})$.

Proof. We argue the unforgeability using the series of hybrids.

$\mathsf{H_0}$: The challenger $\mathcal{B}$ runs the CTS honestly. He gives to the adversary $\mathcal{A}$ the public key pk and signatures with respect to the queried commitments comm_i. In this hybrid, we say $\mathcal{A}$ has advantage $\varepsilon = \mathsf{Adv}_{\mathcal{A}}^{\text{unforge}}(\lambda)$ in the unforgeability game. Then, it holds

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_0}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{\text{unforge}}(\lambda).$$

$\mathsf{H_1}$: The challenger $\mathcal{B}$ runs the identical procedures as H_0, except that he samples $\mathbf{R}_0 \xleftarrow{\$} S_1^{2\times k}$ and $\{\mathbf{R}_i\}_{i\in[\ell]} \xleftarrow{\$} S_1^{2\times k}$, and set $\boldsymbol{b}_i^\top = \boldsymbol{d}^\top \cdot \mathbf{R}_i + h_i \cdot \boldsymbol{g}_\delta^\top \in R_{q_2}^{1\times k}$ for $i \in \{0, 1, \ldots, \ell\}$, where h_i is included in a subfield $\mathcal{S}_{q_2}$ of R_{q_2} of order q_2. According to the Ring-LWE assumption, we know that H_0 and H_1 are computational indistinguishability. Then, it holds

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_0}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_1}(\lambda)| \leq \ell \cdot \mathsf{Adv}_{\mathcal{A}}^{\text{Ring-}\mathsf{LWE}}(\lambda).$$

$\mathsf{H_2}$: The challenger $\mathcal{B}$ runs the identical procedures as H_1, except that except that we add an abort event that is independent of the adversary's view. Specifically, in the final challenge phase, the adversary outputs $(\boldsymbol{m}^*, \mathsf{Rand}^*, \sigma^*)$ as the forgery. $\mathcal{B}$ does the abort check: $h_0 + \langle \boldsymbol{m}_i, \boldsymbol{h}\rangle \neq 0 \bmod q_2 R$ and $h_0 + \langle \boldsymbol{m}_i, \boldsymbol{h}\rangle = 0 \bmod q_2 R$, where $\boldsymbol{h} = (h_1, \cdots, h_\ell) \in \mathcal{S}_{q_2}^\ell$. If the condition does not hold, $\mathcal{B}$ aborts the game.

The only difference between H_1 and H_2 is the abort event. We argue that the adversary still has non-negligible advantage in H_2 even though the abort event happens.

Lemma E.7 *Let I be a $Q_1 + 1$ tuple $(\boldsymbol{m}^*, \boldsymbol{m}_1, ..., \boldsymbol{m}_{Q_1})$ denoted the challenge message $\boldsymbol{m}^*$ along with the queried message's, and $\varepsilon(I)$ define the probability that an abort does not happen in hybrid H_i. Assuming $\varepsilon(I) \in [\varepsilon_{min}, \varepsilon_{max}]$, then we have*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_2}(\lambda) \geq \varepsilon_{min} \cdot \mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_1}(\lambda) - \frac{1}{2}(\varepsilon_{max} - \varepsilon_{min}).$$

$\mathsf{H_3}$: The challenger $\mathcal{B}$ runs the identical procedures as H_2, except that he samples $\boldsymbol{a} \xleftarrow{\$} R^k$, and $\mathcal{B}$ answers the signature queries through using Lemma A.7, rather than Lemma A.8. According to the Ring-LWE assumption, we know that H_2 and H_3 are computational indistinguishability. Then, it holds

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_3}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_2}(\lambda)| \leq \mathsf{Adv}_{\mathcal{A}}^{\text{Ring-}\mathsf{LWE}}(\lambda).$$

Besides, we denote the challenger in H_2 as $\mathcal{B}^*$. Thus, we have

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_3}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{\text{unforge}^*}(\lambda).$$

Lemma E.8 *Let $\mathcal{A}$ be a* PPT *adversary with advantage ε in the adaptive unforgeability game with respect to $\mathcal{B}^*$ for the exact commitment relation $\hat{L}_{q_2}$ and relation $\hat{L}_{\hat{\gamma},q_2,\bar{c}}$, i.e., $\mathsf{Adv}_{\mathcal{A}}^{unforge^*}(\lambda) = \varepsilon$. Let Q_2 be a bound on the number*

of random oracle queries made by $\mathcal{A}$. Let $\nu = 2k\ell N\alpha\sqrt{B^2 + k^2N^2\delta^2}$ and $\nu' = \frac{k\ell\gamma'\sqrt{2NB^2+2k^2N^3\delta^2}}{\sqrt{k+2}}$. Then there exists a reduction algorithm $\mathcal{R}$ for $\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu}$ or $\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu'}$ such that

$$\mathsf{Adv}^{\mathsf{M\text{-}SIS}}_{\mathcal{R}}(\lambda) \geq \varepsilon(\frac{\varepsilon}{Q_2} - 2^{-\lambda}).$$

Proof. According to our construction, the verifier need to consider two cases: original signature and transferred signature. Thus, we need to prove the unforgeability for both cases. Overall, both of them have the similar proof process, and are based on the hardness of $\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu}$ and $\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu'}$ problems, respectively. Below, we present the details for both cases in an unified form, and just separate in their different points.

