Atlantis: Lattice-based Anonymous Tokens with Private Metadata Bit

Foteini Baldimtsi
George Mason University
& Mysten Labs
foteini@gmu.edu

Aayush Yadav
George Mason University
ayadav5@gmu.edu

Abstract

Anonymous tokens with private metadata bit (ATPM) allow an issuer to embed a hidden trust flag, as a single bit, within issued tokens. The bit remains hidden from the clients, but verifiers can read the bit and rate-limit or discard tokens marked suspect. A series of ATPM constructions exist in the literature, however all current constructions rely on classical hardness assumptions such as RSA groups, pairings, or elliptic-curve VRFs and do not provide any post-quantum security guarantees.

In this work we present, the first ATPM scheme based on lattice assumptions. Tokens generated with our scheme are publicly verifiable, and privately bit-extractable given partial knowledge of the issuing authority's secret. Our design follows the Fischlin blind-signature paradigm and enriches it with lattice-based linearly-homomorphic encryption to carry the hidden bit.

We also instantiate our scheme from Falcon-512 and the efficient LNP22 lattice NIZK proof system (Lyubashevsky et. al, Crypto '22). The resulting protocol, which we call Atlantis, requires 70 KB of client-issuer communication and yields 129 KB tokens.

Disclosure. We have been alerted to attacks affecting the one-more unforgeability and metadata-bit privacy of our lattice-based instantiations. We have identified potential minimal repairs and are currently evaluating them. An updated draft with the repaired construction is forthcoming. In the meantime, an explanation of the attacks is given in Appendix D. We thank Baiyu Li and Jonathan Katz for identifying these issues.

1 Introduction

Anonymous tokens are a lightweight yet powerful primitive for propagating trust signals on the Web while protecting users' identities. A typical anonymous tokens protocol involves three roles: an *issuer*, a *client* and a *verifier* (or redemptee). The issuer issues tokens to trustworthy clients, and the clients can then show the tokens to verifiers, who only need to verify the validity of the token. The main goal of such schemes is to guarantee token unforgeability while issuance and redemption transcripts remain unlinkable.

Anonymous tokens have been proposed for a variety of applications such as CAPTCHA-free access to Content Delivery Networks (CDNs) [DGS+18], privacy-preserving web browsing [App21], private contact tracing [SS22], fraud detection [SS22], and privacy-safe click measurement [Wil21], to name only a few. At the same time, major internet companies are considering anonymous-token technology: projects like Google's Private State Tokens [Goo24] and Cloudflare's Privacy Pass [MRFH24] have sparked widespread interest, and an IETF working group supported by Google, Apple, and Cloudflare (among others) is now working toward a unified standard [IET24].

One approach for building anonymous token schemes is to use ideas from anonymous-credential systems [Cha90, CL01]. Although both primitives aim to protect users' privacy, anonymous credentials provide richer functionality as they can embed user attributes, allow selective disclosure, and be reused, with presentations carried out via zero-knowledge proofs. Anonymous tokens, by design, are far simpler since they contain no attributes, are intended for a single redemption, and must satisfy strict efficiency requirements so their presentation protocols are generally light-weight.

One-time use anonymous tokens were introduced in Privacy Pass [DGS+18] to prevent DDoS attacks on content delivery networks (CDN) while also maintaining a high degree of privacy for clients of Cloudflare-based services without compromising the user experience. However, a flaw in Privacy Pass and similar anonymous token protocols is that once an issuer realizes that a client is malicious, they will simply refuse their future issuance; thus, alerting the clients to their detection. Such a signal could be exploited by bad-actors to bypass the servers' security mechanisms.

To address the issue of overt rejection signals, Kreuter et al. proposed an enhanced design of an anonymous token featuring an embedded private metadata bit [KLOR20]. This novel idea of *anonymous tokens with private metadata bit* (ATPM), extends the original Privacy Pass protocol by enabling issuers to embed a discrete trust signal into each token. Using CDNs as a running example, adopting anonymous tokens with private metadata bit allows the issuer to secretly set the metadata bit (for example, $0 = \mathsf{trusted}$ and $1 = \mathsf{suspect}$) while the client still receives what looks like an ordinary, unlinkable token. Because the client never learns the bit's value the issuer can downgrade or throttle suspicious traffic later, when the token is redeemed, without tipping off the attacker. The verifier can read the bit and apply more fine-grained policies such as rate-limiting, selective CAPTCHAs, or silent dropping, all while preserving the anonymity and unforgeability guarantees of the underlying scheme.

A number of constructions of anonymous tokens with private metadata bit have appeared in the academic literature [KLOR20, BLOR22, SS22, CDV23, OTZZ24, BHNY25]. Proposed schemes consider a variety of features such as public and private verifiability, support for additional public metadata, non-interactive issuance, or built-in rate-limit enforcement, and they are constructed based on various cryptographic building blocks such as verifiable oblivious pseudorandom functions (VOPRFs), algebraic MACs, and blind signatures. Yet *none* of the existing designs is secure against quantum adversaries, leaving **post-quantum ATPM constructions** as an emerging open problem. The urgency is heightened by NIST's transition roadmap, echoing the National Security Memorandum 10, which sets the year 2035 as the target date for phasing out pre-quantum public-key algorithms [NIS24].

Our contributions. The contributions of this work are summarized as follows:

- We give the *first* ATPM construction from *lattice assumptions* (in the random oracle model). Our construction is based on blind signatures and supports public verifiability of tokens. Additionally, we also give a way to extend our construction to support public metadata with no overhead.
- We present Atlantis[1], a practical instantiation of our ATPM construction using [PFH+22] and the efficient NIZK for general linear relations due to [LNP22]. Our scheme gives 70 KB transcripts and 129 KB token size assuming hardness of (standard) LWE and one-more ISIS assumptions [AKSY22].
- As a side contribution, we make the observation that our scheme is in fact an instantiation to the well-known Fischlin's paradigm [Fis06], and applies more widely towards upgrading any Fischlin-type round-optimal blind signature scheme to an anonymous token scheme with private metadata bit. This partially addresses an open problem raised in a recent work by Amjad, Yeo and Yung [AYY23] concerning the extension of blind signature schemes to additionally support a private metadata bit.

1.1 Related work

The original construction of ATPMs due to Kreuter et al. [KLOR20] relies on verifiable oblivious pseudo-random functions (VOPRFs) and is proven secure in the random oracle model, maintaining similar security guarantees as Privacy Pass while offering a subtler method of misconduct detection. However, limitation of VOPRF based techniques is that verification of the token can only be performed by parties with the secret issuing key. In contrast, publicly verifiable anonymous tokens with private metadata bit [BLOR22] offer an alternative approach where the validity of the token can be checked by any party using the issuer's public key, and yet the embedded metadata remains hidden without the secret key. Although publicly verifiable schemes can also be constructed using techniques such as RSA blind signatures [AYY23], the current RSA-based variant does not support the inclusion of private metadata. This trade-off illustrates how design choices affect both verification flexibility and the ability to conceal internal indicators of trust.

Building on these ideas, Chase et al. [CDV23] developed a protocol for anonymous tokens with hidden metadata bit from algebraic MACs, which is tailored for scenarios in which the issuer and verifier are the same entity. Their construction leverages algebraic MACs and achieves security in the generic group model (GGM). In addition, it can be naturally extended to support tokens that carry public metadata if required.

In a recent work, Baldimtsi et al. [BHNY25] proposed the notion of non-interactive anonymous tokens (NIAT), where the idea is to omit the first round message from the client by instead allowing the client and issuer to jointly compute the final message in an unlinkable way. Their NIAT construction is in pairing-friendly groups and is also proven secure in the GGM. They further provide an extension to their protocol that allows them to identify clients that attempt to double-spend a token by revealing an associated public key.

Other related ideas include anonymous counting tokens [BRS23] which limits the number of tokens per-message that are redeemed with any given verifier; and non-transferable anonymous tokens [DMTV24] where the tokens are cryptographically bound to the original client, preventing sharing of tokens.

Lattice-based Anonymous Tokens. Two recent works have considered lattice-based variants for anonymous tokens. In [PWFW23], authors construct a post-quantum secure anonymous credential scheme using a STARK-friendly Dilithium signature protocol [DKL+18] and use it to instantiate an anonymous token

[1] Atlantis is an acronym for Anonymous tokens from lattices with NTRU and ISIS

scheme (albeit without a private metadata bit). More recently, Yan et al. proposed a [YCN+25] an ATPM scheme with private verification under the hashed-MLWE assumption [BLNS23].

Lattice-based VOPRFs. Albrecht et al. have proposed two iterations of reasonably practical VOPRFs from structured lattice assumptions [ADDS21, AG24]. We remark that it is quite plausible that these ideas can be used to construct ATPMs, however, as with other VOPRF based schemes, these would only give privately verifiable tokens. In this work, we focus on the more desirable goal of public verifiability and thus propose a blind signature based construction.

1.2 Overview

We now give an overview of the main technical ideas of this work. To motivate our construction, we recall that an anonymous token with public verifiability (and *without* a private metadata bit) is nothing but a blind signature. Thus, as a first attempt we will try to generically upgrade a blind signature scheme to support the embedding of a private metadata bit and explain the challenges with this simple approach.

ATPM from blind signatures. The natural idea for embedding a private bit b inside a round-optimal blind signature scheme would be to have the issuer encrypt b into a ciphertext $\hat{\phi}$ (and prove correctness of encryption[2]), so that it can later extract the bit from $\hat{\phi}$ using a decryption key that only it knows. The astute reader will immediately identify the issue with this straw-man approach, namely that it will require having the client send the same ciphertext in the clear as part of the final signature which allows the issuer to break unlinkability.

Instead of sending $\hat{\phi}$ directly, we wish for the client to be able to homomorphically evaluate it in a way that the resulting ciphertext is unlinkable to $\hat{\phi}$, but also (somehow) preserves information about the original bit so that the signer is still able to decrypt the randomized ciphertext and learn b. Our main technical insight is that the resolution to this dilemma lies in one crucial object that we explain next.

Our candidate construction for upgrading any round-optimal blind signature scheme to an ATPM scheme requires an IND-CPA secure *linearly homomorphic* public key encryption scheme. A linearly homomorphic PKE scheme LinHE has the unique property that, For $b_0, b_1 \in \{0,1\}$,

$$\mathsf{LinHE.Enc}(\mathsf{pk}, b_0) \;+\; \mathsf{LinHE.Enc}(\mathsf{pk}, b_1) \;=\; \mathsf{LinHE.Enc}(\mathsf{pk}, b_0 \oplus b_1)\;,$$

and this is the property that we can leverage to embed the private bit. More concretely, the issuer computes $\hat{\phi} \leftarrow \mathsf{LinHE.Enc}(\mathsf{pk}, b)$, signs it along with the blinded message and sends over $\hat{\phi}$, the (pre)signature $\hat{\sigma}$ over to the client along with a proof $\hat{\pi}$ of well-formedness of the ciphertext. After verifying the signature and the proof, the client evaluates two ciphertexts, ϕ_0 and ϕ_1 as

$$\forall b' \in \{0,1\} \;:\; \phi_{b'} \leftarrow \hat{\phi} + \mathsf{LinHE.Enc}(\mathsf{pk}, b')\;,$$

and additionally proves that each $\phi_{b'}$ was computed correctly. Due to the linearly homomorphic nature of the encryption scheme, it holds that $\phi_{b'} = \mathsf{LinHE}(\mathsf{pk}, b \oplus b')$. Then, on receiving ϕ_0 and ϕ_1 as part of the final signature, the issuer can easily extract b since for any $b' \in \{0,1\}$, $b = b'$ if and only if $\mathsf{LinHE.Dec}(\mathsf{sk}, \phi_{b'}) = 0$. Clearly, the privacy of the bit is maintained.

[2]This is not a huge overhead since an ATPM scheme anyway requires the issuer to prove that the embedded value is a bit.

An issue with unlinkability. Unfortunately, this approach is not automatically unlinkable, since the client cannot trivially adapt the signature $\hat{\sigma}$ from being over the blinded message and $\hat{\phi}$ to being over the original message and $\phi_{b'}$ (for each b'). Although, it seems plausible that by using equivalence class signatures [FHS19] one can overcome this barrier, the approach would not be generic, so we leave the design of such a scheme as an interesting future research direction. Rather, we observe that this issue can be avoided completely if one begins with the generic blind signature scheme due to Fischlin [Fis06].

ATPM from Fischlin-type blind signature. At a high level, Fischlin's generic paradigm requires a commitment scheme COM, a signature scheme SIG, a public key encryption (PKE) scheme PKE and a general-purpose non-interactive zero-knowledge (NIZK) proof system NIZK. Then, assuming a common reference string (CRS) containing the CRS, crs, for the NIZK as well as the PKE public key, pk, a client can blindly obtain an issuer's signature on a message, m, as follows—it creates a commitment c to m and sends it over to the issuer, who signs c and returns the resulting signature $\hat{\sigma}$; if the signature verifies under the issuer's verification key, the client then computes a NIZK proof π stating that it knows $\hat{\sigma}$, and a commitment c such that $\hat{\sigma}$ is a valid signature on the commitment c of m. It then sends the proof π as the blind signature on m.

As we previously suggested, this lets us avoid the issue with unlinkability since the final signature is a zero-knowledge proof and the client can simply hide the original ciphertext $\hat{\phi}$ in the ciphertext. Notably, due to the zero-knowledge property of the proof system, this scheme is unlinkable for an $\mathsf{IND\text{-}CPA}$ secure linearly homomorphic PKE scheme.

1.3 Lattice-based ATPM

We are now ready to describe our main construction. Our starting point for our lattice-based ATPM is the efficient round-optimal blind signature scheme of Agrawal et al. [AKSY22] which, in turn, is based on Fischlin's paradigm [Fis06].

To obtain a blind signature on a message m in the [AKSY22] scheme, the client creates a commitment $\mathbf{v} = \mathbf{C} \cdot \mathbf{u} + \mathsf{H}(\mathsf{m})$ for some public matrix $\mathbf{C}$ and a "short" uniform vector $\mathbf{u}$, and sends $\mathbf{v}$ to the issuer. The issuer then uses its secret trapdoor to sample a short preimage vector $\mathbf{z}$ such that $\mathbf{A} \cdot \mathbf{z} = \mathbf{v}$, for the issuer's public key matrix $\mathbf{A}$ and sends it to the client who, after verifying that $\mathbf{z}$ is a valid preimage, computes a NIZK proof π stating that it knows two short vectors $\mathbf{u}$ and $\mathbf{z}$ such that $\mathbf{A} \cdot \mathbf{z} = \mathbf{C} \cdot \mathbf{u} + \mathsf{H}(\mathsf{m})$; note that a verifier can compute $\mathsf{H}(\mathsf{m})$ given the message. The client then sends the proof π as the blind signature on m. As a technical necessity, the NIZK scheme must be straight-line extractable in order for the proof of one-more unforgeability to go through. The scheme is proven secure in the random oracle model under the (non-standard) one-more ISIS assumption (OM-ISIS) [AKSY22], an interactive analogue of the (standard) inhomogeneous short integer solution (I-SIS) assumption.

With this background, let us now apply our generic transformation to this construction. In particular, the issuer must now encrypt the hidden bit b under a suitable lattice-based linearly homomorphic PKE scheme[3] to obtain $\mathbf{x} \leftarrow \mathsf{LinHE.Enc}(\mathsf{pk}_{\mathsf{LinHE}}, b; r)$ using randomness r, and then sign, both $[\mathbf{v} \ \mathbf{x}]^\top$. It must then send $\mathbf{x}$, $\mathbf{z}$ as well as a proof $\hat{\pi}$ that $b \in \{0,1\}$ and that $\mathbf{x}$ is an encryption of b under $\mathsf{pk}_{\mathsf{LinHE}}$ with (secret) randomness r. Finally, to obtain the signature on m[4], the client will compute $\mathbf{w}_{\hat{b}} = \mathbf{x} + \mathbf{c}_{\hat{b}}$ where $\mathbf{c}_{\hat{b}} \leftarrow \mathsf{LinHE.Enc}(\mathsf{pk}_{\mathsf{LinHE}}, \hat{b}; \nu_{\hat{b}})$, for each $\hat{b} \in \{0,1\}$, and then send $\mathbf{w}_0$, $\mathbf{w}_1$ and a NIZK proof π proving that it knows $\mathbf{u}, \mathbf{x}, \mathbf{z}, \nu_0$ and ν_1 such that

[3] Fortunately, the classical public key encryption scheme [Reg05] from learning with errors (LWE) is already linearly homomorphic. See § 2.3 for details.

[4] In ATPM, the "message" is a so-called 'tag' that is chosen arbitrarily by the client.

$$\mathbf{A} \cdot \mathbf{z} = \begin{bmatrix} \mathbf{C} \cdot \mathbf{u} + \mathsf{H}(\mathsf{m}) \\ \mathbf{x} \end{bmatrix}, \ \mathbf{w}_0 = \mathbf{x} + \mathsf{LinHE.Enc}(\mathsf{pk}_{\mathsf{LinHE}}, 0; \nu_0) \ ,$$
$$\mathbf{w}_1 = \mathbf{x} + \mathsf{LinHE.Enc}(\mathsf{pk}_{\mathsf{LinHE}}, 1; \nu_1), \text{ and } \|\mathbf{u}\|, \|\mathbf{z}\| \text{ are short} \ .$$

The public verification of the signature can be performed simply by verification of the NIZK proof, and bit extraction proceeds exactly as described above, i.e., by checking which $\hat{b} \in \{0, 1\}$ yields $\mathsf{LinHE.Dec}(\mathsf{sk}, \phi_{b'}) = 0$.

We remark that while our idea of using linearly homomorphic PKE applies generally, the lattice instantiation introduces some technical issues. In particular, the issuer holding the secret decryption key $\mathsf{sk}_{\mathsf{LinHE}}$ can additionally recover the noise. This allows a malicious issuer to choose its own encryption randomness r in a way that it can skew the distribution of the recovered noise and break anonymity of the client. Interestingly, this is generally not the case for group-based linearly homomorphic PKE such as Paillier, because the randomness is (in a very loose sense) "wiped" during decryption, so the issuer does not get the additional distributional data as in the lattice case. To address this, some care is needed in how the encryption randomness ν_0 and ν_1 are chosen by the client. We explain these details in Section 3.

2 Preliminaries

Notation. We use λ to denote the security parameter. We use $\leftarrow\!\!\$$ to denote the output of a randomized algorithm, $\leftarrow$ to denote output of a deterministic algorithm, and $\coloneqq$ for assignment. Following common convention, we use lowercase bold-face letters to denote vectors and uppercase bold-face letters for matrices. For a vector $\mathbf{x}$, x_i denotes its ith element. For a vector $\mathbf{x}$, we write its ℓ_2 norm as $\|\mathbf{x}\|_2$, often dropping the subscript and writing it simply as $\|\mathbf{x}\|$. We write the ℓ_∞ norm of $\mathbf{x}$ as $\|\mathbf{x}\|_\infty$.

We write polynomials $u(X)$ in indeterminate X simply as u when it is clear from context. In all that follows, $\mathbb{Z}$ is the set of integers, and $\mathbb{Z}_q$ denotes the set of integers modulo q. For a power of two d, we denote by $\mathcal{R}$ the ring $\mathbb{Z}[X]/(X^d + 1)$. We write the ring $\mathbb{Z}_q[X]/(X^d + 1) \cong \mathcal{R}/q\mathcal{R}$ as $\mathcal{R}_{q,d}$. When the degree is omitted from the subscript, it is assumed to be the fixed protocol parameter d.

2.1 Lattice preliminaries

An m-dimensional lattice $\mathcal{L}$ is a discrete additive subgroup of $\mathbb{R}^m$. Given positive integers n, m, q and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we let $\Lambda_q^{\perp}(\mathbf{A})$ denote the lattice $\{\mathbf{z} \in \mathbb{Z}^m \ : \ \mathbf{A} \cdot \mathbf{z} = \mathbf{0} \mod q\}$. For $\mathbf{t} \in \mathbb{Z}_q^n$, we let $\Lambda_q^{\mathbf{t}}(\mathbf{A})$ denote the coset $\{\mathbf{z} \in \mathbb{Z}^m \ : \ \mathbf{A} \cdot \mathbf{z} = \mathbf{t} \mod q\}$.

Definition 2.1 (Discrete gaussians.)**.** *Let $\mathfrak{s}$ be any positive real number. The Gaussian distribution $\Gamma_{\mathfrak{s}}$ with parameter $\mathfrak{s}$ is defined by the probability distribution function $\rho_{\mathfrak{s}}(\mathbf{z}) = \exp(-\pi\|\mathbf{z}\|^2/\mathfrak{s}^2)$. For any set $\mathcal{L} \subset \mathbb{R}^m$, define $\rho_{\mathfrak{s}}(\mathcal{L}) = \sum_{\mathbf{z} \in \mathcal{L}} \rho_{\mathfrak{s}}(\mathbf{z})$. The discrete Gaussian distribution $\mathfrak{D}_{\mathcal{L},\mathfrak{s}}$ over $\mathcal{L}$ with parameter $\mathfrak{s}$ is defined by the probability distribution function $\rho_{\mathcal{L},\mathfrak{s}}(\mathbf{z}) = \rho_{\mathfrak{s}}(\mathbf{z})/\rho_{\mathfrak{s}}(\mathcal{L})$ for all $\mathbf{z} \in \mathcal{L}$.*

The following lemma (Lemma 4.4 of [MR04, GPV08]) shows that if the parameter $\mathfrak{s}$ of a discrete Gaussian distribution is small, then any vector drawn from this distribution will be short (with high probability).

Lemma 2.2. *Let m, n, q be positive integers with $m > n$, $q \geqslant 2$. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a matrix of dimensions $n \times m$, $\mathfrak{s} = \tilde{\Omega}(n)$ and $\mathcal{L} = \Lambda_q^{\perp}(\mathbf{A})$. Then*

$$\Pr[\|\mathbf{z}\| > \sqrt{m} \cdot \mathfrak{s} : \mathbf{z} \leftarrow\!\!\$ \, \Gamma_{\mathcal{L},\mathfrak{s}}] \leqslant \mathsf{negl}(n).$$

We will also (implicitly) require the following lemma (Lemma 4.4 of [Lyu12]) concerning the minimum-entropy of the discrete Gaussian distribution.

Lemma 2.3. *Let $\Gamma_{\mathbb{Z}^m,\mathfrak{s}}$ be the discrete Gaussian distribution over $\mathbb{Z}^m$ for any $m > 1$, with variance $\mathfrak{s}$. Then, for any $\mathfrak{s} \geq 3/\sqrt{2\pi}$ we have that $\mathbf{H}_\infty\left(\Gamma_{\mathbb{Z}^m,\mathfrak{s}}\right) \geq m$.*

Lattices with trapdoors are lattices that are statistically indistinguishable from randomly chosen lattices, but have a "trapdoor" that allow efficient solutions to hard lattice problems. Formally,

Definition 2.4 (Lattice trapdoors [Ajt99, GPV08].)**.** *For lattice parameters n, m, q with $m \geq O(n \log q)$, a trapdoor lattice sampler consists of algorithms* $\mathsf{TrapGen}$ *and* $\mathsf{SamplePre}$ *with the following syntax:*

- $\mathsf{TrapGen}(1^n, 1^m, q) \to (\mathbf{A}, \mathbf{T_A})$*:* *The lattice generation algorithm is a randomized algorithm that takes as input the matrix dimensions n, m, modulus q, and outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ together with a trapdoor $\mathbf{T_A}$.*
- $\mathsf{SamplePre}(\mathbf{A}, \mathbf{T_A}, \mathbf{t}, \mathfrak{s}) \to \mathbf{z}$*:* *The presampling algorithm takes as input a matrix $\mathbf{A}$, trapdoor $\mathbf{T_A}$, a vector $\mathbf{t} \in \mathbb{Z}_q^n$ and a parameter $\mathfrak{s} \in \mathbb{R}$ (which determines the length of the output vectors). It outputs a vector $\mathbf{z} \in \mathbb{Z}_q^m$ such that $\mathbf{A} \cdot \mathbf{z} = \mathbf{t}$ and $\|\mathbf{z}\| \leqslant \sqrt{m} \cdot \mathfrak{s}$.*

These algorithms must satisfy the following properties:

i. **Well-Distributedness of matrix.** *The following distributions are statistically indistinguishable:*

$$\{\mathbf{A} : (\mathbf{A}, \mathbf{T_A}) \hookleftarrow\$ \mathsf{TrapGen}(1^n, 1^m, q)\} \approx_s \left\{\mathbf{A} : \mathbf{A} \hookleftarrow\$ \mathbb{Z}_q^{n\times m}\right\} \ .$$

ii. **Preimage sampling.** *For all $(\mathbf{A}, \mathbf{T_A}) \hookleftarrow\$ \mathsf{TrapGen}(1^n, 1^m, q)$, if $\mathfrak{s} = \omega(\sqrt{n \cdot \log q \cdot \log m})$, then the following distributions are statistically indistinguishable:*

$$\left\{\mathbf{z} : \mathbf{U} \hookleftarrow\$ \mathbb{Z}_q^n, \mathbf{z} \hookleftarrow\$ \mathsf{SamplePre}(\mathbf{A}, \mathbf{T_A}, \mathbf{U}, \mathfrak{s})\right\} \approx_s \Gamma_{\mathbb{Z}^m,\mathfrak{s}} \ .$$

2.2 Randomness Extraction

Lemma 2.5 (Leftover hash lemma [DORS03, DRS04])**.** *Let $\mathcal{H} = \{\mathsf{H} : X \to Y\}_{\mathsf{H}\in\mathcal{H}}$ be a universal hash family, then for any random variable W taking values in X, the following holds*

$$\Delta_s\left((\mathsf{H}, \mathsf{H}(W)), (\mathsf{H}, \mathfrak{U}_Y)\right) \leqslant \frac{1}{2}\sqrt{2^{-\mathbf{H}_\infty(W)} \cdot |Y|},$$

where $\mathfrak{U}_Y$ denotes uniform distribution over Y.

We will use the following corollary, which follows from the Leftover Hash Lemma.

Corollary 2.6. *Let $\ell > m \cdot n \log_2 q + \omega(\log n)$, q a prime, and m, n are positive integers. Let $\mathbf{R}$ be a $k \times m$ matrix chosen as per distribution $\mathfrak{D}$, where $k = k(n)$ is polynomial in n and $\mathbf{H}_\infty(\mathfrak{D}) = \ell$. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices chosen uniformly in $\mathbb{Z}_q^{n\times k}$ and $\mathbb{Z}_q^{n\times m}$, respectively. Then the statistical distance between the following distributions is negligible in n.*

$$\{(\mathbf{A}, \mathbf{A} \cdot \mathbf{R})\} \approx_s \{(\mathbf{A}, \mathbf{B})\}$$

Lemma 2.7 (Smudging lemma [AJL+12, Lemma 2.1])**.** *Let $\mathfrak{B} = \mathfrak{B}(\lambda)$ and $\mathfrak{s} = \mathfrak{s}(\lambda)$ be two positive integers and let $\mathfrak{D} = \{\mathfrak{D}(\lambda)\}_\lambda$ be any $\mathfrak{B}$-bounded distribution family. Let $\Gamma_\mathfrak{s} = \{\Gamma_\mathfrak{s}(\lambda)\}_\lambda$ denote the zero-centered discrete Gaussian distribution with standard deviation $\mathfrak{s}$. Then the distributions $\mathfrak{D} + \Gamma_\mathfrak{s}$ and $\Gamma_\mathfrak{s}$ are statistically indistinguishable if for all $\lambda \in \mathbb{N}$, $\mathfrak{s} \geq \mathfrak{B} \cdot 2^{\omega(\lambda)}$.*

2.3 Public key encryption

A public key encryption (PKE) scheme PKE consists of the following polynomial-time algorithms.

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$. The setup algorithm takes as input the security parameter λ, and outputs a public-secret key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}(\mathsf{pk}, \mu) \to \psi$. The encryption algorithm takes as input a public key pk and a message μ, and outputs a ciphertext ψ.
- $\mathsf{Dec}(\mathsf{sk}, \psi) \to \mu$. The decryption algorithm takes as input a secret key sk and a ciphertext ψ, and outputs a message μ.

The scheme satisfies correctness if for all λ, m, $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{Setup}(1^\lambda)$, and every ciphertext $c \leftarrow\!\$\ \mathsf{Enc}(\mathsf{pk}, m)$, we have that $\mathsf{Dec}(\mathsf{sk}, c) = m$. Further, the standard IND-CPA security notion is defined as follows.

