Adaptive Oblivious Transfer with Access Control from Lattice Assumptions

Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, Huaxiong Wang

Adaptive Oblivious Transfer with Access Control from Lattice Assumptions

Benoît Libert[1,2], San Ling[3], Fabrice Mouhartem[2], Khoa Nguyen[3], and Huaxiong Wang[3]

[1] CNRS, Laboratoire LIP, France
[2] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France
[3] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore

Abstract Adaptive oblivious transfer (OT) is a protocol where a sender initially commits to a database $\{M_i\}_{i=1}^N$. Then, a receiver can query the sender up to k times with private indexes $\rho_1, \ldots, \rho_k$ so as to obtain $M_{\rho_1}, \ldots, M_{\rho_k}$ and nothing else. Moreover, for each $i \in [k]$, the receiver's choice ρ_i may depend on previously obtained messages $\{M_{\rho_j}\}_{j<i}$. Oblivious transfer with access control (OT-AC) is a flavor of adaptive OT where database records are protected by distinct access control policies that specify which credentials a receiver should obtain in order to access each M_i. So far, all known OT-AC protocols only support access policies made of conjunctions or rely on *ad hoc* assumptions in pairing-friendly groups (or both). In this paper, we provide an OT-AC protocol where access policies may consist of any branching program of polynomial length, which is sufficient to realize any access policy in $\mathsf{NC1}$. The security of our protocol is proved under the Learning-with-Errors (LWE) and Short-Integer-Solution (SIS) assumptions. As a result of independent interest, we provide protocols for proving the correct evaluation of a committed branching program on a committed input.

Keywords. Lattice assumptions, standard assumptions, zero-knowledge arguments, adaptive oblivious transfer.

1 Introduction

Oblivious transfer (OT) is a central cryptographic primitive coined by Rabin [57] and extended by Even *et al.* [25]. It involves a sender S with a database of messages $M_1, \ldots, M_N$ and a receiver R with an index $\rho \in \{1, \ldots, N\}$. The protocol allows R to retrieve the ρ-th entry M_ρ from S without letting S infer anything on R's choice ρ. Moreover, R only obtains M_ρ learns nothing about $\{M_i\}_{i \neq \rho}$.

In its adaptive flavor [52], OT allows the receiver to interact k times with S to retrieve $M_{\rho_1}, \ldots, M_{\rho_k}$ in such a way that, for each $i \in \{2, \ldots, k\}$, the i-th index ρ_i may depend on the messages $M_{\rho_1}, \ldots, M_{\rho_{i-1}}$ previously obtained by R.

OT is known to be a complete building block for cryptography (see, e.g., [30]) in that, if it can be realized, then any secure multiparty computation can be. In its adaptive variant, OT is motivated by applications in privacy-preserving access to sensitive databases (e.g., medical records or financial data) stored in encrypted form on remote servers, oblivious searches or location-based services.

As far as efficiency goes, adaptive OT protocols should be designed in such a way that, after an inevitable initialization phase with linear communication complexity in N and the security parameter λ, the complexity of each transfer is at most poly-logarithmic in N. At the same time, this asymptotic efficiency should not come at the expense of sacrificing ideal security properties. The most efficient adaptive OT protocols that satisfy the latter criterion stem from the work of Camenisch, Neven and shelat [16] and its follow-ups [33,34,35].

In its basic form, (adaptive) OT does not restrict in any way the population of users who can obtain specific records. In many sensitive databases (e.g., DNA databases or patients' medical history), however, not all users should be able to download all records: it is vital access to certain entries be conditioned on the receiver holding suitable credentials delivered by authorities. At the same time, privacy protection mandates that authorized users be able to query database records while leaking as little as possible about their interests or activities. In medical datasets, for example, the specific entries retrieved by a given doctor could reveal which disease his patients are suffering from. In financial or patent datasets, the access pattern of a company could betray its investment strategy or the invention it is developing. In order to combine user-privacy and fine-grained database security, it is thus desirable to enrich adaptive OT protocols with refined access control mechanisms in many of their natural use cases.

This motivated Camenisch, Dubovitskaya and Neven [14] to introduce a variant named *oblivious transfer with access control* (OT-AC), where each database record is protected by a different access control policy $P : \{0,1\}^* \rightarrow \{0,1\}$. Based on their attributes, users can obtain credentials generated by pre-determined authorities, which entitle them to anonymously retrieve database records of which the access policy accepts their certified attributes: in other words, the user can only download the records for which he has a valid credential Cred_x for an attribute string $x \in \{0,1\}^*$ such that $P(x) = 1$. During the transfer phase, the user demonstrates possession of a pair (Cred_x, x) and simultaneously convinces the sender that he is querying some record M_ρ associated with a policy P such that $P(x) = 1$. The only information that the database holder eventually learns is that some user retrieved some record which he was authorized to obtain.

Camenisch *et al.* formalized the OT-AC primitive and provided a construction in groups with a bilinear map [14]. While efficient, their solution "only" supports access policies consisting of conjunctions: each policy P is specified by a list of attributes that a given user should obtain a credential for in order to complete the transfer. Several subsequent works [62,15,13] considered more expressive access policies while even hiding the access policies in some cases [15,13]. Unfortunately, all of them rely on non-standard assumptions (known as "q-type assumptions") in groups with a bilinear maps. For the sake of not putting all one's eggs in the same basket, a primitive as powerful as OT-AC ought to have alternative realizations based on firmer foundations.

In this paper, we propose a solution based on lattice assumptions where access policies consist of any branching program of width 5, which is known [6] to suffice for the realization of any access policy in $\mathsf{NC1}$. As a result of independent interest, we provide protocols for proving the correct evaluation of a committed branching program. More precisely, we give zero-knowledge arguments for demonstrating possession of a secret input $\mathbf{x} \in \{0,1\}^{\kappa}$ and a secret (and possibly certified) branching program BP such that $\mathsf{BP}(\mathbf{x}) = 1$.

Related Work. Oblivious transfer with adaptive queries dates back to the work of Naor and Pinkas [52], which requires $O(\log N)$ interaction rounds per transfer. Naor and Pinkas [54] also gave generic constructions of (adaptive) k-out-of-N OT from private information retrieval (PIR) [19]. The constructions of [52,54], however, are only secure in the half-simulation model, where simulation-based security is only considered for one of the two parties (receiver security being formalized in terms of a game-based definition). Moreover, the constructions of Adaptive OT from PIR [54] require a complexity $O(N^{1/2})$ at each transfer where Adaptive OT allows for $O(\log N)$ cost. Before 2007, many OT protocols (e.g., [53,3,61]) were analyzed in terms of half-simulation.

While several efficient fully simulatable protocols appeared the last 15 years (e.g., [21,47,56] and references therein), full simulatability remained elusive in the adaptive k-out-of-N setting [52] until the work [16] of Camenisch, Neven and shelat, who introduced the “assisted decryption” paradigm. The latter consists in having the sender obliviously decrypt a re-randomized version of one of the original ciphertexts contained in the database. This technique served as a blueprint for many subsequent protocols [33,34,35,40], including those with access control [14,15,13,1] and those presented in this paper. In the adaptive k-out-of-N setting (which we denote as $\mathcal{OT}_{k\times 1}^{N}$), the difficulty is to achieve full simulatability without having to transmit a $O(N)$ bits at each transfer. To our knowledge, except the oblivious-PRF-based approach of Jarecki and Liu [40], all known fully simulatable $\mathcal{OT}_{k\times 1}^{N}$ protocols rely on bilinear maps[4]. A recent work of Döttling *et al.* [23] uses non-black-box techniques to realize LWE-based 2-round oblivious PRF (OPRF) protocols [27]. However, while fully simulatable OPRFs imply [40] fully simulatable adaptive OT, the OPRF construction of [23] does not satisfy the standard notion of full simulation-based security against malicious adversaries (which is impossible to achieve in two rounds). It also relies on the full power of homomorphic encryption, which we do not require.

A number of works introduced various forms of access control in OT. Priced OT [3] assigns variable prices to all database records. In conditional OT [22], access to a record is made contingent on the user’s secret satisfying some predicate. Restricted OT [36] explicitly protects each record with an independent access policy. Still, none of these OT flavors aims at protecting the anonymity of users. The model of Coull, Green and Hohenberger [20] does consider user anonymity via stateful credentials. For the applications of OT-AC, it would nevertheless require re-issuing user credentials at each transfer.

[4] Several pairing-free candidates were suggested in [42,43] but, as pointed out in [35], they do not achieve full simulatability in the sense of [16]. In particular, the sender can detect if the receiver fetches the same record in two distinct transfers.

While efficient, the initial OT-AC protocol of Camenisch *et al.* [14] relies on non-standard assumptions in groups with a bilinear map and only realizes access policies made of conjunctions. Abe *et al.* [1] gave a different protocol which they proved secure under more standard assumptions in the universal composability framework [17]. Their policies, however, remain limited to conjunctions. It was mentioned in [14,1] that disjunctions and DNF formulas can be handled by duplicating database entries. Unfortunately, this approach rapidly becomes prohibitively expensive in the case of (t, n)-threshold policies with $t \approx n/2$. Moreover, securing the protocol against malicious senders requires them to prove that all duplicates encrypt the same message. More expressive policies were considered by Zhang *et al.* [62] who gave a construction based on attribute-based encryption [59] that extends to access policies expressed by any Boolean formulas (and thus $\mathsf{NC1}$ circuits). Camenisch, Dubovitskaya, Neven and Zaverucha [15] generalized the OT-AC functionality so as to hide the access policies. In [13], Camenisch *et al.* gave a more efficient construction with hidden policies based on the attribute-based encryption scheme of [55]. At the expense of a proof in the generic group model, [13] improves upon the expressiveness of [15] in that its policies extend into CNF formulas. While the solutions of [15,13] both hide the access policies to users (and the successful termination of transfers to the database), their policies can only live in a proper subset of $\mathsf{NC1}$. As of now, threshold policies can only be efficiently handled by the ABE-based construction of Zhang *et al.* [62], which requires *ad hoc* assumptions in groups with a bilinear map.

Our Results and Techniques. We describe the first OT-AC protocol based on lattice assumptions. Our construction supports access policies consisting of any branching program of width 5 and polynomial length – which suffices to realize any $\mathsf{NC1}$ circuit – and prove it secure under the SIS and LWE assumptions. We thus achieve the same level of expressiveness as [62] with the benefit of relying on well-studied assumptions. In its initial version, our protocol requires the database holder to communicate $\Theta(N)$ bits to each receiver so as to prove that the database is properly encrypted. In the random oracle model, we can eliminate this burden via the Fiat-Shamir heuristic and make the initialization cost linear in the database size N, regardless of the number of users.

As a first step, we build an ordinary $\mathcal{OT}_{k\times 1}^{N}$ protocol (i.e., without access control) via the "assisted decryption" approach of [16]. In short, the sender encrypts all database entries using a semantically secure cryptosystem. At each transfer, the receiver gets the sender to obliviously decrypt one of the initial ciphertexts without learning which one. Security against malicious adversaries is achieved by adding zero-knowledge (ZK) or witness indistinguishable (WI) arguments that the protocol is being followed. The desired ZK or WI arguments are obtained using the techniques of [44] and we prove that this basic protocol satisfies the full simulatability definitions of [16] under the SIS and LWE assumptions. To our knowledge, this protocol is also the first $\mathcal{OT}_{k\times 1}^{N}$ realization to achieve the standard simulation-based security requirements under lattice assumptions.

So far, all known "beyond-conjunctions" OT-AC protocols [62,13] rely on ciphertext-policy attribute-based encryption (CP-ABE) and proceed by attaching each database record to a CP-ABE ciphertext. Our construction departs from the latter approach for two reasons. First, the only known LWE-based CP-ABE schemes are obtained by applying a universal circuit to a key-policy system, making them very inefficient. Second, the ABE-based approach requires a fully secure ABE (i.e., selective security and semi-adaptive security are insufficient) since the access policies are dictated by the environment *after* the generation of the issuer's public key, which contains the public parameters of the underlying ABE in [62,13]. Even with the best known LWE-based ABE candidates [31], a direct adaptation of the techniques in [62,13] would incur to rely on a complexity leveraging argument [8] and a universal circuit. Instead, we take a different approach and directly prove in a zero-knowledge manner the correct evaluation of a committed branching program for a hidden input.