Particularly, we prove that if the adversary $\mathcal{A}$ can forge a valid original/transferred signature in the selective way, then we can construct an efficient reduction algorithm $\mathcal{B}$ to solve the
$\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu}/\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu'}$ problem. In particular, $\mathcal{B}$ is given an uniformly random matrix $\boldsymbol{x}^\top = [x_1, x_2, \cdots, x_{k+4}] \in R_{q_2}^{k+4}$, and need to output a vector $\boldsymbol{y}$ such that $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = 0 \bmod q_2$ and $\|\boldsymbol{y}\| \leq \nu = 2k\ell N\alpha\sqrt{B^2 + k^2N^2\delta^2}$ or $\|\boldsymbol{y}\| \leq \nu' = \frac{k\ell\gamma'\sqrt{2NB^2+2k^2N^3\delta^2}}{\sqrt{k+2}}$. Similar to the consideration in [23], we choose to use $\boldsymbol{x} = [x_1, x_2, x_3, \cdots, x_{k+2}, 1, x_{k+3}]$, since one of x_i will have an inverse with high probability.

In this case, $\mathcal{B}$ conducts the following steps:

1. Choose $\boldsymbol{x}_1' \xleftarrow{\$} R_{q_1}^2$ and set $\boldsymbol{a}_1^T = (1, \boldsymbol{x}_1'^\top) \in R_{q_1}^3$.
2. Set $\boldsymbol{a}_2^\top = (1, x_{k+3}) \in R_{q_2}^2$.
3. Set $\mathbf{A} = \begin{bmatrix} \boldsymbol{a}_1^\top \\ 0, \boldsymbol{a}_2^\top \end{bmatrix}$ and send it to $\mathcal{A}$.

Clearly, $\mathbf{A}$ is a valid public parameter output by $\Gamma.\mathsf{CKeyGen}$.

Next, we need to argue that $\mathcal{B}$ can simulate the environment of $\mathcal{A}$ successfully for the exact commitment relation $\hat{L}_{q_2,\bar{\mathcal{C}}}$. In particular, we use the following Claim B.5 to specify the case.

Claim E.9 *$\mathcal{B}$ can simulate the environment of $\mathcal{A}$ successfully in the unforgeability game with respect to the exact commitment relation $\hat{L}_{q_2}$ and relation $\hat{L}_{\hat{\gamma},q_2,\bar{\mathcal{C}}}$.*

Proof. $\mathcal{B}$ can set the public parameters in the following way:

1. Set $\boldsymbol{d}^\top = (x_1, x_2) \in R_{q_2}^2$, $\boldsymbol{a}^\top = (x_3, x_4, \cdots, x_{2+k}) \in R_{q_2}^k$.
2. For $i \in [\ell]$, sample $\mathbf{R}_i \xleftarrow{\$} S_1^{2\times k}$, and set $\boldsymbol{b}_i^\top = \boldsymbol{d}^\top \cdot \mathbf{R}_i + h_i \cdot (1, \delta, \cdots, \delta^{k-1}) \in R_{q_2}^{1\times k}$, where $h_i \in S_{q_2}$. Sample $\mathbf{R}_0 \xleftarrow{\$} S_1^{2\times k}$, and set $\boldsymbol{b}_0^\top = \boldsymbol{d}^\top \cdot \mathbf{R}_0 + h_0 \cdot (1, \delta, \cdots, \delta^{k-1}) \in R_{q_2}^{1\times k}$;
3. Send $\mathsf{pk} := (\boldsymbol{a}, \boldsymbol{b}_0, \{\boldsymbol{b}_i\}_{i\in[\ell]})$ to $\mathcal{A}$.

According to the uniformity of $x_3, x_4, \cdots, x_{2+k}$ and the distribution of $\mathbf{R}_0$, pk is a valid public key of our commit-transferrable signature, which follows from the Ring-LWE assumption.