Definition 2.8 (IND-CPA)**.** *A public key encryption scheme* PKE *is IND-CPA secure if for every stateful* PPT *adversary* $\mathcal{A}$*, there exists a negligible functions* $\epsilon(\cdot)$*, such that for every* $\lambda \in \mathbb{N}$

$$\Pr\left[\mathcal{A}(\psi) = b : \begin{array}{r} (\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{Setup}(1^\lambda) \\ b \leftarrow\!\$\ \{0,1\} \\ (\mu_0, \mu_1) \leftarrow \mathcal{A}(\mathsf{pk}) \\ \psi \leftarrow\!\$\ \mathsf{Enc}(\mathsf{pk}, \mu_b) \end{array}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\lambda).$$

Linearly-homomorphic PKE from LWE. The following is a linearly homorphic PKE scheme LinHE:

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$. The setup algorithm takes as input the security parameter λ and computes the integers $n, m, \mathfrak{s}$ and q as the functions of the security parameter. It then samples a matrix $\mathbf{B} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m}$, and vectors $\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n$ and $\boldsymbol{\varepsilon} \leftarrow\!\$\ \Gamma_{\mathbb{Z}^m, \mathfrak{s}}$, and outputs a public-secret key pair $(\mathsf{pk} \coloneqq (\mathbf{B}, \mathbf{s}^\top \mathbf{B} + \boldsymbol{\varepsilon}), \mathsf{sk} \coloneqq \mathbf{s})$.
- $\mathsf{Enc}(\mathsf{pk}, b) \to \mathbf{x}$. The encryption algorithm takes as input a public key pk, interpreted as a matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ and a vector $\mathbf{c} \in \mathbb{Z}_q^m$, and a message $b \in \{0,1\}$. It samples a random vector $\mathbf{r} \leftarrow\!\$\ \{0,1\}^m$, and outputs a ciphertext $\mathbf{x}$ computed as

$$\mathbf{x} \coloneqq \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^\top \end{bmatrix} \cdot \mathbf{r} + b \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix}.$$

- $\mathsf{Dec}(\mathsf{sk}, \mathbf{x}) \to b$. The decryption algorithm takes as input a secret key sk interpreted as a vector $\mathbf{s} \in \mathbb{Z}_q^n$ and a ciphertext $\mathbf{x} \in \mathbb{Z}_q^{n+1}$. It outputs 0 if $\left|\begin{bmatrix} -\mathbf{s}^\top & 1 \end{bmatrix} \cdot \mathbf{x}\right| \leq q/4$, and 1 otherwise.

If $m > n \log(q)$ and $m\mathfrak{B} \ll q/4$ for $\mathfrak{B} = \mathfrak{s}\sqrt{m}$, then the scheme is secure under the $\mathsf{LWE}_{n,m,q,\mathfrak{s}}$ assumption. More importantly, for our purposes, notice that if $m\mathfrak{B} \ll q/8$ then $\mathsf{LinHE.Enc}(\mathsf{pk}, b_0) + \mathsf{LinHE.Enc}(\mathsf{pk}, b_1)$ is equal to

$$\begin{aligned} &\begin{bmatrix} \mathbf{B} \\ \mathbf{c}^\top \end{bmatrix} \cdot \mathbf{r}_0 + b_0 \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^\top \end{bmatrix} \cdot \mathbf{r}_1 + b_1 \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix} \\ = &\begin{bmatrix} \mathbf{B} \\ \mathbf{c}^\top \end{bmatrix} \cdot (\mathbf{r}_0 + \mathbf{r}_1) + (b_0 + b_1) \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix} \end{aligned}$$

which is $x \coloneqq \mathsf{LinHE.Enc}(\mathsf{pk}, b_0 \oplus b_1)$, when $m\mathfrak{B} \ll q/8$ for suitable choice of q, so that

$$\left|\begin{bmatrix}-\mathbf{s}^\top & 1\end{bmatrix} \cdot \hat{\mathbf{x}}\right| = \left|\boldsymbol{\varepsilon}^\top \cdot (\mathbf{r}_0 + \mathbf{r}_1) + (b_0 \oplus b_1) \cdot \lfloor q/2 \rfloor\right| \leq |2m\mathfrak{B} + (b_0 \oplus b_1) \cdot \lfloor q/2 \rfloor|$$

is at most $q/4$ if and only if $b_0 = b_1$.

2.4 Non-interactive zero-knowledge

A non-interactive zero-knowledge (NIZK) proof system NIZK for relation $\mathcal{R}$ consists of the following polynomial time algorithms:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$. The setup algorithm takes as input the security parameter λ, and outputs a common reference string crs.
- $\mathsf{Prove}(\mathsf{crs}, \mathbb{x}, \mathbb{w}) \to \pi$. The prover algorithm takes as input the crs, an instance $\mathbb{x} \in \mathcal{L}_\mathcal{R}$, and a witness $\mathbb{w}$. It outputs a proof π.
- $\mathsf{Verify}(\mathsf{crs}, \mathbb{x}, \pi) \to 0/1$. The verification algorithm takes as input the crs, an instance $\mathbb{x}$, and a proof π. It outputs either 1 (accept) or 0 (reject).

Definition 2.9 (Non-interactive zero-knowledge)**.** *A NIZK proof system* NIZK *must satisfy the following properties:*

i. **Completeness.** *For every* $\lambda \in \mathbb{N}$*, crs* $\mathsf{crs} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$*, any instance* $\mathbb{x} \in \mathcal{L}_\mathcal{R}$ *with corresponding witness* $\mathbb{w}$*,*

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, \mathbb{x}, \pi) = 1 \ : \ \pi \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, \mathbb{x}, \mathbb{w})\right] = 1.$$

ii. **Soundness.** *For every stateful* PPT *adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\begin{array}{l}\mathsf{Verify}(\mathsf{crs}, \mathbb{x}, \pi) = 1 \\ \nexists \mathbb{w} \ : \ (\mathbb{x}, \mathbb{w}) \in \mathcal{R}\end{array} \ : \ \begin{array}{r}\mathsf{crs} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda) \\ (\mathbb{x}, \pi) \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs})\end{array}\right] \leqslant \mathsf{negl}(\lambda) \ .$$

iii. **(Multi-theorem) zero-knowledge.** *There exists a stateful* PPT *simulator* $\mathcal{S}$ *such that for every stateful* PPT *adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\begin{array}{l}\mathcal{A}\left(\{\pi_{i,b}\}_i\right) = b \\ \wedge\, \forall i \in [\ell] \ : \ (\mathbb{x}_i, \mathbb{w}_i) \in \mathcal{R}\end{array} \ : \ \begin{array}{r}b \leftarrow\!\!\$\ \{0,1\} \\ \mathsf{crs}_0 \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda) \\ \mathsf{crs}_1 \leftarrow\!\!\$\ \mathcal{S}(1^\lambda) \\ \{\mathbb{x}_i, \mathbb{w}_i\}_{i \in [\ell]} \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs}_b) \\ \forall i \in [\ell] \ : \ \pi_{i,0} \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}_0, \mathbb{x}_i, \mathbb{w}_i) \\ \{\pi_{i,1}\}_i \leftarrow\!\!\$\ \mathcal{S}(\mathsf{crs}_1, \{\mathbb{x}_i\}_i)\end{array}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\lambda) \ .$$

Argument of knowledge. *A NIZK proof system* NIZK *is an argument of knowledge (NIZKAoK) if it additionally satisfies the following extraction property:*

iv. **Knowledge extraction.** *There exists a* PPT *extractor* $\mathcal{E}$ *such that for every stateful* PPT *adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\begin{array}{l} \mathsf{Verify}(\mathsf{crs}, \mathbb{x}, \pi) = 1 \\ \wedge\ (\mathbb{x}, \mathbb{w}) \notin \mathcal{R} \end{array} : \begin{array}{r} \tau \leftarrow\!\$\ \{0,1\}^\lambda \\ \mathsf{crs} \leftarrow \mathcal{E}(1^\lambda; \tau) \\ (\mathbb{x}, \pi) \leftarrow \mathcal{A}(1^\lambda, \mathsf{crs}) \\ \mathbb{w} \leftarrow \mathcal{E}(\tau, \mathbb{x}, \pi) \end{array}\right] \leqslant \mathsf{negl}(\lambda) \ ,$$

where τ *denotes the randomness used for running the setup algorithm with* $\mathcal{E}$*, and we assume without any loss of generality that* $|\tau| = \lambda$*.*

Straight-line knowledge extraction. A straight-line knowledge extractor (also known as online extractor) can be constructed generically using PKE by encrypting the witness under a shared public encryption key and sending the ciphertext along with the NIZK proof. Importantly, the NIZK proof itself must contain a proof of well-formedness of the ciphertext.

2.5 Anonymous Tokens with Private Metadata Bit

We recall the anonymous tokens (AT) interface defined by Chase et al. [CDV23]. Formally, it is comprised of polynomial time algorithms Setup, KeyGen, ReadBit and Verify, and a probabilistic polynomial time (PPT) interactive token issuance protocol between Client and Issuer.

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$. The setup algorithm takes as input the security parameter λ and outputs the protocol's public parameters pp.
- $\mathsf{KeyGen}(\mathsf{pp}) \to (\mathsf{pk}, \mathsf{sk})$. The issuer's key generation algorithm the public parameters pp as input and generates the public key pk and secret key sk of the issuer.
- $\langle \mathsf{Client}(\mathsf{pk}, \mu), \mathsf{Issuer}(\mathsf{sk}, b, \mu) \rangle \to (t, \sigma)$ or $\perp$. This is the interactive token issuance protocol between Client and Issuer where the client's input is the issuer's public key pk and some public metadata $\mu \in \mathcal{M}$, and the issuer's input is its secret key sk, the hidden metadata bit $b \in \{0,1\}$ and the public metadata μ. It can be broken down into three algorithms.
  - $\mathsf{ClientQuery}(\mathsf{pk}, \mu) \to (\mathsf{query}, \mathsf{state})$. Client initiates the protocol by sending query to the issuer.
  - $\mathsf{IssueToken}(\mathsf{sk}, b, \mu, \mathsf{query}) \to \mathsf{resp}$. Issuer replies with resp.
  - $\mathsf{ClientObtain}(\mathsf{state}, \mathsf{resp}) \to (t, \sigma)$. Client locally computes the token from the issuer's response.
- $\mathsf{ReadBit}(\mathsf{sk}, t, \sigma, \mu) \to \{0,1\}$ or $\perp$. The issuer's bit extraction algorithm takes as input the issuer's secret key sk, a token (t, σ), the public metadata μ and outputs $b \in \{0,1\}$ or $\perp$.
- $\mathsf{Verify}(\mathsf{pk}, t, \sigma, \mu) \to \{0,1\}$ or $\perp$. The public verification algorithm takes as input the issuer's public key pk, a token (t, σ), the public metadata μ and outputs $b \in \{0,1\}$ or $\perp$.

Correctness of an AT scheme holds if for any given execution of the interactive protocol, the output of the Issuer's ReadBit algorithm is the same as the embedded metadata bit b. In terms of security, an AT scheme must satisfy *one-more unforgeability*, *unlinkability*, and *privacy of the metadata bit*.

One-more unforgeability ensures that for an issuer running ℓ times, on some fixed (b, μ) pair, an adversary should not be able to produce more than ℓ tokens with pairwise distinct tags.

Definition 2.10 (One-more unforgeability)**.** *Let* $\mathsf{Adv}^{\mathsf{OM\text{-}Unf}}_{\mathsf{ATPM},\mathcal{A}} = \Pr[\mathsf{OM\text{-}Unf}_{\mathsf{ATPM},\mathcal{A}}(\lambda) = \mathbf{accept}]$ *be the advantage of an adversary* $\mathcal{A}$ *in the experiment defined in Figure 1. We say an ATPM scheme* ATPM *is one-more unforgeable if for any* PPT *adversary* $\mathcal{A}$ *this advantage is negligible.*

Game $\mathsf{OM\text{-}Unf}_{\mathsf{ATPM},\mathcal{A}}(\lambda)$

1 : $\mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda)$
2 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{KeyGen}(\mathsf{pp})$
3 : $b^*, \mu^*, \{(t_i, \sigma_i)\}_{i\in[N]} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Read}}}(\mathsf{pk})$
4 : Define ℓ as the multiplicity of (b^*, μ^*) in Q
5 : **if** $\#\{t_i\}_{i\in[N]} \le \ell$ **then reject**
6 : **if** $\bigvee_{i\in[N]} \mathsf{ReadBit}(\mathsf{sk}, t_i, \sigma_i, \mu^*) \neq b^*$ **then reject**
7 : **accept**

Oracle $\mathcal{O}_{\mathsf{Issue}}(b^{(i)}, \mu^{(i)}, \mathsf{query}^{(i)})$

1 : **if** multiset Q is uninitialized **then**
2 : $\quad Q := \varnothing$
3 : $Q := Q \cup \{(b^{(i)}, \mu^{(i)})\}$
4 : $\mathsf{resp}^{(i)} \leftarrow\!\$\ \mathsf{IssueToken}(\mathsf{sk}, b^{(i)}, \mu^{(i)}, \mathsf{query}^{(i)})$
5 : **return** $\mathsf{resp}^{(i)}$

Oracle $\mathcal{O}_{\mathsf{Read}}(t^{(i)}, \sigma^{(i)}, \mu^{(i)})$

1 : **return** $\mathsf{ReadBit}(\mathsf{sk}, t^{(i)}, \sigma^{(i)}, \mu^{(i)})$

Figure 1: The one-more unforgeability experiment for ATPM.

Unlinkability ensures that a malicious issuer can not link tokens to any given client, and more generally to the corresponding issuing sessions.

Definition 2.11 (κ-unlinkability)**.** *Let* $\mathsf{Adv}^{\mathsf{UNLINK}}_{\mathsf{ATPM},\mathcal{A},n} = \Pr[\mathsf{UNLINK}_{\mathsf{ATPM},\mathcal{A},n}(\lambda) = \mathbf{accept}]$ *be the advantage of an adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ *for parameter* $n \in \mathbb{N}$ *in the experiment defined in Figure 2. We say an ATPM scheme* ATPM *is* κ*-unlinkable if for any* PPT *adversary* $\mathcal{A}$ *and* $n \in \mathbb{N}$*, this advantage at most* $\kappa/n + \mathsf{negl}(\lambda)$.

Finally, privacy of the metadata bit ensures that an adversary can do no better than guess the hidden metadata bit with trivial probability.

Definition 2.12 (Privacy of metadata bit)**.** *Let*

$$\mathsf{Adv}^{\mathsf{PMB}}_{\mathcal{A}} = \left|\ \Pr[\mathsf{PMB}_{\mathsf{ATPM},\mathcal{A},1}(\lambda) = \mathbf{accept}] - \Pr[\mathsf{PMB}_{\mathsf{ATPM},\mathcal{A},0}(\lambda) = \mathbf{accept}]\ \right|$$

be the advantage of an adversary $\mathcal{A}$ *in the experiment defined in Figure 3. We say an ATPM scheme* ATPM *preserves privacy of the metadata bit if for any* PPT *adversary* $\mathcal{A}$*, this advantage negligible.*

3 Lattice-based Anonymous Tokens with Private Metadata bit

In this section, we give our lattice-based ATPM construction. The idea here is similar to the commit-sign-prove approach for lattice blind signatures [AKSY22, BLNS23]. The non-trivial part of our construction is facilitating the extraction of the embedded metadata bit. As discussed in our overview, at a high level, we address this challenge by encrypting the bit under a linearly homomorphic encryption scheme, and then using the homomorphic evaluations of the ciphertext under bits 0 and 1 to extract the originally embedded bit. We present the full construction below.

Note that for ease of presentation, we have omitted the public metadata, μ, in this construction. We give the complete picture later in Section 5.

Game $\mathsf{UNLINK}_{\mathsf{ATPM},\mathcal{A},n}(\lambda)$

1 : $\mathsf{pp} \leftarrow\$ \mathsf{Setup}(1^\lambda)$
2 : $(\mathsf{pk}, \mathsf{st}_1) \leftarrow \mathcal{A}_1(\mathsf{pp})$
3 : $(Q^*, \{\mathsf{resp}_i\}_{i \in Q^*}, \mathsf{st}_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{Query}}, \mathcal{O}_{\mathsf{Obtain}}}(\mathsf{st}_1)$
4 : **if** $Q^* \not\subseteq Q_{\mathsf{Query}} \setminus Q_{\mathsf{Obtain}}$ **then reject**
5 : **if** $|Q^*| \neq n$ **then reject**
6 : **foreach** $i \in Q^*$ **do**
7 : $\quad \mathsf{out}_i \leftarrow\$ \mathsf{ClientObtain}(\mathsf{state}^{(i)}, \mathsf{resp}_i)$
8 : $\quad$ **if** $\mathsf{out}_i = \perp$ **then reject**
9 : $\hat{i} \leftarrow\$ Q^*$
10 : Pick a random permutation φ of Q^*
11 : $i^* \leftarrow \mathcal{A}_3\left(\mathsf{st}_2, \mathsf{out}_{\hat{i}}, \{\mathsf{out}_{\varphi(i)}\}_{i \in Q^* \setminus \{\hat{i}\}}\right)$
12 : **if** $i^* = \hat{i}$ **then accept**

Oracle $\mathcal{O}_{\mathsf{Query}}(i, \mu^{(i)})$

1 : **if** Q_{Query} is uninitialized **then**
2 : $\quad Q_{\mathsf{Query}} \coloneqq \varnothing$
3 : $Q_{\mathsf{Query}} \coloneqq Q_{\mathsf{Query}} \cup \{i\}$
4 : $(\mathsf{query}^{(i)}, \mathsf{state}^{(i)}) \leftarrow\$ \mathsf{ClientQuery}(\mathsf{pk}, \mu^{(i)})$
5 : **return** $\mathsf{query}^{(i)}$

Oracle $\mathcal{O}_{\mathsf{Obtain}}(j^{(i)}, \mathsf{resp}^{(i)})$

1 : **if** Q_{Obtain} is uninitialized **then**
2 : $\quad Q_{\mathsf{Obtain}} \coloneqq \varnothing$
3 : **if** $j^{(i)} \in Q_{\mathsf{Obtain}} \ \vee \ j^{(i)} \notin Q_{\mathsf{Query}}$ **then**
4 : $\quad$ **return** $\perp$
5 : $Q_{\mathsf{Obtain}} \coloneqq Q_{\mathsf{Obtain}} \cup \{j^{(i)}\}$
6 : $\mathsf{out}^{(i)} \leftarrow\$ \mathsf{ClientObtain}(\mathsf{state}^{(i)}, \mathsf{resp}^{(i)})$
7 : **return** $\mathsf{out}^{(i)}$

Figure 2: The unlinkability experiment for ATPM.

Game $\mathsf{PMB}_{\mathcal{A},\hat{b}}(\lambda)$

1 : $\mathsf{pp} \leftarrow\$ \mathsf{Setup}(1^\lambda)$
2 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow\$ \mathsf{KeyGen}(\mathsf{pp})$
3 : $\mathsf{flag} \coloneqq \mathsf{False}$
4 : $b^* \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Read}}, \mathcal{O}_{\mathsf{Valid}}, \mathcal{O}_{\mathsf{Challenge}}}(\mathsf{pp}, \mathsf{pk})$
5 : **if** $b^* = \hat{b}$ **then accept**

Oracle $\mathcal{O}_{\mathsf{Issue}}(b^{(i)}, \mu^{(i)}, \mathsf{query}^{(i)})$

1 : $\mathsf{resp}^{(i)} \leftarrow\$ \mathsf{IssueToken}(\mathsf{sk}, b^{(i)}, \mu^{(i)}, \mathsf{query}^{(i)})$
2 : **return** $\mathsf{resp}^{(i)}$

Oracle $\mathcal{O}_{\mathsf{Read}}(t^{(i)}, \sigma^{(i)}, \mu^{(i)})$

1 : **if** $\mathsf{flag} \wedge \mu^{(i)} = \hat{\mu}$ **then return** $\perp$
2 : **return** $\mathsf{ReadBit}(\mathsf{sk}, t^{(i)}, \sigma^{(i)}, \mu^{(i)})$

Oracle $\mathcal{O}_{\mathsf{Valid}}(t^{(i)}, \sigma^{(i)}, \mu^{(i)})$

1 : $b^{(i)} \leftarrow \mathsf{ReadBit}(\mathsf{sk}, t^{(i)}, \sigma^{(i)}, \mu^{(i)})$
2 : **return** $(b^{(i)} \neq \perp)$

Oracle $\mathcal{O}_{\mathsf{Challenge}}(\mu^{(i)}, \mathsf{query}^{(i)})$

1 : **if** flag **then return** $\perp$
2 : $\mathsf{flag} \coloneqq \mathsf{True}$
3 : $\hat{\mu} \coloneqq \mu^{(i)}$
4 : $\mathsf{resp}^{(i)} \leftarrow\$ \mathsf{IssueToken}(\mathsf{sk}, \hat{b}, \hat{\mu}, \mathsf{query}^{(i)})$
5 : **return** resp

Figure 3: The privacy of metadata bit experiment for ATPM.

3.1 Construction

Let system parameters $n, m, m', \mathfrak{s}_{\boldsymbol{\varepsilon}}, \mathfrak{B}_{\boldsymbol{\varepsilon}} = \mathfrak{s}_{\boldsymbol{\varepsilon}}\sqrt{m}, \mathfrak{s}_{\mathbf{z}}, \mathfrak{B}_{\mathbf{z}} = \mathfrak{s}_{\mathbf{z}}\sqrt{m'}, \mathfrak{s}_{\boldsymbol{\nu}}, \mathfrak{B}_{\boldsymbol{\nu}} = \mathfrak{s}_{\boldsymbol{\nu}}\sqrt{m}$ and a prime number q be functions of the security parameter λ such that the one-more ISIS instance $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ and the LWE instance $\mathsf{LWE}_{q,\mathfrak{s}_{\boldsymbol{\varepsilon}}}$ are hard. These parameters must satisfy the following constraints:

$$n = \mathsf{poly}(\lambda),\ m, m' > n \log q + \lambda,\ \frac{\mathfrak{s}_{\mathbf{z}}}{2m'} = \Omega(1),\ m\mathfrak{B}_{\boldsymbol{\varepsilon}} \ll q/8,\ \mathfrak{s}_{\boldsymbol{\nu}} = O(2^{\lambda})\ . \tag{1}$$

Our construction requires a hash function $\mathsf{H} : \{0,1\}^{\lambda} \to \mathbb{Z}_q^n$, a lattice trapdoor $\mathcal{T} = (\mathsf{TrapGen}, \mathsf{SamplePre})$, a NIZK $\mathsf{NIZK}_{\mathsf{I}} = (\mathsf{NIZK}_{\mathsf{I}}.\mathsf{Setup}, \mathsf{NIZK}_{\mathsf{I}}.\mathsf{Prove}, \mathsf{NIZK}_{\mathsf{I}}.\mathsf{Verify})$ and a NIZK argument of knowledge (NIZKAoK) with straight-line extraction $\mathsf{NIZK}_{\mathsf{C}} = (\mathsf{NIZK}_{\mathsf{C}}.\mathsf{Setup}, \mathsf{NIZK}_{\mathsf{C}}.\mathsf{Prove}, \mathsf{NIZK}_{\mathsf{C}}.\mathsf{Verify})$ respectively for the following relations:

Relation $\mathscr{R}_{\mathsf{I}}$
Instance: Each instance $\mathbb{x}$ is interpreted as matrices $\mathbf{A}$ and $\mathbf{B}$, and vectors $\mathbf{c}$ and $\mathbf{x}$.
Witness: Witness $\mathbb{w}$ consists of vectors $\mathbf{s}$ and $\mathbf{r}$, and a bit b.
Membership: $\mathbb{w}$ is a valid witness for $\mathbb{x}$ if:

$$b \in \{0,1\} \ \wedge\ \mathbf{x} = \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^{\top} \end{bmatrix} \cdot \mathbf{r} + b \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix}$$

Relation $\mathscr{R}_{\mathsf{C}}$
Instance: Each instance $\mathbb{x}$ is interpreted as matrices $\mathbf{A}$ and $\mathbf{B}$, vectors $\mathbf{c}, \mathbf{w}_0$ and $\mathbf{w}_1$, and a string t.
Witness: Witness $\mathbb{w}$ consists of vectors $\mathbf{u}, \mathbf{x}, \mathbf{z}, \boldsymbol{\nu}_0$ and $\boldsymbol{\nu}_1$.
Membership: $\mathbb{w}$ is a valid witness for $\mathbb{x}$ if:

$$\mathbf{A}_{\top} \cdot \mathbf{z} - \mathbf{C} \cdot \mathbf{u} = \mathsf{H}(t) \ \wedge\ \mathbf{A}_{\perp} \cdot \mathbf{z} = \mathbf{x} \ \wedge\ \|\mathbf{u}\| \le \frac{\mathfrak{B}_{\mathbf{z}}}{2m'} \ \wedge\ \|\mathbf{z}\| \le \mathfrak{B}_{\mathbf{z}}$$
$$\wedge\ \mathbf{w}_0 = \mathbf{x} + \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^{\top} \end{bmatrix} \cdot \boldsymbol{\nu}_0 \ \wedge\ \mathbf{w}_1 = \mathbf{x} + \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^{\top} \end{bmatrix} \cdot \boldsymbol{\nu}_1 + \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix} \ \wedge\ \|\boldsymbol{\nu}_0\|, \|\boldsymbol{\nu}_1\| \le \mathfrak{B}_{\boldsymbol{\nu}}$$

System setup. Given the system parameters $\lambda, n, m, m', q, \mathfrak{s}_{\mathbf{z}}, \mathfrak{s}_{\boldsymbol{\varepsilon}}, \mathfrak{s}_{\boldsymbol{\nu}}$ as well as the hash function H modeled as a random oracle, the setup algorithm samples a random matrix $\mathbf{C} \in \mathbb{Z}_q^{n \times m'}$. It also runs the NIZK setup algorithms for the relations $\mathscr{R}_{\mathsf{I}}$ and $\mathscr{R}_{\mathsf{C}}$ to generate the corresponding $\mathsf{crs}_{\mathsf{I}}$ and $\mathsf{crs}_{\mathsf{C}}$.

$\mathsf{KeyGen}(1^{\lambda})$

1 : $(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow\$ \mathsf{TrapGen}(2n+1, m', q, \mathfrak{s}_{\mathbf{z}})$
2 : $\mathbf{B} \leftarrow\$ \mathbb{Z}_q^{n \times m},\ \mathbf{s} \leftarrow\$ \mathbb{Z}_q^n,\ \boldsymbol{\varepsilon} \leftarrow\$ \Gamma_{\mathbb{Z}^m, \mathfrak{s}_{\boldsymbol{\varepsilon}}}$
3 : **return** $\mathsf{pk} \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{s}^{\top}\mathbf{B} + \boldsymbol{\varepsilon}),\ \mathsf{sk} \coloneqq (\mathbf{T}_{\mathbf{A}}, \mathbf{s})$

$\mathsf{Setup}(1^{\lambda})$

1 : $\mathbf{C} \leftarrow\$ \mathbb{Z}_q^{n \times m'}$
2 : $\mathsf{crs}_{\mathsf{I}} \leftarrow\$ \mathsf{NIZK}_{\mathsf{I}}.\mathsf{Setup}(1^{\lambda})$
3 : $\mathsf{crs}_{\mathsf{C}} \leftarrow\$ \mathsf{NIZK}_{\mathsf{C}}.\mathsf{Setup}(1^{\lambda})$
4 : **return** $\mathsf{pp} \coloneqq (\mathbf{C}, \mathsf{crs}_{\mathsf{I}}, \mathsf{crs}_{\mathsf{C}})$

Key generation. The issuer's key generation algorithm uses trapdoor sampling to create a random matrix and its trapdoor $\mathbf{A}, \mathbf{T}_{\mathbf{A}}$ as the public and secret keys respectively. It also samples a uniform public matrix $\mathbf{B}$ and additionally outputs $\mathbf{B}$ and $\mathbf{s}^{\top}\mathbf{B} + \boldsymbol{\varepsilon}$ as part of the public key, for uniformly sampled $\mathbf{s}$ and $\boldsymbol{\varepsilon}$ sampled from a discrete Gaussian with bound $\mathfrak{B}_{\boldsymbol{\varepsilon}}$ such that $m\mathfrak{B}_{\boldsymbol{\varepsilon}} \ll q/8$.