At a high level, our OT-AC protocol works as follows. For each $i \in [N]$, the database entry $M_i \in \{0,1\}^t$ is associated with branching program BP_i. In the initialization step, the database holder generates a Regev ciphertext $(\mathbf{a}_i, \mathbf{b}_i)$ of M_i, and issues a certificate for the pair $\big((\mathbf{a}_i, \mathbf{b}_i), \mathsf{BP}_i\big)$, using the signature scheme from [44]. At each transfer, the user U who wishes to get a record $\rho \in [N]$ must obtain a credential $\mathsf{Cred}_{\mathbf{x}}$ for an attribute string $\mathbf{x} \in \{0,1\}^\kappa$ such that $\mathsf{BP}_\rho(\mathbf{x}) = 1$. Then, U modifies $(\mathbf{a}_\rho, \mathbf{b}_\rho)$ into an encryption of $M_\rho \oplus \mu \in \{0,1\}^t$, for some random string $\mu \in \{0,1\}^t$, and re-randomizes the resulting ciphertext into a fresh encryption $(\mathbf{c}_0, \mathbf{c}_1)$ of $M_\rho \oplus \mu$. At this point, U proves that $(\mathbf{c}_0, \mathbf{c}_1)$ was obtained by transforming one of the original ciphertexts $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$ by arguing possession of a valid certificate for $\big((\mathbf{a}_\rho, \mathbf{b}_\rho), \mathsf{BP}_\rho\big)$ and knowledge of all randomness used in the transformation that yields $(\mathbf{c}_0, \mathbf{c}_1)$. At the same time, U proves possession of $\mathsf{Cred}_{\mathbf{x}}$ for a string $\mathbf{x}$ which is accepted by the committed BP_ρ. To demonstrate these statements in zero-knowledge, we develop recent techniques [49,46,44] for lattice-based analogues [41,48] of Stern's protocol [60].

As a crucial component of our OT-AC protocol, we need to prove knowledge of an input $\mathbf{x} = (x_0, \ldots, x_{\kappa-1})^\top \in \{0,1\}^\kappa$ satisfying a hidden BP of length L, where L and κ are polynomials in the security parameter. For each $\theta \in [L]$, we need to prove that the computation of the θ-th state

$$\eta_\theta = \pi_{\theta,0}(\eta_{\theta-1}) \cdot (1 - x_{\mathrm{var}(\theta)}) + \pi_{\theta,1}(\eta_{\theta-1}) \cdot x_{\mathrm{var}(\theta)}, \tag{1}$$

is done correctly, for permutations $\pi_{\theta,0}, \pi_{\theta,1} : [0,4] \to [0,4]$ and for integer $\mathrm{var}(\theta) \in [0, \kappa-1]$ specified by BP. To date, equations of the form (1) have not been addressed in the context of zero-knowledge proofs for lattice-based cryptography. In this work, we are not only able to handle L such equations, but also manage to do so with a reasonable asymptotic cost. Below, we will explain the latter aspect.

In order to compute η_θ as in (1), we have to fetch the value $x_{\mathrm{var}(\theta)}$ in the input $(x_0, \ldots, x_{\kappa-1})^\top$ and provide evidence that the searching process is conducted honestly. If we perform a naive left-to-right search in the array $x_0, \ldots, x_{\kappa-1}$, the expected complexity is $\mathcal{O}(\kappa)$ for each step, and the total complexity amounts to $\mathcal{O}(L \cdot \kappa)$. If we instead perform a dichotomic search over $x_0, \ldots, x_{\kappa-1}$, we can decrease the complexity at each step down to $\mathcal{O}(\log \kappa)$. However, in this case, we need to prove in zero-knowledge a statement "I obtained $x_{\mathrm{var}(\theta)}$ by conducting a correct dichotomic search in my secret array."

We solve this problem as follows. For each $i \in [0, \kappa-1]$, we employ a SIS-based commitment scheme [41] to commit to x_i as com_i, and prove that the committed bits are consistent with the ones involved in the credential $\mathsf{Cred}_{\mathbf{x}}$ mentioned above. Then we build a SIS-based Merkle hash tree [46] of depth $\delta_\kappa = \lceil \log \kappa \rceil$ on top of the commitments $\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1}$. Now, for each $\theta \in [L]$, we consider the binary representation $d_{\theta,1}, \ldots, d_{\theta,\delta_\kappa}$ of $\mathrm{var}(\theta)$. We then prove knowledge of a bit y_θ such that these conditions hold: "If one starts at the root of the tree and follows the path determined by the bits $d_{\theta,1}, \ldots, d_{\theta,\delta_\kappa}$, then one will reach the leaf associated with the commitment opened to y_θ." The idea is that, if the Merkle tree and the commitment scheme are both secure, then it should be true that $y_\theta = x_{\mathrm{var}(\theta)}$. In other words, this enables us to provably perform a "binary search" for $x_{\mathrm{var}(\theta)} = y_\theta$. Furthermore, this process can be done in zero-knowledge, by adapting the recent techniques from [46]. As a result, we obtain a protocol with communication cost just $\mathcal{O}(L \cdot \log \kappa + \kappa)$.

We believe that our idea of bringing Merkle trees into the context of privacy-preserving protocols for branching programs to decrease the asymptotic complexity will be of independent interest. For instance, it may find applications in the private computation of branching programs [50,39].

2 Background and Definitions

Vectors are denoted in bold lower-case letters and bold upper-case letters will denote matrices. The Euclidean and infinity norm of any vector $\mathbf{b} \in \mathbb{R}^n$ will be denoted by $\|\mathbf{b}\|$ and $\|\mathbf{b}\|_\infty$, respectively. The Euclidean norm of matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ with columns $(\mathbf{b}_i)_{i \leq n}$ is $\|\mathbf{B}\| = \max_{i \leq n} \|\mathbf{b}_i\|$. When $\mathbf{B}$ has full column-rank, we let $\widetilde{\mathbf{B}}$ denote its Gram-Schmidt orthogonalization.

When S is a finite set, we denote by $U(S)$ the uniform distribution over S, and by $x \hookleftarrow U(S)$ the action of sampling x according to this distribution. Finally for any integers A, B, N, we let $[N]$ and $[A, B]$ denote the sets $\{1, \ldots, N\}$ and $\{A, A+1, \ldots, B\}$, respectively.

2.1 Lattices

A lattice L is the set of integer linear combinations of linearly independent basis vectors $(\mathbf{b}_i)_{i \leq n}$ living in $\mathbb{R}^m$. We work with q-ary lattices, for some prime q.

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

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

For a lattice L, let $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi \|\mathbf{x} - \mathbf{c}\|^2 / \sigma^2)$ for a vector $\mathbf{c} \in \mathbb{R}^m$ and a real $\sigma > 0$. The discrete Gaussian of support L, center $\mathbf{c}$ and parameter σ is $D_{L,\sigma,\mathbf{c}}(\mathbf{y}) = \rho_{\sigma,\mathbf{c}}(\mathbf{y}) / \rho_{\sigma,\mathbf{c}}(L)$ for any $\mathbf{y} \in L$, where $\rho_{\sigma,\mathbf{c}}(L) = \sum_{\mathbf{x} \in L} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. The distribution centered in $\mathbf{c} = \mathbf{0}$ is denoted by $D_{L,\sigma}(\mathbf{y})$.

It is well known that one can efficiently sample from a Gaussian distribution with lattice support given a sufficiently short basis of the lattice.

Lemma 1 ([11, Le. 2.3]). *There exists a PPT algorithm* $\mathsf{GPVSample}$ *that takes as inputs a basis* $\mathbf{B}$ *of a lattice* $L \subseteq \mathbb{Z}^n$ *and a rational* $\sigma \geq \|\widetilde{\mathbf{B}}\| \cdot \Omega(\sqrt{\log n})$*, and outputs vectors* $\mathbf{b} \in L$ *with distribution* $D_{L,\sigma}$*.*

We also rely on the trapdoor generation algorithm of Alwen and Peikert [4], which refines the technique of Gentry *et al.* [28].

Lemma 2 ([4, Th. 3.2]). *There is a PPT algorithm* $\mathsf{TrapGen}$ *that takes as inputs* 1^n*,* 1^m *and an integer* $q \geq 2$ *with* $m \geq \Omega(n \log q)$*, and outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *such that* $\mathbf{A}$ *is within statistical distance* $2^{-\Omega(n)}$ *to* $U(\mathbb{Z}_q^{n \times m})$*, and* $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{n \log q})$*.*

We use the basis delegation algorithm [18] that inputs a trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and produces a trapdoor for any $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ containing $\mathbf{A}$ as a submatrix. A technique from Agrawal *et al.* [2] is sometimes used in our proofs.

Lemma 3 ([18, Le. 3.2]). *There is a PPT algorithm* $\mathsf{ExtBasis}$ *that inputs* $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ *whose first* m *columns span* $\mathbb{Z}_q^n$*, and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *where* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *is a submatrix of* $\mathbf{B}$*, and outputs a basis* $\mathbf{T_B}$ *of* $\Lambda_q^{\perp}(\mathbf{B})$ *with* $\|\widetilde{\mathbf{T_B}}\| \leq \|\widetilde{\mathbf{T_A}}\|$*.*

Lemma 4 ([2, Th. 19]). *There is a PPT algorithm* $\mathsf{SampleRight}$ *that inputs* $\mathbf{A}, \mathbf{C} \in \mathbb{Z}_q^{n \times m}$*, a small-norm* $\mathbf{R} \in \mathbb{Z}^{m \times m}$*, a short basis* $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ *of* $\Lambda_q^{\perp}(\mathbf{C})$*, a vector* $\mathbf{u} \in \mathbb{Z}_q^n$ *and a rational* σ *such that* $\sigma \geq \|\widetilde{\mathbf{T_C}}\| \cdot \Omega(\sqrt{\log n})$*, and outputs* $\mathbf{b} \in \mathbb{Z}^{2m}$ *such that* $\left[\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R} + \mathbf{C} \right] \cdot \mathbf{b} = \mathbf{u} \bmod q$ *and with distribution statistically close to* $D_{L,\sigma}$ *where* $L = \{\mathbf{x} \in \mathbb{Z}^{2m} : \left[\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R} + \mathbf{C} \right] \cdot \mathbf{x} = \mathbf{u} \bmod q\}$*.*

2.2 Hardness Assumptions

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

If $q \geq \sqrt{n}\beta$ and $m, \beta \leq \mathsf{poly}(n)$, then standard worst-case lattice problems with approximation factors $\gamma = \widetilde{\mathcal{O}}(\beta\sqrt{n})$ reduce to $\mathsf{SIS}_{n,m,q,\beta}$ (see, e.g., [28, Se. 9]).

Definition 3. *Let* $n, m \geq 1$*,* $q \geq 2$*, and let* χ *be a probability distribution on* $\mathbb{Z}$*. For* $\mathbf{s} \in \mathbb{Z}_q^n$*, let* $A_{\mathbf{s},\chi}$ *be the distribution obtained by sampling* $\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n)$ *and* $e \hookleftarrow \chi$*, and outputting* $(\mathbf{a}, \mathbf{a}^T \cdot \mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$*. The Learning With Errors problem* $\mathsf{LWE}_{n,q,\chi}$ *asks to distinguish* m *samples chosen according to* $\mathcal{A}_{\mathbf{s},\chi}$ *(for* $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$*) and* m *samples chosen according to* $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$*.*

If q is a prime power, $B \geq \sqrt{n}\omega(\log n)$, $\gamma = \widetilde{\mathcal{O}}(nq/B)$, then there exists an efficient sampleable B-bounded distribution χ (i.e., χ outputs samples with norm at most B with overwhelming probability) such that $\mathsf{LWE}_{n,q,\chi}$ is as least as hard as SIVP_γ (see, e.g., [58,11]).

2.3 Adaptive Oblivious Transfer

In the syntax of [16], an adaptive k-out-of-N OT scheme $\mathcal{OT}_k^N$ is a tuple of stateful PPT algorithms $(\mathsf{S}_\mathsf{I}, \mathsf{R}_\mathsf{I}, \mathsf{S}_\mathsf{T}, \mathsf{R}_\mathsf{T})$. The sender $\mathsf{S} = (\mathsf{S}_\mathsf{I}, \mathsf{S}_\mathsf{T})$ consists of two interactive algorithms S_I and S_T and the receiver has a similar representation as algorithms R_I and R_T. In the *initialization phase*, the sender and the receiver run interactive algorithms S_I and R_I, respectively, where S_I takes as input messages $M_1, \ldots, M_N$ while R_I has no input. This phase ends with the two algorithms S_I and R_I outputting their state information S_0 and R_0 respectively.

During the i-th *transfer*, $1 \leq i \leq k$, both parties run an interactive protocol via the R_T and S_T algorithms. The sender starts runs $\mathsf{S}_\mathsf{T}(S_{i-1})$ to obtain its updated state information S_i while the receiver runs $\mathsf{R}_\mathsf{T}(R_{i-1}, \rho_i)$ on input of its previous state R_{i-1} and the index $\rho_i \in \{1, \ldots, N\}$ of the message it wishes to retrieve. At the end, R_T outputs an updated state R_i and a message M'_{ρ_i}.

Correctness mandates that, for all $M_1, \ldots, M_N$, for all $\rho_1, \ldots, \rho_k \in [N]$ and all coin tosses ϖ of the (honestly run) algorithms, we have $M'_{\rho_i} = M_{\rho_i}$ for all i.

We consider protocols that are secure (against static corruptions) in the sense of simulation-based definitions. The security properties against a cheating sender and a cheating receiver are formalized via the "real-world/ideal-world" paradigm. The security definitions of [16] are recalled in Appendix B.1.

2.4 Adaptive Oblivious Transfer with Access Control

Camenisch *et al.* [14] define oblivious transfer with access control (OT-AC) as a tuple of PPT algorithms/protocols $(\mathsf{ISetup}, \mathsf{Issue}, \mathsf{DBSetup}, \mathsf{Transfer})$ such that:

ISetup: takes as inputs public parameters pp specifying a set $\mathcal{P}$ of access policies and generates a key pair (PK_I, SK_I) for the issuer.