Then, the $\mathcal{A}$ can conduct signature queries and get responds from $\mathcal{B}$. In particular, after receiving the signature query $(\mathsf{comm}, \boldsymbol{m}, \{\boldsymbol{r}_{i,j}\}_{i\in[\ell],j\in[k]})$ from $\mathcal{A}$, where $\mathsf{comm} = \{(\mathsf{comm}_{i,1}, \mathsf{comm}_{i,2}, \cdots, \mathsf{comm}_{i,k})\}_{i\in[\ell]}$ and

$$\mathsf{comm}_{i,1} := \begin{bmatrix} t_{1,1}^{(i)} \\ t_{2,1}^{(i)} \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_{i,1} + \begin{bmatrix} 0 \\ m_i \end{bmatrix},$$

$$\mathsf{comm}_{i,2} := \begin{bmatrix} t_{1,2}^{(i)} \\ t_{2,2}^{(i)} \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_{i,2} + \begin{bmatrix} 0 \\ m_i\delta \end{bmatrix},$$

$$\cdots$$

$$\mathsf{comm}_{i,k} := \begin{bmatrix} t_{1,k}^{(i)} \\ t_{2,k}^{(i)} \end{bmatrix} = \mathbf{A} \cdot \boldsymbol{r}_{i,k} + \begin{bmatrix} 0 \\ m_i\delta^{k-1} \end{bmatrix}.$$

$\mathcal{B}$ can compute

$$\begin{aligned}
\mathbf{F}_{\mathsf{comm}} &= \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{b}_0 + \sum_{i\in[\ell]} \left((t_{2,1}^{(i)}, t_{2,2}^{(i)}, \cdots, t_{2,k}^{(i)}) \cdot \mathbf{G}^{-1}(\boldsymbol{b}_i) \right) \middle| \boldsymbol{a}_2 \right] \\
&= \Bigg[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \Big| \boldsymbol{d}^\top \cdot \mathbf{R}_0 + h_0 \cdot \mathbf{G} \\
&\quad + \sum_{i\in[\ell]} \left((\boldsymbol{a}_2^\top \cdot \mathbf{R}_{i,2} + m_i\mathbf{G}) \cdot \mathbf{G}^{-1}(\boldsymbol{d}^\top \cdot \mathbf{R}_i + h_i\mathbf{G}) \right) \Big| \boldsymbol{a}_2 \Bigg] \\
&= \Bigg[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \Big| \boldsymbol{d}^\top \cdot \mathbf{R}_0 + h_0 \cdot \mathbf{G} + \sum_{i\in[\ell]} \Big(\boldsymbol{a}_2^\top \cdot \mathbf{R}_{i,2} \cdot \mathbf{G}^{-1}(\boldsymbol{d}^\top \cdot \mathbf{R}_i \\
&\quad + h_i\mathbf{G}) + \boldsymbol{d}^\top \cdot m_i\mathbf{R}_{i,2} + m_i h_i \mathbf{G} \Big) \Big| \boldsymbol{a}_2 \Bigg] \\
&= \Bigg[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \Big| \boldsymbol{a}_2^\top \cdot \sum_{i\in[\ell]} \left(\mathbf{R}_{i,2} \cdot \mathbf{G}^{-1}(\boldsymbol{d}^\top \cdot \mathbf{R}_i + h_i\mathbf{G}) \right) + \\
&\quad + \boldsymbol{d}^\top \cdot \left(\mathbf{R}_0 + \sum_{i\in[\ell]} m_i \mathbf{R}_{i,2} \right) + (h_0 + \langle \boldsymbol{m}, \boldsymbol{h} \rangle) \cdot \mathbf{G} \Big| \boldsymbol{a}_2 \Bigg],
\end{aligned}$$

where we denote $\mathbf{R}_i = \begin{bmatrix} \mathbf{R}_{i,1} \\ \mathbf{R}_{i,2} \end{bmatrix} = \left[\boldsymbol{r}_{i,1}, \boldsymbol{r}_{i,2}, \cdots, \boldsymbol{r}_{i,k} \right] \in R^{3\times k}$ with $\mathbf{R}_{i,2} \in R^{2\times k}$.

For any $h_0 + \langle \boldsymbol{m}, \boldsymbol{h} \rangle \neq 0 \bmod q_2 R$, we know that $h_0 + \langle \boldsymbol{m}, \boldsymbol{h} \rangle$ is invertible over the subfield $\mathcal{S}_{q_2}$ of ring R_{q_2}. According to the algorithm in Lemma A.7, the challenger can get a short vector $\boldsymbol{z} \in R^{12}$ such that $\mathbf{F}_{\mathsf{comm}} \cdot \boldsymbol{z} = 0$. □

From above Claim B.5, we know that $\mathcal{B}$ can simulate the environment of $\mathcal{A}$ successfully.

Next, for the challenge query of the form $(\mathsf{comm}^*, \boldsymbol{m}^*, \{\boldsymbol{r}^*_{i,j}\}_{i\in[\ell],j\in[k]})$, we have

$$\mathbf{F}_{\mathsf{comm}^*} = \Bigg[[\boldsymbol{d}^\top|\boldsymbol{a}^\top]\big|\boldsymbol{a}_2^\top \cdot \sum_{i\in[\ell]} \Big(\mathbf{R}^*_{i,2}\mathbf{G}^{-1}(\boldsymbol{d}^\top \cdot \mathbf{R}_i + h_i\mathbf{G})\Big) + \boldsymbol{d}^\top \cdot (\mathbf{R}_0 + \sum_{i\in[\ell]} m_i^*\mathbf{R}^*_{i,2})\big|\boldsymbol{a}_2^\top\Bigg].$$

Below, according to the fact that the adversary's forgery is for original signature or transferred one, we need to separate the following proof into two cases.