Presignature issuance and token generation. We show the presignature issuance and token generation protocols in detail in Figure 4. Given some query $\mathbf{v}$, the issuer embeds the metadata bit b inside the vector $\mathbf{x}$, which in turn can be viewed as (LWE-based) encryption of the b under the public key $(\mathbf{B}, \mathbf{c})$. For a suitable choice of the modulus q, the shown encryption scheme is linearly homomorphic in the sense we explained

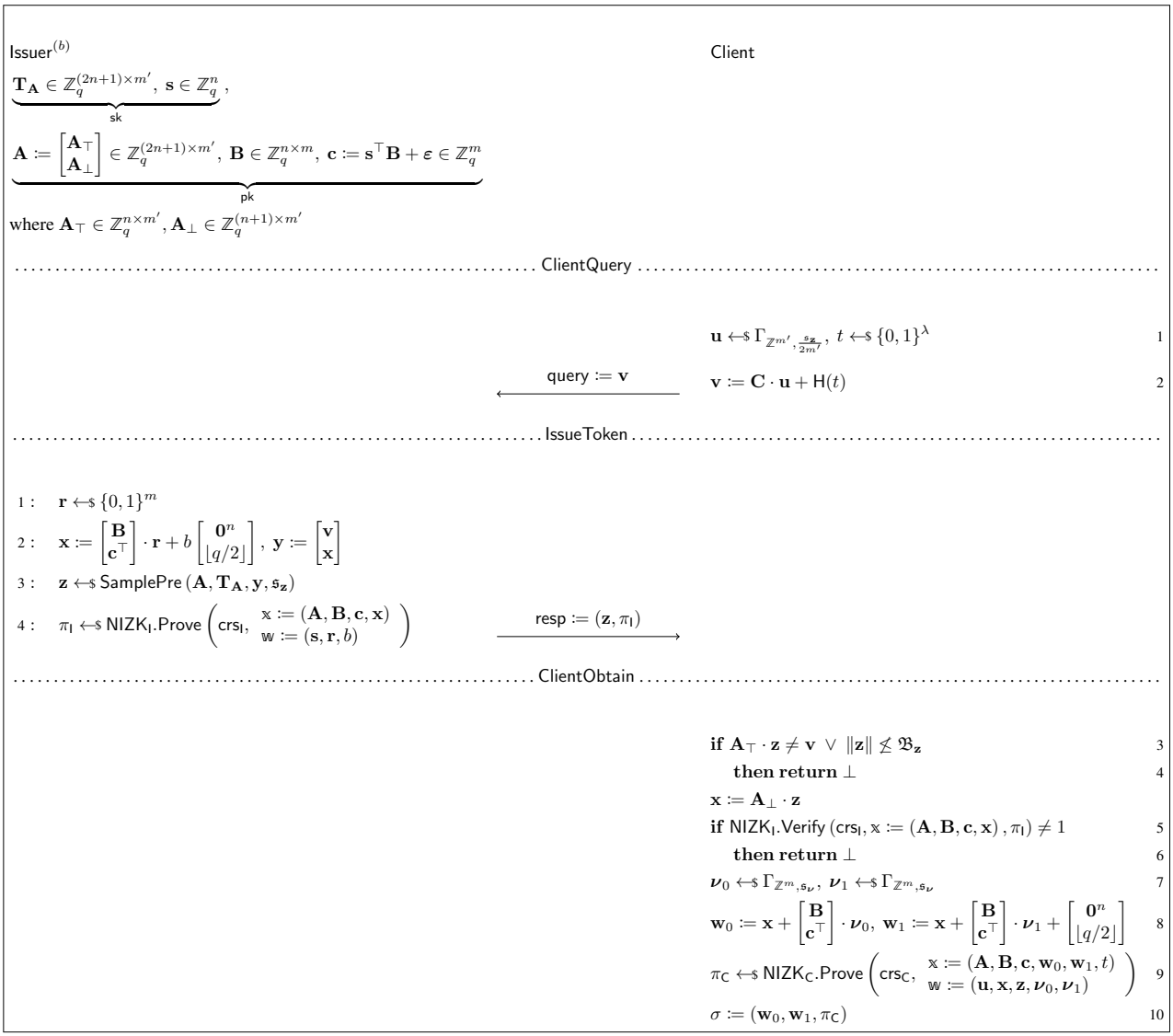

Figure 4: Lattice-based construction for ATPM presignature issuance and token generation.

previously. The issuer then samples a preimage $\mathbf{z}$ on input $\mathbf{y} := \begin{bmatrix}\mathbf{v}\\ \mathbf{x}\end{bmatrix}$ using the lattice trapdoor $\mathbf{T_A}$. The full issuance includes $\mathbf{z}$ and a NIZK proof π_I proving $b \in \{0,1\}$ and the well-formedness of the vector $\mathbf{x}$.

In order to create the token, the client first verifies signature, as well as the NIZK proof. This is then followed by the client computing $\mathbf{w}_0 := \mathbf{x} + \overline{\mathbf{w}}_0$ (similarly $\mathbf{w}_1$), where $\overline{\mathbf{w}}_0$ (resp. $\overline{\mathbf{w}}_1$) is essentially the encryption of 0 (resp. 1) under the public key $(\mathbf{B}, \mathbf{c})$ and randomness $\boldsymbol{\nu}_0$ (resp. $\boldsymbol{\nu}_1$). Notably, instead of sampling $\boldsymbol{\nu}_{\hat{b}}$ for each $\hat{b} \in \{0,1\}$ from the same distribution as $\mathbf{r}$, the client samples them from a discrete Gaussian with variance $\mathfrak{s}_{\boldsymbol{\nu}} = O(2^\lambda)$. Looking ahead to the security proof, this allows us to leverage the smudging lemma (Lemma 2.7) so that the distribution over $(\mathbf{r}+\boldsymbol{\nu}_{\hat{b}})$ is statistically indistinguishable from $\boldsymbol{\nu}_{\hat{b}}$, thereby preventing a malicious signer (who can decrypt $\mathbf{w}_{\hat{b}}$ to learn $\boldsymbol{\varepsilon}^\top\cdot(\mathbf{r}+\boldsymbol{\nu}_{\hat{b}})$) from breaking unlinkability by skewing $(\mathbf{r}+\boldsymbol{\nu}_{\hat{b}})$ with its choice of $\mathbf{r}$.

Now, due to the linear homomorphism of the encryption scheme, $\mathbf{w}_0$ (resp. $\mathbf{w}_1$) is an encryption of $b \oplus 0$ (resp. $b \oplus 1$). Finally, the client creates a NIZK proof π_{C} for proving knowledge of a short $\mathbf{z}$ as a signature on $\mathbf{z}$ as well as the well-formedness of $\mathbf{w}_0, \mathbf{w}_1$. The final token is the tag t and the signature comprised of the ciphertexts $\mathbf{w}_0$ and $\mathbf{w}_1$ along with the NIZK proof π_{C}.

Public verification (token redemption). The public verification algorithm simply verifies the NIZK proof π_{C}.

$\mathsf{Verify}(\mathsf{pk} \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{c}), t, \sigma \coloneqq (\mathbf{w}_0, \mathbf{w}_1, \pi_{\mathsf{C}}))$

1 : **return** $\mathsf{NIZK}_{\mathsf{C}}.\mathsf{Verify}(\mathsf{crs}_{\mathsf{C}}, \mathbb{x} \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{w}_0, \mathbf{w}_1, t), \pi_{\mathsf{C}})$

Bit extraction. Extraction of the private metadata bit proceeds by first calling the verification algorithm. Then the issuer essentially decrypts both $\mathbf{w}_0$ and $\mathbf{w}_1$, and outputs the bit corresponding to whichever of the two is 0.

$\mathsf{ReadBit}(\mathsf{sk} \coloneqq (\mathbf{T}_{\mathbf{A}}, \mathbf{s}), t, \sigma \coloneqq (\mathbf{w}_0, \mathbf{w}_1, \pi_{\mathsf{C}}))$

1 : **if** $\mathsf{Verify}(\mathsf{pk}, t, \sigma) \neq 1$ **then**
2 : **return** $\perp$
3 : **foreach** $\hat{b} \in \{0, 1\}$ **do**
4 : **if** $\left|\begin{bmatrix} -\mathbf{s}^\top & 1 \end{bmatrix} \cdot \mathbf{w}_{\hat{b}}\right| \leq q/4$ **then**
5 : **return** $\hat{b}$
6 : **return** $\perp$

Correctness. By completeness of $\mathsf{NIZK}_{\mathsf{I}}$, $\mathbf{x}$ is a valid encryption of b under $(\mathbf{B}, \mathbf{c})$, and by correctness of pre-image sampling, $\mathbf{z}$ is a short preimage of $\begin{bmatrix} \mathbf{C} \cdot \mathbf{u} + \mathsf{H}(t) \\ \mathbf{x} \end{bmatrix}$. Furthermore, by completeness of $\mathsf{NIZK}_{\mathsf{C}}$ we have that for each $\hat{b} \in \{0, 1\}$,

$$\mathbf{w}_{\hat{b}} \;=\; \mathbf{x} + \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^\top \end{bmatrix} \cdot \boldsymbol{\nu}_{\hat{b}} + \hat{b} \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix} = \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^\top \end{bmatrix} \cdot \left(\mathbf{r} + \boldsymbol{\nu}_{\hat{b}}\right) + (b + \hat{b}) \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix}$$

which is an encryption of $b \oplus \hat{b}$ under $(\mathbf{B}, \mathbf{c})$ when $m\mathfrak{B}_{\boldsymbol{\varepsilon}} \ll q/8$ for suitable choice of q. Also notice that since each $\mathbf{w}_{\hat{b}}$ is an encryption of $b \oplus \hat{b}$, for the metadata bit b, and $m\mathfrak{B}_{\boldsymbol{\varepsilon}} \ll q/8$, it follows that

$$\left|\begin{bmatrix} -\mathbf{s}^\top & 1 \end{bmatrix} \cdot \mathbf{w}_{\hat{b}}\right| = \left|\boldsymbol{\varepsilon}^\top \cdot (\mathbf{r} + \boldsymbol{\nu}_{\hat{b}}) + (b \oplus \hat{b}) \cdot \lfloor q/2 \rfloor\right| \leq \left|2m\mathfrak{B}_{\boldsymbol{\varepsilon}} + (b \oplus \hat{b}) \cdot \lfloor q/2 \rfloor\right|$$

is at most $q/4$ if and only if $\hat{b} = b$. Therefore, the above ATPM construction satisfies correctness.

Security. We now present the main security theorem for this section and sketch the proofs. The complete security proofs have been provided in Appendix A.

Theorem 3.1. *Let system parameters $n, m, m', \mathfrak{s}_{\boldsymbol{\varepsilon}}, \mathfrak{B}_{\boldsymbol{\varepsilon}} = \mathfrak{s}_{\boldsymbol{\varepsilon}}\sqrt{m}, \mathfrak{s}_{\mathbf{z}}, \mathfrak{B}_{\mathbf{z}} = \mathfrak{s}_{\mathbf{z}}\sqrt{m'}, \mathfrak{s}_{\boldsymbol{\nu}}, \mathfrak{B}_{\boldsymbol{\nu}} = \mathfrak{s}_{\boldsymbol{\nu}}\sqrt{m}$ and a prime number q be functions of the security parameter λ satisfying the constraints defined in (1), then construction 3 is:*

(i) one-more unforgeable (in the random oracle model) assuming $\mathsf{NIZK}_{\mathsf{I}}$ *satisfies zero-knowledge,* $\mathsf{NIZK}_{\mathsf{C}}$ *are straight-line extractable and the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ *problem is hard;*

(ii) κ*-unlinkable for* $\kappa = 2$*, assuming* $\mathsf{NIZK}_{\mathsf{C}}$ *satisfies zero-knowledge; and*

(iii) has private metadata bit assuming $\mathsf{NIZK}_{\mathsf{I}}$ *satisfies zero-knowledge,* $\mathsf{NIZK}_{\mathsf{C}}$ *is straight-line extractable, and the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ *and* $\mathsf{LWE}_{q,n,m,\mathfrak{s}_{\boldsymbol{\epsilon}}}$ *problems are hard.*

Pf. sketch. The proof of one-more unforegeability (Lemma A.1) of the construction essentially follows via a reduction to the OM-ISIS assumption. At a high-level, the reduction sets $\mathbf{C} \coloneqq \mathbf{A}_{\top} \cdot \mathbf{R}$ for $\mathbf{R}$ chosen uniformly from $\{0,1\}^{m' \times m'}$. We can use the leftover hash lemma (Corollary 2.6) to argue that this change is statistically indistinguishable to an adversary. Then, whenever the adversary makes a query for the hash of some value t, the reduction programs the random oracle such that $\mathsf{H}(t) \coloneqq \mathbf{t}$ where $\mathbf{t}$ is the response of a *syndrome query* to the OM-ISIS challenger. Then, for each Issue query $\mathbf{v}$, the reduction makes a *preimage query* on $\mathbf{v}$ to the OM-ISIS challenger and forwards the corresponding preimage, $\mathbf{z}$ along with a simulated proof π_{I}. Finally, when the adversary outputs its (one-more) forgeries, the reduction can extract $(\mathbf{u}_i, \mathbf{z}_i)$ from each proof $\pi_{\mathsf{C},i}$. The reduction then computes $\mathbf{z}'_i \coloneqq \mathbf{z}_i - \mathbf{R} \cdot \mathbf{x}_i$ and $\mathbf{t}_i \coloneqq \mathbf{A}_{\top} \cdot \mathbf{z}_i - \mathbf{C} \cdot \mathbf{x}_i$ and outputs $\{(\mathbf{z}'_i, \mathbf{t}_i)\}_i$ as the OM-ISIS solution. By choosing $\mathbf{u}$ from a suitable distribution, we can ensure that $\|\mathbf{z}'_i\| \leq 2\mathfrak{B}_{\mathbf{z}}$, as desired.

To prove κ-unlinkability (Lemma A.3) of the construction, the idea is to make the final output values independent of the query as well as the responses. In order to accomplish this, we will firstly have the challenger simulate all its proofs. Following this, we can make a strictly information theoretic argument to reason that any query $\mathbf{v}$ is indistinguishable from uniformly random. Moreover, the corresponding final challenge output, $\mathsf{out}_i \coloneqq (t_i, \mathbf{w}_{0,i}, \mathbf{w}_{1,i}, \pi_{\mathsf{C},i})$ is also independent of the response $(\mathbf{z}_i, \pi_{\mathsf{I},i})$ by additional application of the smudging lemma (Lemma 2.7). It follows that the only meaningful information the adversary can learn in this case is the private bit embedded in $\mathbf{w}_{0,i}$ and $\mathbf{w}_{1,i}$. So the best adversarial strategy is to create some n_b responses with bit b for each bit ($n_0 + n_1 = n$), read the bit $b_{\hat{i}}$ of $\mathsf{out}_{\hat{i}}$ and output a random value i^* from the set of all $n_{b_{\hat{i}}}$ indices where the embedded bit was equal to $b_{\hat{i}}$. The probability that the adversary wins is

$$\sum_{b \in \{0,1\}} \Pr\left[b_{i^*} = b_{\hat{i}}\right] \cdot \Pr\left[i^* = \hat{i}\right] = \sum_{b \in \{0,1\}} \frac{n_b}{n} \cdot \frac{1}{n_b} = \frac{2}{n} \ .$$

Lastly, we can leverage the IND-CPA security (from LWE) of our linearly homomorphic encryption scheme for privacy of the metadata bit (Lemma A.4) of our construction. We also need to be able to carefully answer any Read or Valid queries, which can be done by extracting $\mathbf{x}$ from any π_{C} and looking up the corresponding bit $\hat{b}$ stored during issuance. Importantly, if no such entry is found, we can once again reduce the security to OM-ISIS (as in one-more unforgeability) since the adversary must have created a valid token without the issuer. □

4 Atlantis: Anonymous Tokens from Lattices with NTRU and SIS

In this section, we explain our instantiation of our ATPM construction and provide concrete estimates under suitable parameter choices.

Following the efficient blind signature scheme due to Agrawal, et al. [AKSY22], our scheme is instantiated with SHA-3-256 as the hash function, Falcon-512 [PFH+22] for pre-image sampling and the efficient NIZK for general linear relations (and norm relations) due to [LNP22]. The LNP22 proof system, however,

does not provide a straight-line extraction which is crucial for our purpose. As we previously remarked, straight-line witness extraction from a NIZK can be generically achieved using an IND-CPA secure PKE by encrypting the witness, and then additionally proving correct computation of the ciphertext inside the NIZK. This construction is secure if the secret key is discarded after setup, and the public key is given as part of the public parameters. To extract a witness, a challenger runs the setup algorithm and withholds the secret key. Zero-knowledge follows from that of the NIZK and IND-CPA security of the PKE. Concretely, we use the PKE scheme in [LPS10] (similar to CRYSTALS-Kyber). We provide the expanded protocol at the end of this section, and its security proof in Appendix B.

Now, in order to evaluate the concrete costs of our (expanded) construction, we instantiate it in a manner identical to [AKSY22]. Recall that the modulus for the zero-knowledge proof should factor into primes congruent to $5 \mod 8$ so that the polynomial $x^{128}+1$ generating the quotient for the ring used in the zero-knowledge proof splits exactly into two factors (cf. [LNP22, Lemma 2.6]). Further, by choosing the moduli for the signature scheme and PKE as multiples of the zero-knowledge modulus we can simplify the relation.

Note that the base modulus q must be chosen to be sufficiently large so as to account for the smudging lemma. In particular, for around 80-bits of hardness (i.e., roughly 2^{-40} statistical distance between the "good" and "bad" distributions), we set q to be the smallest prime greater than 2^{50} that is $5 \mod 8$. For brevity, we will summarize the salient aspects of the instantiation below.

▷ ClientQuery :
Client's commitment. The client samples $t \in \{0,1\}^{256}$ and $u_1, u_2 \in \mathcal{R}_{q,256}$ such that $\|(u_1,\ u_2)\|_\infty = 2^{22}$, and then for public polynomial $c \in \mathcal{R}_{q,256}$ computes the commitment

$$v \coloneqq c \cdot u_1 + u_2 + \mathsf{H}(t)\ .$$

Consequently, the size of the client's query is 1.5 KB.

▷ IssueToken :
Trapdoor generation and preimage sampling. We instantiate Falcon-512 [PFH+22] over the ring $\mathcal{R}_{q,512}$,

$$\mathfrak{s}_z = \frac{1.17}{\pi}\sqrt{q \cdot \log\left(4 \cdot 512 \cdot \left(1+\sqrt{\lambda \cdot 2^{64}}\right)/2\right)} \text{ and } \mathfrak{B}_z = 1.1 \cdot \mathfrak{s}_\mathbf{z}\sqrt{2 \cdot 512}\ .$$

Let $x \in \mathcal{R}_q^{256}$ be the encryption of the private metadata bit b. The signer runs the preimage sampling algorithm over $v||x \in \mathcal{R}_{q,512}$ with respect to its public verification key polynomial $a_1 \in \mathcal{R}_{q,512}$, to obtain $z_1, z_2 \in \mathcal{R}_{q,512}$ such that

$$a_1 \cdot z_1 + z_2 = v||x\ .$$

Issuer's ZK proof. The client must prove the relation $\mathcal{R}_{\mathsf{I}}$. We set the modulus $q_{\mathsf{ZK}}^{(\mathsf{I})} = q \cdot q_1^{(\mathsf{I})} \approx 2^{118}$, and choose the Module-LWE parameter $\hat{m}_2 = 58$ and Module-SIS parameter $\hat{n} = 14$ for soundness of the zero-knowledge proof, and carry over the other parameters from [LNP22]. The length of the committed vector over $\mathcal{R}_{q_{\mathsf{ZK}}^{(\mathsf{I})},128}$ is $\hat{m}_1 = 4$. The final proof size after compression is 64.5 KB.

The issuer sends z_1, x (the client can compute z_2) as well as the proof. Given this, the estimated size of the issuer's response is around $4.6 + 64.5 = 69.1$ KB.

▷ ClientObtain :

Client's ciphertext. The client computes w_0, w_1 using randomness ν_0 and ν_1 sampled according to $\Gamma_{\mathcal{R}_q, \mathfrak{s}_\nu}$, where $\mathfrak{s}_\nu \approx 2^{40}$ for roughly 2^{-40} statistical distance. For straight-line extraction, the client must encrypt the 6 witness polynomials $u_1, u_2, x, z_1, \nu_0, \nu_1 \in \mathcal{R}_{q,512}$ (z_2 can be computed using the instance and the extracted values). The PKE scheme is instantiated over $\mathcal{R}_{q^{(\mathsf{O})}_{\mathsf{PKE}},128}$ for $q^{(\mathsf{O})}_{\mathsf{PKE}} = q \cdot q^{(\mathsf{O})}_1 \approx 2^{62}$. The reduced degree polynomial-ring allows the protocol to remain compatible with the degree of the polynomial-ring in the zero-knowledge proof system of [LNP22]. Since $\mathcal{R}_{q,512} \subset \mathcal{R}_{q^{(\mathsf{Q})}_{\mathsf{PKE}},512} \cong \mathcal{R}^4_{q^{(\mathsf{Q})}_{\mathsf{PKE}},128}$, the rank of the plaintext space is $6 \cdot 4 = 24$. Choosing the infinity norm bound $\tau = 3$ on the encryption randomness $(s, e_1, \mathbf{e}_2)$, and the Module-LWE rank to be 19 for sufficient hardness results in a final ciphertext size of approximately 42.2 KB.

Client's ZK proof. The client must prove the relation $\mathcal{R}_{\mathsf{C}}$. We set the modulus $q^{(\mathsf{O})}_{\mathsf{ZK}} = q \cdot q^{(\mathsf{O})}_1 \cdot q^{(\mathsf{O})}_2 \approx 2^{79}$, and choose the Module-LWE parameter $\hat{m}_2 = 46$ and Module-SIS parameter $\hat{n} = 12$ for soundness of the zero-knowledge proof, and carry over the other parameters from [LNP22]. The length of the committed vector over $\mathcal{R}_{q^{(\mathsf{O})}_{\mathsf{ZK}},128}$ is 69. The resulting proof can then be further compressed using [DKL+18], essentially by cutting low-order bits of the commitment. The final proof size after compression is 83.9 KB.

Thus, the transcript size is 70.6 KB, and the final token size is 129.1 KB, including 32 bytes for the message and 3 KB for w_0 and w_1.

Concrete security. In Table 1, we summarize the concrete security of the given instantiation estimated using the Python script provided by [AKSY22][5].

Security Property	Assumption	Bit security (Core-SVP Hardness)
Hiding for commitment	MLWE	106 bits
Unforgeability (Falcon)	MSIS	121 bits
Ciphertext security (ClientObtain)	MLWE	105 bits
NIZK proof (IssueToken)	MSIS and MLWE	101 and 102 bits (resp.)
NIZK proof (ClientObtain)	MSIS and MLWE	100 and 104 bits (resp.)
One-more Unforgeability	rOM-ISIS	140 bits

Table 1: Concrete security of token generation.

4.1 Expanded ATPM from Lattice Assumptions

We now give our expanded lattice-based ATPM construction which will allow us to extract the witness from any (general-purpose) NIZK proof in straight-line manner.

[5] Our estimation code can be found at https://github.com/aayux/atlantis-estimates. In it, we also provide parameters for 128-bit hardness of smudging and find that the resulting transcript size is 102.7 KB and the final token size is around 176.3 KB.

Let system parameters $n, m, m', \mathfrak{s}_{\boldsymbol{\varepsilon}}, \mathfrak{B}_{\boldsymbol{\varepsilon}} = \mathfrak{s}_{\boldsymbol{\varepsilon}}\sqrt{m}, \mathfrak{s}_{\mathbf{z}}, \mathfrak{B}_{\mathbf{z}} = \mathfrak{s}_{\mathbf{z}}\sqrt{m'}, \mathfrak{s}_{\boldsymbol{\nu}}, \mathfrak{B}_{\boldsymbol{\nu}} = \mathfrak{s}_{\boldsymbol{\nu}}\sqrt{m}$ and a prime number q be functions of the security parameter λ such that the one-more ISIS instance $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ and the LWE instance $\mathsf{LWE}_{q,\mathfrak{s}_{\boldsymbol{\varepsilon}}}$ are hard. These parameters must satisfy the constraints prescribed in (1).

Our construction requires a hash function $\mathsf{H} : \{0,1\}^{\lambda} \to \mathbb{Z}_q^n$, a lattice trapdoor $\mathcal{T} = (\mathsf{TrapGen}, \mathsf{SamplePre})$, a public key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$, and NIZKs $\mathsf{NIZK_I} = (\mathsf{NIZK_C.Setup}, \mathsf{NIZK_I.Prove}, \mathsf{NIZK_I.Verify})$ and $\mathsf{NIZK_C} = (\mathsf{NIZK_C.Setup}, \mathsf{NIZK_C.Prove}, \mathsf{NIZK_C.Verify})$ respectively for the following relations:

Relation $\mathscr{R}'_{\mathsf{I}}$
Instance: Each instance $\mathbb{x}$ is interpreted as matrices $\mathbf{A}$ and $\mathbf{B}$, and vectors $\mathbf{c}$, $\mathbf{v}$, $\mathbf{x}$ and $\mathbf{z}$.
Witness: Witness $\mathbb{w}$ consists of vectors $\mathbf{s}$ and $\mathbf{r}$, and a bit b.
Membership: $\mathbb{w}$ is a valid witness for $\mathbb{x}$ if:

$$b \in \{0,1\} \ \wedge \ \mathbf{x} = \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^{\top} \end{bmatrix} \cdot \mathbf{r} + b \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix}$$

Relation $\mathscr{R}'_{\mathsf{C}}$
Instance: Each instance $\mathbb{x}$ is interpreted as matrices $\mathbf{A}$ and $\mathbf{B}$, vectors $\mathbf{c}$, w_0 and w_1, and a string t.
Witness: Witness $\mathbb{w}$ consists of vectors $\mathbf{u}, \mathbf{x}, \mathbf{z}, \boldsymbol{\nu}_0$ and $\boldsymbol{\nu}_1$.
Membership: $\mathbb{w}$ is a valid witness for $\mathbb{x}$ if:

$$\mathbf{A}_{\top} \cdot \mathbf{z} - \mathbf{C} \cdot \mathbf{u} = \mathsf{H}(t) \ \wedge \ \mathbf{A}_{\perp} \cdot \mathbf{z} = \mathbf{x} \ \wedge \ \|\mathbf{u}\| \leq \frac{\mathfrak{B}_{\mathbf{z}}}{2m'} \ \wedge \ \|\mathbf{z}\| \leq \mathfrak{B}_{\mathbf{z}}$$
$$\wedge\, w_0 = \mathbf{x} + \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^{\top} \end{bmatrix} \cdot \boldsymbol{\nu}_0 \ \wedge \ w_1 = \mathbf{x} + \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^{\top} \end{bmatrix} \cdot \boldsymbol{\nu}_1 + \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix} \ \wedge \ \|\boldsymbol{\nu}_0\|, \|\boldsymbol{\nu}_1\| \leq \mathfrak{B}_{\boldsymbol{\nu}}$$
$$\wedge\, \boldsymbol{\psi} = \mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, \mathbf{u}||\mathbf{x}||\mathbf{z}||\boldsymbol{\nu}_0||\boldsymbol{\nu}_1; \boldsymbol{\rho})$$

System setup. Given the system parameters $\lambda, n, m, m', q, \mathfrak{s}_{\mathbf{z}}, \mathfrak{s}_{\boldsymbol{\varepsilon}}, \mathfrak{s}_{\boldsymbol{\nu}}$ as well as the hash function H modeled as a random oracle, the setup algorithm samples a random matrix $\mathbf{C} \in \mathbb{Z}_q^{n \times m'}$ and generates the PKE public and secret keys. It also runs the NIZK setup algorithms for the relations $\mathscr{R}'_{\mathsf{I}}$ and $\mathscr{R}'_{\mathsf{C}}$ to generate the corresponding $\mathsf{crs_I}$ and $\mathsf{crs_C}$.

$\mathsf{Setup}(1^{\lambda})$

1 : $\mathbf{C} \leftarrow\$ \mathbb{Z}_q^{n \times m'}$
2 : $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\$ \mathsf{PKE.KeyGen}(1^{\lambda})$
3 : $\mathsf{crs_I} \leftarrow\$ \mathsf{NIZK_I.Setup}(1^{\lambda})$
4 : $\mathsf{crs_C} \leftarrow\$ \mathsf{NIZK_C.Setup}(1^{\lambda})$
5 : **return** $\mathsf{pp} := \begin{pmatrix} \mathbf{C}, \mathsf{pk}_{\mathsf{PKE}}, \\ \mathsf{crs_I}, \mathsf{crs_C} \end{pmatrix}$

$\mathsf{KeyGen}(1^{\lambda})$

1 : $(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow\$ \mathsf{TrapGen}(2n+1, m', q, \mathfrak{s}_{\mathbf{z}})$
2 : $\mathbf{B} \leftarrow\$ \mathbb{Z}_q^{n \times m}, \mathbf{s} \leftarrow\$ \mathbb{Z}_q^n, \boldsymbol{\varepsilon} \leftarrow\$ \Gamma_{\mathbb{Z}^m, \mathfrak{s}_{\varepsilon}}$
3 : **return** $\mathsf{pk} := (\mathbf{A}, \mathbf{B}, \mathbf{s}^{\top}\mathbf{B} + \boldsymbol{\varepsilon})$, $\mathsf{sk} := (\mathbf{T}_{\mathbf{A}}, \mathbf{s})$

Key generation. No change is required to the issuer's key generation algorithm.