Issue: is an interactive protocol between the issuer I and a stateful user U under common input (pp, x), where x is an attribute string. The issuer I takes as inputs its key pair (PK_I, SK_I) and a user pseudonym P_U. The user takes as inputs its state information st_U. The user U outputs either an error symbol $\perp$ or a credential Cred_U, and an updated state st'_U.

$\mathsf{DBSetup}$: is an algorithm that takes as input the issuer's public key PK_I, a database $DB = (M_i, \mathsf{AP}_i)_{i=1}^N$ containing records M_i whose access is restricted by an access policy AP_i and outputs a database public key PK_{DB}, an encryption of the records $(ER_i)_{i=1}^N$ and a database secret key SK_{DB}.

$\mathsf{Transfer}$: is a protocol between the database DB and a user U with common inputs (PK_I, PK_{DB}). DB inputs SK_{DB} and U inputs $(\rho, st_\mathsf{U}, ER_\rho, \mathsf{AP}_\rho)$, where $\rho \in [N]$ is a record index to which U is requesting access. The interaction ends with U outputting $\perp$ or a string $M_{\rho'}$ and an updated state st'_U.

We assume private communication links, so that communications between a user and the issuer are authenticated, and those between a user and the database are anonymized: otherwise, anonymizing the $\mathsf{Transfer}$ protocol is impossible.

The security definitions formalize two properties called *user anonymity* and *database security*. The former captures that the database should be unable to tell which honest user is making a query and neither can tell which records are being accessed. This should remain true even if the database colludes with corrupted users and the issuer. As for database security, the intuition is that a cheating user cannot access a record for which it does not have the required credentials, even when colluding with other dishonest users. In case the issuer is colluding with these cheating users, they cannot obtain more records from the database than they retrieve. Precise security definitions [14] are recalled in Appendix B.2.

2.5 Vector Decompositions

We will employ the decomposition technique from [48,44], which allows transforming vectors with infinity norm larger than 1 into vectors with infinity norm 1.

For any $B \in \mathbb{Z}_+$, define the number $\delta_B := \lfloor \log_2 B \rfloor + 1 = \lceil \log_2(B+1) \rceil$ and the sequence $B_1, \ldots, B_{\delta_B}$, where $B_j = \lfloor \frac{B+2^{j-1}}{2^j} \rfloor$, $\forall j \in [1, \delta_B]$. This sequence satisfies $\sum_{j=1}^{\delta_B} B_j = B$ and any integer $v \in [0, B]$ can be decomposed to $\mathsf{idec}_B(v) = (v^{(1)}, \ldots, v^{(\delta_B)})^\top \in \{0,1\}^{\delta_B}$ such that $\sum_{j=1}^{\delta_B} B_j \cdot v_j = v$. We describe this decomposition procedure in a deterministic manner as follows:

1. Set $v' := v$; For $j = 1$ to δ_B do:
   If $v' \geq B_j$ then $v^{(j)} := 1$, else $v^{(j)} := 0$; $v' := v' - B_j \cdot v^{(j)}$.
2. Output $\mathsf{idec}_B(v) = (v^{(1)}, \ldots, v^{(\delta_B)})^\top$.

For any positive integers $\mathfrak{m}, B$, we define $\mathbf{H}_{\mathfrak{m},B} := \mathbf{I}_{\mathfrak{m}} \otimes [B_1 | \ldots | B_{\delta_B}] \in \mathbb{Z}^{\mathfrak{m} \times \mathfrak{m}\delta_B}$ and the following injective functions:

(i) $\mathsf{vdec}_{\mathfrak{m},B} : [0, B]^{\mathfrak{m}} \to \{0,1\}^{\mathfrak{m}\delta_B}$ that maps vector $\mathbf{v} = (v_1, \ldots, v_{\mathfrak{m}})$ to vector $\big(\mathsf{idec}_B(v_1)^\top \| \ldots \| \mathsf{idec}_B(v_{\mathfrak{m}})^\top\big)^\top$. Note that $\mathbf{H}_{\mathfrak{m},B} \cdot \mathsf{vdec}_{\mathfrak{m},B}(\mathbf{v}) = \mathbf{v}$.

(ii) $\mathsf{vdec}'_{\mathfrak{m},B} : [-B, B]^{\mathfrak{m}} \to \{-1,0,1\}^{\mathfrak{m}\delta_B}$ that maps vector $\mathbf{w} = (w_1, \ldots, w_{\mathfrak{m}})$ to vector $\big(\sigma(w_1) \cdot \mathsf{idec}_B(|w_1|)^\top \| \ldots \| \sigma(w_{\mathfrak{m}}) \cdot \mathsf{idec}_B(|w_{\mathfrak{m}}|)^\top\big)^\top$, where for each $i = 1, \ldots, \mathfrak{m}$: $\sigma(w_i) = 0$ if $w_i = 0$; $\sigma(w_i) = -1$ if $w_i < 0$; $\sigma(w_i) = 1$ if $w_i > 0$. Note that $\mathbf{H}_{\mathfrak{m},B} \cdot \mathsf{vdec}'_{\mathfrak{m},B}(\mathbf{w}) = \mathbf{w}$.

3 Building Blocks

We will use two distinct signature schemes because one of them only needs to be secure in the sense of a weaker security notion and can be more efficient. This weaker notion is sufficient to sign the database entries and allows a better efficiency in the scheme of Section 4. In particular, by making it stateful (which also suffices since all database entries are signed at once), we can reduce the public key size to $\log N$ matrices if N is the number of database entries. The second scheme must be stateful and secure in the standard EUF-CMA sense since the issuer uses it to certify users' attributes. The signature scheme of Section 3.1 is only used in the OT-AC protocol of Section 4 while the scheme of Section 3.2 is used in the adaptive OT protocol of Section 5 as well.

3.1 Signatures Supporting Efficient Zero-Knowledge Proofs

We use a signature scheme proposed by Libert *et al.* [44] who extended the Böhl *et al.* signature [7] in order to sign messages comprised of multiple blocks while keeping the scheme compatible with zero-knowledge proofs.

$\mathsf{Keygen}(1^\lambda, 1^{N_b})$: Given a security parameter $\lambda > 0$ and the number of blocks $N_b = \mathsf{poly}(\lambda)$, choose $n = \mathcal{O}(\lambda)$, a prime modulus $q = \widetilde{\mathcal{O}}(N \cdot n^4)$, a dimension $m = 2n\lceil \log q \rceil$; an integer $\ell = \mathsf{poly}(n)$ and Gaussian parameters $\sigma = \Omega(\sqrt{n \log q}\log n)$. Define the message space as $\mathcal{M} = (\{0,1\}^{m_I})^{N_b}$.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$. This basis allows computing short vectors in $\Lambda_q^{\perp}(\mathbf{A})$ with a Gaussian parameter σ. Next, choose $\ell + 1$ random $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell \hookleftarrow U(\mathbb{Z}_q^{n \times m})$.
2. Choose random matrices $\mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n \times m/2})$, $\mathbf{D}_0 \hookleftarrow U(\mathbb{Z}_q^{n \times m}), \mathbf{D}_j \hookleftarrow U(\mathbb{Z}_q^{n \times m_I})$ for $j = 1, \ldots, N_b$, as well as a random vector $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.

The private signing key consists of $SK := \mathbf{T_A}$ while the public key is comprised of $PK := \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \{\mathbf{D}_k\}_{k=0}^{N_b}, \mathbf{u}\big)$.

$\mathsf{Sign}\big(SK, \mathsf{Msg}\big)$: To sign an N_b-block $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_{N_b}) \in (\{0,1\}^{m_I})^{N_b}$,

1. Choose $\tau \hookleftarrow U(\{0,1\}^\ell)$. Using $SK := \mathbf{T_A}$, compute a short basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$ for $\Lambda_q^{\perp}(\mathbf{A}_\tau)$, where $\mathbf{A}_\tau = [\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j]\mathbf{A}_j] \in \mathbb{Z}_q^{n \times 2m}$.
2. Sample $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$. Compute the vector $\mathbf{c}_M \in \mathbb{Z}_q^n$ as a chameleon hash of $(\mathfrak{m}_1, \ldots, \mathfrak{m}_{N_b})$. Namely, compute $\mathbf{c}_M = \mathbf{D}_0 \cdot \mathbf{r} + \sum_{k=1}^{N_b} \mathbf{D}_k \cdot \mathfrak{m}_k \in \mathbb{Z}_q^n$, which is used to define $\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{c}_M) \in \mathbb{Z}_q^n$. Using the delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$, sample a vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $D_{\Lambda_q^{\mathbf{u}_M}(\mathbf{A}_\tau),\sigma}$.

Output the signature $sig = (\tau, \mathbf{v}, \mathbf{r}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^m$.

$\mathsf{Verify}\big(PK, \mathsf{Msg}, sig\big)$: Given $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_{N_b}) \in (\{0,1\}^{m_I})^{N_b}$ and $sig = (\tau, \mathbf{v}, \mathbf{r}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^m$, return 1 if $\|\mathbf{v}\| < \sigma\sqrt{2m}$, $\|\mathbf{r}\| < \sigma\sqrt{m}$ and

$$\mathbf{A}_\tau \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{D}_0 \cdot \mathbf{r} + \sum_{k=1}^{N_b} \mathbf{D}_k \cdot \mathfrak{m}_k) \bmod q. \tag{2}$$

3.2 A Simpler Variant with Bounded-Message Security and Security Against Non-Adaptive Chosen-Message Attacks

We consider a stateful variant of the scheme in Section 3.1 where a bound $Q \in \mathsf{poly}(n)$ on the number of signed messages is fixed at key generation time. In the context of $\mathcal{OT}_{k \times 1}^{N}$, this is sufficient and leads to efficiency improvements. In the modified scheme hereunder, the string $\tau \in \{0,1\}^\ell$ is an ℓ-bit counter maintained by the signer to keep track of the number of previously signed messages. This simplified variant resembles the SIS-based signature scheme of Böhl *et al.* [7].

In this version, the message space is $\{0,1\}^{n\lceil \log q \rceil}$ so that vectors of $\mathbb{Z}_q^n$ can be signed by first decomposing them using $\mathsf{vdec}_{n,q-1}(.)$.

Keygen$(1^\lambda, 1^Q)$**:** Given $\lambda > 0$ and the maximal number $Q \in \mathsf{poly}(\lambda)$ of signatures, choose $n = \mathcal{O}(\lambda)$, a prime $q = \widetilde{\mathcal{O}}(Q \cdot n^4)$, $m = 2n\lceil \log q \rceil$, an integer $\ell = \lceil \log Q \rceil$ and Gaussian parameters $\sigma = \Omega(\sqrt{n \log q} \log n)$. The message space is $\{0,1\}^{m_d}$, for some $m_d \in \mathsf{poly}(\lambda)$ with $m_d \geq m$.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$, which allows sampling short vectors in $\Lambda_q^{\perp}(\mathbf{A})$ with a Gaussian parameter σ. Next, choose $\ell + 1$ random $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell \hookleftarrow U(\mathbb{Z}_q^{n \times m})$.
2. Choose $\mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n \times m_d})$ as well as a random vector $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.

The counter τ is initialized to $\tau = 0$. The private key consists of $SK := \mathbf{T_A}$ and the public key is $PK := \big(\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \mathbf{D},\ \mathbf{u}\big)$.

Sign$\big(SK, \tau, \mathfrak{m}\big)$**:** To sign a message $\mathfrak{m} \in \{0,1\}^{m_d}$,

1. Increment the counter by setting $\tau := \tau + 1$ and interpret it as a string $\tau \in \{0,1\}^\ell$. Then, using $SK := \mathbf{T_A}$, compute a short delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$ for the matrix $\mathbf{A}_\tau = [\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j] \mathbf{A}_j] \in \mathbb{Z}_q^{n \times 2m}$.
2. Compute the vector $\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathfrak{m} \in \mathbb{Z}_q^n$. Then, using the delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$, sample a short vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $D_{\Lambda_q^{\mathbf{u}_M}(\mathbf{A}_\tau), \sigma}$.

Output the signature $sig = (\tau, \mathbf{v}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m}$.

Verify$\big(PK, \mathfrak{m}, sig\big)$**:** Given PK, $\mathfrak{m} \in \{0,1\}^{m_d}$ and a signature $sig = (\tau, \mathbf{v}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m}$, return 1 if $\|\mathbf{v}\| < \sigma\sqrt{2m}$ and $\mathbf{A}_\tau \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathfrak{m} \bmod q$.

For our purposes, the scheme only needs to satisfy a notion of bounded-message security under non-adaptive chosen-message attack. In this relaxed model, the adversary only obtains a bounded number of signatures for messages that are chosen non-adaptively (i.e., all at once and before seeing the public key) by the adversary. This security notion is sufficient for signing the N database entries. Note that the queries are non-adaptive but the adversary can adaptively choose its forgery message.