For the case of original one. If the adversary can forge a valid signature $\mathsf{Sig}_{\mathsf{comm}^*} := \begin{bmatrix} \boldsymbol{s}^*_{1,1} \\ \boldsymbol{s}^*_{1,2} \\ \boldsymbol{s}^*_2 \\ \boldsymbol{s}^*_3 \end{bmatrix}$ with $\boldsymbol{s}^*_3 = (s^*_{3,1}, s^*_{3,2})^T \in R^2$, such that

$$\begin{aligned}
&\mathbf{F}_{\mathsf{comm}^*} \cdot \mathsf{Sig}_{\mathsf{comm}^*} = \\
&\left[[\boldsymbol{d}^\top|\boldsymbol{a}^\top]\big|\boldsymbol{a}_2^\top \cdot \sum_{i\in[\ell]} \mathbf{R}^*_{i,2}\mathbf{G}^{-1}(\boldsymbol{p}) + \boldsymbol{d}^\top \cdot \mathbf{R}^*\big|\boldsymbol{a}_2\right] \cdot \begin{bmatrix} \boldsymbol{s}^*_{1,1} \\ \boldsymbol{s}^*_{1,2} \\ \boldsymbol{s}^*_2 \\ \boldsymbol{s}^*_3 \end{bmatrix} \\
=&\langle \boldsymbol{d}, \boldsymbol{s}^*_{1,1}\rangle + \langle \boldsymbol{a}, \boldsymbol{s}^*_{1,2}\rangle + \langle \boldsymbol{a}_2^\top \cdot \sum_{i\in[\ell]} \mathbf{R}^*_{i,2}\mathbf{G}^{-1}(\boldsymbol{p}) + \boldsymbol{d}^\top \cdot \mathbf{R}^*, \boldsymbol{s}^*_2\rangle \\
&+ \langle \boldsymbol{a}_2, \boldsymbol{s}^*_3\rangle \\
=&\, 0,
\end{aligned}$$

where $\boldsymbol{p} = \boldsymbol{d}^\top \cdot \mathbf{R}_i + h_i\mathbf{G}, \mathbf{R}^* = \mathbf{R}_0 + \sum_{i\in[\ell]} m_i^*\mathbf{R}^*_{i,2}$, then $\mathcal{B}$ can compute $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{s}^*_{1,1} + \mathbf{R}^* \cdot \boldsymbol{s}^*_2 \\ \boldsymbol{s}^*_{1,2} \\ \boldsymbol{s}^*_3 + \sum_{i\in[\ell]} \mathbf{R}^*_{i,2}\mathbf{G}^{-1}(\boldsymbol{p}) \cdot \boldsymbol{s}^*_2 \end{bmatrix}$ as a solution to the $\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu}$ problem defined by $[x_1, x_2, x_3, \cdots, x_{k+2}, 1, x_{k+3}]$. And the ℓ_2 norm of this solution is less than $\|\boldsymbol{y}\| \le \alpha\sqrt{(k+4)N} + (k+\sqrt{2k})N\alpha\sqrt{(\ell B+1)^2 + k^2N^2\ell^2\delta^2} \le 2k\ell N\alpha\sqrt{B^2 + k^2N^2\delta^2}$.

For the case of transferred one. If the adversary can forge a valid proof for the language $L_{\gamma',q_2,\bar{\mathcal{C}}}$, then the reduction algorithm $\mathcal{B}$ can run the extractor of the $\mathsf{NIZKPoK}$ system Π_2, and get a ℓ_2 norm short vector $\mathsf{Sig}'_{\mathsf{comm}^*} := \begin{bmatrix} \boldsymbol{s}^*_{1,1} \\ \boldsymbol{s}^*_{1,2} \\ \boldsymbol{s}^*_2 \\ \boldsymbol{s}^*_3 \end{bmatrix}$

with $\boldsymbol{s}_3^* = (s_{3,1}^*, s_{3,2}^*)^\top \in R^2$, such that

$$\begin{aligned}
&\mathbf{F}_{\mathsf{comm}^*} \cdot \mathsf{Sig}_{\mathsf{comm}^*} \\
&= \left[[\boldsymbol{d}^\top | \boldsymbol{a}^\top] \middle| \boldsymbol{a}_2^\top \cdot \sum_{i\in[\ell]} \mathbf{R}_{i,2}^* \mathbf{G}^{-1}(\boldsymbol{p}) + \boldsymbol{d}^\top \cdot \mathbf{R}^* \middle| \boldsymbol{a}_2 \right] \cdot \begin{bmatrix} \boldsymbol{s}_{1,1}^* \\ \boldsymbol{s}_{1,2}^* \\ \boldsymbol{s}_2^* \\ \boldsymbol{s}_3^* \end{bmatrix} \\
&= \langle \boldsymbol{d}, \boldsymbol{s}_{1,1}^* \rangle + \langle \boldsymbol{a}, \boldsymbol{s}_{1,2}^* \rangle + \langle \boldsymbol{a}_2^\top \cdot \sum_{i\in[\ell]} \mathbf{R}_{i,2}^* \mathbf{G}^{-1}(\boldsymbol{p}) + \boldsymbol{d}^\top \cdot \mathbf{R}^*, \boldsymbol{s}_2^* \rangle \\
&\quad + \langle \boldsymbol{a}_2, \boldsymbol{s}_3^* \rangle \\
&= 0,
\end{aligned}$$