Presignature issuance and token generation. We show the presignature issuance and token generation protocols in detail in Figure 5. The notable difference from the basic construction is that the NIZK proof sent by the client is accompanied by a ciphertext encrypting the client's secrets. Looking ahead to the security proofs, the challenger will possess a secret decryption key corresponding to the public encryption key enabling it to directly extract all the secrets from the ciphertext in a straight-line manner.

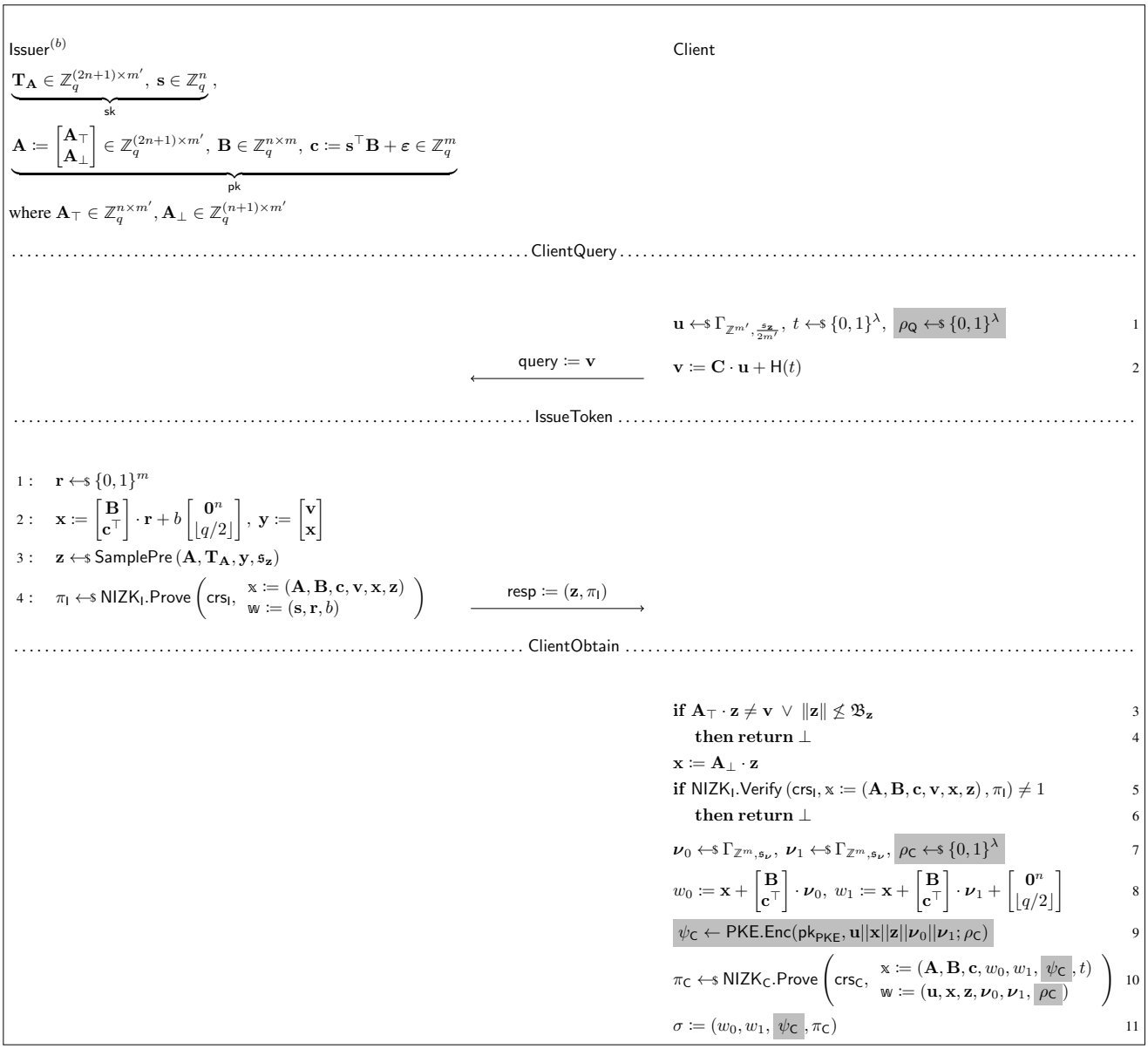

Figure 5: Expanded lattice-based construction for ATPM presignature issuance and token generation.

Public verification (token redemption). As before, the public verification algorithm simply verifies the NIZK proof π_C, but with ψ_C as an additional input in σ.

$$\frac{\mathsf{Verify}(\mathsf{pk} \coloneqq (\mathbf{A},\mathbf{B},\mathbf{c}), t, \sigma \coloneqq (w_0, w_1, \psi_\mathsf{C}, \pi_\mathsf{C}))}{1: \quad \textbf{return } \mathsf{NIZK_C.Verify}(\mathsf{crs_C}, \mathbb{x} \coloneqq (\mathbf{A},\mathbf{B},\mathbf{c}, w_0, w_1, \psi_\mathsf{C}, t), \pi_\mathsf{C})}$$

Bit extraction. Extraction of the private metadata bit proceeds similarly as it did before, but now takes ψ_C as an additional input in σ.

$\mathsf{ReadBit}(\mathsf{sk} \coloneqq (\mathbf{T}_{\mathbf{A}}, \mathbf{s}), t, \sigma \coloneqq (w_0, w_1, \psi_{\mathsf{C}}, \pi_{\mathsf{C}}))$

1 : **if** $\mathsf{NIZK}_{\mathsf{C}}.\mathsf{Verify}(\mathsf{pk}, t, \sigma) \neq 1$ **then**
2 : **return** $\perp$
3 : **foreach** $\hat{b} \in \{0,1\}$ **do**
4 : **if** $\left|\begin{bmatrix}-\mathbf{s}^\top & 1\end{bmatrix} \cdot w_{\hat{b}}\right| \leq q/4$ **then**
5 : **return** $\hat{b}$
6 : **return** $\perp$

5 Lattice-based ATPM with Public Metadata

We now show a simple way to extend our previous construction to allow for the inclusion of public attribute μ in the the presignature issuance, token generation and verification algorithms. Instead of providing redundant details, our emphasis will be on simply highlighting the changes from the previous construction.

5.1 Construction

Let system parameters $n, m, m', \mathfrak{s}_{\boldsymbol{\varepsilon}}, \mathfrak{B}_{\boldsymbol{\varepsilon}} = \mathfrak{s}_{\boldsymbol{\varepsilon}}\sqrt{m}, \mathfrak{s}_{\mathbf{z}}, \mathfrak{B}_{\mathbf{z}} = \mathfrak{s}_{\mathbf{z}}\sqrt{m'}, \mathfrak{s}_{\boldsymbol{\nu}}, \mathfrak{B}_{\boldsymbol{\nu}} = \mathfrak{s}_{\boldsymbol{\nu}}\sqrt{m}$ and a prime number q be functions of the security parameter λ such that the one-more ISIS instance $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},3\mathfrak{B}_{\mathbf{z}}}$ and the LWE instance $\mathsf{LWE}_{q,\mathfrak{s}_{\boldsymbol{\varepsilon}}}$ are hard. These parameters must satisfy the constraints prescribed in (1).

Our construction requires hash functions $\mathsf{H}_1, \mathsf{H}_2 : \{0,1\}^\lambda \to \mathbb{Z}_q^n$, a lattice trapdoor $\mathcal{T} = (\mathsf{TrapGen}, \mathsf{SamplePre})$, a NIZK $\mathsf{NIZK}_{\mathsf{I}} = (\mathsf{NIZK}_{\mathsf{C}}.\mathsf{Setup}, \mathsf{NIZK}_{\mathsf{I}}.\mathsf{Prove}, \mathsf{NIZK}_{\mathsf{I}}.\mathsf{Verify})$ and a NIZK$\underline{\text{AoK}}$ with straight-line extraction $\mathsf{NIZK}_{\mathsf{C}} = (\mathsf{NIZK}_{\mathsf{C}}.\mathsf{Setup}, \mathsf{NIZK}_{\mathsf{C}}.\mathsf{Prove}, \mathsf{NIZK}_{\mathsf{C}}.\mathsf{Verify})$ respectively for $\mathcal{R}_{\mathsf{I}}^{\mu} \equiv \mathcal{R}_{\mathsf{I}}$ and the relation:

Relation $\mathcal{R}_{\mathsf{C}}^{\mu}$
Instance: Each instance $\mathbb{x}$ is interpreted as matrices $\mathbf{A}$ and $\mathbf{B}$, vectors $\mathbf{c}$, $\mathbf{w}_0$ and $\mathbf{w}_1$, and strings t and μ.
Witness: Witness $\mathbb{w}$ consists of vectors $\mathbf{u}, \mathbf{x}, \mathbf{z}, \boldsymbol{\nu}_0$ and $\boldsymbol{\nu}_1$.
Membership: $\mathbb{w}$ is a valid witness for $\mathbb{x}$ if:

$$\mathbf{A}_\top \cdot \mathbf{z} - \mathbf{C} \cdot \mathbf{u} = \mathsf{H}_1(t) + \mathsf{H}_2(\mu) \ \wedge \ \mathbf{A}_\perp \cdot \mathbf{z} = \mathbf{x} \ \wedge \ \|\mathbf{u}\| \leq \frac{\mathfrak{B}_{\mathbf{z}}}{3m'} \ \wedge \ \|\mathbf{z}\| \leq \mathfrak{B}_{\mathbf{z}}$$
$$\mathbf{w}_0 = \mathbf{x} + \begin{bmatrix}\mathbf{B}\\ \mathbf{c}^\top\end{bmatrix} \cdot \boldsymbol{\nu}_0 \ \wedge \ \mathbf{w}_1 = \mathbf{x} + \begin{bmatrix}\mathbf{B}\\ \mathbf{c}^\top\end{bmatrix} \cdot \boldsymbol{\nu}_1 + \begin{bmatrix}\mathbf{0}^n\\ \lfloor q/2 \rfloor\end{bmatrix} \ \wedge \ \|\boldsymbol{\nu}_0\|, \|\boldsymbol{\nu}_1\| \leq \mathfrak{B}_{\boldsymbol{\nu}}$$

Presignature issuance and token generation. We show the presignature issuance and token generation protocols in detail in Figure 6. The $\mathsf{ClientQuery}$ algorithm does not change and is therefore omitted. The high level idea remains the same as in the previous construction, however, the preimage sampling algorithm runs on input $\mathbf{y} \coloneqq \begin{bmatrix}\mathbf{v} + \mathsf{H}_2(\mu)\\ \mathbf{x}\end{bmatrix}$ for public metadata μ; and the client verifies signature by including $\mathsf{H}_2(\mu)$. Lastly, the NIZK proof π_{C} now also includes μ.

Public verification (token redemption). As before, the public verification algorithm simply verifies the NIZK proof π_{C}, but with the public metadata μ as an additional input.

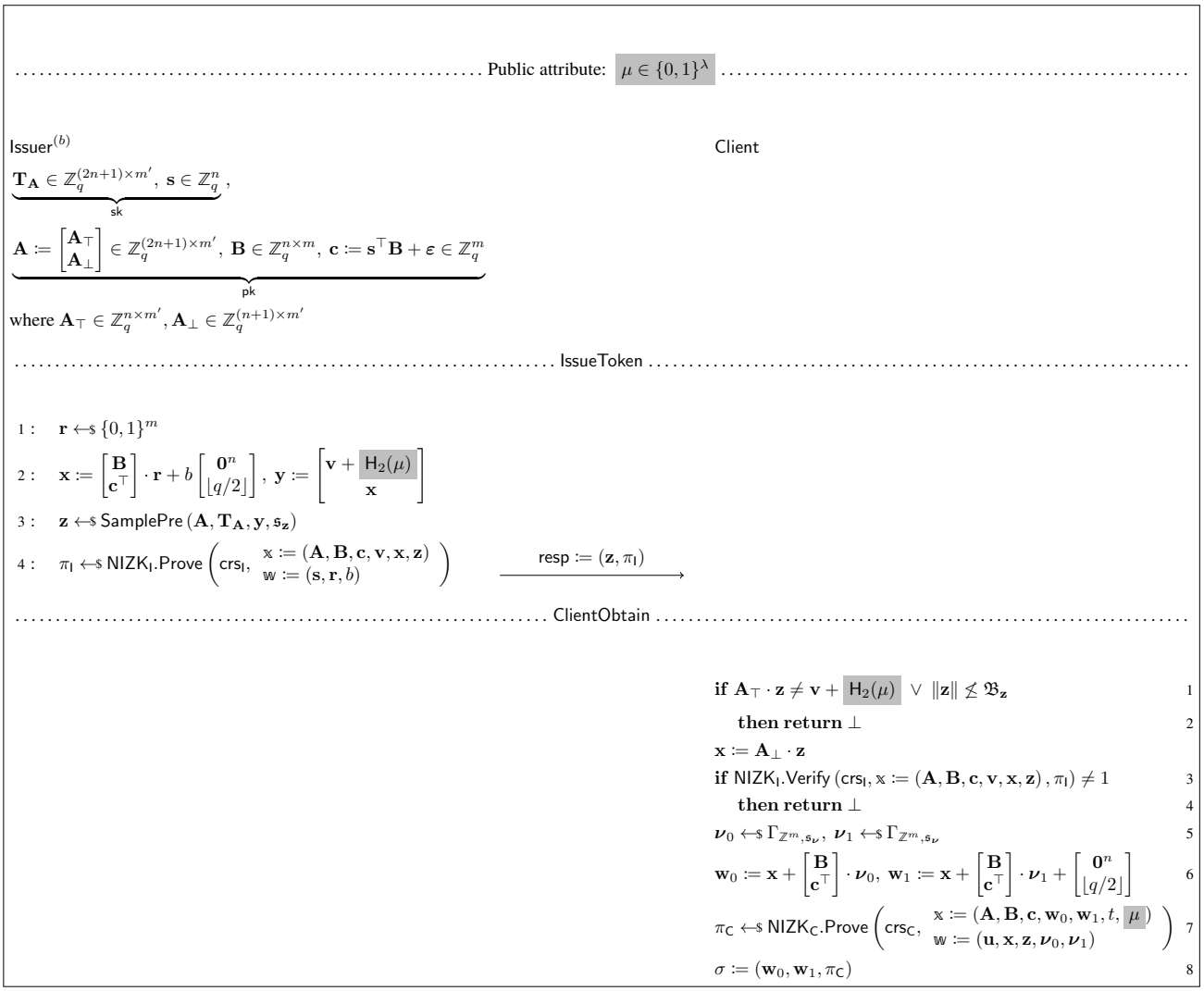

Figure 6: Lattice-based construction for ATPM with public metadata presignature issuance and token generation.

$\mathsf{Verify}(\mathsf{pk} \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{c}), t, \sigma \coloneqq (\mathbf{w}_0, \mathbf{w}_1, \pi_\mathsf{C}), \mu)$

1 : **return** $\mathsf{NIZK_C.Verify}(\mathsf{crs_C}, \mathbb{x} \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{w}_0, \mathbf{w}_1, t, \mu), \pi_\mathsf{C})$

Bit extraction. Extraction of the private metadata bit proceeds similarly as it did before, but now takes the public metadata μ as an additional input.

$\mathsf{ReadBit}(\mathsf{sk} \coloneqq (\mathbf{T}_{\mathbf{A}}, \mathbf{s}), t, \sigma \coloneqq (\mathbf{w}_0, \mathbf{w}_1, \pi_{\mathsf{C}}), \boxed{\mu})$

1 : **if** $\mathsf{NIZK}_{\mathsf{C}}.\mathsf{Verify}(\mathsf{pk}, t, \sigma, \boxed{\mu}) \neq 1$ **then**
2 : **return** $\perp$
3 : **foreach** $\hat{b} \in \{0,1\}$ **do**
4 : **if** $\left|\begin{bmatrix}-\mathbf{s}^{\top} & 1\end{bmatrix} \cdot \mathbf{w}_{\hat{b}}\right| \leq q/4$ **then**
5 : **return** $\hat{b}$
6 : **return** $\perp$

Security. We now present the main security theorems for this section. The full security proofs have been provided in Appendix C.

Theorem 5.1. *Let system parameters $n, m, m', \mathfrak{s}_{\boldsymbol{\varepsilon}}, \mathfrak{B}_{\boldsymbol{\varepsilon}} = \mathfrak{s}_{\boldsymbol{\varepsilon}}\sqrt{m}, \mathfrak{s}_{\mathbf{z}}, \mathfrak{B}_{\mathbf{z}} = \mathfrak{s}_{\mathbf{z}}\sqrt{m'}, \mathfrak{s}_{\boldsymbol{\nu}}, \mathfrak{B}_{\boldsymbol{\nu}} = \mathfrak{s}_{\boldsymbol{\nu}}\sqrt{m}$ and a prime number q be functions of the security parameter λ satisfying the constraints defined in (1), then construction 5 is:*

(i) one-more unforgeable (in the random oracle model) assuming $\mathsf{NIZK}_{\mathsf{I}}$ satisfies zero-knowledge, $\mathsf{NIZK}_{\mathsf{C}}$ is straight-line extractable and the $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},3\mathfrak{B}_{\mathbf{z}}}$ problem is hard;

(ii) κ-unlinkable for $\kappa = 2$, assuming $\mathsf{NIZK}_{\mathsf{I}}$ is sound and $\mathsf{NIZK}_{\mathsf{C}}$ satisfies zero-knowledge; and

(iii) has private metadata bit assuming $\mathsf{NIZK}_{\mathsf{I}}$ satisfies zero-knowledge, $\mathsf{NIZK}_{\mathsf{C}}$ is straight-line extractable, and the $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},3\mathfrak{B}_{\mathbf{z}}}$ and $\mathsf{LWE}_{q,n,m,\mathfrak{s}_{\boldsymbol{\varepsilon}}}$ problems are hard.

Acknowledgements. We are grateful to Damien Stehlé for alerting us to the noise recovery issue in an earlier version of the paper. This work was supported by NSF awards #2143287 and #2247304.

References

[ADDS21] Martin R. Albrecht, Alex Davidson, Amit Deo, and Nigel P. Smart. Round-optimal verifiable oblivious pseudorandom functions from ideal lattices. In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 261–289. Springer, Cham, May 2021.

[AG24] Martin R. Albrecht and Kamil Doruk Gür. Verifiable oblivious pseudorandom functions from lattices: Practical-ish and thresholdisable. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 205–237. Springer, Singapore, December 2024.

[AJL+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 483–501. Springer, Berlin, Heidelberg, April 2012.

[Ajt99] Miklós Ajtai. Determinism versus non-determinism for linear time RAMs (extended abstract). In *31st ACM STOC*, pages 632–641. ACM Press, May 1999.

[AKSY22] Shweta Agrawal, Elena Kirshanova, Damien Stehlé, and Anshu Yadav. Practical, round-optimal lattice-based blind signatures. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 39–53. ACM Press, November 2022.

[App21] Apple. icloud private relay overview. https://www.apple.com/icloud/docs/iCloud_Private_Relay_Overview_Dec2021.pdf, 2021. Accessed: 2024-04-29.

[AYY23] Ghous Amjad, Kevin Yeo, and Moti Yung. RSA blind signatures with public metadata. Cryptology ePrint Archive, Report 2023/1199, 2023.

[BHNY25] Foteini Baldimtsi, Lucjan Hanzlik, Quan Nguyen, and Aayush Yadav. Non-interactive anonymous tokens with private metadata bit. Cryptology ePrint Archive, Report 2025/430, 2025.

[BLNS23] Ward Beullens, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Lattice-based blind signatures: Short, efficient, and round-optimal. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 16–29. ACM Press, November 2023.

[BLOR22] Fabrice Benhamouda, Tancrède Lepoint, Michele Orrù, and Mariana Raykova. Publicly verifiable anonymous tokens with private metadata bit. Cryptology ePrint Archive, Report 2022/004, 2022.

[BRS23] Fabrice Benhamouda, Mariana Raykova, and Karn Seth. Anonymous counting tokens. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part II*, volume 14439 of *LNCS*, pages 245–278. Springer, Singapore, December 2023.

[CDV23] Melissa Chase, F. Betül Durak, and Serge Vaudenay. Anonymous tokens with stronger metadata bit hiding from algebraic MACs. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 418–449. Springer, Cham, August 2023.

[Cha90] David Chaum. Showing credentials without identification transferring signatures between unconditionally unlinkable pseudonyms. In Jennifer Seberry and Josef Pieprzyk, editors, *AUSCRYPT'90*, volume 453 of *LNCS*, pages 246–264. Springer, Berlin, Heidelberg, January 1990.

[CL01] Jan Camenisch and Anna Lysyanskaya. An identity escrow scheme with appointed verifiers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 388–407. Springer, Berlin, Heidelberg, August 2001.

[DGS+18] Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. Privacy pass: Bypassing internet challenges anonymously. *PoPETs*, 2018(3):164–180, July 2018.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018(1):238–268, 2018.

[DMTV24] F. Betül Durak, Laurane Marco, Abdullah Talayhan, and Serge Vaudenay. Non-transferable anonymous tokens by secret binding. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 2460–2474. ACM Press, October 2024.

[DORS03] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. Cryptology ePrint Archive, Report 2003/235, 2003.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 523–540. Springer, Berlin, Heidelberg, May 2004.

[FHS19] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *Journal of Cryptology*, 32(2):498–546, April 2019.

[Fis06] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77. Springer, Berlin, Heidelberg, August 2006.

[Goo24] Google. Private state tokens. https://developers.google.com/privacy-sandbox/protections/private-state-tokens, 2024. Accessed: 2024-04-29.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[IET24] IETF. Privacy pass (privacypass). https://datatracker.ietf.org/wg/privacypass/about/, 2024. Accessed: 2024-04-29.

[KLOR20] Ben Kreuter, Tancrède Lepoint, Michele Orrù, and Mariana Raykova. Anonymous tokens with private metadata bit. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 308–336. Springer, Cham, August 2020.

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 71–101. Springer, Cham, August 2022.

[LPS10] Vadim Lyubashevsky, Adriana Palacio, and Gil Segev. Public-key cryptographic primitives provably as secure as subset sum. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 382–400. Springer, Berlin, Heidelberg, February 2010.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Berlin, Heidelberg, April 2012.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381. IEEE Computer Society Press, October 2004.

[MRFH24] Thibault Meunier, Cefan Daniel Rubin, and Armando Faz-Hernández. Privacy pass: upgrading to the latest protocol version. https://blog.cloudflare.com/privacy-pass-standard, 2024. Accessed: 2024-04-29.

[NIS24] NIST. Transition to post-quantum cryptography standards. https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf, 2024. Accessed: 2025-05-12.

[OTZZ24] Michele Orrù, Stefano Tessaro, Greg Zaverucha, and Chenzhi Zhu. Oblivious issuance of proofs. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part IX*, volume 14928 of *LNCS*, pages 254–287. Springer, Cham, August 2024.

[PFH+22] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[PWFW23] Guru-Vamsi Policharla, Bas Westerbaan, Armando Faz-Hernández, and Christopher A Wood. Post-quantum privacy pass via post-quantum anonymous credentials. Cryptology ePrint Archive, Report 2023/414, 2023.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[SS22] Tjerand Silde and Martin Strand. Anonymous tokens with public metadata and applications to private contact tracing. In Ittay Eyal and Juan A. Garay, editors, *FC 2022*, volume 13411 of *LNCS*, pages 179–199. Springer, Cham, May 2022.

[Wil21] John Wilander. Introducing private click measurement, pcm. https://webkit.org/blog/11529/introducing-private-click-measurement-pcm/, 2021. Accessed: 2024-04-29.

[YCN+25] Yingfei Yan, Sherman S. M. Chow, Lucien K. L. Ng, Harry W. H. Wong, Yongjun Zhao, and Baocang Wang. Batch anonymous MAC tokens from lattices. In Ruben Niederhagen and Markku-Juhani O. Saarinen, editors, *Post-Quantum Cryptography - 16th International Workshop, PQCrypto 2025, Part I*, pages 349–384. Springer, Cham, April 2025.

A Security Proofs for Section 3

In this section, we prove our main theorem (Theorem 3.1) for ATPM security of construction 3.

Lemma A.1 (One-more unforgeability)**.** *Construction 3 is one-more unforgeable (in the random oracle model) assuming* $\mathsf{NIZK}_{\mathsf{I}}$ *satisfies zero-knowledge,* $\mathsf{NIZK}_{\mathsf{C}}$ *is straight-line extractable and the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ *problem is hard.*

Proof. The proof proceeds through a sequence of games:

Game $\mathbf{G}_0$: This is the original ATPM one-more unforgeability experiment $\mathsf{OM\text{-}Unf}_{\mathsf{ATPM},\mathcal{A}}$ defined with respect to construction 3 (without public metadata).

Game $\mathsf{OM\text{-}Unf}_{\mathsf{ATPM},\mathcal{A}}(\lambda)$

1 : $(\mathbf{C}, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{I}}, \mathsf{crs}_{\mathsf{C}}) =: \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda)$
2 : $\begin{pmatrix} \mathsf{pk} := (\mathbf{A}, \mathbf{B}, \mathbf{c}), \\ \mathsf{sk} := (\mathbf{T}_{\mathbf{A}}, \mathbf{s}) \end{pmatrix} \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$
3 : $b^*, \{(t_i, \sigma_i)\}_{i\in[N]} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Read}}}(\mathsf{pk})$
4 : Define ℓ as the multiplicity of b^* in Q
5 : **if** $\#\{t_i\}_{i\in[N]} \le \ell \vee b^* \notin \{0,1\}$ **then reject**
6 : **if** $\bigvee_{i\in[N]} \mathsf{ReadBit}(\mathsf{sk}, t_i, \sigma_i) \neq b^*$ **then reject**
7 : **accept**

Oracle $\mathcal{O}_{\mathsf{Issue}}(b^{(i)}, \mathsf{query}^{(i)} := \mathbf{v}^{(i)})$

1 : **if** multiset Q is uninitialized **then**
2 : $\quad Q := \varnothing$
3 : $Q := Q \cup \{b^{(i)}\}$
4 : $(\mathbf{z}^{(i)}, \pi_{\mathsf{I}}^{(i)}) =: \mathsf{resp}^{(i)} \leftarrow\!\$\ \mathsf{IssueToken}(\mathsf{sk}, b^{(i)}, \mathsf{query}^{(i)})$
5 : **return** $\mathsf{resp}^{(i)}$

Oracle $\mathcal{O}_{\mathsf{Read}}(t^{(i)}, \sigma^{(i)} := (\mathbf{w}_0^{(i)}, \mathbf{w}_1^{(i)}, \pi_{\mathsf{C}}^{(i)}))$

1 : **return** $\mathsf{ReadBit}(\mathsf{sk}, t^{(i)}, \sigma^{(i)})$

Figure 7: The one-more unforgeability Game $\mathbf{G}_0$.

Game $\mathbf{G}_1$: This is the same as $\mathbf{G}_0$, except for each Issue query, the challenger simulates the corresponding proof $\pi_{\mathsf{I}}^{(i)}$ without a witness. It is easily shown that if $\mathsf{NIZK}_{\mathsf{I}}$ satisfies zero-knowledge, then $\mathbf{G}_1$ and $\mathbf{G}_2$ are indistinguishable to the adversary.

Game $\mathbf{G}_2$: This is the same as $\mathbf{G}_1$, except for each of the ith Read queries, instead of running the ReadBit algorithm as before, the challenger runs the extractor for $\mathsf{NIZK}_{\mathsf{C}}$ to extract the witness from the corresponding proof $\pi_{\mathsf{C}}^{(i)}$ and verifies the relation $\mathcal{R}_{\mathsf{C}}$ locally. Similarly, for each of the ith (one-more) forgery, it extracts the witness from the corresponding proof $\pi_{\mathsf{C},i}$ and verifies $\mathcal{R}_{\mathsf{C}}$ locally. The indistinguishability of games $\mathbf{G}_2$ and $\mathbf{G}_3$ follows from the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$.

Now, let $\mathsf{Adv}_{\mathcal{A}}^{j}$ be the advantage of a PPT adversary $\mathcal{A}$ in Game $\mathbf{G}_j$. We will now show that the games $\mathbf{G}_1$ and $\mathbf{G}_2$ are indistinguishable to an adversary, or rather that $\left|\mathsf{Adv}_{\mathcal{A}}^{2} - \mathsf{Adv}_{\mathcal{A}}^{1}\right|$ is negligible.