Theorem 1. *The scheme is bounded message secure under non-adaptive chosen-message attacks if the* SIS *assumption holds.* (The proof is in Appendix C.1.)

4 A Fully Simulatable Adaptive OT Protocol

Our basic $\mathcal{OT}_{k \times 1}^N$ protocol builds on the "assisted decryption" technique [16]. The databases holder encrypts all entries using a multi-bit variant [56] of Regev's cryptosystem [58] and proves the well-formedness of its public key and all ciphertexts. In addition, all ciphertexts are signed using a signature scheme. At each transfer, the receiver statistically re-randomizes a blinded version of the desired ciphertext, where the blinding is done via the additive homomorphism of Regev. Then, the receiver provides a witness indistinguishable (WI) argument that the modified ciphertext (which is submitted for oblivious decryption) is a transformation of one of the original ciphertexts by arguing knowledge of a signature on this hidden ciphertext. In response, the sender obliviously decrypts the modified ciphertext and argues in zero-knowledge that the response is correct.

Adapting the technique of [16] to the lattice setting requires the following building blocks: (i) A signature scheme allowing to sign ciphertexts while remaining compatible with ZK proofs; (ii) A ZK protocol allowing to prove knowledge of a signature on some hidden ciphertext which belongs to a public set and was transformed into a given ciphertext; (iii) A protocol for proving the correct decryption of a ciphertext; (iv) A method of statistically re-randomizing an LWE-encrypted ciphertext in a way that enables oblivious decryption. The first three ingredients can be obtained from [44]. Since component (i) only needs to be secure against random-message attacks as long as the adversary obtains at most N signatures, we use the simplified SIS-based signature scheme of Section 3.2. The statistical re-randomization of Regev ciphertexts is handled via the noise flooding technique [5], which consists in drowning the initial noise with a super-polynomially larger noise. While recent results [24,9] provide potentially more efficient alternatives, we chose the flooding technique for simplicity because it does not require the use of FHE (and also because the known multi-bit version [37] of the GSW FHE [29] incurs an *ad hoc* circular security assumption).

4.1 Description

Our scheme works with security parameter λ, modulus q, lattice dimensions $n = \mathcal{O}(\lambda)$ and $m = 2n\lceil \log q \rceil$. Let $B_\chi = \widetilde{\mathcal{O}}(\sqrt{n})$, and let χ be a B_χ-bounded distribution. We also define an integer B as a randomization parameter such that $B = n^{\omega(1)} \cdot (m+1)B_\chi$ and $B + (m+1)B_\chi \leq q/5$ (to ensure decryption correctness). Our basic $\mathcal{OT}_{k\times 1}^N$ protocol goes as follows.

$\mathsf{\textbf{Initialization}}\big(\mathsf{S_I}(1^\lambda, \mathrm{DB}), \mathsf{R_I}(1^\lambda)\big)$**:** In this protocol, the sender $\mathsf{S_I}$ has a database $\mathrm{DB} = (M_1, \ldots, M_N)$ of N messages, where $M_i \in \{0,1\}^t$ for each $i \in [N]$, for some $t \in \mathsf{poly}(\lambda)$. It interacts with the receiver $\mathsf{R_I}$ as follows.

1. Generate a key pair for the signature scheme of Section 3.2 in order to sign $Q = N$ messages of length $m_d = (n+t) \cdot \lceil \log q \rceil$ each. This key pair consists of $SK_{sig} = \mathbf{T_A} \in \mathbb{Z}^{m\times m}$ and $PK_{sig} := \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{u}\big)$, where $\ell = \log N$ and $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell \in U(\mathbb{Z}_q^{n\times m})$, $\mathbf{D} \in U(\mathbb{Z}_q^{n\times m_d})$. The counter is initialized to $\tau = 0$.
2. Choose $\mathbf{S} \hookleftarrow \chi^{n\times t}$ that will serve as a secret key for an LWE-based encryption scheme. Then, sample $\mathbf{F} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$, $\mathbf{E} \hookleftarrow \chi^{m\times t}$ and compute
$$\mathbf{P} = [\mathbf{p}_1 | \ldots | \mathbf{p}_t] = \mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E} \ \in \mathbb{Z}_q^{m\times t}, \qquad (3)$$
so that $(\mathbf{F}, \mathbf{P}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{m\times t}$ forms a public key for a t-bit variant of Regev's encryption scheme [58].
3. Sample vectors $\mathbf{a}_1, \ldots, \mathbf{a}_N \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{x}_1, \ldots, \mathbf{x}_N \hookleftarrow \chi^t$ to compute
$$(\mathbf{a}_i, \mathbf{b}_i) = \big(\mathbf{a}_i,\ \mathbf{S}^\top \cdot \mathbf{a}_i + \mathbf{x}_i + M_i \cdot \lfloor q/2 \rfloor\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t \qquad \forall i \in [N]. \qquad (4)$$
4. For each $i \in [N]$, generate a signature $(\tau_i, \mathbf{v}_i) \leftarrow \mathsf{Sign}(SK_{sig}, \tau, \mathfrak{m}_i)$ on the decomposition $\mathfrak{m}_i = \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_i^\top | \mathbf{b}_i^\top)^\top \in \{0,1\}^{m_d}$.

5. $\mathsf{S_I}$ sends $R_0 = \big(PK_{sig},\ (\mathbf{F},\mathbf{P}),\ \{(\mathbf{a}_i,\mathbf{b}_i),(\tau_i,\mathbf{v}_i)\}_{i=1}^N\big)$ to $\mathsf{R_I}$ and interactively proves knowledge of small-norm $\mathbf{S} \in \mathbb{Z}^{n\times t}$, $\mathbf{E} \in \mathbb{Z}^{m\times t}$, short vectors $\{\mathbf{x}_i\}_{i=1}^N$ and t-bit messages $\{M_i\}_{i=1}^N$, for which (3) and (4) hold. To this end, $\mathsf{S_I}$ plays the role of the prover in the ZK argument system described in Section 6.3. If the argument of knowledge does not verify or if there exists $i \in [N]$ such that $(\tau_i,\mathbf{v}_i)$ is an invalid signature on the message $\mathfrak{m}_i = \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_i^\top|\mathbf{b}_i^\top)^\top$ w.r.t. PK_{sig}, then $\mathsf{R_I}$ aborts.
6. Finally $\mathsf{S_I}$ defines $S_0 = \big((\mathbf{S},\mathbf{E}),(\mathbf{F},\mathbf{P}),PK_{sig}\big)$, which it keeps to itself.

Transfer$\big(\mathsf{S_T}(S_{i-1}),\mathsf{R_T}(R_{i-1},\rho_i)\big)$**:** At the i-th transfer, the receiver $\mathsf{R_T}$ has state R_{i-1} and an index $\rho_i \in [1,N]$. It interacts as follows with the sender $\mathsf{S_T}$ that has state S_{i-1} in order to obtain M_{ρ_i} from DB.

1. $\mathsf{R_T}$ samples vectors $\mathbf{e} \hookleftarrow U(\{-1,0,1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and a random $\nu \hookleftarrow U([-B,B]^t)$ to compute
$$(\mathbf{c}_0,\mathbf{c}_1) = \big(\mathbf{a}_{\rho_i} + \mathbf{F}\cdot\mathbf{e},\ \mathbf{b}_{\rho_i} + \mathbf{P}^\top\cdot\mathbf{e} + \mu\cdot\lfloor q/2\rfloor + \nu\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t, \qquad (5)$$
which is a re-randomization of $(\mathbf{a}_{\rho_i},\mathbf{b}_{\rho_i}+\mu\cdot\lfloor q/2\rfloor)$. The ciphertext $(\mathbf{c}_0,\mathbf{c}_1)$ is sent to $\mathsf{S_T}$. In addition, $\mathsf{R_T}$ provides an interactive WI argument that $(\mathbf{c}_0,\mathbf{c}_1)$ is indeed a transformation of $(\mathbf{a}_{\rho_i},\mathbf{b}_{\rho_i})$ for some $\rho_i \in [N]$, and $\mathsf{R_T}$ knows a signature on $\mathfrak{m} = \mathsf{vdec}_{n+1,q-1}(\mathbf{a}_{\rho_i}^\top|\mathbf{b}_{\rho_i}^\top)^\top \in \{0,1\}^{m_d}$. To this end, $\mathsf{R_T}$ runs the prover in the ZK argument system in Section 6.5.
2. If the argument of step 1 verifies, $\mathsf{S_T}$ uses $\mathbf{S}$ to decrypt $(\mathbf{c}_0,\mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ and obtain $M' = \lfloor(\mathbf{c}_1 - \mathbf{S}^\top\cdot\mathbf{c}_0)/(q/2)\rceil \in \{0,1\}^t$, which is sent back to $\mathsf{R_T}$. In addition, $\mathsf{S_T}$ provides a zero-knowledge argument of knowledge of vector $\mathbf{y} = \mathbf{c}_1 - \mathbf{S}^\top\cdot\mathbf{c}_0 - M'\cdot\lfloor q/2\rfloor \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \le q/5$ and small-norm matrices $\mathbf{E} \in \mathbb{Z}^{m\times t}$, $\mathbf{S} \in \mathbb{Z}^{n\times t}$ satisfying (modulo q)
$$\mathbf{P} = \mathbf{F}^\top\cdot\mathbf{S} + \mathbf{E}\ , \qquad \mathbf{c}_0^\top\cdot\mathbf{S} + \mathbf{y}^\top = \mathbf{c}_1^\top - {M'}^\top\cdot\lfloor q/2\rfloor. \qquad (6)$$
To this end, $\mathsf{S_T}$ runs the prover in the ZK argument system in Section 6.4.
3. If the ZK argument produced by $\mathsf{S_T}$ does not properly verify at step 2, $\mathsf{R_T}$ halts and outputs $\perp$. Otherwise, $\mathsf{R_T}$ recalls the random string $\mu \in \{0,1\}^t$ that was chosen at step 1 and computes $M_{\rho_i} = M' \oplus \mu$. The transfer ends with $\mathsf{S_T}$ and $\mathsf{R_T}$ outputting $S_i = S_{i-1}$ and $R_i = R_{i-1}$, respectively.

In the initialization phase, the sender has to repeat step 5 with each receiver to prove that $\{(\mathbf{a}_i,\mathbf{b}_i)\}_{i=1}^N$ are well-formed. Using the Fiat-Shamir heuristic [26], we can decrease this initialization cost from $O(N\cdot U)$ to $O(N)$ (regardless of the number of users U) by making the proof non-interactive. This modification also reduces each transfer to 5 communication rounds since, even in the transfer phase, the sender's ZK arguments can be non-interactive and the receiver's arguments only need to be WI, which is preserved when the basic ZK protocol (which has a ternary challenge space) is repeated $\omega(\log n)$ times in parallel. To keep the security proof simple, we derive the matrix $\mathbf{F} \in \mathbb{Z}_q^{n\times m}$ from a second random oracle. Knowing a short basis of $\Lambda_q^\perp(\mathbf{F})$, the simulator can extract the columns of $\mathbf{S}$ from the public key $\mathbf{P} \in \mathbb{Z}_q^{n\times m}$. Details are given in Appendix D.

4.2 Security

The security of the above $\mathcal{OT}_{k\times 1}^{N}$ protocol against static corruptions is stated by the following theorems.

Theorem 2. *The $\mathcal{OT}_{k\times 1}^{N}$ protocol provides receiver security under the* SIS *assumption.* (The proof is available in Appendix E.1.)

Theorem 3. *The $\mathcal{OT}_{k\times 1}^{N}$ protocol provides sender security under the* SIS *and* LWE *assumptions.* (The proof is available in Appendix E.2.)

5 OT with Access Control for Branching Programs

In this section, we extend our protocol of Section 4 into a protocol where database entries can be protected by access control policies consisting of branching programs. In a nutshell, the construction goes as follows.

When the database is set up, the sender signs (a binary representation of) each database entry $(\mathbf{a}_i, \mathbf{b}_i)$ together with a hash value $\mathbf{h}_{\mathsf{BP},i} \in \mathbb{Z}_q^n$ of the corresponding branching program. For each possessed attribute $\mathbf{x} \in \{0,1\}^\kappa$, the user U obtains a credential $\mathsf{Cred}_{\mathsf{U},\mathbf{x}}$ from the issuer.

If U has a credential $\mathsf{Cred}_{\mathsf{U},\mathbf{x}}$ for an attribute $\mathbf{x}$ satisfying the ρ-th branching program, U can re-randomize $(\mathbf{a}_\rho, \mathbf{b}_\rho)$ into $(\mathbf{c}_0, \mathbf{c}_1)$, which is given to the sender, while proving that: (i) He knows a signature $(\tau, \mathbf{v})$ on some message $(\mathbf{a}_\rho, \mathbf{b}_\rho, \mathbf{h}_{\mathsf{BP},\rho})$ such that $(\mathbf{c}_0, \mathbf{c}_1)$ is a re-randomization of $(\mathbf{a}_\rho, \mathbf{b}_\rho)$; (ii) The corresponding $\mathbf{h}_{\mathsf{BP},\rho}$ is the hash value of (the binary representation of) a branching program BP_ρ that accepts an attribute $\mathbf{x} \in \{0,1\}^\kappa$ for which he has a valid credential $\mathsf{Cred}_{\mathsf{U},\mathbf{x}}$ (i.e., $\mathsf{BP}_\rho(\mathbf{x}) = 1$).