then $\mathcal{B}$ can compute $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{s}_{1,1}^* + \mathbf{R}^* \cdot \boldsymbol{s}_2^* \\ \boldsymbol{s}_{1,2}^* \\ \sum_{i\in[\ell]} \mathbf{R}_{i,2}^* \mathbf{G}^{-1}(\boldsymbol{p}) \cdot \boldsymbol{s}_2^* + \boldsymbol{s}_3^* \end{bmatrix}$ as a solution to the $\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu'}$ problem defined by $[x_1, x_2, x_3, \cdots, x_{k+2}, 1, x_{k+3}]$. And the ℓ_2 norm of this solution is less than $\|\boldsymbol{y}\| \leq \alpha' \sqrt{(k+4)N} + (k + \sqrt{2k}) N\alpha' \cdot \sqrt{(\ell B + 1)^2 + k^2 N^2 \delta^2} \leq 2k\ell N \alpha' \sqrt{B^2 + k^2 N^2 \delta^2}$, with $\alpha' = \gamma' / \sqrt{(2k+4)N}$.

Furthermore, according to the forking lemma of [8, 53], $\mathcal{R}$ can complete the above reduction with probability at least $\varepsilon(\frac{\varepsilon}{Q_2} - 2^{-\lambda})$.

Summing up all above arguments, we conclude that our commit transferrable signature satisfies unforgeable in the adaptive way. □

Completing the Proof. Recall that $|Q_1|$ is the upper bound of the number of the adversary's signing queries, and ε_1 is the advantage of the adversary in H_1. By Lemma E.1 and E.2, we can know that

$$\begin{aligned}
&\Pr_H \left[H(\boldsymbol{m}^*) = 0 \bigwedge H(\boldsymbol{m}_1) \neq 0 \bigwedge \cdots \bigwedge H(\boldsymbol{m}_{|Q_1|}) \neq 0 \right] \\
&\in \left(\frac{1}{q_2^\tau}(1 - \frac{Q_1}{q_2^\tau}), \frac{1}{q_2^\tau} \right).
\end{aligned}$$

Thus, we know that for any (Q_1+1)-tuple I denoting a challenge $\boldsymbol{m}^*$ along with signing queries, we have $\varepsilon(I) \in \left(\frac{1}{q_2^\tau}(1 - \frac{Q_1}{q_2^\tau}), \frac{1}{q_2^\tau} \right)$. Then by setting $[\varepsilon_{min}, \varepsilon_{max}] = \left[\frac{1}{q_2^\tau}(1 - \frac{Q_1}{q_2^\tau}), \frac{1}{q_2^\tau} \right]$ in Lemma E.7, we have

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_2}(\lambda) \geq \frac{1}{q_2^\tau}(1 - \frac{Q_1}{q_2^\tau})\varepsilon_1 - \frac{Q_1}{2q_2^{2\tau}}.$$

By our parameter setting, $|Q| \leq \frac{1}{2}\varepsilon_1 q_2^\tau$, we have that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_2}(\lambda) \geq \frac{1}{q_2^\tau}(1 - \frac{Q_1}{q_2^\tau})\varepsilon_1 - \frac{Q_1}{2q_2^{2\tau}} \geq \frac{1}{4q_2^\tau} \cdot \mathsf{Adv}_{\mathcal{A}}^{\mathsf{H}_1}(\lambda).$$

In summary, we have that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{H}_0}_{\mathcal{A}}(\lambda) &\leq \mathsf{Adv}^{\mathsf{H}_1}_{\mathcal{A}}(\lambda) + \ell \cdot \mathsf{Adv}^{\text{Ring-LWE}}_{\mathcal{A}}(\lambda) \\ &\leq \ell \cdot \mathsf{Adv}^{\text{Ring-LWE}}_{\mathcal{A}}(\lambda) + 4q_2^{\tau}\mathsf{Adv}^{\mathsf{H}_2}_{\mathcal{A}}(\lambda) \\ &\leq (\ell + 4q_2^{\tau}) \cdot \mathsf{Adv}^{\text{Ring-LWE}}_{\mathcal{A}}(\lambda) + 4q_2^{\tau}\mathsf{Adv}^{\mathsf{H}_3}_{\mathcal{A}}(\lambda) \\ &\leq (\ell + 4q_2^{\tau}) \cdot \mathsf{Adv}^{\text{Ring-LWE}}_{\mathcal{A}}(\lambda) + 4q_2^{\tau}\sqrt{(\mathsf{Adv}^{\mathsf{M\text{-}SIS}}_{\mathcal{R}}(\lambda) + \frac{1}{2^{\lambda}})Q_2},\end{aligned}$$