Suppose, for contradiction, that $\left|\mathsf{Adv}_{\mathcal{A}}^{2} - \mathsf{Adv}_{\mathcal{A}}^{1}\right|$ was non-negligible, then there is at least one query $(t^{(j)}, \sigma^{(j)} := (\cdot, \cdot, \pi_{\mathsf{C}}^{(j)}))$ to the Read oracle such that $\mathbf{G}_2$ returns $\perp$ where $\mathbf{G}_1$ does not (by completeness of $\mathsf{NIZK}_{\mathsf{C}}$, the opposite case is not possible). In other words, there does *not* exist a valid witness $\mathbb{w}^{(j)} \leftarrow \mathsf{NIZK}_{\mathsf{C}}.\mathcal{E}(\tau_{\mathsf{C}}, \mathbb{x}^{(j)}, \pi_{\mathsf{C},j})$ such that $(\mathbb{x}^{(j)}, \mathbb{w}^{(j)}) \in \mathcal{R}_{\mathsf{C}}$ where $\mathbb{x}^{(j)} := (\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{w}_0^{(j)}, \mathbf{w}_1^{(j)}, t^{(j)})$ and $\mathbb{w}^{(j)} := (\mathbf{u}^{(j)}, \mathbf{x}^{(j)}, \mathbf{z}^{(j)}, \boldsymbol{\nu}_0^{(j)}, \boldsymbol{\nu}_1^{(j)})$, but $\mathsf{NIZK}_{\mathsf{C}}.\mathsf{Verify}(\mathsf{crs}_{\mathsf{C}}, \mathbb{x}^{(j)}, \pi_{\mathsf{C}}^{(j)})$ accepts.

For any such j, we can construct a reduction $\mathcal{B}_{\mathsf{KSnd}}$ such that if $\epsilon(\lambda)$ is non-negligible, then $\mathcal{B}_{\mathsf{KSnd}}$ can break the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$ with non-negligible probability. In particular, the reduction does the following:

- ▷ The NIZK challenger C samples $\tau_{\mathsf{C}} \leftarrow\!\$\ \{0,1\}^\lambda$, obtains the reference string $\mathsf{crs}_{\mathsf{C}} \leftarrow \mathsf{NIZK}_{\mathsf{C}}.\mathcal{E}(\{0,1\}^\lambda; \tau_{\mathsf{C}})$ and sends $\mathsf{crs}_{\mathsf{C}}$ to the reduction, $\mathcal{B}_{\mathsf{KSnd}}$. The reduction, in turn, runs Setup_2 as in Game $\mathbf{G}_2$ except that

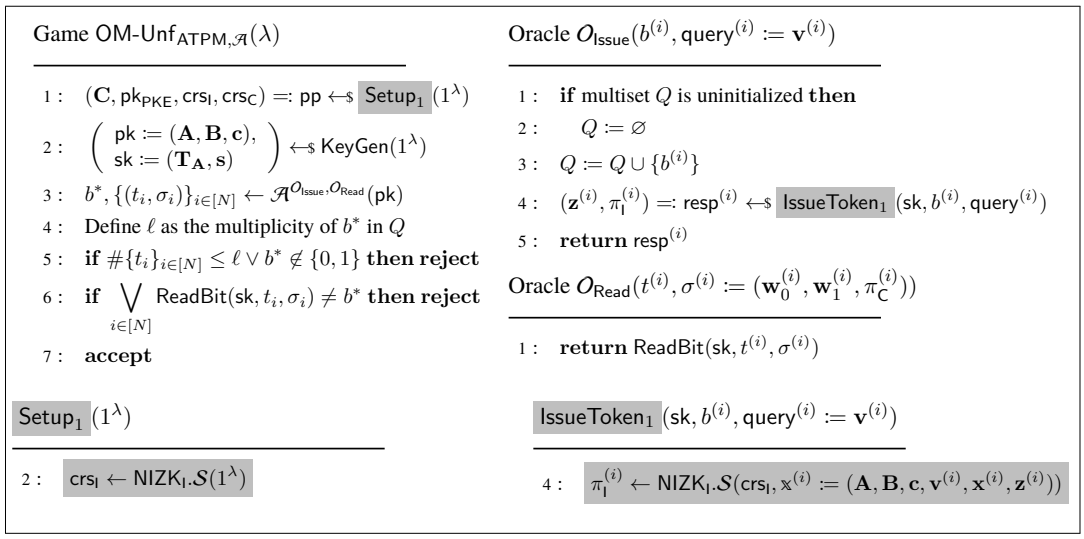

Figure 8: The one-more unforgeability Game $\mathbf{G}_1$ with differences from the previous game highlighted. For Setup and IssueToken, only the lines that change are shown.

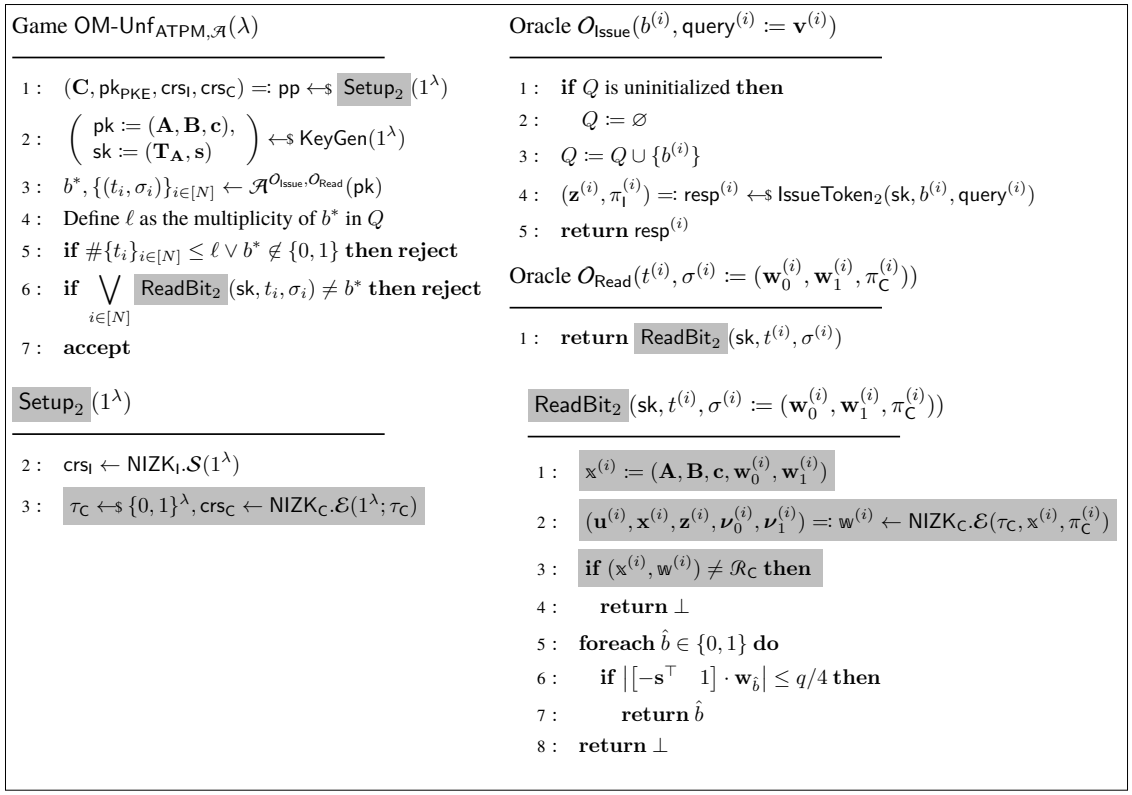

Figure 9: The one-more unforgeability Game $\mathbf{G}_2$ with differences from the previous game highlighted. For Setup, only the lines that change are shown.

it does not compute $\mathsf{crs}_{\mathsf{C}}$ itself but sets it as the value received from C, and sends the resulting pp to the adversary $\mathcal{A}$.

- ▷ The rest of the experiment proceeds as in Game $\mathbf{G}_2$ and for all Read queries to which $\mathsf{ReadBit}$ outputs $\perp$, $\mathcal{B}_{\mathsf{KSnd}}$ stores the queries and responses in Q_{Read}.
- ▷ Finally, when $\mathcal{A}$ outputs its forgeries, $\mathcal{B}_{\mathsf{KSnd}}$ samples $j \leftarrow\!\$\ [|Q_{\mathsf{Read}}|]$, sets $\mathbb{x}^{(j)} := (\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{w}_0^{(j)}, \mathbf{w}_1^{(j)}, t^{(j)})$ drawn from Q_{Read} and sends $(\mathbb{x}^{(j)}, \pi_{\mathsf{C}}^{(j)})$ to C.

By assumption, $(\mathbb{x}^{(j)}, \pi_{\mathsf{C}}^{(j)})$ breaks the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$ with probability $(1/|Q_{\mathsf{Read}}|) \cdot \epsilon(\lambda)$, which is non-negligible for $|Q_{\mathsf{Read}}| = \mathsf{poly}(\lambda)$ if $\epsilon(\lambda)$ is non-negligible.

Game $\mathbf{G}_3$: This is the same as $\mathbf{G}_2$, except instead of sampling $\mathbf{C}$ uniformly during Setup, the challenger first runs KeyGen to obtain $\mathbf{A}$, and then sets $\mathbf{C} := \mathbf{A}_\top \cdot \mathbf{R}$ where $\mathbf{R} \leftarrow\!\$\ \{0,1\}^{m' \times m'}$. We argue via the leftover hash lemma (Corollary 2.6) that $(\mathbf{A}, \mathbf{A} \cdot \mathbf{R})$ and $(\mathbf{A}, \mathbf{C})$ are statistically indistinguishable.

Game $\mathsf{OM\text{-}Unf}_{\mathsf{ATPM},\mathcal{A}}(\lambda)$

1 : $\left(\begin{array}{l}\mathsf{pk} := \left(\mathbf{A} := \begin{bmatrix}\mathbf{A}_\top \\ \mathbf{A}_\perp\end{bmatrix}, \mathbf{B}, \mathbf{c}\right), \\ \mathsf{sk} := (\mathbf{T}_{\mathbf{A}}, \mathbf{s})\end{array}\right) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$

2 : $(\mathbf{C}, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{I}}, \mathsf{crs}_{\mathsf{C}}) =: \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}_3(1^\lambda, \mathbf{A}_\top)$

3 : $b^*, \{(t_i, \sigma_i)\}_{i \in [N]} \leftarrow \mathcal{A}^{O_{\mathsf{Issue}}, O_{\mathsf{Read}}}(\mathsf{pk})$

4 : Define ℓ as the multiplicity of b^* in Q

5 : **if** $\#\{t_i\}_{i \in [N]} \leq \ell \vee b^* \notin \{0,1\}$ **then reject**

6 : **if** $\bigvee_{i \in [N]} \mathsf{ReadBit}_2(\mathsf{sk}, t_i, \sigma_i, \mathbf{x}_i) \neq b^*$ **then reject**

7 : **accept**

$\mathsf{Setup}_3(1^\lambda, \mathbf{A}^\top)$

1 : $\mathbf{R} \leftarrow\!\$\ \{0,1\}^{m' \times m'}, \mathbf{C} := \mathbf{A}_\top \cdot \mathbf{R}$

Oracle $O_{\mathsf{Issue}}(b^{(i)}, \mathsf{query}^{(i)} := \mathbf{v}^{(i)})$

1 : **if** Q is uninitialized **then**

2 : $\quad Q := \varnothing$

3 : $Q := Q \cup \{b^{(i)}\}$

4 : $(\mathbf{z}^{(i)}, \pi_{\mathsf{I}}^{(i)}) =: \mathsf{resp}^{(i)} \leftarrow\!\$\ \mathsf{IssueToken}_4(\mathsf{sk}, b^{(i)}, \mathsf{query}^{(i)})$

5 : **return** $\mathsf{resp}^{(i)}$

Oracle $O_{\mathsf{Read}}(t^{(i)}, \sigma^{(i)} := (\mathbf{w}_0^{(i)}, \mathbf{w}_1^{(i)}, \pi_{\mathsf{C}}^{(i)}))$

1 : **return** $\mathsf{ReadBit}_2(\mathsf{sk}, t^{(i)}, \sigma^{(i)})$

Figure 10: The one-more unforgeability Game $\mathbf{G}_3$ with differences from the previous game highlighted. For Setup, only the lines that change are shown.

More formally, since $\mathbf{R} \leftarrow\!\$\ \{0,1\}^{m' \times m'}$, we have $\mathbf{H}_\infty(\mathbf{R}) = m' \cdot m'$. As $m' > n \log q + \lambda$, we obtain from the leftover hash lemma the fact that following distributions are statistically indistinguishable:

$$\left\{(\mathbf{A}', \mathbf{A}' \cdot \mathbf{R}) \ : \ \mathbf{A}' \leftarrow\!\$\ \mathbb{Z}_q^{n \times m'}, \mathbf{R} \leftarrow\!\$\ \{0,1\}^{m' \times m'}\right\}$$
$$\approx_{\mathsf{s}} \left\{(\mathbf{A}', \mathbf{C}) \ : \ \mathbf{A}' \leftarrow\!\$\ \mathbb{Z}_q^{n \times m'}, \mathbf{C} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m'}\right\}$$

Then, by well-distributedness of $\mathbf{A}_\top$ (Definition 2.4), $\mathbf{G}_2$ and $\mathbf{G}_3$ are also statistically indistinguishable.

Finally, we will show that an adversary's advantage in $\mathbf{G}_3$ is negligible. To do so, we will prove that any adversary with non-negligible advantage $\mathsf{Adv}^3_{\mathcal{A}}$, there exists a reduction that either breaks the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$, or the one-more ISIS assumption. More formally, we have the following lemma.

Claim A.2. *Assuming the straight-line extractability of* $\mathsf{NIZK}_{\mathsf{C}}$ *and that the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ *problem is hard,* $\mathsf{Adv}^{3}_{\mathcal{A}}$ *is negligible for all* PPT *adversaries* $\mathcal{A}$.

Proof. We can separate the analysis into the following two cases:

- **Case 1.** There exists some $j \in [N]$ such that $\mathsf{ReadBit}(\mathsf{sk}, t_j, \sigma_j \coloneqq (\mathbf{w}_{0,j}, \mathbf{w}_{1,j}, \pi_{\mathsf{C},j})) = b^{*}(\neq \perp)$ but there does *not* exist a valid witness $\mathbb{w}_j \leftarrow \mathsf{NIZK}_{\mathsf{C}}.\mathcal{E}(\tau, \mathbb{x}_j, \pi_{\mathsf{C},j})$ such that $(\mathbb{x}_j, \mathbb{w}_j) \in \mathcal{R}_{\mathsf{C}}$ where $\mathbb{x}_j \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{w}_{0,j}, \mathbf{w}_{1,j}, t_j)$ and $\mathbb{w}_j \coloneqq (\mathbf{u}_j, \mathbf{x}_j, \mathbf{z}_j, \boldsymbol{\nu}_{0,j}, \boldsymbol{\nu}_{1,j})$.

- **Case 2.** For all $j \in [N]$, $\mathsf{ReadBit}(\mathsf{sk}, t_j, \sigma_j \coloneqq (\mathbf{w}_{0,j}, \mathbf{w}_{1,j}, \pi_{\mathsf{C},j})) = b^{*}(\neq \perp)$ and there exists a valid witness $\mathbb{w}_j \leftarrow \mathsf{NIZK}_{\mathsf{C}}.\mathcal{E}(\tau, \mathbb{x}_j, \pi_{\mathsf{C},j})$ such that $(\mathbb{x}_j, \mathbb{w}_j) \in \mathcal{R}_{\mathsf{C}}$ with $\mathbb{x}_j$ and $\mathbb{w}_j$ as defined previously.

Let $\epsilon(\lambda)$ be the adversary's advantage in the first case. Then, we can construct a reduction $\mathcal{B}_{\mathsf{KSnd}}$ such that if $\epsilon(\lambda)$ is non-negligible, then $\mathcal{B}_{\mathsf{KSnd}}$ can break the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$ with non-negligible probability. In particular, the reduction does the following:

- ▷ The NIZK challenger $\mathcal{C}$ samples $\tau \leftarrow\!\!\$\ \{0,1\}^{\lambda}$, obtains the reference string $\mathsf{crs}_{\mathsf{C}} \leftarrow \mathsf{NIZK}_{\mathsf{C}}.\mathcal{E}(\{0,1\}^{\lambda}; \tau)$ and sends $\mathsf{crs}_{\mathsf{C}}$ to the reduction, $\mathcal{B}_{\mathsf{KSnd}}$. The reduction, in turn, runs Setup_2 as in Game $\mathbf{G}_3$ except that it does not compute $\mathsf{crs}_{\mathsf{C}}$ itself but sets it as the value received from $\mathcal{C}$, and sends the resulting pp to the adversary $\mathcal{A}$.
- ▷ The rest of the experiment proceeds as in Game $\mathbf{G}_3$.
- ▷ Finally, when $\mathcal{A}$ outputs its (one-more) forgeries $\{t_i, \sigma_i \coloneqq \mathbf{w}_{0,i}, \mathbf{w}_{1,i}, \pi_{\mathsf{C},i})\}$, $\mathcal{B}_{\mathsf{KSnd}}$ samples $j \leftarrow\!\!\$\ [N]$, sets $\mathbb{x}_j \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{w}_{0,j}, \mathbf{w}_{1,j}, t_j)$ and sends $(\mathbb{x}_j, \pi_{\mathsf{C},j})$ to $\mathcal{C}$.

By assumption, $(\mathbb{x}_j, \pi_{\mathsf{C},j})$ breaks the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$ with probability $(1/N) \cdot \epsilon(\lambda)$, which is non-negligible for $N = \mathsf{poly}(\lambda)$ if $\epsilon(\lambda)$ is non-negligible.

Similarly, let $\epsilon(\lambda)$ be the adversary's advantage in the second case. We show that we can construct a reduction $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ such that if $\epsilon(\lambda)$ is non-negligible, then $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ can break OM-ISIS assumption with non-negligible probability. In particular, the reduction does the following:

- ▷ It sets $\mathbf{A} \coloneqq \begin{bmatrix} \mathbf{A}_{\top} \\ \mathbf{A}_{\perp} \end{bmatrix}$ for $\mathbf{A}_{\top}$ given to it by the OM-ISIS challenger $\mathcal{C}$, and $\mathbf{A}_{\perp} \leftarrow\!\!\$\ \mathbb{Z}_q^{(n+1)\times m'}$. It computes the rest of the pk and sk (except $\mathbf{T}_{\mathbf{A}_{\top}}$) as before.
- ▷ On receiving a *fresh* random oracle query for $\mathsf{H}(t^{(i)})$, the reduction $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ in turn queries $\mathcal{C}$ for a syndrome $\mathbf{t}^{(i)}$ (i.e., makes a *syndrome query*) and forwards it to $\mathcal{A}$. Let $\mathcal{R}$ be an ordered set containing the random oracle queries $t^{(i)}$ and $\mathcal{S}$ the ordered set containing the corresponding syndromes $\mathbf{t}^{(i)}$ returned by the OM-ISIS challenger.
- ▷ On receiving a Issue query $(b^{(i)}, \mathsf{query}^{(i)} \coloneqq \mathbf{v}^{(i)})$, it inserts $b^{(i)}$ into Q, and in turn queries $\mathcal{C}$ for a preimage of $\mathbf{v}^{(i)}$ (i.e., makes a *preimage query*) , to which $\mathcal{C}$ responds with a preimage $\mathbf{z}^{(i)}$. $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ forwards $\mathbf{z}^{(i)}$ and the simulated proof $\pi_{\mathsf{I}}^{(i)}$ to $\mathcal{A}$.
- ▷ On receiving a Read query $(t^{(i)}, \sigma^{(i)})$, it answers it as explained in Game $\mathbf{G}_4$.

▷ When $\mathcal{A}$ outputs its forgeries $\{t_i, \sigma_i := (\mathbf{w}_{0,i}, \mathbf{w}_{1,i}, \pi_{\mathsf{C},i})\}_{i\in[N]}$, the reduction $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ chooses a random subset $\{t_i, \sigma_i\}_{i\in[\ell+1]}$ of the $(\ell+1) \leq N$ *distinct* tags t_i. From this, it extracts $\mathbf{u}_i, \mathbf{z}_i$ and computes $\mathbf{z}'_i := \mathbf{z}_i - \mathbf{R}\cdot\mathbf{x}_i$ and $\mathbf{t}_i := \mathbf{A}_\top\cdot\mathbf{z}_i - \mathbf{C}\cdot\mathbf{x}_i$ for each $i \in [\ell+1]$. It then outputs $\{(\mathbf{z}'_i, \mathbf{t}_i)\}_{i\in[\ell+1]}$ as its OM-ISIS preimages.

Clearly, since each $(\mathbb{x}_i, \mathbb{w}_i)$ satisfies the relation $\mathcal{R}_\mathsf{C}$, $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ must succeed if $\mathbf{t}_i \neq \mathbf{t}_j$ and $\mathbf{t}_i \in \mathcal{S}$ for all $i \neq j \in [\ell+1]$ where $\mathcal{S}$ is the set of challenge syndromes. Now, consider the following events:

- **Case 2.1** For all $i \in [\ell+1]$, $t_i \in \mathcal{R}$, i.e., every tag in the forgery was queried by $\mathcal{A}$ to the random oracle.
- **Case 2.2** There exists a $j \in [\ell+1]$, such that $t_j \notin \mathcal{R}$, i.e., some tag t_j in the forgery was *not* queried by $\mathcal{A}$ to the random oracle.

Let us first consider the latter "bad" case, which implies that $\mathcal{A}$ must never have queried the random oracle on t_j for some $j \in [\ell+1]$. However, the random oracle model dictates that $\mathsf{H}(t_j)$ must be undefined, and therefore completely random from the point of view of the adversary. Then, since it must hold that $\mathsf{H}(t_i) = \mathbf{A}_\top\cdot\mathbf{z}_i - \mathbf{C}\cdot\mathbf{u}_i$, it follows that the probability of the "bad" event is at most negligible.

Now, in the former "good" case, since for each $i \in [\ell+1]$, $\mathbf{t}_i = \mathsf{H}(t_i)$, it necessarily holds that each $\mathbf{t}_i$ is distinct and is in the set $\mathcal{S}$. Moreover, we have that

$$\begin{aligned}\mathbf{t}_i &= \mathbf{A}_\top\cdot\mathbf{z}_i - \mathbf{C}\cdot\mathbf{u}_i \\ &= \mathbf{A}_\top\cdot(\mathbf{z}_i - \mathbf{R}\cdot\mathbf{u}_i) \quad .\end{aligned}$$

Since $\|\mathbf{z}_i\| \leq \mathfrak{B}_\mathbf{z}$ and $\|\mathbf{u}_i\| \leq \mathfrak{B}_\mathbf{z}/m$, and $\mathbf{R}$ is an $m' \times m'$ binary matrix,

$$\|\mathbf{z}_i - \mathbf{R}\cdot\mathbf{u}_i\| \leq 2\mathfrak{B}_\mathbf{z} \quad .$$

Thus, each $(\mathbf{z}_i - \mathbf{R}\cdot\mathbf{u}_i)$ is a valid $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_\mathbf{z},2\mathfrak{B}_\mathbf{z}}$ solution with probability nearly $\epsilon(\lambda)$. □

Thus, construction 3 is one-more unforgeable.

□

Lemma A.3 (κ-unlinkability)**.** *Construction 3 is κ-unlinkable for $\kappa = 2$, assuming NIZK_I is sound and NIZK_C satisfies zero-knowledge.*

Proof. We proceed through a sequence of games:

Game $\mathbf{G}_0$: This is the original ATPM unlinkability experiment $\mathsf{UNLINK}_{\mathsf{ATPM},\mathcal{A},n}$ defined with respect to construction 3 (without public metadata).

Game $\mathbf{G}_1$: This is the same as $\mathbf{G}_0$, except the challenger simulates each NIZK proof π_C without a witness. It is easily shown that if NIZK_C satisfies zero-knowledge, then $\mathbf{G}_0$ and $\mathbf{G}_1$ are indistinguishable to the adversary.

Game $\mathbf{G}_2$: This is the same as $\mathbf{G}_1$, except on every Query oracle call by the adversary, instead of running ClientQuery as usual, the challenger will instead sample uniform vector $\boldsymbol{v} \leftarrow\!\!\$\ \mathbb{Z}_q^n$ and set the corresponding $\mathsf{query}^{(i)} := \boldsymbol{v} + \mathsf{H}(t^{(i)})$. We argue via the leftover hash lemma (Corollary 2.6) that $(\mathbf{C}, \mathbf{C}\cdot\mathbf{u})$ and $(\mathbf{C}, \boldsymbol{v})$ are statistically indistinguishable.

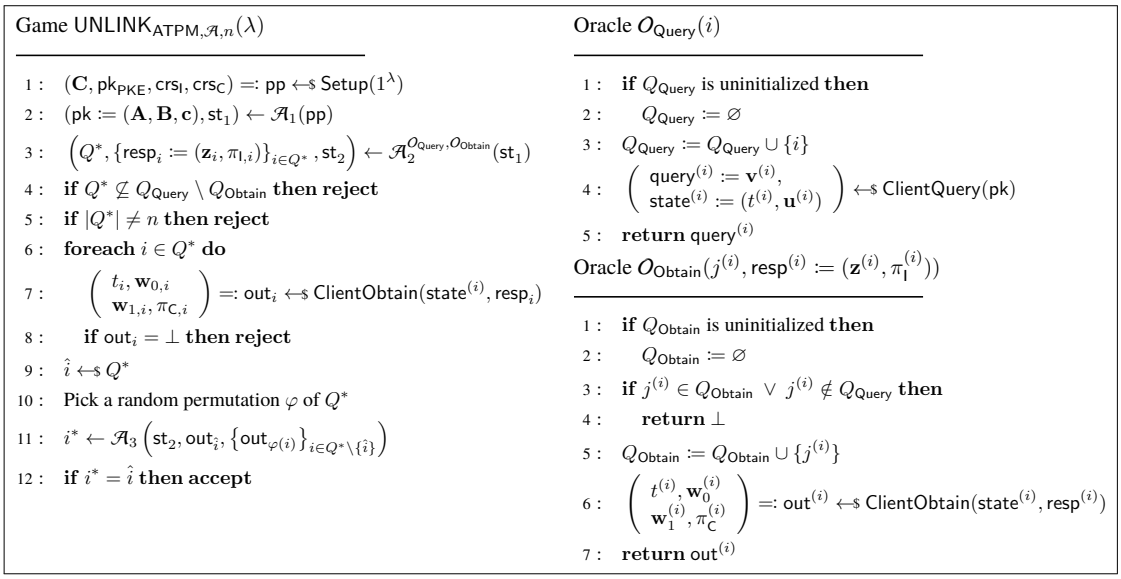

Figure 11: The κ-unlinkability Game $\mathbf{G}_0$.

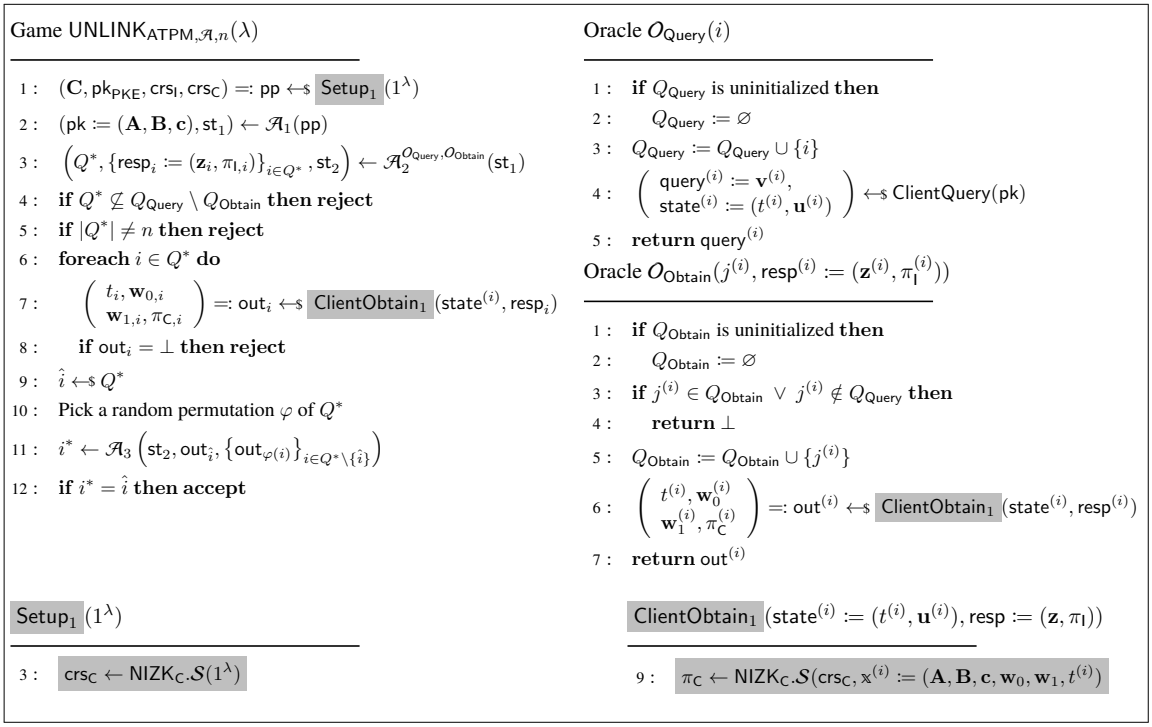

Figure 12: The κ-unlinkability Game $\mathbf{G}_1$ with differences from the previous game highlighted. For Setup and ClientObtain, only the lines that change are shown.