While statement (i) can be proved as in Section 4, handling (ii) requires a method of proving the possession of a (committed) branching program BP and a (committed) input $\mathbf{x} \in \{0,1\}^\kappa$ such that $\mathsf{BP}(\mathbf{x}) = 1$ while demonstrating possession of a credential for $\mathbf{x}$.

Recall that a branching program BP of length L, input space $\{0,1\}^\kappa$ and width 5 is specified by L tuples of the form $(\mathrm{var}(\theta), \pi_{\theta,0}, \pi_{\theta,1})$ where

- var : $[L] \to [0, \kappa - 1]$ is a function that associates the θ-th tuple with the coordinate $x_{\mathrm{var}(\theta)} \in \{0,1\}$ of the input $\mathbf{x} = (x_0, \ldots, x_{\kappa-1})^\top$.
- $\pi_{\theta,0}, \pi_{\theta,1} : \{0,1,2,3,4\} \to \{0,1,2,3,4\}$ are permutations that determine the θ-th step of the evaluation.

On input $\mathbf{x} = (x_0, \ldots, x_{\kappa-1})^\top$, BP computes its output as follows. For each bit $b \in \{0,1\}$, let $\bar{b}$ denote the bit $1-b$. Let η_θ denote the state of computation at step θ. The initial state is $\eta_0 = 0$ and, for $\theta \in [1, L]$, the state η_θ is computed as

$$\eta_\theta = \pi_{\theta, x_{\mathrm{var}(\theta)}}(\eta_{\theta-1}) = \pi_{\theta,0}(\eta_{\theta-1}) \cdot \bar{x}_{\mathrm{var}(\theta)} + \pi_{\theta,1}(\eta_{\theta-1}) \cdot x_{\mathrm{var}(\theta)}.$$

Finally, the output of evaluation is $\mathsf{BP}(\mathbf{x}) = 1$ if $\eta_L = 0$, otherwise $\mathsf{BP}(\mathbf{x}) = 0$.

We now let $\delta_\kappa = \lceil \log_2 \kappa \rceil$ and note that each integer in $[0, \kappa - 1]$ can be determined by δ_κ bits. In particular, for each $\theta \in [L]$, let $d_{\theta,1}, \ldots, d_{\theta,\delta_\kappa}$ be the bits representing $\mathrm{var}(\theta)$. Then, we consider the following representation of BP:

$$\mathbf{z}_{\mathsf{BP}} = \big(d_{1,1}, \ldots, d_{1,\delta_\kappa}, \ldots, d_{L,1}, \ldots, d_{L,\delta_\kappa}, \pi_{1,0}(0), \ldots, \pi_{1,0}(4), \pi_{1,1}(0), \ldots,$$
$$\pi_{1,1}(4), \ldots, \pi_{L,0}(0), \ldots, \pi_{L,0}(4), \pi_{L,1}(0), \ldots, \pi_{L,1}(4)\big)^\top \in [0,4]^\zeta, \qquad (7)$$

where $\zeta = L(\delta_\kappa + 10)$.

5.1 The OT-AC Protocol

We assume public parameters pp consisting of a modulus q, integers n, m such that $m = 2n\lceil \log q \rceil$, a public matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$, the maximal length $L \in \mathsf{poly}(n)$ of branching programs and their desired input length $\kappa \in \mathsf{poly}(n)$.

ISetup(pp): Given public parameters $\mathsf{pp} = \{q, n, m, \bar{\mathbf{A}}, L, \kappa\}$, generate a key pair $(PK_I, SK_I) \leftarrow \mathsf{Keygen}(\mathsf{pp}, 1)$ for the signature scheme in Section 3.1 in order to sign single-block messages (i.e., $N_b = 1$) of length $m_I = n \cdot \lceil \log q \rceil + \kappa$. Letting $\ell_I = \mathcal{O}(n)$, this key pair contains $SK_I = \mathbf{T}_{\mathbf{A}_I} \in \mathbb{Z}^{m \times m}$ and

$$PK_I := \big(\mathbf{A}_I,\ \{\mathbf{A}_{I,j}\}_{j=0}^{\ell_I},\ \mathbf{D}_I,\ \{\mathbf{D}_{I,0}, \mathbf{D}_{I,1}\},\ \mathbf{u}_I\big).$$

Issue$\big(\mathsf{I}(\mathsf{pp}, SK_I, PK_I, P_{\mathsf{U}}, \mathbf{x}) \leftrightarrow \mathsf{U}(\mathsf{pp}, \mathbf{x}, st_{\mathsf{U}})\big)$**:** On common input $\mathbf{x} \in \{0,1\}^\kappa$, the issuer I and the user U interact in the following way:

1. If $st_{\mathsf{U}} = \emptyset$, U creates a pseudonym $P_{\mathsf{U}} = \bar{\mathbf{A}} \cdot \mathbf{e}_{\mathsf{U}} \in \mathbb{Z}_q^n$, for a randomly chosen $\mathbf{e}_{\mathsf{U}} \hookleftarrow U(\{0,1\}^m)$, which is sent to I. It sets $st_{\mathsf{U}} = (\mathbf{e}_{\mathsf{U}}, P_{\mathsf{U}}, 0, \emptyset, \emptyset)$. Otherwise, U parses its state st_{U} as $(\mathbf{e}_{\mathsf{U}}, P_{\mathsf{U}}, f_{DB}, C_{\mathsf{U}}, \mathsf{Cred}_{\mathsf{U}})$.
2. The issuer I defines the message $\mathfrak{m}_{\mathsf{U},\mathbf{x}} = (\mathsf{vdec}_{n,q-1}(P_{\mathsf{U}})^\top | \mathbf{x}^\top)^\top \in \{0,1\}^{m_I}$. Then, it runs the signing algorithm of Section 3.1 to obtain and return $\mathsf{cert}_{\mathsf{U},\mathbf{x}} = \big(\tau_{\mathsf{U}}, \mathbf{v}_{\mathsf{U}}, \mathbf{r}_{\mathsf{U}}\big) \leftarrow \mathsf{Sign}(SK_I, \mathfrak{m}_{\mathsf{U},\mathbf{x}}) \in \{0,1\}^{\ell_I} \times \mathbb{Z}^{2m} \times \mathbb{Z}^m$, which binds U's pseudonym P_{U} to the attribute string $\mathbf{x} \in \{0,1\}^\kappa$.
3. U checks that $\mathsf{cert}_{\mathsf{U},\mathbf{x}}$ satisfies (2) and that $\|\mathbf{v}_{\mathsf{U}}\| \leq \sigma\sqrt{2m}, \mathbf{r}_{\mathsf{U}} \leq \sigma\sqrt{m}$. If so, U sets $C_{\mathsf{U}} := C_{\mathsf{U}} \cup \{\mathbf{x}\}$, $\mathsf{Cred}_{\mathsf{U}} := \mathsf{Cred}_{\mathsf{U}} \cup \{\mathsf{cert}_{\mathsf{U},\mathbf{x}}\}$ and updates its state $st_{\mathsf{U}} = (\mathbf{e}_{\mathsf{U}}, P_{\mathsf{U}}, f_{DB}, C_{\mathsf{U}}, \mathsf{Cred}_{\mathsf{U}})$. If $\mathsf{cert}_{\mathsf{U},\mathbf{x}}$ does not properly verify, U aborts the interaction and leaves st_{U} unchanged.

DBSetup$\big(PK_I, \mathrm{DB} = \{(M_i, \mathsf{BP}_i)\}_{i=1}^N\big)$**:** The sender DB has $\mathrm{DB} = \{(M_i, \mathsf{BP}_i)\}_{i=1}^N$ which is a database of N pairs made of a message $M_i \in \{0,1\}^t$ and a policy realized by a length-L branching program $\mathsf{BP}_i = \{\mathrm{var}_i(\theta), \pi_{i,\theta,0}, \pi_{i,\theta,1}\}_{\theta=1}^L$.

1. Choose a random matrix $\mathbf{A}_{\mathrm{HBP}} \hookleftarrow U\big(\mathbb{Z}_q^{n \times \zeta}\big)$ which will be used to hash the description of branching programs.
2. Generate a key pair for the signature scheme of Section 3.2 in order to sign $Q = N$ messages of length $m_d = (2n+t) \cdot \lceil \log q \rceil$ each. This key pair consists of $SK_{sig} = \mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$ and $PK_{sig} := \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{u}\big)$, where $\ell = \lceil \log N \rceil$ and $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell \in U(\mathbb{Z}_q^{n \times m})$, $\mathbf{D} \in U(\mathbb{Z}_q^{n \times m_d})$ with $m = 2n\lceil \log q \rceil$, $m_d = (2n+t)\lceil \log q \rceil$. The counter is initialized to $\tau = 0$.

3. Sample $\mathbf{S} \hookleftarrow \chi^{n \times t}$ which will serve as a secret key for an LWE-based encryption scheme. Then, sample $\mathbf{F} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$, $\mathbf{E} \hookleftarrow \chi^{m \times t}$ to compute
$$\mathbf{P} = [\mathbf{p}_1 | \ldots | \mathbf{p}_t] = \mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E} \ \in \mathbb{Z}_q^{m \times t} \tag{8}$$
so that $(\mathbf{F}, \mathbf{P})$ forms a public key for a t-bit variant of Regev's system.
4. Sample vectors $\mathbf{a}_1, \ldots, \mathbf{a}_N \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{x}_1, \ldots, \mathbf{x}_N \hookleftarrow \chi^t$ to compute
$$(\mathbf{a}_i, \mathbf{b}_i) = \big(\mathbf{a}_i,\ \mathbf{a}_i^\top \cdot \mathbf{S} + \mathbf{x}_i + M_i \cdot \lfloor q/2 \rfloor\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t \qquad \forall i \in [N] \tag{9}$$
5. For each $i = 1$ to N, $(\mathbf{a}_i, \mathbf{b}_i)$ is bound to BP_i as follows.
   a. Let $\mathbf{z}_{\mathsf{BP},i} \in [0,4]^\zeta$ be the binary representation of the branching program. Compute its digest $\mathbf{h}_{\mathsf{BP},i} = \mathbf{A}_{\mathrm{HBP}} \cdot \mathbf{z}_{\mathsf{BP},i} \in \mathbb{Z}_q^n$.
   b. Using SK_{sig}, generate a signature $(\tau_i, \mathbf{v}_i) \leftarrow \mathsf{Sign}(SK_{sig}, \tau, \mathfrak{m}_i)$ on the message $\mathfrak{m}_i = \mathsf{vdec}_{2n+t,q-1}(\mathbf{a}_i | \mathbf{b}_i | \mathbf{h}_{\mathsf{BP},i}) \in \{0,1\}^{m_d}$ obtained by decomposing $(\mathbf{a}_i^\top | \mathbf{b}_i^\top | \mathbf{h}_{\mathsf{BP},i}^\top)^\top \in \mathbb{Z}_q^{2n+t}$.
6. The database's public key is defined as $PK_{\mathrm{DB}} = \big(PK_{sig},\ (\mathbf{F}, \mathbf{P}),\ \mathbf{A}_{\mathrm{HBP}}\big)$ while the encrypted database is $\{ER_i = \big(\mathbf{a}_i, \mathbf{b}_i, (\tau_i, \mathbf{v}_i)\big), \mathsf{BP}_i\}_{i=1}^N$. The sender DB outputs $\big(PK_{\mathrm{DB}}, \{ER_i, \mathsf{BP}_i\}_{i=1}^N\big)$ and keeps $SK_{\mathrm{DB}} = \big(SK_{sig}, \mathbf{S}\big)$.

Transfer$\big(\mathsf{DB}(SK_{\mathrm{DB}}, PK_{\mathrm{DB}}, PK_I), \mathsf{U}(\rho, st_{\mathsf{U}}, PK_I, PK_{\mathrm{DB}}, ER_\rho, \mathsf{BP}_\rho)\big)$**:** Given an index $\rho \in [N]$, a record $ER_\rho = \big(\mathbf{a}_\rho, \mathbf{b}_\rho, (\tau_\rho, \mathbf{v}_\rho)\big)$ and a policy BP_ρ, the user U parses st_{U} as $(\mathbf{e}_{\mathsf{U}}, P_{\mathsf{U}}, f_{DB}, C_{\mathsf{U}}, \mathsf{Cred}_{\mathsf{U}})$. If C_{U} does not contain any $\mathbf{x} \in \{0,1\}^\kappa$ s.t. $\mathsf{BP}_\rho(\mathbf{x}) = 1$ and $\mathsf{Cred}_{\mathsf{U}}$ contains the corresponding $\mathsf{cert}_{\mathsf{U},\mathbf{x}}$, U outputs $\perp$. Otherwise, he selects such a pair $(\mathbf{x}, \mathsf{cert}_{\mathsf{U},\mathbf{x}})$ and interacts with DB:

1. If $f_{DB} = 0$, U interacts with DB for the first time and requires DB to prove knowledge of small-norm $\mathbf{S} \in \mathbb{Z}^{n \times t}$, $\mathbf{E} \in \mathbb{Z}^{m \times t}$, $\{\mathbf{x}_i\}_{i=1}^N$ and t-bit messages $\{M_i\}_{i=1}^N$ satisfying (8)-(9). To do this, DB uses the ZK argument in Section 6.3. If there exists $i \in [N]$ such that $(\tau_i, \mathbf{v}_i)$ is an invalid signature on $\mathsf{vdec}_{2n+t,q-1}(\mathbf{a}_i^\top | \mathbf{b}_i^\top | \mathbf{h}_{\mathsf{BP},i}^\top)^\top$ or if the ZK argument does not verify, U aborts. Otherwise, U updates st_{U} and sets $f_{DB} = 1$.