which completes the proof. □

E.4 Instantiation of NIZKPoK for Construction E.3

In this part, we instantiate the NIZKPoK involved in Construction E.3. There are three relations needed to prove:
(1)

$$\begin{aligned}L_{\gamma',q,\bar{\mathcal{C}}} = \Big\{&(\mathbf{F}_{\mathsf{comm}'}, 0) \in R_q^{1\times(2k+4)} \times R_q : \exists\ \boldsymbol{x} \in R_q^{(2k+4)} \text{ and} \\ &f \in \bar{\mathcal{C}} \text{ such that } \|\boldsymbol{x}\|_2 \leq \gamma' \text{ and } \mathbf{F}_{\mathsf{comm}'} \cdot \boldsymbol{x} = 0\Big\},\end{aligned}$$

(2)

$$\begin{aligned}\hat{L}_{q_2} := \Big\{&\mathsf{comm} = \{\mathsf{comm}_{i,j}\}_{i\in[\ell],j\in[4]} : \exists((m_i)_{i\in[\ell]}, q_2, \{\boldsymbol{r}_{i,j}\}_{i\in[\ell],j\in[4]}) \\ &\text{such that } \boldsymbol{r}_{i,j} \in S_1^3 \text{ with } \|\boldsymbol{r}_{i,j}\|_\infty \leq 1 \text{ and} \\ &\mathsf{comm}_{i,j} = \mathsf{Commit}(\mathsf{params}, m_i \cdot q_2^{\frac{j-1}{4}}, \boldsymbol{r}_{i,j}) \text{ for } i \in [\ell], j \in [4]\Big\},\end{aligned}$$

(3)

$$\begin{aligned}\hat{L}_{\hat{\gamma},q_2,\bar{\mathcal{C}}} = \Big\{&\Big([\boldsymbol{a}_2^{\top}, 1] \in R_{q_2}^3, \{t_{2,1}^{(i)} \in R_{q_2}\}_{i\in[\ell]}\Big) : \exists\ \{\boldsymbol{r} \in S_1^2 \times \mathcal{M}, f_i \in \bar{\mathcal{C}}\}_{i\in[\ell]} \\ &\text{such that } \|\boldsymbol{r}\|_2 \leq \hat{\gamma} \text{ and } [\boldsymbol{a}_2^{\top}, 1] \cdot \boldsymbol{r} = f_i \cdot t_{2,1}^{(i)}\Big\}.\end{aligned}$$

The first relation can be proved by the same NIZKPoK system Π_1 as Theorem 4.3. For the second relation, we can apply the multi-theorem straight-line extractable NIZKPoK system Π_2 for ℓ times to prove it. The third relation can be proved similar to the first relation except that we need run a similar NIZK system Π_3 as Theorem 4.3 for ℓ times, with the slightly different parameters. Particularly,

Theorem E.10 ([6, 23, 26]) *In the random oracle model, assuming the hardness of* $\mathsf{M\text{-}SIS}_{q_2,1,2k+4,\gamma'}$*, there exists a* NIZKPoK *system* Π *for the relaxed language* $L_{\gamma',q_2,\bar{\mathcal{C}}}$*, with* $\gamma' = 2\sqrt{(2k+4)N} \cdot \eta\kappa \cdot ((\sqrt{2k}+k)k\ell\alpha N^2\delta + \alpha\sqrt{(2k+4)N})$.

Moreover, assuming a t*-time adversary* $\mathcal{A}$ *forging a proof with probability* ε*, there exists a* $O(t/\varepsilon)$*-time extractor, who can successfully extract the witness* $\boldsymbol{x}$ *and* $c \in \bar{\mathcal{C}}$ *with probability* $\frac{1}{2}$.

Theorem E.11 ([6, 23, 26]) *In the random oracle model, assuming the hardness of* $\mathsf{M\text{-}SIS}_{q_2,1,3,\hat{\gamma}}$*, there exists a* NIZK *system* Π_3 *for the relaxed language* $L_{\hat{\gamma},q_2,\bar{\mathcal{C}}}$*, with* $\hat{\gamma} = 2\sqrt{(2k+4)N} \cdot \eta\kappa \cdot (\sqrt{2N + B^2})$.