More formally, since $\mathbf{u} \leftarrow\$ \Gamma_{\mathbb{Z}^{m'}, \frac{s_{\mathbf{Z}}}{2m'}}$, we have $\mathbf{H}_\infty(\mathbf{R}) \geq m'$. As $m' > n \log q + \lambda$, we obtain from the leftover hash lemma the fact that following distributions are statistically indistinguishable:

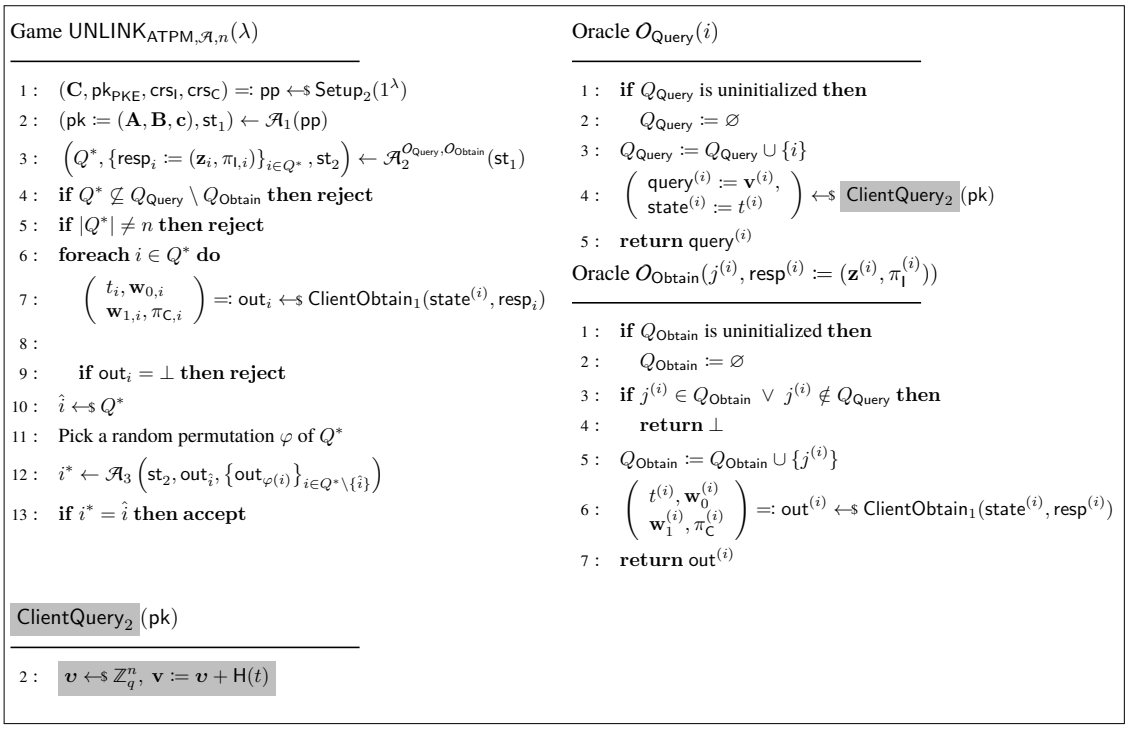

Figure 13: The κ-unlinkability Game $\mathbf{G}_2$ with differences from the previous game highlighted. For ClientQuery, only the lines that change are shown.

$$\left\{(\mathbf{C}, \mathbf{C} \cdot \mathbf{u}) \ : \ \mathbf{C} \leftarrow\$ \mathbb{Z}_q^{n \times m'}, \mathbf{u} \leftarrow\$ \Gamma_{\mathbb{Z}^{m'}, \frac{s_{\mathbf{Z}}}{2m'}}\right\} \approx_{\mathsf{s}} \left\{(\mathbf{C}, \boldsymbol{\upsilon}) \ : \ \mathbf{C} \leftarrow\$ \mathbb{Z}_q^{n \times m'}, \boldsymbol{\upsilon} \leftarrow\$ \mathbb{Z}_q^n\right\}$$

It therefore follows that $\mathbf{G}_1$ and $\mathbf{G}_2$ are also indistinguishable to the adversary.

Let us now analyze the following two cases:

- **Case 1.** For all $i \in Q^*$, there exists a witness $\mathbb{w}_i$ such that $(\mathbb{x}_i, \mathbb{w}_i) \in \mathcal{R}_{\mathsf{I}}$ and $\mathsf{ClientObtain}_1(\mathsf{state}^{(i)}, \mathsf{resp}_i)$ does not abort.
- **Case 2.** For some $j \in Q^*$, there does *not* exist a witness $\mathbb{w}_j$ such that $(\mathbb{x}_j, \mathbb{w}_j) \in \mathcal{R}_{\mathsf{I}}$, but $\mathsf{ClientObtain}_1(\mathsf{state}^{(j)}, \mathsf{resp}_j)$ does not abort.

Considering, first, the latter of the two cases, let $\epsilon(\lambda)$ be the adversary's advantage. Then, we can construct a reduction $\mathcal{B}_{\mathsf{Snd}}$ such that if $\epsilon(\lambda)$ is non-negligible, then $\mathcal{B}_{\mathsf{Snd}}$ can break the soundness of $\mathsf{NIZK}_{\mathsf{I}}$ with non-negligible probability. In particular, the reduction does the following:

▷ The NIZK challenger C runs the setup to obtain $\mathsf{crs}_{\mathsf{I}}$ it over to the reduction, $\mathcal{B}_{\mathsf{Snd}}$. The reduction, in turn, runs Setup_2 as in Game $\mathbf{G}_3$ except that it does not compute $\mathsf{crs}_{\mathsf{I}}$ itself but sets it as the value received from C, and sends the resulting pp to the adversary $\mathcal{A}$.

▷ The rest of the experiment proceeds as in Game $\mathbf{G}_3$.

▷ Finally, when $\mathcal{A}$ outputs $\{\mathsf{resp}_i := (\mathbf{z}_i, \pi_{\mathsf{I},i})\}_{i \in Q^*}$, $\mathcal{B}_{\mathsf{Snd}}$ samples $j \leftarrow\$ Q^*$, sets $\mathbb{x}_j := (\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{v}^{(j)}, \mathbf{A}_\perp \cdot \mathbf{z}_j, \mathbf{z}_j)$ where $\mathbf{v}^{(j)} =: \mathsf{query}^{(j)}$ and sends $(\mathbb{x}_j, \pi_{\mathsf{I},j})$ to C.

By assumption, $(\mathbb{x}_j, \pi_{\mathsf{I},j})$ breaks soundness of $\mathsf{NIZK}_{\mathsf{C}}$ with probability $(1/n) \cdot \epsilon(\lambda)$, which is non-negligible for $n = \mathsf{poly}(\lambda)$ if $\epsilon(\lambda)$ is non-negligible.

Returning now to the first case, we can make a strictly information theoretic argument to reason that $\mathbf{v}^{(i)}$ is uniformly random for each $i \in Q^*$. Moreover, by the smudging lemma (Lemma 2.7), we have that $(\mathbf{r}^\star + \boldsymbol{\nu}_{0,i})$ is distributed as $\boldsymbol{\nu}_{0,i}$ (similarly for $\boldsymbol{\nu}_{1,i}$). Therefore, the corresponding final challenge output, $\mathsf{out}_i := (t_i, \mathbf{w}_{0,i}, \mathbf{w}_{1,i}, \pi_{\mathsf{C},i})$ is also independent of the response $(\mathbf{z}_i, \pi_{\mathsf{I},i})$. It follows that the only meaningful information the adversary can learn in this case is the private bit embedded in $\mathbf{w}_{0,i}$ and $\mathbf{w}_{1,i}$. So the best adversarial strategy is to create some n_b responses with bit b for each bit ($n_0 + n_1 = n$), read the bit $b_{\hat{i}}$ of $\mathsf{out}_{\hat{i}}$ and output a random value i^* from the set of all $n_{b_{\hat{i}}}$ indices where the embedded bit was equal to $b_{\hat{i}}$. The probability that the adversary wins is

$$\sum_{b \in \{0,1\}} \Pr\left[b_{i^*} = b_{\hat{i}}\right] \cdot \Pr\left[i^* = \hat{i}\right] = \sum_{b \in \{0,1\}} \frac{n_b}{n} \cdot \frac{1}{n_b} = \frac{2}{n} \ .$$

Let $\mathsf{Adv}^j_{\mathcal{A}}$ be the advantage of a PPT adversary $\mathcal{A}$ in Game $\mathbf{G}_j$. Then, our analysis shows that $\mathsf{Adv}^2_{\mathcal{A}} \leq \frac{2}{n} + \mathsf{negl}(\lambda)$.

Thus, construction 3 is 2-unlinkable. □

Lemma A.4 (Privacy of metadata bit)**.** *Construction 3 has private metadata bit assuming* $\mathsf{NIZK}_{\mathsf{I}}$ *satisfies zero-knowledge,* $\mathsf{NIZK}_{\mathsf{C}}$ *is straight-line extractable, and the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ *and* $\mathsf{LWE}_{q,n,m,\mathfrak{s}_{\boldsymbol{\epsilon}}}$ *problems are hard.*

Proof. We proceed through a sequence of games:

Game $\mathbf{G}_0$: This is the original ATPM metadata bit privacy experiment $\mathsf{PMB}_{\mathsf{ATPM},\mathcal{A},\hat{b}}$ for $\hat{b} \in \{0,1\}$ defined with respect to construction 3 (without public metadata).

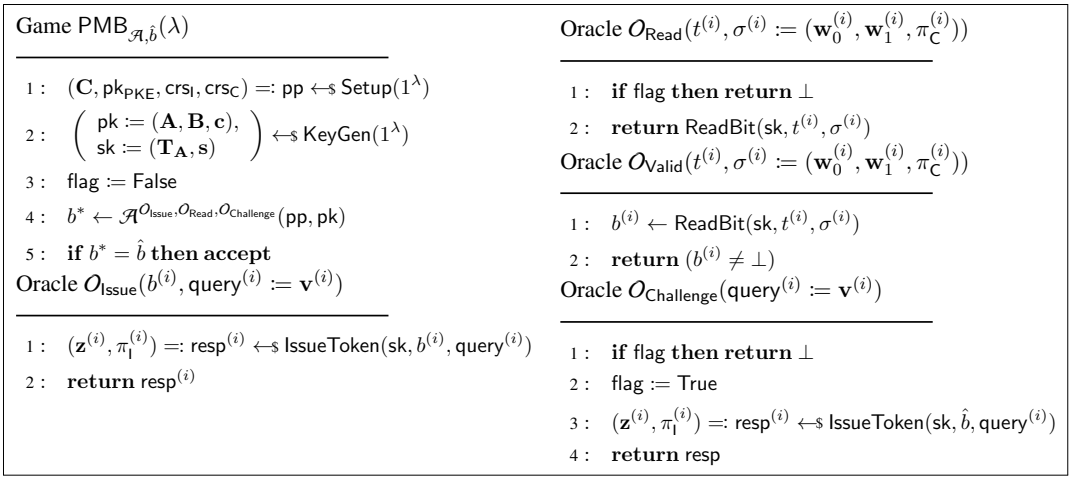

Figure 14: The privacy of metadata bit Game $\mathbf{G}_0$.

Game $\mathbf{G}_1$: This is the same as $\mathbf{G}_0$, except the challenger simulates each NIZK proof without a witness. It is easily shown that if $\mathsf{NIZK}_{\mathsf{I}}$ satisfies zero-knowledge, then $\mathbf{G}_0$ and $\mathbf{G}_1$ are indistinguishable to the adversary.

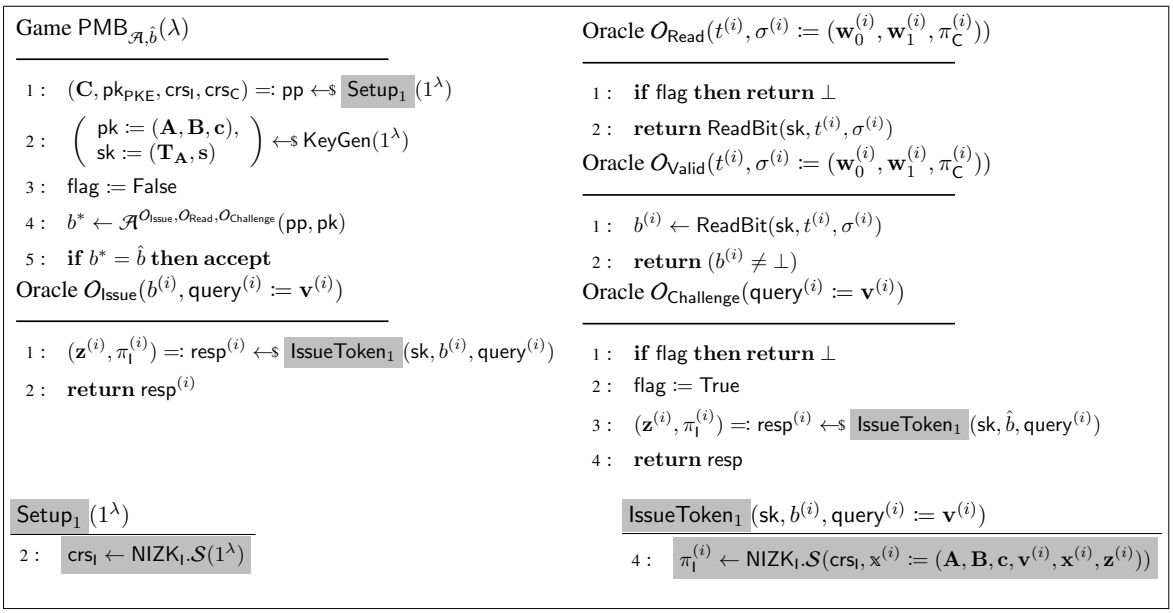

Figure 15: The privacy of metadata bit Game $\mathbf{G}_1$ with differences from the previous game highlighted. For Setup and IssueToken, only the lines that change are shown.

Game $\mathbf{G}_2$: This is the same as $\mathbf{G}_1$, except for each of the ith Read or Valid queries, instead of running the ReadBit algorithm as before, the challenger runs the extractor for $\mathsf{NIZK}_{\mathsf{C}}$ to extract the witness from the corresponding proof $\pi_{\mathsf{C}}^{(i)}$ and verifies the relation $\mathcal{R}_{\mathsf{C}}$ locally. Similarly, for each of the ith (one-more) forgery, it extracts the witness from the corresponding proof $\pi_{\mathsf{C},i}$ and verifies $\mathcal{R}_{\mathsf{C}}$ locally. The indistinguishability of games $\mathbf{G}_1$ and $\mathbf{G}_2$ follows from the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$ as shown in the corresponding game (see Figure 9) in the one-more unforgeability proof above.

Game $\mathbf{G}_3$: This is the same as $\mathbf{G}_2$, except for each Issue query, the challenger stores the corresponding $(\mathbf{x}^{(i)}, b^{(i)})$ pair in Q_{BitMap}. Clearly, this affects no change to the adversary's point-of-view.

Game $\mathbf{G}_4$: This is the same as $\mathbf{G}_3$, except for each call to ReadBit, instead of directly attempting to decrypt $\mathbf{w}_0$ and $\mathbf{w}_1$, the challenger first uses the extracted $\mathbf{x}$ value to look up the corresponding bit in Q_{BitMap} and returns it if one is found and aborts otherwise. Notice that games $\mathbf{G}_3$ and $\mathbf{G}_4$ are distinguishable only if an adversary makes a Read or Valid query for a $(t^\star, \sigma^\star)$ pair such that the extracted $\mathbf{x}^\star$ does not have a corresponding entry in Q_{BitMap}. In other words, the adversary was never issued a $\mathbf{z}^\star$ with $\mathbf{x}^\star = \mathbf{A}_\perp \cdot \mathbf{z}^\star$ such that $\mathbf{A}_\top \cdot \mathbf{z}^\star = \mathbf{C} \cdot \mathbf{u}^\star + \mathsf{H}(t^\star)$. Thus, we can use the same argument as was given in Claim A.2 to show that $\mathbf{G}_3$ and $\mathbf{G}_4$ are indistinguishable to an adversary assuming the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$ and that the $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ problem is hard.

Now, let $\mathsf{Adv}_{\mathcal{A}}^{j}$ be the advantage of a PPT adversary $\mathcal{A}$ in Game $\mathbf{G}_j$. We will show that the adversary's advantage in $\mathbf{G}_4$ is negligible. More formally, we have the following lemma.

Claim A.5. *Assuming that the* $\mathsf{LWE}_{q,n,m,\mathfrak{s}_{\boldsymbol{\varepsilon}}}$ *problem is hard,* $\mathsf{Adv}_{\mathcal{A}}^{4}$ *is negligible for all* PPT *adversaries* $\mathcal{A}$.

Proof. We will show that each $\mathbf{x}^{(i)}$ (necessarily computed as $\mathbf{A}_\perp \cdot \mathbf{z}^{(i)}$) by the adversary is indistinguishable from uniform. First, by the hardness of $\mathsf{LWE}_{q,n,m,\mathfrak{s}_{\boldsymbol{\varepsilon}}}$, we see that

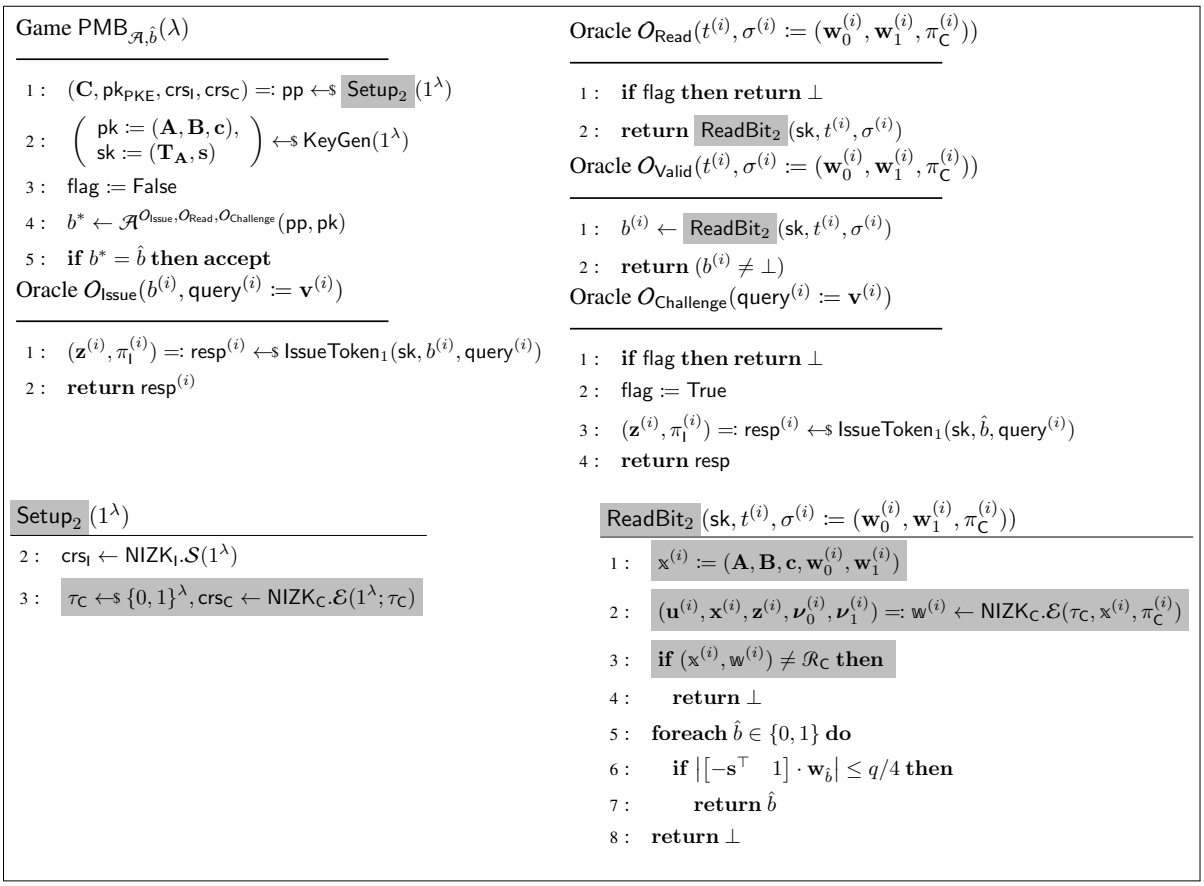

Figure 16: The privacy of metadata bit Game $\mathbf{G}_2$ with differences from the previous game highlighted. For Setup and IssueToken, only the lines that change are shown.

$$\left(\mathbf{B}, \mathbf{s}^\top\mathbf{B} + \boldsymbol{\varepsilon} \ : \ \mathbf{B} \leftarrow\$ \mathbb{Z}_q^{n\times m}, \mathbf{s} \leftarrow\$ \mathbb{Z}_q^n, \varepsilon \leftarrow\$ \Gamma_{\mathbb{Z}^m, \mathfrak{s}_{\boldsymbol{\varepsilon}}}\right) \approx_{\mathsf{s}} \left(\mathbf{B}, \mathbf{c} \ : \ \mathbf{B} \leftarrow\$ \mathbb{Z}_q^{n\times m}, \mathbf{c} \leftarrow\$ \mathbb{Z}_q^m\right) \ .$$

Consequently, $\begin{bmatrix} \mathbf{B} \\ \mathbf{c}^\top \end{bmatrix}$ is statistically indistinguishable from a uniform matrix $\mathbf{U} \in \mathbb{Z}_q^{(n+1)\times m}$. Using this, we argue via the leftover hash lemma (Corollary 2.6) that $(\mathbf{U}, \mathbf{U}\cdot\mathbf{r})$ and $(\mathbf{U}, \boldsymbol{\rho})$ are statistically indistinguishable for uniform $\boldsymbol{\rho}$. More formally, since $\mathbf{r} \leftarrow\$ \mathbb{Z}_q^m$, we have $\mathbf{H}_\infty(\mathbf{r}) = m$. As $m > n\log q + \lambda$, we obtain from the leftover hash lemma the fact that following distributions are statistically indistinguishable:

$$\left\{(\mathbf{U}, \mathbf{U}\cdot\mathbf{r}) \ : \ \mathbf{U} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times m}, \mathbf{r} \leftarrow\$ \mathbb{Z}_q^m\right\} \approx_{\mathsf{s}} \left\{(\mathbf{U}, \boldsymbol{\rho}) \ : \ \mathbf{U} \leftarrow\$ \mathbb{Z}_q^{(n+1)\times m}, \boldsymbol{\rho} \leftarrow\$ \mathbb{Z}_q^{n+1}\right\} \ .$$

It therefore follows that $\mathbf{x} := \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^\top \end{bmatrix} \cdot \mathbf{r} + b \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix}$ is statistically indistinguishable from uniform. Further, since the distribution over $\mathbf{z}$ is statistically close to $\Gamma_{\mathbb{Z}^m, \mathfrak{s}_{\mathbf{z}}}$ (Definition 2.4), we have $\mathbf{H}_\infty(\mathbf{z}) \geq m$. As $m > n\log q + \lambda$, we obtain from the leftover hash lemma the fact that following distributions are statistically indistinguishable:

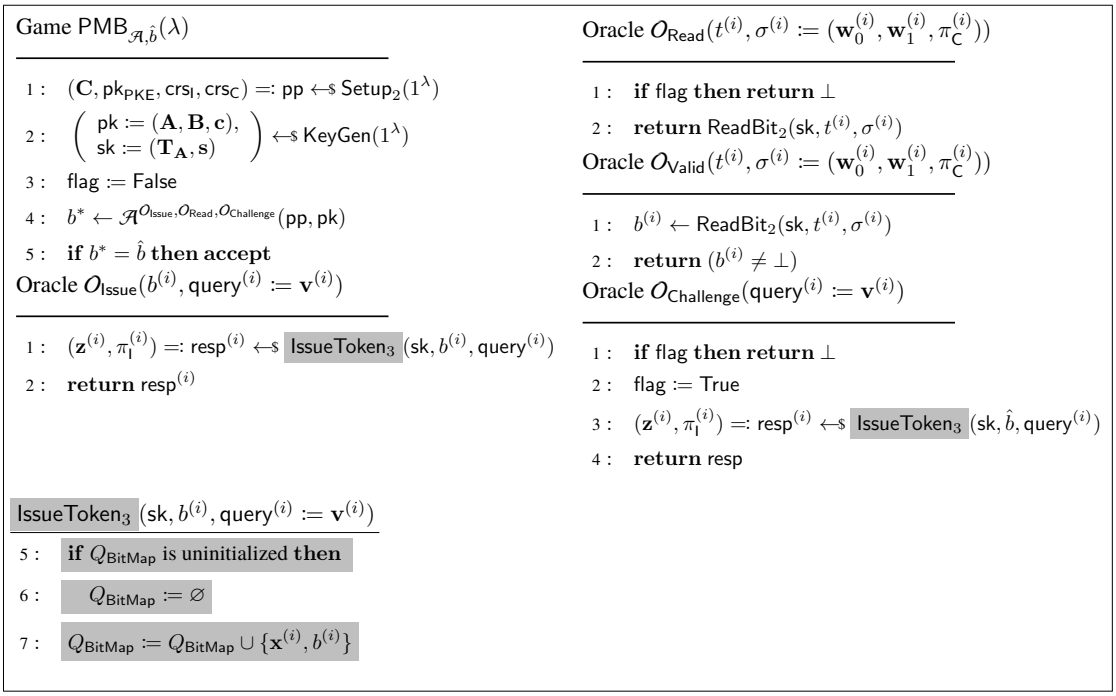

Figure 17: The privacy of metadata bit Game $\mathbf{G}_3$ with differences from the previous game highlighted. For Setup and IssueToken, only the lines that change are shown.

$$\left\{(\mathbf{A}_\perp, \mathbf{x}) \ : \ \mathbf{A}_\perp \leftarrow\$ \mathbb{Z}_q^{(n+1)\times m}, \mathbf{x} \leftarrow\$ \mathbb{Z}_q^{n+1}\right\} \approx_{\mathsf{s}}$$
$$\left\{(\mathbf{A}_\perp, \mathbf{A}_\perp \cdot \mathbf{z}) \ : \ \mathbf{A}_\perp \leftarrow\$ \mathbb{Z}_q^{(n+1)\times m}, \mathbf{z} \leftarrow\$ \Gamma_{\mathbb{Z}^m, \mathfrak{s}_{\mathbf{z}}}\right\}$$

This concludes the proof. □

Thus, construction 3 satisfies privacy of metadata bit. □

B Proofs for Section 4

Below, we prove ATPM security of our expanded construction.