2. U re-randomizes the pair $(\mathbf{a}_\rho, \mathbf{b}_\rho)$ contained in ER_ρ. It samples vectors $\mathbf{e} \hookleftarrow U(\{-1,0,1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and $\nu \hookleftarrow U([-B,B]^t)$ to compute
$$(\mathbf{c}_0, \mathbf{c}_1) = \big(\mathbf{a}_\rho + \mathbf{F} \cdot \mathbf{e},\ \mathbf{b}_\rho + \mathbf{P}^\top \cdot \mathbf{e} + \mu \cdot \lfloor q/2 \rfloor + \nu\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t, \tag{10}$$
which is sent to DB as a re-randomization of $(\mathbf{a}_\rho, \mathbf{b}_\rho + \mu \cdot \lfloor q/2 \rfloor)$. Then, U provides an interactive WI argument that $(\mathbf{c}_0, \mathbf{c}_1)$ is a re-randomization of some $(\mathbf{a}_\rho, \mathbf{b}_\rho)$ associated with a policy BP_ρ for which U has a credential $\mathsf{cert}_{\mathsf{U},x}$ for some $\mathbf{x} \in \{0,1\}^\kappa$ such that $\mathsf{BP}_\rho(\mathbf{x}) = 1$. In addition, U demonstrates possession of: (i) a preimage $\mathbf{z}_{\mathsf{BP},\rho} \in [0,4]^\zeta$ of $\mathbf{h}_{\mathsf{BP},\rho} = \mathbf{A}_{\mathrm{HBP}} \cdot \mathbf{z}_{\mathsf{BP},\rho} \in \mathbb{Z}_q^n$; (ii) a credential $\mathsf{Cred}_{\mathsf{U},\mathbf{x}}$ for the corresponding $\mathbf{x} \in \{0,1\}^\kappa$ and the private key $\mathbf{e}_{\mathsf{U}} \in \{0,1\}^m$ for the pseudonym P_{U} to which $\mathbf{x}$ is bound; (iii) the coins leading to the randomization of some

$(\mathbf{a}_\rho, \mathbf{b}_\rho)$. Then entire step is conducted by arguing knowledge of

$$\begin{cases} \mathbf{e}_{\mathsf{U}} \in \{0,1\}^m, \mathfrak{m}_{\mathsf{U},\mathbf{x}} \in \{0,1\}^{m_I},\ \mathbf{x} \in \{0,1\}^\kappa,\ \widehat{\mathfrak{m}}_{\mathsf{U},\mathbf{x}} \in \{0,1\}^{m/2} \\ \tau_{\mathsf{U}} \in \{0,1\}^{\ell_I},\ \mathbf{v}_{\mathsf{U}} = (\mathbf{v}_{\mathsf{U},1}^\top | \mathbf{v}_{\mathsf{U},2}^\top)^\top \in [-\beta,\beta]^{2m},\ \mathbf{r}_{\mathsf{U}} \in [-\beta,\beta]^m \\ \qquad\qquad\qquad\qquad // \text{ signature on } \mathfrak{m}_{\mathsf{U},\mathbf{x}} = (\mathsf{vdec}_{n,q-1}(P_{\mathsf{U}})^\top | \mathbf{x}^\top)^\top \\ \mathbf{z}_{\mathsf{BP},\rho} \in [0,4]^\zeta \qquad\qquad // \text{ representation of } \mathsf{BP}_\rho \\ \mathfrak{m} \in \{0,1\}^{m_d},\ \tau \in \{0,1\}^\ell,\ \mathbf{v} = (\mathbf{v}_1^\top | \mathbf{v}_2^\top)^\top \in \mathbb{Z}^{2m} \\ \qquad\qquad\qquad\qquad // \text{ signature on } \mathfrak{m} = \mathsf{vdec}_{2n+t,q-1}(\mathbf{a}_i^\top | \mathbf{b}_i^\top | \mathbf{h}_{\mathsf{BP},\rho}^\top)^\top \\ \mathbf{e} \in \{-1,0,1\}^t,\ \mu \in \{0,1\}^t,\ \nu \in [-B,B]^t, \\ \qquad\qquad\qquad\qquad // \text{ coins allowing the re-randomization of } (\mathbf{a}_\rho, \mathbf{b}_\rho) \end{cases}$$

satisfying the relations (modulo q)

$$\begin{cases} \mathbf{H}_{2n+t,q-1} \cdot \mathfrak{m} + \left[\begin{array}{c|c|c|c} \mathbf{F} & & & \\ \hline \mathbf{P}^\top & \mathbf{I}_t \cdot \lfloor q/2 \rfloor & \mathbf{I}_t & \\ \hline & & & -\mathbf{A}_{\mathrm{HBP}} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{e} \\ \hline \mu \\ \hline \nu \\ \hline \mathbf{z}_{\mathsf{BP},\rho} \end{array}\right] = \left[\begin{array}{c} \mathbf{c}_0 \\ \hline \mathbf{c}_1 \\ \hline \mathbf{0}^n \end{array}\right] \\ \qquad // \text{ (recall that } (\mathbf{a}_\rho^\top | \mathbf{b}_\rho^\top | \mathbf{h}_{\mathsf{BP},\rho}^\top)^\top = \mathbf{H}_{2n+t,q-1} \cdot \mathfrak{m}) \\ \mathbf{A} \cdot \mathbf{v}_1 + \mathbf{A}_0 \cdot \mathbf{v}_2 + \sum_{j=1}^{\ell} \mathbf{A}_j \cdot (\tau[j] \cdot \mathbf{v}_2) - \mathbf{D} \cdot \mathfrak{m} = \mathbf{u} \\ \mathbf{A}_I \cdot \mathbf{v}_{\mathsf{U},1} + \mathbf{A}_{I,0} \cdot \mathbf{v}_{\mathsf{U},2} + \sum_{j=1}^{\ell_I} \mathbf{A}_{I,j} \cdot (\tau_{\mathsf{U}}[j] \cdot \mathbf{v}_{\mathsf{U},2}) - \mathbf{D}_I \cdot \widehat{\mathfrak{m}}_{\mathsf{U},\mathbf{x}} = \mathbf{u}_I \\ \mathbf{D}_{I,0} \cdot \mathbf{r}_{\mathsf{U}} + \mathbf{D}_{I,1} \cdot \mathfrak{m}_{\mathsf{U},\mathbf{x}} - \mathbf{H}_{n,q-1} \cdot \widehat{\mathfrak{m}}_{\mathsf{U},\mathbf{x}} = \mathbf{0} \\ \left[\begin{array}{c|c} \mathbf{H}_{n,q-1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_\kappa \end{array}\right] \cdot \mathfrak{m}_{\mathsf{U},\mathbf{x}} + \begin{bmatrix} -\bar{\mathbf{A}} \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{e}_{\mathsf{U}} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{I}_\kappa \end{bmatrix} \cdot \mathbf{x} = \mathbf{0} \end{cases} \tag{11}$$

and such that $\mathbf{z}_{\mathsf{BP},\rho} \in [0,4]^\zeta$ encodes BP_ρ such that $\mathsf{BP}_\rho(\mathbf{x}) = 1$. This is done by running the argument system described in Section 6.6.

3. If the ZK argument of step 2 verifies, DB decrypts $(\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ to obtain $M' = \lfloor (\mathbf{c}_1 - \mathbf{S}^\top \cdot \mathbf{c}_0)/(q/2) \rceil \in \{0,1\}^t$, which is returned to U. Then, DB argues knowledge of $\mathbf{y} = \mathbf{c}_1 - \mathbf{S}^\top \cdot \mathbf{c}_0 - M' \cdot \lfloor q/2 \rfloor \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \leq q/5$ and small-norm $\mathbf{E} \in \mathbb{Z}^{m \times t}$, $\mathbf{S} \in \mathbb{Z}^{n \times t}$ satisfying (modulo q)

$$\mathbf{P} = \mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E}\ , \qquad \mathbf{c}_0^\top \cdot \mathbf{S} + \mathbf{y}^\top = \mathbf{c}_1^\top - {M'}^\top \cdot \lfloor q/2 \rfloor.$$

   To this end, DB uses the ZK argument system of Section 6.4.
4. If the ZK argument produced by DB does not verify, U outputs $\perp$. Otherwise, U recalls the string $\mu \in \{0,1\}^t$ and outputs $M_{\rho_i} = M' \oplus \mu$.

Like our construction of Section 4, the above protocol requires the DB to repeat a ZK proof of communication complexity $\Omega(N)$ with each user U during the initialization phase. By applying the Fiat-Shamir heuristic as in Appendix D, the cost of the initialization phase can be made independent of the number of users: the sender can publicize $\big(PK_{\mathrm{DB}}, \{ER_i, \mathsf{BP}_i\}_{i=1}^N\big)$ along with a with a universally verifiable non-interactive proof of well-formedness.

The security of the above protocol against static corruptions is proved in Appendix F, under the SIS and LWE assumptions.

6 Our Zero-Knowledge Arguments of Knowledge

This section provides all the zero-knowledge arguments of knowledge (ZKAoK) used as building blocks in our two adaptive OT schemes. Our argument systems operate in the framework of Stern's protocol [60], which was originally introduced in the context of code-based cryptography but has been developed [48,49,46,44,45] into a useful tool for lattice-based cryptography.

In Section 6.1, we first recall Stern's protocol in a generalized, abstract manner suggested in [44]. Then, using various transformations, we will demonstrate that all the required ZKAoKs can be obtained from this abstract protocol. Our basic strategy and techniques are summarized in Section 6.2, while the details of the protocols are given in the next subsections. In particular, our treatment of hidden branching programs in Section 6.6 is rather sophisticated as it requires to handle a number of secret objects nested together via branching programs, commitments, encryptions, signatures and Merkle trees. This protocol introduces new techniques and insights of which we provide the intuition hereafter.

6.1 Abstracting Stern's Protocol

Let K, D, q be positive integers with $D \geq K$ and $q \geq 2$, and let VALID be a subset of $\mathbb{Z}^D$. Suppose that $\mathcal{S}$ is a finite set such that every $\phi \in \mathcal{S}$ can be associated with a permutation Γ_ϕ of D elements satisfying the following conditions:

$$\begin{cases} \mathbf{w} \in \mathsf{VALID} \iff \Gamma_\phi(\mathbf{w}) \in \mathsf{VALID}, \\ \text{If } \mathbf{w} \in \mathsf{VALID} \text{ and } \phi \text{ is uniform in } \mathcal{S}, \text{ then } \Gamma_\phi(\mathbf{w}) \text{ is uniform in } \mathsf{VALID}. \end{cases} \tag{12}$$

We aim to construct a statistical ZKAoK for the following abstract relation:

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

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

The conditions in (12) play a crucial role in proving in ZK that $\mathbf{w} \in \mathsf{VALID}$. To this end, the prover samples a random $\phi \hookleftarrow U(\mathcal{S})$ and lets the verifier check that $\Gamma_\phi(\mathbf{w}) \in \mathsf{VALID}$ without learning any additional information about $\mathbf{w}$ due to the randomness of ϕ. Furthermore, to prove in a zero-knowledge manner that the linear equation is satisfied, the prover samples a masking vector $\mathbf{r}_w \hookleftarrow U(\mathbb{Z}_q^D)$, and convinces the verifier instead that $\mathbf{M} \cdot (\mathbf{w} + \mathbf{r}_w) = \mathbf{M} \cdot \mathbf{r}_w + \mathbf{v} \bmod q$.

The interaction between prover $\mathcal{P}$ and verifier $\mathcal{V}$ is described in Figure 1. The protocol uses a statistically hiding and computationally binding string commitment scheme COM (e.g., the SIS-based scheme from [41]).