E.5 Parameter Settings of Construction E.3

In this part, we set the concrete parameters for Construction E.3 and the straight-line extractable NIZKPoK system, according to the related requirements in correctness and security. For clarity, we denote the straight-line extractable NIZKPoK system for $\hat{L}_{q_2}$ in Section 5 as Π_1, denote the NIZKPoK system for $L_{\gamma',q_2,\bar{\mathcal{C}}}$ in Section 4.2 as Π_2, and denote the NIZKPoK system for $\hat{L}_{\hat{\gamma},q_2,\bar{\mathcal{C}}}$ as Π_3.
Requirements for Correctness. We require the following:

- The SamplePre in the Sign step needs to work properly. according to Lemma A.3, we need to set $\alpha \geq 2\sqrt{\delta^2+1}\cdot((\sqrt{k}+\sqrt{2})\sqrt{N}+1)$.
- The valid original signature $\mathsf{Sig}_{\mathsf{comm}}$ can be verified successfully. According to Lemma A.2, we need to set $\gamma = \alpha\sqrt{(2k+4)\cdot N}$.
- The valid transferred signature $\mathsf{Sig}'_{\mathsf{comm}'}$ can be verified successfully. According to Lemma E.4 and the relaxed language in Theorem 4.3, we need to set $\gamma' = 2\sqrt{(2k+4)N}\cdot\eta\kappa\cdot((\sqrt{2k}+k)k\ell\alpha N^2\delta + \alpha\sqrt{(2k+4)N})$.
- Ciphertexts of Construction C.2 can be decrypted correctly. According to the corresponding analysis, we need set $\hat{N}|N$, $k'\hat{N} \leq \lfloor\sqrt{q_1}\rceil/4$.

Requirements for Security. We require the following:

- The ring R is cyclotomic, i.e., $R = \mathbb{Z}[X]/(\Phi_m(X))$, where $\Phi_m(X)$ is the m^{th} cyclotomic polynomial, and denote $N = \varphi(m)$. Here, we consider the cyclotomic polynomials $\Phi_m(X) = X^N + 1$ with N to be a power of 2.
- There exists an exact NIZKPoK system Π_1 for the commitment relation $\hat{L}_{q_2}$. Hence, according to Theorem 5.2 and Corollary 2, the problems

  1. $\mathsf{M\text{-}SIS}_{q_1,1,3,\eta_1}$, $\mathsf{M\text{-}LWE}_{q_2,1,1}$ and $\mathsf{M\text{-}SIS}_{q_1,1,3,5.6\cdot\kappa^2\cdot 3\cdot N}$ over $R = \mathbb{Z}[X]/\langle X^N+1\rangle$, with $\eta_1 = 66\cdot\kappa\cdot N$, N to be a power of 2, prime q_1 such that x^N+1 is fully-splitting, prime $q_2 = 3$ or 5 mod 8,
  2. $\mathsf{M\text{-}SIS}_{q_1,n,k,\eta_2}$, $\mathsf{M\text{-}LWE}_{q_1,\hat{\lambda},k}$ and $\mathsf{M\text{-}SIS}_{q_1,n,k,88\cdot\kappa^2\cdot k\cdot\hat{N}}$ over $\hat{R} = \mathbb{Z}[X]/\langle X^{\hat{N}}+1\rangle$, with $\eta_2 = 242\cdot\kappa\cdot\hat{N}$, $\hat{N}$ to be power of 2, prime q_1 such that $X^{\hat{N}}+1$ to be fully-splitting,

  need to be hard.
- There exists an NIZKPoK system Π_2 for the language $L_{\gamma',q_2,\bar{\mathcal{C}}}$. According to Theorem 4.3, the problem $\mathsf{M\text{-}SIS}_{q_2,1,2k+4,\gamma'}$ needs to be hard.
- There exists an NIZKPoK system Π_3 for the language $L_{\hat{\gamma},q_2,\bar{\mathcal{C}}}$. According to Theorem 4.3, the problem $\mathsf{M\text{-}SIS}_{q_2,1,3,\hat{\gamma}}$ needs to be hard.
- The constructed CTS satisfies unforgeability in Definition 3.3. Particularly,
  - For Definition 3.3 with respect to the exact commitment relation $\hat{L}_{q_2,\bar{\mathcal{C}}}$, according to Lemma E.8, and Claim E.9, we need to set $\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu}$ problem and $\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu'}$ being hard with $\nu = 2k\ell N\alpha\sqrt{B^2+k^2N^2\delta^2}$ and $\nu' = \frac{k\ell\gamma'\sqrt{2NB^2+2k^2N^3\delta^2}}{\sqrt{k+2}}$.
- The underlying BDLOP satisfies hiding and binding. Hence, according to Section 2.3, we need to set $\mathsf{M\text{-}LWE}_{q_2,1,1}$ and $\mathsf{M\text{-}SIS}_{q_1,1,3,88\cdot\kappa^2\cdot 3\cdot N}$ being hard.
- The successful simulation of the adversary in Claim E.9. Here, according to the Lemma A.7, we need to set $\alpha \geq 2\sqrt{\delta^2+1}\cdot((\sqrt{k}+\sqrt{2})\cdot\sqrt{N}(\ell B+k\ell N\delta+1)+1)$.

More specifically, we have the concreted parameter setting in the following Table 15.

Below, we roughly explain about the calculations of the above table.