Theorem B.1. *Let system parameters $n, m, m', \mathfrak{s}_\varepsilon, \mathfrak{B}_\varepsilon = \mathfrak{s}_\varepsilon\sqrt{m}, \mathfrak{s}_{\mathbf{z}}, \mathfrak{B}_{\mathbf{z}} = \mathfrak{s}_{\mathbf{z}}\sqrt{m'}$ and a prime number q be functions of the security parameter λ satisfying the constraints defined in (1), then construction 4.1 is:*

(i) one-more unforgeable (in the random oracle model) assuming $\mathsf{NIZK}_{\mathsf{I}}$ *satisfies zero-knowledge,* $\mathsf{NIZK}_{\mathsf{C}}$ *is sound and the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ *problem is hard;*

(ii) κ-unlinkable for $\kappa = 2$, assuming PKE *is* IND-CPA *secure,* $\mathsf{NIZK}_{\mathsf{I}}$ *is sound and* $\mathsf{NIZK}_{\mathsf{C}}$ *satisfies zero-knowledge; and*

(iii) has private metadata bit assuming $\mathsf{NIZK}_{\mathsf{I}}$ *satisfies zero-knowledge,* $\mathsf{NIZK}_{\mathsf{C}}$ *is sound, and the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ *and* $\mathsf{LWE}_{q,n,m,\mathfrak{s}_\varepsilon}$ *problems are hard.*

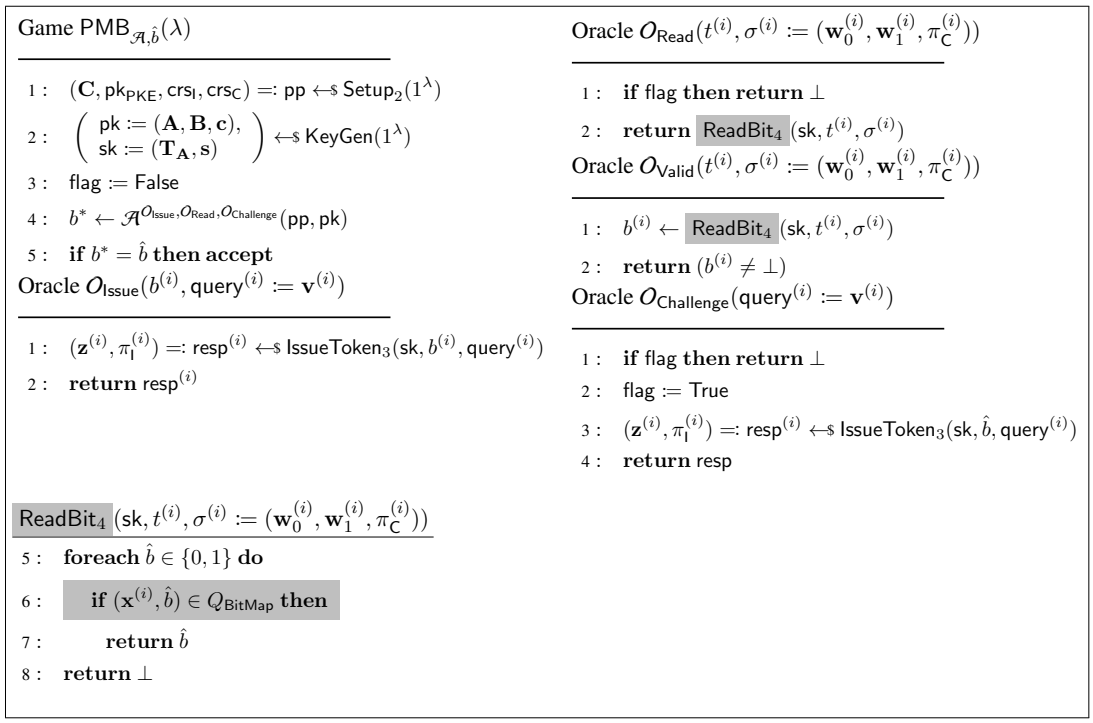

Figure 18: The privacy of metadata bit Game $\mathbf{G}_4$ with differences from the previous game highlighted. For Setup and IssueToken, only the lines that change are shown.

Lemma B.2 (One-more unforgeability)**.** *Construction 4.1 is one-more unforgeable (in the random oracle model) assuming* $\mathsf{NIZK}_{\mathsf{I}}$ *satisfies zero-knowledge,* $\mathsf{NIZK}_{\mathsf{C}}$ *is sound and the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ *problem is hard.*

Proof. The proof proceeds through a sequence of games:

Game $\mathbf{G}_0$: This is the original ATPM one-more unforgeability experiment $\mathsf{OM\text{-}Unf}_{\mathsf{ATPM},\mathcal{A}}$ defined with respect to construction 4.1 (without public metadata; see Figure 7).

Game $\mathbf{G}_1$: This is the same as $\mathbf{G}_0$, except for each Issue query, the challenger simulates the corresponding proof $\pi_{\mathsf{I}}^{(i)}$ without a witness (see Figure 8). It is easily shown that if $\mathsf{NIZK}_{\mathsf{I}}$ satisfies zero-knowledge, then $\mathbf{G}_1$ and $\mathbf{G}_2$ are indistinguishable to the adversary.

Game $\mathbf{G}_2$: This is the same as $\mathbf{G}_1$, except for each of the ith Read queries, instead of running the ReadBit algorithm as before, the challenger uses $\mathsf{sk}_{\mathsf{PKE}}$ to extract the witness from the corresponding proof $\pi_{\mathsf{C}}^{(i)}$ and verifies the relation $\mathcal{R}'_{\mathsf{C}}$ locally. Similarly, for each of the ith (one-more) forgery, it extracts the witness from the corresponding proof $\pi_{\mathsf{C},i}$ and verifies $\mathcal{R}'_{\mathsf{C}}$ locally (see Figure 9). The indistinguishability of games $\mathbf{G}_1$ and $\mathbf{G}_2$ follows from the soundness of $\mathsf{NIZK}_{\mathsf{C}}$.

Game $\mathbf{G}_3$: This is the same as $\mathbf{G}_2$, except instead of sampling $\mathbf{C}$ uniformly during Setup, the challenger first runs KeyGen to obtain $\mathbf{A}$, and then sets $\mathbf{C} := \mathbf{A}_\top \cdot \mathbf{R}$ where $\mathbf{R} \leftarrow\$ \{0,1\}^{m' \times m'}$ (see Figure 10). As with Lemma A.1 (Game $\mathbf{G}_3$), we can argue via the leftover hash lemma (Corollary 2.6) that $(\mathbf{A}, \mathbf{A} \cdot \mathbf{R})$ and $(\mathbf{A}, \mathbf{C})$ are statistically indistinguishable.

Finally, by the same argument as Lemma A.1, we can show that an adversary's advantage in $\mathbf{G}_3$ is negligible. Thus, construction 4.1 is one-more unforgeable. □

Lemma B.3 (κ-unlinkability)**.** *Construction 4.1 is κ-unlinkable for $\kappa = 2$, assuming* PKE *is* IND-CPA *secure,* NIZK_I *is sound and* NIZK_C *satisfies zero-knowledge.*

Proof. We proceed through a sequence of games:

Game $\mathbf{G}_0$: This is the original ATPM unlinkability experiment $\mathsf{UNLINK}_{\mathsf{ATPM},\mathcal{A},n}$ defined with respect to construction 4.1 (without public metadata; see Figure 11).

Game $\mathbf{G}_1$: This is the same as $\mathbf{G}_0$, except the challenger simulates each NIZK proof π_C without a witness (see Figure 12). It is easily shown that if NIZK_C satisfies zero-knowledge, then $\mathbf{G}_0$ and $\mathbf{G}_1$ are indistinguishable to the adversary.

Game $\mathbf{G}_2$: This is the same as $\mathbf{G}_1$, except instead of computing each ψ_C as an encryption of the client's secrets during ClientObtain, the challenger computes them as encryptions of $\mathbf{0}$. We will show that assuming PKE is IND-CPA secure, games $\mathbf{G}_1$ and $\mathbf{G}_2$ are indistinguishable to an adversary.

Game $\mathsf{UNLINK}_{\mathsf{ATPM},\mathcal{A},n}(\lambda)$

1 : $(\mathbf{C}, \mathsf{pk}_\mathsf{PKE}, \mathsf{crs}_\mathsf{I}, \mathsf{crs}_\mathsf{C}) =: \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}_1(1^\lambda)$
2 : $(\mathsf{pk} := (\mathbf{A}, \mathbf{B}, \mathbf{c}), \mathsf{st}_1) \leftarrow \mathcal{A}_1(\mathsf{pp})$
3 : $\left(Q^*, \{\mathsf{resp}_i := (\mathbf{z}_i, \pi_{\mathsf{I},i})\}_{i \in Q^*}, \mathsf{st}_2\right) \leftarrow \mathcal{A}_2^{O_\mathsf{Query}, O_\mathsf{Obtain}}(\mathsf{st}_1)$
4 : **if** $Q^* \not\subseteq Q_\mathsf{Query} \setminus Q_\mathsf{Obtain}$ **then reject**
5 : **if** $|Q^*| \neq n$ **then reject**
6 : **foreach** $i \in Q^*$ **do**
7 : $\quad \begin{pmatrix} \mathbf{w}_{0,i}, \mathbf{w}_{1,i}, \\ \psi_{\mathsf{C},i}, \pi_{\mathsf{C},i} \end{pmatrix} =: \mathsf{out}_i \leftarrow\!\$\ \mathsf{ClientObtain}_2(\mathsf{state}^{(i)}, \mathsf{resp}_i)$
8 : $\quad$ **if** $\mathsf{out}_i = \bot$ **then reject**
9 : $\hat{i} \leftarrow\!\$\ Q^*$
10 : Pick a random permutation φ of Q^*
11 : $i^* \leftarrow \mathcal{A}_3\left(\mathsf{st}_2, \mathsf{out}_{\hat{i}}, \{\mathsf{out}_{\varphi(i)}\}_{i \in Q^* \setminus \{\hat{i}\}}\right)$
12 : **if** $i^* = \hat{i}$ **then accept**

Oracle $O_\mathsf{Query}(i)$

1 : **if** Q_Query is uninitialized **then**
2 : $\quad Q_\mathsf{Query} := \varnothing$
3 : $Q_\mathsf{Query} := Q_\mathsf{Query} \cup \{i\}$
4 : $\begin{pmatrix} \mathsf{query}^{(i)} := \mathbf{v}^{(i)}, \\ \mathsf{state}^{(i)} := (\mathbf{u}^{(i)}, t^{(i)}) \end{pmatrix} \leftarrow\!\$\ \mathsf{ClientQuery}(\mathsf{pk})$
5 : **return** $\mathsf{query}^{(i)}$

Oracle $O_\mathsf{Obtain}(j^{(i)}, \mathsf{resp}^{(i)} := (\mathbf{z}^{(i)}, \pi_\mathsf{I}^{(i)}))$

1 : **if** Q_Obtain is uninitialized **then**
2 : $\quad Q_\mathsf{Obtain} := \varnothing$
3 : **if** $j^{(i)} \in Q_\mathsf{Obtain} \vee j^{(i)} \notin Q_\mathsf{Query}$ **then**
4 : $\quad$ **return** $\bot$
5 : $Q_\mathsf{Obtain} := Q_\mathsf{Obtain} \cup \{j^{(i)}\}$
6 : $\begin{pmatrix} \mathbf{w}_0^{(i)}, \mathbf{w}_1^{(i)}, \\ \psi_\mathsf{C}^{(i)}, \pi_\mathsf{C}^{(i)} \end{pmatrix} =: \mathsf{out}^{(i)} \leftarrow\!\$\ \mathsf{ClientObtain}_2(\mathsf{state}^{(i)}, \mathsf{resp}^{(i)})$
7 : **return** $\mathsf{out}^{(i)}$

$\mathsf{ClientObtain}_2(\mathsf{state} := t, \mathsf{resp} := (\mathbf{z}, \pi_\mathsf{I}))$

9 : $\psi_\mathsf{C} \leftarrow\!\$\ \mathsf{PKE.Enc}(\mathsf{pk}_\mathsf{PKE}, \mathbf{0})$
10 : $\pi_\mathsf{C} \leftarrow\!\$\ \mathsf{NIZK}_\mathsf{C}.\mathsf{Prove}\left(\mathsf{crs}_\mathsf{C}, \begin{matrix} \mathbb{x} := (\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{w}_0, \mathbf{w}_1, t, \psi_\mathsf{C}) \\ \mathbb{w} := (\mathbf{u}, \mathbf{x}, \mathbf{z}, \boldsymbol{\nu}_0, \boldsymbol{\nu}_1) \end{matrix}\right)$
11 : $\sigma := (\mathbf{w}_0, \mathbf{w}_1, \pi_\mathsf{C}, \psi_\mathsf{C})$

Figure 19: The κ-unlinkability Game $\mathbf{G}_2$ with differences from the previous game highlighted. For ClientObtain, only the lines that change are shown.

We further divide the game into a sequence of $|Q^*| = n$ intermediate games so that Game $\mathbf{G}_{1_k}$ is the same as Game $\mathbf{G}_{1_{k-1}}$, except that for $k \in [n]$, the encryptions $\psi_{\mathsf{C},(1)}, \psi_{\mathsf{C},(2)}, \ldots, \psi_{\mathsf{C},(k)}$ are computed as encryptions

of $\mathbf{0}$ and the remaining $\psi_{\mathsf{C},(k+1)} \ldots, \psi_{\mathsf{C},(n)}$ are computed as encryptions of the secrets in their respective $\mathsf{state}^{(i)}$. Note that $\mathbf{G}_{1_0}$ is just $\mathbf{G}_1$, and $\mathbf{G}_{1_n}$ is just $\mathbf{G}_2$.

Now, suppose there exists some PPT adversary $\mathcal{A}$ has non-negligible absolute advantage between games $\mathbf{G}_{1_{k-1}}$ and $\mathbf{G}_{1_k}$. We give a reduction algorithm $\mathcal{B}_{\mathsf{CPA}}$ that breaks the IND-CPA security of PKE with non-negligible advantage as follows:

- ▷ The IND-CPA challenger chooses a bit $\hat{b} \leftarrow\!\$\ \{0,1\}$ and samples $(\mathsf{pk}_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow\!\$\ \mathsf{PKE.Setup}(1^\lambda)$ and sends $\mathsf{pk}_{\mathsf{PKE}}$ to the reduction $\mathcal{B}_{\mathsf{CPA}}$.
- ▷ The rest of the experiment proceeds exactly as in $\mathbf{G}_{1_{k-1}}$ except, when $\mathcal{A}$ outputs $\{Q^*, \{\mathsf{resp}_i\}\}_{i \in Q^*}$, it computes the encryptions $\psi_{\mathsf{C},(1)}, \psi_{\mathsf{C},(2)}, \ldots, \psi_{\mathsf{C},(k-1)}, \psi_{\mathsf{C},(k+1)}, \ldots, \psi_{\mathsf{C},(n))}$ as it does in Game $\mathbf{G}_{1_{k-1}}$. However, instead of computing $\psi_{\mathsf{C},(k)}$ the same way, it sets $\mathbf{m}_0 := \mathbf{u}^{(k)} || \mathbf{x}_k || \mathbf{z}_k || \boldsymbol{\nu}_{0,k} || \boldsymbol{\nu}_{1,k}$ and $\mathbf{m}_1 := \mathbf{0}$, and sends $(\mathbf{m}_0, \mathbf{m}_1)$ to the challenger who responds with the ciphertext $\psi_{\mathsf{C},(k)}$ (to $\mathbf{m}_{\hat{b}}$).
- ▷ Finally, when $\mathcal{A}$ outputs i^*, $\mathcal{B}_{\mathsf{CPA}}$ outputs 0 if $i^* = \hat{i}$ indicating its guess that it received an encryption of $\mathbf{0}$ from the IND-CPA challenger; otherwise, it outputs 1.

If challenger encrypts $\mathbf{m}_0$ in $\psi_{\mathsf{C},(k)}$, $\mathcal{B}_{\mathsf{CPA}}$ perfectly simulates Game $\mathbf{G}_{1_{k-1}}$ to $\mathcal{A}$; otherwise, it simulates Game $\mathbf{G}_{1_k}$. Thus, if $\mathcal{A}$ can distinguish the two games with non-negligible advantage, then $\mathcal{B}_{\mathsf{CPA}}$ has non-negligible advantage against the IND-CPA challenger. Applying this argument to each $k \in [n]$, we get that games $\mathbf{G}_1$ and $\mathbf{G}_2$ are indistinguishable.

Game $\mathbf{G}_3$: This is the same as $\mathbf{G}_2$, except on every Query oracle call by the adversary, instead of running ClientQuery as usual, the challenger will instead sample uniform vector $\boldsymbol{v} \leftarrow\!\$\ \mathbb{Z}_q^n$ and set the corresponding $\mathsf{query}^{(i)} := \boldsymbol{v} + \mathsf{H}(t^{(i)})$. As in the proof of Lemma A.3 (Game $\mathbf{G}_2$), we can argue via the leftover hash lemma (Corollary 2.6) that $(\mathbf{C}, \mathbf{C} \cdot \mathbf{u})$ and $(\mathbf{C}, \boldsymbol{v})$ are statistically indistinguishable and, as a result, so are games $\mathbf{G}_3$ and $\mathbf{G}_4$.

Now, let $\mathsf{Adv}^j_{\mathcal{A}}$ be the advantage of a PPT adversary $\mathcal{A}$ in Game $\mathbf{G}_j$. Then, proceeding similarly as we did in the proof of Lemma A.3, we can show that $\mathsf{Adv}^3_{\mathcal{A}} \leq \frac{2}{n} + \mathsf{negl}(\lambda)$. Therefore, construction 4.1 is 2-unlinkable. □

Lemma B.4 (Privacy of metadata bit)**.** *Construction 4.1 has private metadata bit assuming* $\mathsf{NIZK}_{\mathsf{I}}$ *satisfies zero-knowledge,* $\mathsf{NIZK}_{\mathsf{C}}$ *is sound and the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ *and* $\mathsf{LWE}_{q,n,m,\mathfrak{s}_{\boldsymbol{\varepsilon}}}$ *problems are hard.*

Proof. We proceed through a sequence of games:

Game $\mathbf{G}_0$: This is the original ATPM metadata bit privacy experiment $\mathsf{PMB}_{\mathsf{ATPM},\mathcal{A},\hat{b}}$ for $\hat{b} \in \{0,1\}$ defined with respect to construction 4.1 (without public metadata; see Figure 14).

Game $\mathbf{G}_1$: This is the same as $\mathbf{G}_0$, except the challenger simulates each NIZK proof without a witness (see Figure 15). It is easily shown that if $\mathsf{NIZK}_{\mathsf{I}}$ satisfies zero-knowledge, then $\mathbf{G}_0$ and $\mathbf{G}_1$ are indistinguishable to the adversary.

Game $\mathbf{G}_2$: This is the same as $\mathbf{G}_1$, except for each of the ith Read or Valid queries, instead of running the ReadBit algorithm as before, the challenger runs the extractor for $\mathsf{NIZK}_{\mathsf{C}}$ to extract the witness from the corresponding proof $\pi^{(i)}_{\mathsf{C}}$ and verifies the relation $\mathcal{R}_{\mathsf{C}}$ locally (see Figure 16). Similarly, for each of the ith (one-more) forgery, it extracts the witness from the corresponding proof $\pi_{\mathsf{C},i}$ and verifies $\mathcal{R}_{\mathsf{C}}$ locally. The indistinguishability of games $\mathbf{G}_1$ and $\mathbf{G}_2$ follows from soundness of $\mathsf{NIZK}_{\mathsf{C}}$.

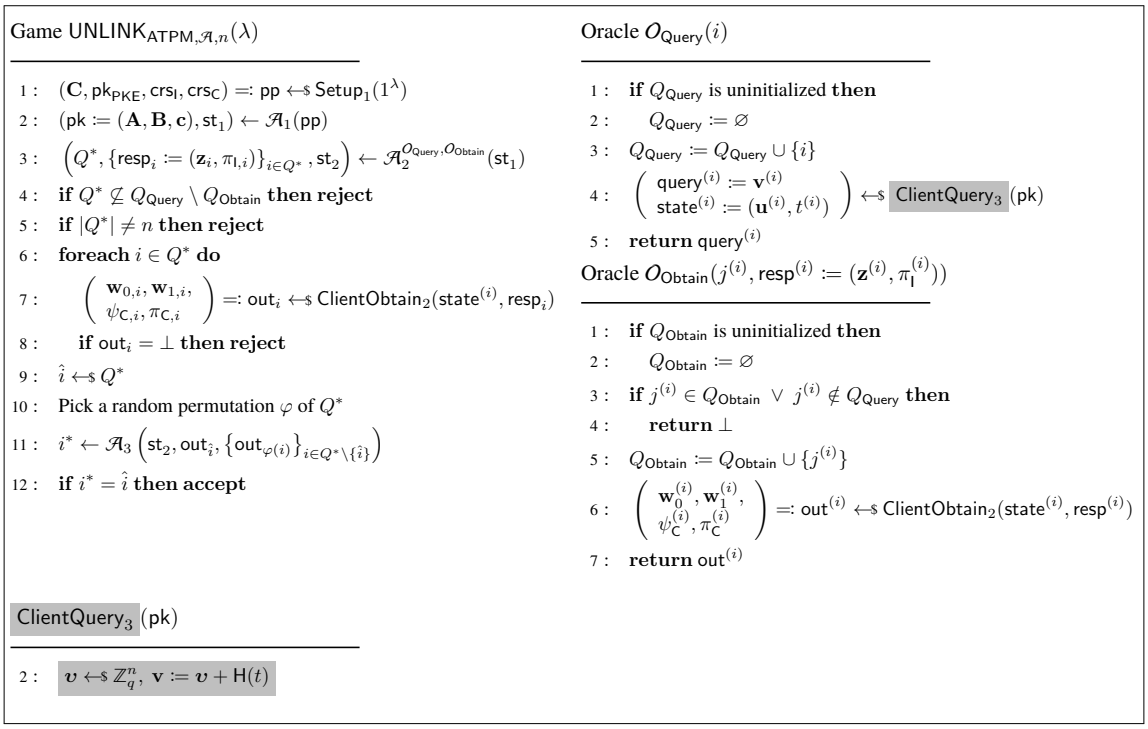

Figure 20: The κ-unlinkability Game $\mathbf{G}_3$ with differences from the previous game highlighted. For ClientQuery, only the lines that change are shown.

Game $\mathbf{G}_3$: This is the same as $\mathbf{G}_2$, except for each Issue query, the challenger stores the corresponding $(\mathbf{x}^{(i)}, b^{(i)})$ pair in Q_{BitMap} (see Figure 17). Clearly, this affects no change to the adversary's point-of-view.

Game $\mathbf{G}_4$: This is the same as $\mathbf{G}_3$, except for each call to ReadBit, instead of directly attempting to decrypt $\mathbf{w}_0$ and $\mathbf{w}_1$, the challenger first uses the extracted $\mathbf{x}$ value to look up the corresponding bit in Q_{BitMap} and returns it if one is found and aborts otherwise (see Figure 18). We can use the same argument as was given in the proof of Lemma A.4 (Game $\mathbf{G}_4$) to show that $\mathbf{G}_3$ and $\mathbf{G}_4$ are indistinguishable to an adversary assuming the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$ and that the $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},2\mathfrak{B}_{\mathbf{z}}}$ problem is hard.

Finally, by the same argument as Claim A.5, we can show that an adversary's advantage in $\mathbf{G}_4$ is negligible assuming hardness of $\mathsf{LWE}_{q,n,m,\mathfrak{s}_{\boldsymbol{\varepsilon}}}$. Thus, construction 4.1 satisfies privacy of metadata bit. □

C Proofs for Section 5

In this section, we prove our main theorem (Theorem 5.1) for ATPM security of construction 5.

Lemma C.1 (One-more unforgeability)**.** *Construction 5 is one-more unforgeable (in the random oracle model) assuming* $\mathsf{NIZK}_{\mathsf{I}}$ *satisfies zero-knowledge,* $\mathsf{NIZK}_{\mathsf{C}}$ *is straight-line extractable and the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},3\mathfrak{B}_{\mathbf{z}}}$ *problem is hard.*

Proof. The proof proceeds through a sequence of games:

Game $\mathbf{G}_0$: This is the original ATPM one-more unforgeability experiment $\mathsf{OM\text{-}Unf}_{\mathsf{ATPM},\mathcal{A}}$ defined with respect to construction 5.

Game $\mathsf{OM\text{-}Unf}_{\mathsf{ATPM},\mathcal{A}}(\lambda)$

$(\mathbf{C}, \mathsf{pk}_{\mathsf{PKE}}, \mathsf{crs}_{\mathsf{I}}, \mathsf{crs}_{\mathsf{C}}) =: \mathsf{pp} \leftarrow\!\$\ \mathsf{Setup}(1^\lambda)$

$\left(\begin{array}{l}\mathsf{pk} := (\mathbf{A}, \mathbf{B}, \mathbf{c}),\\ \mathsf{sk} := (\mathbf{T}_{\mathbf{A}}, \mathbf{s})\end{array}\right) \leftarrow\!\$\ \mathsf{KeyGen}(1^\lambda)$

$\mu^*, b^*, \{(t_i, \sigma_i)\}_{i\in[N]} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Read}}}(\mathsf{pk})$

Define ℓ as the multiplicity of b^* in Q

if $\#\{t_i\}_{i\in[N]} \le \ell \vee b^* \notin \{0,1\}$ **then reject**

if $\bigvee_{i\in[N]} \mathsf{ReadBit}(\mathsf{sk}, t_i, \sigma_i, \mu^*) \neq b^*$ **then reject**

accept

Oracle $\mathcal{O}_{\mathsf{Issue}}(b^{(i)}, \mu^{(i)}, \mathsf{query}^{(i)} := \mathbf{v}^{(i)})$

if multiset Q is uninitialized **then**

  $Q := \varnothing$

$Q := Q \cup \{(b^{(i)}, \mu^{(i)})\}$

$(\mathbf{z}^{(i)}, \pi_{\mathsf{I}}^{(i)}) =: \mathsf{resp}^{(i)} \leftarrow\!\$\ \mathsf{IssueToken}(\mathsf{sk}, b^{(i)}, \mu^{(i)}, \mathsf{query}^{(i)})$

return $\mathsf{resp}^{(i)}$

Oracle $\mathcal{O}_{\mathsf{Read}}(t^{(i)}, \sigma^{(i)} := (\mathbf{w}_0^{(i)}, \mathbf{w}_1^{(i)}, \pi_{\mathsf{C}}^{(i)}), \mu^{(i)})$

return $\mathsf{ReadBit}(\mathsf{sk}, t^{(i)}, \sigma^{(i)}, \mu^{(i)})$

Figure 21: The one-more unforgeability Game $\mathbf{G}_0$.

Game $\mathbf{G}_1$: This is the same as $\mathbf{G}_0$, except for each Issue query, the challenger simulates the corresponding proof $\pi_{\mathsf{I}}^{(i)}$ without a witness (see Figure 8). It is easily shown that if $\mathsf{NIZK}_{\mathsf{I}}$ satisfies zero-knowledge, then $\mathbf{G}_1$ and $\mathbf{G}_2$ are indistinguishable to the adversary.

Game $\mathbf{G}_2$: This is the same as $\mathbf{G}_1$, except for each of the ith Read queries, instead of running the ReadBit algorithm as before, the challenger uses $\mathsf{sk}_{\mathsf{PKE}}$ to extract the witness from the corresponding proof $\pi_{\mathsf{C}}^{(i)}$ and verifies the relation $\mathcal{R}'_{\mathsf{C}}$ locally. Similarly, for each of the ith (one-more) forgery, it extracts the witness from the corresponding proof $\pi_{\mathsf{C},i}$ and verifies $\mathcal{R}'_{\mathsf{C}}$ locally (see Figure 9). The indistinguishability of games $\mathbf{G}_1$ and $\mathbf{G}_2$ follows from the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$.

Game $\mathbf{G}_3$: This is the same as $\mathbf{G}_2$, except instead of sampling $\mathbf{C}$ uniformly during Setup, the challenger first runs KeyGen to obtain $\mathbf{A}$, and then sets $\mathbf{C} := \mathbf{A}_\top \cdot \mathbf{R}$ where $\mathbf{R} \leftarrow\!\$\ \{0,1\}^{m'\times m'}$ (see Figure 10). As with Lemma A.1 (Game $\mathbf{G}_3$), we can argue via the leftover hash lemma (Corollary 2.6) that $(\mathbf{A}, \mathbf{A}\cdot\mathbf{R})$ and $(\mathbf{A}, \mathbf{C})$ are statistically indistinguishable.

Finally, with only a slight modification to argument as the one in Claim A.2, we can show that an adversary's advantage in $\mathbf{G}_3$ is negligible. As before, we must prove that any adversary with non-negligible advantage $\mathsf{Adv}^4_{\mathcal{A}}$, there exists a reduction that either breaks the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$, or the one-more ISIS assumption, i.e.,

Claim C.2. *Assuming the straight-line extractability of* $\mathsf{NIZK}_{\mathsf{C}}$ *and that the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},3\mathfrak{B}_{\mathbf{z}}}$ *problem is hard,* $\mathsf{Adv}^3_{\mathcal{A}}$ *is negligible for all* PPT *adversaries* $\mathcal{A}$.

Proof. We can separate the analysis into the following two cases:

- **Case 1.** There exists some $j \in [N]$ such that $\mathsf{ReadBit}(\mathsf{sk}, t_j, \sigma_j := (\mathbf{w}_{0,j}, \mathbf{w}_{1,j}, \pi_{\mathsf{C},j})) = b^* (\neq \perp)$ but there does *not* exist a valid witness $\mathbb{w}_j \leftarrow \mathsf{NIZK}_{\mathsf{C}}.\mathcal{E}(\tau, \mathbb{x}_j, \pi_{\mathsf{C},j})$ such that $(\mathbb{x}_j, \mathbb{w}_j) \in \mathcal{R}_{\mathsf{C}}$ where $\mathbb{x}_j := (\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{w}_{0,j}, \mathbf{w}_{1,j}, t_j, \mu^*)$ and $\mathbb{w}_j := (\mathbf{u}_j, \mathbf{x}_j, \mathbf{z}_j, \boldsymbol{\nu}_{0,j}, \boldsymbol{\nu}_{1,j})$.

- **Case 2.** For all $j \in [N]$, $\mathsf{ReadBit}(\mathsf{sk}, t_j, \sigma_j := (\mathbf{w}_{0,j}, \mathbf{w}_{1,j}, \pi_{\mathsf{C},j}), \mu^*) = b^* (\neq \perp)$ and there exists a valid witness $\mathbb{w}_j \leftarrow \mathsf{NIZK}_{\mathsf{C}}.\mathcal{E}(\tau, \mathbb{x}_j, \pi_{\mathsf{C},j})$ such that $(\mathbb{x}_j, \mathbb{w}_j) \in \mathcal{R}_{\mathsf{C}}$ with $\mathbb{x}_j$ and $\mathbb{w}_j$ as defined previously.