Theorem 4. *The protocol in Figure 1 is a statistical* ZKAoK *with perfect completeness, soundness error* $2/3$*, and communication cost* $\mathcal{O}(D \log q)$*. Namely:*

1. **Commitment:** Prover samples $\mathbf{r}_w \leftarrow U(\mathbb{Z}_q^D)$, $\phi \leftarrow U(\mathcal{S})$ and randomness ρ_1, ρ_2, ρ_3 for COM. Then he sends $\mathrm{CMT} = \big(C_1, C_2, C_3\big)$ to the verifier, where

$$C_1 = \mathsf{COM}(\phi, \mathbf{M}\cdot\mathbf{r}_w \bmod q; \rho_1), \;\; C_2 = \mathsf{COM}(\mathit{\Gamma}_\phi(\mathbf{r}_w); \rho_2),$$
$$C_3 = \mathsf{COM}(\mathit{\Gamma}_\phi(\mathbf{w} + \mathbf{r}_w \bmod q); \rho_3).$$

2. **Challenge:** The verifier sends a challenge $Ch \leftarrow U(\{1,2,3\})$ to the prover.
3. **Response:** Depending on Ch, the prover sends RSP computed as follows:
   - $Ch = 1$: Let $\mathbf{t}_w = \mathit{\Gamma}_\phi(\mathbf{w})$, $\mathbf{t}_r = \mathit{\Gamma}_\phi(\mathbf{r}_w)$, and $\mathrm{RSP} = (\mathbf{t}_w, \mathbf{t}_r, \rho_2, \rho_3)$.
   - $Ch = 2$: Let $\phi_2 = \phi$, $\mathbf{w}_2 = \mathbf{w} + \mathbf{r}_w \bmod q$, and $\mathrm{RSP} = (\phi_2, \mathbf{w}_2, \rho_1, \rho_3)$.
   - $Ch = 3$: Let $\phi_3 = \phi$, $\mathbf{w}_3 = \mathbf{r}_w$, and $\mathrm{RSP} = (\phi_3, \mathbf{w}_3, \rho_1, \rho_2)$.

Verification: Receiving RSP, the verifier proceeds as follows:

- $Ch = 1$: Check that $\mathbf{t}_w \in \mathsf{VALID}$, $C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2)$, $C_3 = \mathsf{COM}(\mathbf{t}_w + \mathbf{t}_r \bmod q; \rho_3)$.
- $Ch = 2$: Check that $C_1 = \mathsf{COM}(\phi_2, \mathbf{M}\cdot\mathbf{w}_2 - \mathbf{v} \bmod q; \rho_1)$, $C_3 = \mathsf{COM}(\mathit{\Gamma}_{\phi_2}(\mathbf{w}_2); \rho_3)$.
- $Ch = 3$: Check that $C_1 = \mathsf{COM}(\phi_3, \mathbf{M}\cdot\mathbf{w}_3; \rho_1)$, $\;C_2 = \mathsf{COM}(\mathit{\Gamma}_{\phi_3}(\mathbf{w}_3); \rho_2)$.

In each case, the verifier outputs 1 if and only if all the conditions hold.

Figure 1: Stern-like ZKAoK for the relation $\mathrm{R}_{\mathrm{abstract}}$.

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

The proof of the theorem relies on standard simulation and extraction techniques for Stern-like protocols [41,48,44]. It is given in Appendix G.

6.2 Our Strategy and Basic Techniques, In a Nutshell

Before going into the details of our protocols, we first summarize our governing strategy and the techniques that will be used in the next subsections.

In each protocol, we prove knowledge of (possibly one-dimensional) integer vectors $\{\mathbf{w}_i\}_i$ that have various constraints (e.g., smallness, special arrangements of coordinates, or correlation with one another) and satisfy a system

$$\Big\{ \sum_i \mathbf{M}_{i,j} \cdot \mathbf{w}_i = \mathbf{v}_j \Big\}_j, \tag{13}$$

where $\{\mathbf{M}_{i,j}\}_{i,j}$, $\{\mathbf{v}_j\}_j$ are public matrices (which are possibly zero or identity matrices) and vectors. Our strategy consists in transforming this entire system into one equivalent equation $\mathbf{M}\cdot\mathbf{w} = \mathbf{v}$, where matrix $\mathbf{M}$ and vector $\mathbf{v}$ are public, while the constraints of the secret vector $\mathbf{w}$ capture those of witnesses $\{\mathbf{w}_i\}_i$ and they are provable in zero-knowledge via random permutations. For this purpose, the Stern-like protocol from Section 6.1 comes in handy.

A typical transformation step is of the form $\mathbf{w}_i \to \bar{\mathbf{w}}_i$, where there exists public matrix $\mathbf{P}_{i,j}$ such that $\mathbf{P}_{i,j} \cdot \bar{\mathbf{w}}_i = \mathbf{w}_i$. This subsumes the decomposition and extension mechanisms which first appeared in [48].

- **Decomposition:** Used when $\mathbf{w}_i$ has infinity norm bound larger than 1 and we want to work more conveniently with $\bar{\mathbf{w}}_i$ whose norm bound is exactly 1. In this case, $\mathbf{P}_{i,j}$ is a decomposition matrix (see Section 2.5).
- **Extension:** Used when we insert "dummy" coordinates to $\mathbf{w}_i$ to obtain $\bar{\mathbf{w}}_i$ whose coordinates are somewhat balanced. In this case, $\mathbf{P}_{i,j}$ is a $\{0,1\}$-matrix with zero-columns corresponding to positions of insertions.

Such a step transforms the term $\mathbf{M}_{i,j} \cdot \mathbf{w}_i$ into $\overline{\mathbf{M}}_{i,j} \cdot \bar{\mathbf{w}}_i$, where $\overline{\mathbf{M}}_{i,j} = \mathbf{M}_{i,j} \cdot \mathbf{P}_{i,j}$ is a public matrix. Also, using the commutativity property of addition, we often group together secret vectors having the same constraints.

After a number of transformations, we will reach a system equivalent to (13):

$$\begin{cases} \mathbf{M}'_{1,1} \cdot \mathbf{w}'_1 + \mathbf{M}'_{1,2} \cdot \mathbf{w}'_2 + \cdots + \mathbf{M}'_{1,k} \cdot \mathbf{w}'_k = \mathbf{v}_1, \\ \vdots \\ \mathbf{M}'_{t,1} \cdot \mathbf{w}'_1 + \mathbf{M}'_{t,2} \cdot \mathbf{w}'_2 + \cdots + \mathbf{M}'_{t,k} \cdot \mathbf{w}'_k = \mathbf{v}_t, \end{cases} \tag{14}$$

where integers t, k and matrices $\mathbf{M}'_{i,j}$ are public. Defining

$$\mathbf{M} = \left(\begin{array}{c|c|c|c} \mathbf{M}'_{1,1} & \mathbf{M}'_{1,2} & \cdots & \mathbf{M}'_{1,k} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \mathbf{M}'_{t,1} & \mathbf{M}'_{t,2} & \cdots & \mathbf{M}'_{t,k} \end{array}\right); \quad \mathbf{w} = \begin{pmatrix} \mathbf{w}'_1 \\ \vdots \\ \mathbf{w}'_k \end{pmatrix}; \quad \mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_t \end{pmatrix},$$

we obtain the unified equation $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$. At this stage, we will use a properly defined composition of random permutations to prove the constraints of $\mathbf{w}$. We remark that the crucial aspect of the above process is in fact the manipulation of witness vectors, while the transformations of public matrices/vectors just follow accordingly. To ease the presentation of the next subsections, we will thus only focus on the secret vectors.

In the process, we will employ various extending and permuting techniques which require introducing some notations. The most frequently used ones are given in Table 1. Some of these techniques appeared (in slightly different forms) in previous works [48,49,46,44,45]. The last three parts of the table summarizes newly-introduced techniques that will enable the treatment of secret-and-correlated objects involved in the evaluation of hidden branching programs.

In particular, the intriguing technique of the last row will be used for proving knowledge of secret integer z of the form $z = x \cdot y$ for some $(x, y) \in [0,4] \times \{0,1\}$ satisfying other relations. The following example illustrates how it works.

Example 1. Let $(x, y) = (2, 1)$ and $(c, b) = (4, 1)$. Then we have:

$$\mathsf{ext}_{5\times 2}(2,1) = \big(0,\ 1,\ 0,\ 0,\ 0,\ 4,\ 0,\ 3,\ 0,\ 2\big)^\top$$
$$T_{5\times 2}[4,1]\big(\mathsf{ext}_{5\times 2}(2,1)\big) = \big(0,\ 0,\ 4,\ 0,\ 3,\ 0,\ 2,\ 0,\ 1,\ 0\big)^\top$$

Note that: $T_{5\times 2}[4,1]\big(\mathsf{ext}_{5\times 2}(2,1)\big) = \mathsf{ext}_{5\times 2}(1,0) = \mathsf{ext}_{5\times 2}(2 + 4 \bmod 5, 1 \oplus 1)$.

Notation	Meaning/Property/Usage/Technique
$\mathsf{B}^2_{\mathfrak{m}}$	– The set of vectors in $\{0,1\}^{2\mathfrak{m}}$ with Hamming weight $\mathfrak{m}$. – $\forall \phi \in \mathcal{S}_{2\mathfrak{m}}, \mathbf{x}' \in \mathbb{Z}^{2\mathfrak{m}} : \mathbf{x}' \in \mathsf{B}^2_{\mathfrak{m}} \Leftrightarrow \phi(\mathbf{x}') \in \mathsf{B}^2_{\mathfrak{m}}$. – To prove $\mathbf{x} \in \{0,1\}^{\mathfrak{m}}$: Extend $\mathbf{x}$ to $\mathbf{x}' \in \mathsf{B}^2_{\mathfrak{m}}$, then permute $\mathbf{x}'$.
$\mathsf{B}^3_{\mathfrak{m}}$	– The set of vectors in $\{-1,0,1\}^{3\mathfrak{m}}$ that have exactly $\mathfrak{m}$ coordinates equal to j, for every $j \in \{-1,0,1\}$. – $\forall \phi \in \mathcal{S}_{3\mathfrak{m}}, \mathbf{x}' \in \mathbb{Z}^{3\mathfrak{m}} : \mathbf{x}' \in \mathsf{B}^3_{\mathfrak{m}} \Leftrightarrow \phi(\mathbf{x}') \in \mathsf{B}^3_{\mathfrak{m}}$. – To prove $\mathbf{x} \in \{-1,0,1\}^{\mathfrak{m}}$: Extend $\mathbf{x}$ to $\mathbf{x}' \in \mathsf{B}^3_{\mathfrak{m}}$, then permute $\mathbf{x}'$.
$\mathsf{ext}_2(\cdot)$ and $T_2[\cdot](\cdot)$	– For $c \in \{0,1\} : \mathsf{ext}_2(c) = (\bar{c}, c)^\top \in \{0,1\}^2$. – For $b \in \{0,1\}$ and $\mathbf{x} = (x_0, x_1)^\top \in \mathbb{Z}^2$: $T_2[b](\mathbf{x}) = (x_b, x_{\bar{b}})^\top$. – Property: $\mathbf{x} = \mathsf{ext}_2(c) \Leftrightarrow T_2[b](\mathbf{x}) = \mathsf{ext}_2(c \oplus b)$. – To prove $c \in \{0,1\}$ simultaneously satisfies many relations: Extend it to $\mathbf{x} = \mathsf{ext}_2(c)$, then permute and use the *same* b at all appearances.
$\mathsf{expand}(\cdot,\cdot)$ and $T_{\mathsf{exp}}[\cdot,\cdot](\cdot)$	– For $c \in \{0,1\}$ and $\mathbf{x} \in \mathbb{Z}^{\mathfrak{m}}$: $\mathsf{expand}(c, \mathbf{x}) = (\bar{c} \cdot \mathbf{x}^\top \mid c \cdot \mathbf{x}^\top)^\top \in \mathbb{Z}^{2\mathfrak{m}}$. – For $b \in \{0,1\}, \phi \in \mathcal{S}_m$, $\mathbf{v} = \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \end{pmatrix} \in \mathbb{Z}^{2\mathfrak{m}}$: $T_{\mathsf{exp}}[b, \phi](\mathbf{v}) = \begin{pmatrix} \phi(\mathbf{v}_b) \\ \phi(\mathbf{v}_{\bar{b}}) \end{pmatrix}$. – Property: $\mathbf{v} = \mathsf{expand}(c, \mathbf{x}) \Leftrightarrow T_{\mathsf{exp}}[b, \phi](\mathbf{v}) = \mathsf{expand}(c \oplus b, \phi(\mathbf{x}))$.
$[\cdot]_5$	For $k \in \mathbb{Z}$: $[k]_5$ denotes the integer $t \in \{0,1,2,3,4\}$, s.t. $t = k \bmod 5$.
$\mathsf{ext}_5(\cdot)$ and $T_5[\cdot](\cdot)$	– For $x \in [0,4] : \mathsf{ext}_5(x) = ([x{+}4]_5, [x{+}3]_5, [x{+}2]_5, [x{+}1]_5, x)^\top \in [0,4]^5$. – For $c \in [0,4]$ and $\mathbf{v} = (v_0, v_1, v_2, v_3, v_4)^\top \in \mathbb{Z}^5$: $$T_5[c](\mathbf{v}) = \big(v_{[-c]_5}, v_{[-c+1]_5}, v_{[-c+2]_5}, v_{[-c+3]_5}, v_{[-c+4]_5}\big)^\top.$$ – Property: $\mathbf{v} = \mathsf{ext}_5(x) \Leftrightarrow T_5[c](\mathbf{v}) = \mathsf{ext}_5(x + c \bmod 5)$. – To prove $x \in [0,4]$ simultaneously satisfies many relations: Extend it to $\mathbf{v} = \mathsf{ext}_5(x)$, then permute and use the *same* c at all appearances.
unit_x	– $\forall x \in [0,4]$: unit_x is the 5-dim unit vector $(v_0, \ldots, v_4)^\top$ with $v_x = 1$. – For $c \in [0,4], \mathbf{v} \in \mathbb{Z}^5$: $\mathbf{v} = \mathsf{unit}_x \Leftrightarrow T_5[c](\mathbf{v}) = \mathsf{unit}_{x+c \bmod 5}$. → Allow proving $\mathbf{v} = \mathsf{unit}_x$ for some $x \in [0,4]$ satisfying other relations.
$\mathsf{ext}_{5\times 2}(\cdot,\cdot)$ and $T_{5\times 2}[\cdot,\cdot](\cdot)$	– For $x \in [0,4]$ and $y \in \{0,1\}$: $$\mathsf{ext}_{5\times 2}(x,y) = ([x+4]_5 \cdot \bar{y}, [x+4]_5 \cdot y, [x+3]_5 \cdot \bar{y}, [x+3]_5 \cdot y, [x+2]_5 \cdot \bar{y}, [x+2]_5 \cdot y, [x+1]_5 \cdot \bar{y}, [x+1]_5 \cdot y, x \cdot \bar{y}, x \cdot y)^\top \in [0,4]^{10}$$ – For $(c,b) \in [0,4] \times \{0,1\}$ and $\mathbf{v} = (v_{0,0}, v_{0,1}, \ldots, v_{4,0}, v_{4,1})^\top \in \mathbb{Z}^{10}$: $$T_{5\times 2}[c,b](\mathbf{v}) = \big(v_{[-c]_5,b}, v_{[-c]_5,\bar{b}}, v_{[-c+1]_5,b}, v_{[-c+1]_5,\bar{b}}, v_{[-c+2]_5,b}, v_{[-c+2]_5,\bar{b}}, v_{[-c+3]_5,b}, v_{[-c+3]_5,\bar{b}}, v_{[-c+4]_5,b}, v_{[-c+4]_5,\bar{b}}\big)^\top.$$ – Property: $\mathbf{v} = \mathsf{ext}_{5\times 2}(x,y) \Leftrightarrow T_{5\times 2}[c,b](\mathbf{v}) = \mathsf{ext}_{5\times 2}(x{+}c \bmod 5, y{\oplus}b)$. → Allow proving $z = x \cdot y$ for some $(x,y) \in [0,4] \times \{0,1\}$ satisfying other relations: Extend z to $\mathbf{v} = \mathsf{ext}_{5\times 2}(x,y)$, then permute and use the *same* c, b at all appearances of x, y, respectively.