- According to the used NIZKPoK system Π_1 in Theorem 5.2 and Corollary 2, we need to set $\eta_1 = 66 \cdot \kappa \cdot N$, $\eta_2 = 242 \cdot \kappa \cdot N$, N to be power of 2, prime q_1 such that $X^N + 1$ to be fully-splitting, prime $q_2 = 3$ or 5 mod 8.
- According to the used NIZKPoK system Π_2 in Theorem E.10, we need to set $\gamma' = 2\sqrt{(2k+4)N} \cdot 11\kappa \cdot ((\sqrt{2k}+k)k\ell\alpha N^2\delta + \alpha\sqrt{(2k+4)N})$, such that the assumption $\mathsf{M\text{-}SIS}_{q_2,1,2k+4,\gamma'}$ is hard, and a NIZKPoK system Π_2 exists for the relaxed language $L_{\gamma',q_2,\bar{\mathcal{C}}}$.
- Given the concrete value of N, we need to fix κ such that the size of the challenge set is larger than 2^{256}, i.e., $\binom{N}{\kappa} \times 2^\kappa \geq 2^{256}$.
- Given N, q_2, κ, we can calculate the values of α, γ, γ' (all these parameters need to be used in the description of our CTS in Construction E.3), according to the above parameter analysis for correctness and security.
- As a reasonable setting, we assume the upper bound of the number of queries that the adversary can make to be 2^{64}.
- We can further compute the values of ν, ν' (all these parameters need to be used to ensure the security proof of our CTS in Construction E.3), as the requirement of security proof.
- In order to obtain much better tradeoff between efficiency and security, we first choose modulus q_2 such that both the hiding (based on $\mathsf{M\text{-}LWE}_{q_2,1,1}$) and the unforgeability (based on $\mathsf{M\text{-}SIS}_{q_2,1,k+4,\nu'}$) properties have the sufficient security level.
- Then, we set n, $\hat{N}$, λ, and q_1, such that the additional underlying assumptions for for NIZKPoK Π_1 also have sufficient hardness.

During the above calculation process, we use the Root-Hermite Factor δ_0 to estimate bit-hardness of the underlying assumptions, i.e., M-SIS and M-LWE, according to the best known attacks, and δ_0 can be determined given N, q_1, q_2, α. Generally, we can use the work [3, 4, 33] to estimate δ_0 and its corresponding hardness of the assumptions.

Our reduction from each building block is essentially tight (by calling the adversary a constant number of times), so the attained security of our construction is essentially the same as that of the underlying M-LWE/M-SIS problem.

Parameters	Description
N	Ring dimension
$\widehat{N}$	Ring dimension for the straight-line extractable NIZKPoK
R	Cyclotomic Ring used in this work
q_1 q_2	Moduli used for BDLOP commitment scheme
$\mathcal{M}$ $\bar{\mathcal{M}}$ ℓ	Message space of the commitment, which consists of ℓ subspace $\bar{\mathcal{M}}$. And $\bar{\mathcal{M}}$ is contained in a subset of the subfield of R_{q_2}, according to Corollary 1
B	ℓ_2 norm of elements in $\bar{\mathcal{M}}$ are bounded by B
δ k	the basis and dimension of the gadget vector $\boldsymbol{g}$ i.e., $\boldsymbol{g}^\top = (1, \delta, \ldots, \delta^{k-1})$
n k'	the numbers of rows and columns of matrix for the straight-line extractable NIZKPoK
$\hat{\lambda}$	The dimension of secret key of M-LWE in the straight-line extractable NIZKPoK
S_β	Set of all elements in R with ℓ_∞ norm at most β
α	Parameter used in SamplePre
γ	ℓ_2 norm parameter used in Verify algorithm for original signature
$\mathcal{C}$ κ	Challenge set of the NIZKPoK system Π $\mathcal{C} = \{c \in R : \|c\|_1 = \kappa, \|c\|_\infty = 1\}$
$\bar{\mathcal{C}}$	The set of differences $\mathcal{C} - \mathcal{C}$ except 0
γ'	ℓ_2 norm parameter for "short" vectors in the language of Π
$\hat{\gamma}$	ℓ_2 norm parameter for "short" vectors in the valid commitment relation, (i.e., $\hat{L}_{\hat{\gamma},q_2,\bar{\mathcal{C}}}$)
δ_0	Root-Hermite Factor
Bit-sec	Bit-security in time

Table 14. Parameters of Adaptive Commit-Transferrable Signature Scheme

	Params Example
N	2^{15}
$\hat{N}$	2^{11}
q_1	$\sim 2^{40}$
q_2	$\sim 2^{140}$
τ	1
δ	140
k	140
B	2^{16}
Q_2	2^{64}
ℓ	16
n	2
$\hat{\lambda}$	4
k'	89
α	$2^{40.528}$
γ	$2^{52.103}$
κ	22
γ'	$2^{110.443}$
$\hat{\gamma}$	$2^{36.494}$
δ_0	1.001142
Bit-sec of underlying assumptions	616.996
Bit-sec of concrete construction	134

Table 15. Concrete Settings for the Parameters and the Related Security in the case of unforgeability with exact relation.