Let $\epsilon(\lambda)$ be the adversary's advantage in the first case. Then, we can construct a reduction $\mathcal{B}_{\mathsf{KSnd}}$ such that if $\epsilon(\lambda)$ is non-negligible, then $\mathcal{B}_{\mathsf{KSnd}}$ can break the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$ with non-negligible probability. In particular, the reduction does the following:

- ▷ The NIZK challenger C samples $\tau \leftarrow\!\$\ \{0,1\}^{\lambda}$, obtains the reference string $\mathsf{crs}_{\mathsf{C}} \leftarrow \mathsf{NIZK}_{\mathsf{C}}.\mathcal{E}(\{0,1\}^{\lambda}; \tau)$ and sends $\mathsf{crs}_{\mathsf{C}}$ to the reduction, $\mathcal{B}_{\mathsf{KSnd}}$. The reduction, in turn, runs Setup_2 as in Game $\mathbf{G}_3$ except that it does not compute $\mathsf{crs}_{\mathsf{C}}$ itself but sets it as the value received from C, and sends the resulting pp to the adversary $\mathcal{A}$.
- ▷ The rest of the experiment proceeds as in Game $\mathbf{G}_3$.
- ▷ Finally, when $\mathcal{A}$ outputs its (one-more) forgeries $\{t_i, \sigma_i \coloneqq \mathbf{w}_{0,i}, \mathbf{w}_{1,i}, \pi_{\mathsf{C},i}), \boxed{\mu^*}\}$, $\mathcal{B}_{\mathsf{KSnd}}$ samples $j \leftarrow\!\$\ [N]$, sets $\mathbb{x}_j \coloneqq (\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{w}_{0,j}, \mathbf{w}_{1,j}, t_j, \boxed{\mu^*})$ and sends $(\mathbb{x}_j, \pi_{\mathsf{C},j})$ to C.

By assumption, $(\mathbb{x}_j, \pi_{\mathsf{C},j})$ breaks the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$ with probability $(1/N) \cdot \epsilon(\lambda)$, which is non-negligible for $N = \mathsf{poly}(\lambda)$ if $\epsilon(\lambda)$ is non-negligible.

Similarly, let $\epsilon(\lambda)$ be the adversary's advantage in the second case. We show that we can construct a reduction $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ such that if $\epsilon(\lambda)$ is non-negligible, then $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ can break OM-ISIS assumption with non-negligible probability. In particular, the reduction does the following:

- ▷ It sets $\mathbf{A} \coloneqq \begin{bmatrix} \mathbf{A}_{\top} \\ \mathbf{A}_{\bot} \end{bmatrix}$ for $\mathbf{A}_{\top}$ given to it by the OM-ISIS challenger C, and $\mathbf{A}_{\bot} \leftarrow\!\$\ \mathbb{Z}_q^{(n+1)\times m'}$. It computes the rest of the pk and sk (except $\mathbf{T}_{\mathbf{A}_{\top}}$) as before.
- ▷ On receiving a *fresh* random oracle query for $\mathsf{H}_1(t^{(i)})$, the reduction $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ in turn queries C for a syndrome $\mathbf{t}^{(i)}$ (i.e., makes a *syndrome query*) and forwards it to $\mathcal{A}$. Let $\mathcal{R}$ be an ordered set containing the random oracle queries $t^{(i)}$ and $\mathcal{S}$ the ordered set containing the corresponding syndromes $\mathbf{t}^{(i)}$ returned by the OM-ISIS challenger.
- ▷ On receiving a *fresh* random oracle query for $\boxed{\mathsf{H}_2(\mu^{(i)})}$, the reduction $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ samples $\mathbf{h}^{(i)} \leftarrow\!\$\ \Gamma_{\mathbb{Z}^{m'}, \mathfrak{s}_{\mathbf{z}}}$ and returns $\mathsf{H}_2(\mu^{(i)}) \coloneqq \mathbf{A}_{\top} \cdot \mathbf{h}^{(i)}$ to the adversary. Let $\mathcal{R}_2$ be an ordered set containing the tuple of random oracle queries and their responses $(\mu^{(i)}, \mathbf{h}^{(i)})$.
- ▷ On receiving a Issue query $(b^{(i)}, \mathsf{query}^{(i)} \coloneqq \mathbf{v}^{(i)}, \boxed{\mu^{(i)}})$, it inserts $(b^{(i)}, \boxed{\mu^{(i)}})$ into Q, and looks for $\mathbf{h}^{(i)}$ corresponding to $\mu^{(i)}$ in $\mathcal{R}_2$. If none is found, it samples $\mathbf{h}^{(i)} \leftarrow\!\$\ \Gamma_{\mathbb{Z}^{m'}, \mathfrak{s}_{\mathbf{z}}}$ and inserts it into $\mathcal{R}_2$. Next, it queries C for a preimage of $\mathbf{v}^{(i)} \boxed{+\mathbf{A}_{\top} \cdot \mathbf{h}^{(i)}}$ (i.e., makes a *preimage query*), to which C responds with a preimage $\mathbf{z}^{(i)}$. $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ forwards $\mathbf{z}^{(i)}$ and the simulated proof $\pi_{\mathsf{I}}^{(i)}$ to $\mathcal{A}$.
- ▷ On receiving a Read query $(t^{(i)}, \sigma^{(i)})$, it answers it as explained in Game $\mathbf{G}_4$.
- ▷ When $\mathcal{A}$ outputs its forgeries $\left\{t_i, \sigma_i \coloneqq (\mathbf{w}_{0,i}, \mathbf{w}_{1,i}, \pi_{\mathsf{C},i}), \boxed{\mu^*}\right\}_{i \in [N]}$, the reduction $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ chooses a random subset $\{t_i, \sigma_i\}_{i \in [\ell+1]}$ of the $(\ell+1) \leq N$ *distinct* tags t_i. From this, it extracts $\mathbf{u}_i, \mathbf{z}_i$ and computes $\mathbf{z}'_i \coloneqq \mathbf{z}_i - \mathbf{R} \cdot \mathbf{x}_i - \mathbf{h}^*$ and $\mathbf{t}_i \coloneqq \mathbf{A}_{\top} \cdot \mathbf{z}_i - \mathbf{C} \cdot \mathbf{x}_i \boxed{-\mathbf{A}_{\top} \cdot \mathbf{h}^*}$ for each $i \in [\ell+1]$, where $\mathbf{h}^*$ is the vector corresponding to $(\mu^*, \cdot) \in \mathcal{R}_2$. It then outputs $\{(\mathbf{z}'_i, \mathbf{t}_i)\}_{i \in [\ell+1]}$ as its OM-ISIS preimages.

Clearly, since each $(\mathbb{x}_i, \mathbb{w}_i)$ satisfies the relation $\mathcal{R}_{\mathsf{C}}$, $\mathcal{B}_{\mathsf{OM\text{-}ISIS}}$ must succeed if $\mathbf{t}_i \neq \mathbf{t}_j$ and $\mathbf{t}_i \in \mathcal{S}$ for all $i \neq j \in [\ell+1]$ where $\mathcal{S}$ is the set of challenge syndromes. Now, consider the following events:

- **Case 2.1** For all $i \in [\ell+1]$, $t_i \in \mathcal{R}$, i.e., every tag in the forgery and the public metadata μ^* was queried by $\mathcal{A}$ to the random oracle.
- **Case 2.2** For all $i \in [\ell+1]$, $t_i \in \mathcal{R}$, i.e., every tag in the forgery and μ^* was *not* queried by $\mathcal{A}$ to the random oracle.
- **Case 2.3** There exists a $j \in [\ell+1]$, such that $t_j \notin \mathcal{R}$, i.e., some tag t_j in the forgery was *not* queried by $\mathcal{A}$ to the random oracle, and μ^* was queried by $\mathcal{A}$ to the random oracle.
- **Case 2.4** There exists a $j \in [\ell+1]$, such that $t_j \notin \mathcal{R}$, i.e., some tag t_j in the forgery and μ^* were *not* queried by $\mathcal{A}$ to the random oracle.

Let us jointly consider the latter three "bad" cases, which imply that $\mathcal{A}$ must never have queried the random oracle on t_j for some $j \in [\ell+1]$ (similarly for μ^*). However, the random oracle model dictates that $\mathsf{H}_1(t_j)$ (similarly for $\mathsf{H}_2(\mu^*)$) must be undefined, and therefore completely random from the point of view of the adversary. Then, since it must hold that $\mathsf{H}_1(t_i) = \mathbf{A}_\top \cdot \mathbf{z}_i - \mathbf{C} \cdot \mathbf{u}_i - \mathsf{H}_2(\mu^*)$, it follows that the probability of the "bad" events is at most negligible.

Now, in the first "good" case, since for each $i \in [\ell+1]$, $\mathbf{t}_i = \mathsf{H}_1(t_i)$, it necessarily holds that each $\mathbf{t}_i$ is distinct and is in the set $\mathcal{S}$. Moreover, we have that

$$\begin{aligned}\mathbf{t}_i &= \mathbf{A}_\top \cdot \mathbf{z}_i - \mathbf{C} \cdot \mathbf{u}_i - \mathbf{A}_\top \cdot \mathbf{h}^* \\ &= \mathbf{A}_\top \cdot \left(\mathbf{z}_i - \mathbf{R} \cdot \mathbf{u}_i - \mathbf{h}^*\right) \quad .\end{aligned}$$

Since $\|\mathbf{z}_i, \mathbf{h}^*\| \leq 2\mathfrak{B}_{\mathbf{z}}$ and $\|\mathbf{u}_i\| \leq \mathfrak{B}_{\mathbf{z}}/m$, and $\mathbf{R}$ is an $m' \times m'$ binary matrix,

$$\|\mathbf{z}_i - \mathbf{R} \cdot \mathbf{u}_i\| \leq 3\mathfrak{B}_{\mathbf{z}} \quad .$$

Thus, each $(\mathbf{z}_i - \mathbf{R} \cdot \mathbf{u}_i - \mathbf{h}^*)$ is a valid $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},3\mathfrak{B}_{\mathbf{z}}}$ solution with probability nearly $\epsilon(\lambda)$. □

Thus, construction 5 is one-more unforgeable. □

Lemma C.3 (κ-unlinkability)**.** *Construction 5 is κ-unlinkable for $\kappa = 2$, assuming $\mathsf{NIZK}_{\mathsf{I}}$ is sound and $\mathsf{NIZK}_{\mathsf{C}}$ satisfies zero-knowledge.*

Proof. We proceed through a sequence of games:

Game $\mathbf{G}_0$: This is the original ATPM unlinkability experiment $\mathsf{UNLINK}_{\mathsf{ATPM},\mathcal{A},n}$ defined with respect to construction 5.

Game $\mathbf{G}_1$: This is the same as $\mathbf{G}_0$, except the challenger simulates each NIZK proof π_{C} without a witness (see Figure 12). It is easily shown that if $\mathsf{NIZK}_{\mathsf{C}}$ satisfies zero-knowledge, then $\mathbf{G}_0$ and $\mathbf{G}_1$ are indistinguishable to the adversary.

Game $\mathbf{G}_2$: This is the same as $\mathbf{G}_1$, except on every Query oracle call by the adversary, instead of running ClientQuery as usual, the challenger will instead sample uniform vector $\boldsymbol{v} \leftarrow\!\$\ \mathbb{Z}_q^n$ and set the corresponding $\mathsf{query}^{(i)} := \boldsymbol{v} + \mathsf{H}_1(t^{(i)})$. As in the proof of Lemma A.3 (Game $\mathbf{G}_2$), we can argue via the leftover hash

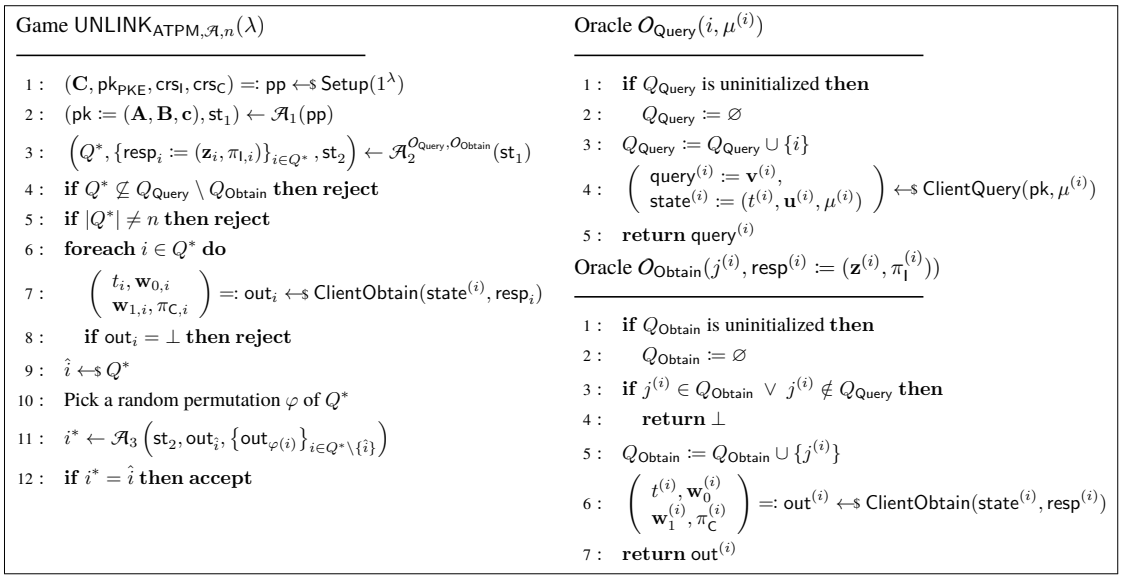

Figure 22: The κ-unlinkability Game $\mathbf{G}_0$.

lemma (Corollary 2.6) that $(\mathbf{C}, \mathbf{C} \cdot \mathbf{u})$ and $(\mathbf{C}, \boldsymbol{v})$ are statistically indistinguishable and, as a result, so are games $\mathbf{G}_1$ and $\mathbf{G}_2$.

Now, let $\mathsf{Adv}^j_{\mathcal{A}}$ be the advantage of a PPT adversary $\mathcal{A}$ in Game $\mathbf{G}_j$. Then, proceeding similarly as we did in the proof of Lemma A.3, we can show that $\mathsf{Adv}^2_{\mathcal{A}} \leq \frac{2}{n} + \mathsf{negl}(\lambda)$. Therefore, construction 5 is 2-unlinkable. □

Lemma C.4 (Privacy of metadata bit)**.** *Construction 5 has private metadata bit assuming* $\mathsf{NIZK}_{\mathsf{I}}$ *satisfies zero-knowledge,* $\mathsf{NIZK}_{\mathsf{C}}$ *is straight-line extractable, and the* $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},3\mathfrak{B}_{\mathbf{z}}}$ *and* $\mathsf{LWE}_{q,n,m,\mathfrak{s}_{\boldsymbol{\epsilon}}}$ *problems are hard.*

Proof. We proceed through a sequence of games:

Game $\mathbf{G}_0$: This is the original ATPM metadata bit privacy experiment $\mathsf{PMB}_{\mathsf{ATPM},\mathcal{A},\hat{b}}$ for $\hat{b} \in \{0, 1\}$ defined with respect to construction 5.

Game $\mathbf{G}_1$: This is the same as $\mathbf{G}_0$, except the challenger simulates each NIZK proof without a witness (see Figure 15). It is easily shown that if $\mathsf{NIZK}_{\mathsf{I}}$ satisfies zero-knowledge, then $\mathbf{G}_0$ and $\mathbf{G}_1$ are indistinguishable to the adversary.

Game $\mathbf{G}_2$: This is the same as $\mathbf{G}_1$, except for each of the ith Read or Valid queries, instead of running the ReadBit algorithm as before, the challenger runs the extractor for $\mathsf{NIZK}_{\mathsf{C}}$ to extract the witness from the corresponding proof $\pi_{\mathsf{C}}^{(i)}$ and verifies the relation $\mathcal{R}_{\mathsf{C}}$ locally (see Figure 16). Similarly, for each of the ith (one-more) forgery, it extracts the witness from the corresponding proof $\pi_{\mathsf{C},i}$ and verifies $\mathcal{R}_{\mathsf{C}}$ locally. The indistinguishability of games $\mathbf{G}_1$ and $\mathbf{G}_2$ follows from the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$.

Game $\mathbf{G}_3$: This is the same as $\mathbf{G}_2$, except for each Issue query, the challenger stores the corresponding $(\mathbf{x}^{(i)}, b^{(i)})$ pair in Q_{BitMap} (see Figure 17). Clearly, this affects no change to the adversary's point-of-view.

Game $\mathbf{G}_4$: This is the same as $\mathbf{G}_3$, except for each call to ReadBit, instead of directly attempting to decrypt $\mathbf{w}_0$ and $\mathbf{w}_1$, the challenger first uses the extracted $\mathbf{x}$ value to look up the corresponding bit in Q_{BitMap} and

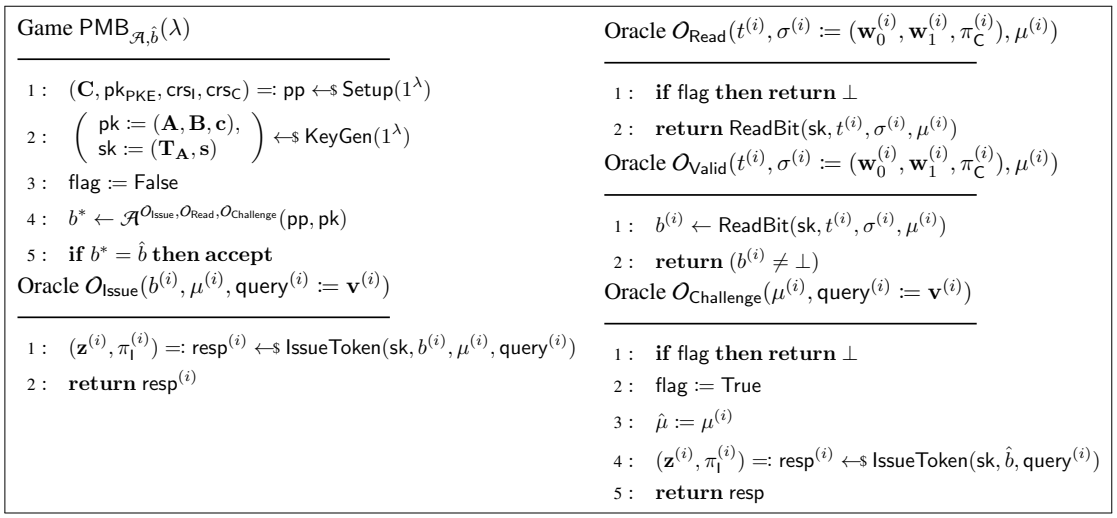

Figure 23: The privacy of metadata bit Game $\mathbf{G}_0$.

returns it if one is found and aborts otherwise (see Figure 18). We can use the same argument as was given in the proof of Lemma A.4 (Game $\mathbf{G}_4$) to show that $\mathbf{G}_3$ and $\mathbf{G}_3$ are indistinguishable to an adversary assuming the straight-line extractability of $\mathsf{NIZK}_{\mathsf{C}}$ and that the $\mathsf{OM\text{-}ISIS}_{q,n,m',\mathfrak{s}_{\mathbf{z}},3\mathfrak{B}_{\mathbf{z}}}$ problem is hard.

Finally, by the same argument as Claim A.5, we can show that an adversary's advantage in $\mathbf{G}_4$ is negligible assuming hardness of $\mathsf{LWE}_{q,n,m,\mathfrak{s}_{\boldsymbol{\epsilon}}}$. Thus, construction 5 satisfies privacy of metadata bit. □

D Attacks on the Security of our Lattice-based ATPM Constructions

The constructions in Sections 3 and 4 (and the public-metadata extension) admit the following attacks that break one-more unforgeability and metadata-bit privacy.

Let $\mathbf{D} = \begin{bmatrix} \mathbf{B} \\ \mathbf{c}^\top \end{bmatrix}$ and $\boldsymbol{\delta} = \begin{bmatrix} \mathbf{0}^n \\ \lfloor q/2 \rfloor \end{bmatrix}$. Then, the client relation $\mathcal{R}_{\mathsf{C}}$ checks $\mathbf{w}_0 = \mathbf{x} + \mathbf{D}\boldsymbol{\nu}_0$ and $\mathbf{w}_1 = \mathbf{x} + \mathbf{D}\boldsymbol{\nu}_1 + \boldsymbol{\delta}$, along with norm bounds. The main issue is that it does not (i) check that the issuer honestly generated $\mathbf{x}$; or (ii) ensure that the rerandomization vectors $\boldsymbol{\nu}_0$ and $\boldsymbol{\nu}_1$ leave the encrypted metadata bit unchanged.

1. Bit-flipping attack 1. Suppose the client finds $\boldsymbol{\nu}^* \in \mathbb{Z}^m$ satisfying

$$\mathbf{B}\boldsymbol{\nu}^* = \mathbf{0}^n, \qquad \mathbf{c}^\top \boldsymbol{\nu}^* = \lfloor q/2 \rfloor \pmod q, \quad \text{and} \quad \|\boldsymbol{\nu}\| \leq \mathfrak{B}_{\boldsymbol{\nu}} .$$

After receiving a presignature for bit b, the client sets $\boldsymbol{\nu}_0 = \boldsymbol{\nu}_1 := \boldsymbol{\nu}^*$. It then proves $\mathcal{R}_{\mathsf{C}}$ using the original $\mathbf{x}, \mathbf{z}$. The proof verifies, but one can check that $\mathbf{w}_0$ now decrypts to $1 - b$, while $\mathbf{w}_1$ decrypts to b. Thus ReadBit returns the opposite of the embedded bit b. As a further consequence, this attack can be used to break one-more unforgeability.

Of course, one still needs to find such a $\boldsymbol{\nu}^* \in \mathbb{Z}^m$. For the suggested 80-bit parameters ($n = 256$, $m = 512$, $q \approx 2^{50}$, $\mathfrak{B}_{\boldsymbol{\nu}} \approx 2^{44.5}$), LLL found such a vector in about 12 hours. The base construction also admits the attack with $m = 12928$ and $\mathfrak{B}_{\boldsymbol{\nu}} \approx 2^{46.83}$.

2. Attack on metadata-bit privacy. Observe that the adversary can still query O_{Valid} after the challenge in the PMB experiment. Now, using the vector $\boldsymbol{\nu}^*$ from above, it sets $\boldsymbol{\nu}_0 := \mathbf{0}^m$ and $\boldsymbol{\nu}_1 := \boldsymbol{\nu}^*$. Since $\mathbf{D}\boldsymbol{\nu}^* = \boldsymbol{\delta}$

$(\mathrm{mod}\ q)$, we have $\mathbf{w}_0 = \mathbf{x}$ and $\mathbf{w}_1 = \mathbf{x} + 2\boldsymbol{\delta}$. If the challenge bit is 0, ReadBit returns 0. On the other hand, if it is 1, both $\mathbf{w}_0$ and $\mathbf{w}_1$ decrypt to 1, and hence ReadBit returns $\perp$. Thus, one query to O_{Valid} reveals the challenge bit, breaking PMB security.

3. Bit-flipping attack 2. For the ring instantiation of Section 4, let us write $x := x_1||x_2 \in \mathcal{R}_{q,256}$, where $x_1, x_2 \in \mathcal{R}_{q,128}$, then the decryption equation is $-sx_1 + x_2$. Now, given a presignature (z_1, z_2) for bit 0, the client forms $x^* := (x_1 + kX^j)||x_2$ and $(z_1^*, z_2^*) := (z_1, z_2 + kX^{j+256})$.

Now, since $a_1 \cdot z_1 + z_2 = v||x$ and since (by definition of coefficient concatenation), $x = x_1 + X^{128}x_2$ and $v||x = v + X^{256}x$, we have

$$a_1 \cdot z_1^* + z_2^* = a_1 \cdot z_1 + z_2 + kX^{j+256} = v||x + kX^{j+256} = v||x^*.$$

Hence the presignature equation still holds for (z_1^*, z_2^*). Next, suppose $\|(z_1^*, z_2^*)\| < \mathfrak{B}_{\mathbf{z}}$ and $ks_{128-j} \bmod q \in (q/4 + \mathfrak{E}, 3q/4 - \mathfrak{E})$, where $1 \le j \le 127$, s_{128-j} is the coefficient of X^{128-j} in s, and $\mathfrak{E}$ bounds the decryption noise. To see that x^* decrypts to 1, let $e := -sx_1 + x_2$ be the original decryption polynomial, where $\|e\|_\infty \le \mathfrak{E}$. The modified decryption polynomial is

$$-s(x_1 + kX^j) + x_2 = e - ksX^j.$$

Since $X^{128} = -1$ in $\mathcal{R}_{q,128}$, the constant coefficient of sX^j is $-s_{128-j}$. Thus, the constant coefficient of the modified decryption polynomial is $e_0 + ks_{128-j} \bmod q$, where $|e_0| \le \mathfrak{E}$. By the condition on ks_{128-j}, this coefficient lies in $(q/4, 3q/4)$ modulo q, and hence its zero-centered representative has absolute value greater than $q/4$. Therefore, x^* decrypts to 1.

This means that the client can prove $\mathcal{R}_{\mathsf{C}}$ using this altered witness and obtain a valid token for the opposite bit. The decryption condition holds with estimated probability $1/2$ when k is sampled uniformly from $[0, q/2]$.

4. Attack on one-more unforgeability. Say, in the one-more unforgeability experiment, the client makes two queries to the oracle O_{Issue} for the bit 1. One using $\mathbf{v}_1 := \mathbf{C}\mathbf{u}_1 + \mathsf{H}(t)$, and another using $\mathbf{v}_2 := \mathbf{C}\mathbf{u}_2$. Let $\mathbf{z}_1, \mathbf{z}_2$ be the corresponding presignatures, with encryption randomness $\mathbf{r}_1, \mathbf{r}_2$. Now, the client sets $\mathbf{z}^* := \mathbf{z}_1 + \mathbf{z}_2 \pmod q$ and $\mathbf{u}^* := \mathbf{u}_1 + \mathbf{u}_2$. Then by linearity, we have

$$\mathbf{A}_\top \cdot \mathbf{z}^* - \mathbf{C} \cdot \mathbf{u}^* = \mathsf{H}(t) \quad \text{and} \quad \mathbf{A}_\perp \cdot \mathbf{z}^* \quad = \mathbf{D} \cdot (\mathbf{r}_1 + \mathbf{r}_2) + 2\boldsymbol{\delta}\ .$$

The second equation shows that the resulting ciphertext decrypts to bit 0. Thus, if the norm bounds hold, the client can prove $\mathcal{R}_{\mathsf{C}}$ using $\mathbf{z}^*, \mathbf{u}^*$ and obtain a valid token for bit 0, even though both presignatures were issued for bit 1. Of course, one still needs to ensure that $\|\mathbf{z}^*\| \le \mathfrak{B}_{\mathbf{z}}$. For the GPV construction of Section 3, each preimage has expected length about $0.4\mathfrak{B}_{\mathbf{z}}$, and hence their sum can satisfy the norm bound. For the Falcon instantiation of Section 4, the sum has expected length about $1.29\mathfrak{B}_{\mathbf{z}}$. In this case, the success of the attack depends on whether the LNP approximate norm proof rejects the larger vector.

Naturally, these issues point towards a gap in the security proofs. The one-more unforgeability proof in Claim A.2 reduces one-more forgery to OM-ISIS. However, the reduction counts issuance queries for the target bit b^*, while $\mathcal{R}_{\mathsf{C}}$ does not ensure that the extracted $\mathbf{x}$ is an honestly generated encryption of b^*. Indeed, the bit-flipping attacks above show that a valid proof of $\mathcal{R}_{\mathsf{C}}$ does not guarantee that the ciphertext decrypts to the bit for which it was issued. Thus, the stated reduction does not establish one-more unforgeability.

The PMB proof has a similar problem. In the transition from Game $\mathbf{G}_3$ to Game $\mathbf{G}_4$, the challenger replaces decryption with a lookup of the extracted $\mathbf{x}$ in Q_{BitMap}. However, $\mathcal{R}_{\mathsf{C}}$ does not ensure that this $\mathbf{x}$ was honestly generated during issuance. Thus, the extracted $\mathbf{x}$ may not appear in Q_{BitMap}, and even if it does,

its decrypted bit may differ from the recorded bit. Hence, the transition from Game $\mathbf{G}_3$ to Game $\mathbf{G}_4$ is not justified. Consequently, the corresponding security claims of Theorems 3.1, B.1, and 5.1 do not follow from the given proofs.