Table 1. Basic notations and extending/permuting techniques used in our protocols.

6.3 Protocol 1

Let n, m, q, N, t, B_χ be the parameters defined in Section 4. The protocol allows the prover to prove knowledge of LWE secrets and the well-formedness of ciphertexts. It is summarized as follows.

Common input: $\mathbf{F} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{P} \in \mathbb{Z}_q^{m \times t}$; $\{\mathbf{a}_i \in \mathbb{Z}_q^n,\ \mathbf{b}_i \in \mathbb{Z}_q^t\}_{i=1}^N$.

Prover's goal is to prove knowledge of $\mathbf{S} \in [-B_\chi, B_\chi]^{n \times t}$, $\mathbf{E} \in [-B_\chi, B_\chi]^{m \times t}$, $\{\mathbf{x}_i \in [-B_\chi, B_\chi]^t, M_i \in \{0,1\}^t\}_{i=1}^N$ such that the following equations hold:

$$\begin{cases} \mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E} = \mathbf{P} \bmod q \\ \forall i \in [N] : \mathbf{S}^\top \cdot \mathbf{a}_i + \mathbf{x}_i + \lfloor q/2 \rfloor \cdot M_i = \mathbf{b}_i \bmod q. \end{cases} \tag{15}$$

For each $j \in [t]$, let $\mathbf{p}_j, \mathbf{s}_j, \mathbf{e}_j$ be the j-th column of matrices $\mathbf{P}, \mathbf{S}, \mathbf{E}$, respectively. For each $(i,j) \in [N] \times [t]$, let $\mathbf{b}_i[j], \mathbf{x}_i[j], M_i[j]$ denote the j-th coordinate of vectors $\mathbf{b}_i, \mathbf{x}_i, M_i$, respectively. Then, observe that (15) can be rewritten as:

$$\begin{cases} \forall j \in [t] : \mathbf{F}^\top \cdot \mathbf{s}_j + \mathbf{I}_m \cdot \mathbf{e}_j = \mathbf{p}_j \bmod q \\ \forall (i,j) \in [N] \times [t] : \mathbf{a}_i^\top \cdot \mathbf{s}_j + 1 \cdot \mathbf{x}_i[j] + \lfloor q/2 \rfloor \cdot M_i[j] = \mathbf{b}_i[j] \bmod q. \end{cases} \tag{16}$$

Then, we form the following vectors:

$$\mathbf{w}_1 = \big(\mathbf{s}_1^\top \mid \ldots \mid \mathbf{s}_t^\top \mid \mathbf{e}_1^\top \mid \ldots \mid \mathbf{e}_t^\top \mid (\mathbf{x}_1[1], \ldots, \mathbf{x}_N[t])\big)^\top \in [-B_\chi, B_\chi]^{(n+m+N)t};$$
$$\mathbf{w}_2 = (M_1[1], \ldots, M_N[t])^\top \in \{0,1\}^{Nt}.$$

Next, we run $\mathsf{vdec}'_{(n+m+N)t, B_\chi}$ to decompose $\mathbf{w}_1$ into $\bar{\mathbf{w}}_1$ and then extend $\bar{\mathbf{w}}_1$ to $\mathbf{w}_1^* \in \mathsf{B}^3_{(n+m+N)t\delta_{B_\chi}}$. We also extend $\mathbf{w}_2$ into $\mathbf{w}_2^* \in \mathsf{B}^2_{Nt}$ and we then form $\mathbf{w} = ((\mathbf{w}_1^*)^\top \mid (\mathbf{w}_2^*)^\top)^\top \in \{-1,0,1\}^D$, where $D = 3(n+m+N)t\delta_{B_\chi} + 2Nt$.

Observe that relations (16) can be transformed into *one* equivalent equation of the form $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$, where $\mathbf{M}$ and $\mathbf{v}$ are built from the common input.

Having performed the above unification, we now define VALID as the set of all vectors $\mathbf{t} = (\mathbf{t}_1^\top \mid \mathbf{t}_2^\top)^\top \in \{-1,0,1\}^D$, where $\mathbf{t}_1 \in \mathsf{B}^3_{(n+m+N)t\delta_{B_\chi}}$ and $\mathbf{t}_2 \in \mathsf{B}^2_{Nt}$. Clearly, our vector $\mathbf{w}$ belongs to the set VALID.

Next, we specify the set $\mathcal{S}$ and permutations of D elements $\{\varGamma_\phi : \phi \in \mathcal{S}\}$, for which the conditions in (12) hold.

- $\mathcal{S} := \mathcal{S}_{3(n+m+N)t\delta_{B_\chi}} \times \mathcal{S}_{2Nt}$.
- For $\phi = (\phi_1, \phi_2) \in \mathcal{S}$ and for $\mathbf{t} = (\mathbf{t}_1^\top \mid \mathbf{t}_2^\top)^\top \in \mathbb{Z}^D$, where $\mathbf{t}_1 \in \mathbb{Z}^{3(n+m+N)t\delta_{B_\chi}}$ and $\mathbf{t}_2 \in \mathbb{Z}^{2Nt}$, we define $\varGamma_\phi(\mathbf{t}) = (\phi_1(\mathbf{t}_1)^\top \mid \phi_2(\mathbf{t}_2)^\top)^\top$.

By inspection, it can be seen that the desired properties in (12) are satisfied. As a result, we can obtain the required ZKAoK by running the protocol from Section 6.1 with common input $(\mathbf{M}, \mathbf{v})$ and prover's input $\mathbf{w}$. The protocol has communication cost $\mathcal{O}(D \log q) = \widetilde{\mathcal{O}}(\lambda) \cdot \mathcal{O}(Nt)$ bits.

While this protocol has linear complexity in N, it is only used in the initialization phase, where $\varOmega(N)$ bits inevitably have to be transmitted anyway.

6.4 Protocol 2

Let n, m, q, N, t, B be system parameters. The protocol allows the prover to prove knowledge of LWE secrets and the correctness of decryption.

Common input: $\mathbf{F} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{P} \in \mathbb{Z}_q^{m\times t}$; $\mathbf{c}_0 \in \mathbb{Z}_q^n$, $\mathbf{c}_1 \in \mathbb{Z}_q^t$, $M' \in \{0,1\}^t$.

Prover's goal is to prove knowledge of $\mathbf{S} \in [-B_\chi, B_\chi]^{n\times t}$, $\mathbf{E} \in [-B_\chi, B_\chi]^{m\times t}$ and $\mathbf{y} \in [-q/5, q/5]^t$ such that the following equations hold:

$$\mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E} = \mathbf{P} \bmod q; \quad \mathbf{c}_0^\top \cdot \mathbf{S} + \mathbf{y}^\top = \mathbf{c}_1^\top - M'^\top \cdot \lfloor q/2 \rfloor \bmod q. \tag{17}$$

For each $j \in [t]$, let $\mathbf{p}_j, \mathbf{s}_j, \mathbf{e}_j$ be the j-th column of matrices $\mathbf{P}, \mathbf{S}, \mathbf{E}$, respectively; and let $\mathbf{y}[j], \mathbf{c}_1[j], M'[j]$ be the j-th entry of vectors $\mathbf{y}, \mathbf{c}_1, M'$, respectively. Then, observe that (17) can be re-written as:

$$\forall j \in [t] : \begin{cases} \mathbf{F}^\top \cdot \mathbf{s}_j + \mathbf{I}_m \cdot \mathbf{e}_j = \mathbf{p}_j \bmod q \\ \mathbf{c}_0^\top \cdot \mathbf{s}_j + 1 \cdot \mathbf{y}[j] = \mathbf{c}_1[j] - M'[j] \cdot \lfloor q/2 \rfloor \bmod q. \end{cases} \tag{18}$$

Next, we form vector $\mathbf{w}_1 = (\mathbf{s}_1^\top \mid \ldots \mid \mathbf{s}_t^\top \mid \mathbf{e}_1^\top \mid \ldots \mid \mathbf{e}_t^\top)^\top \in [-B_\chi, B_\chi]^{(n+m)t}$, then decompose it to $\bar{\mathbf{w}}_1 \in \{-1,0,1\}^{(n+m)t\delta_{B_\chi}}$, and extend $\bar{\mathbf{w}}_1$ to $\mathbf{w}_1^* \in \mathsf{B}^3_{(n+m)t\delta_{B_\chi}}$.

At the same time, we decompose vector $\mathbf{y} = (\mathbf{y}[1], \ldots, \mathbf{y}[t])^\top \in [-q/5, q/5]^t$ to $\bar{\mathbf{y}} \in \{-1,0,1\}^{t\delta_{q/5}}$, and then extend $\bar{\mathbf{y}}$ to $\mathbf{y}^* \in \mathsf{B}^3_{t\delta_{q/5}}$.

Defining the ternary vector $\mathbf{w} = ((\mathbf{w}_1^*)^\top \mid (\mathbf{y}^*)^\top)^\top \in \{-1,0,1\}^D$ of dimension $D = 3(n+m)t\delta_{B_\chi} + 3t\delta_{q/5}$, we finally obtain the equation $\mathbf{M}\cdot\mathbf{w} = \mathbf{v} \bmod q$, for public matrix $\mathbf{M}$ and public vector $\mathbf{v}$. Using similar arguments as in Section 6.3, we can obtain the desired zero-knowledge argument system. The protocol has communication cost $\mathcal{O}(D\log q) = \widetilde{\mathcal{O}}(\lambda)\cdot\mathcal{O}(t)$ bits.

6.5 Protocol 3

Let n, m, m_d, q, t, ℓ, B be the parameters defined in Section 4. The protocol allows the prover to argue that a given ciphertext is a correct randomization of some hidden ciphertext and that he knows a valid signature on that ciphertext. Let β be the infinity norm bound of these valid signatures.

Common input: It consists of matrices $\mathbf{F} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{P} \in \mathbb{Z}_q^{m\times t}$, $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell \in \mathbb{Z}_q^{n\times m}$, $\mathbf{D} \in \mathbb{Z}_q^{n\times m_d}$ and vectors $\mathbf{c}_0 \in \mathbb{Z}_q^n$, $\mathbf{c}_1 \in \mathbb{Z}_q^t$, $\mathbf{u} \in \mathbb{Z}_q^n$.

Prover's goal is to prove knowledge of $\mathfrak{m} \in \{0,1\}^{m_d}$, $\mu \in \{0,1\}^t$, $\mathbf{e} \in \{-1,0,1\}^t$, $\nu \in [-B,B]^t$, $\tau = (\tau[1], \ldots, \tau[\ell])^\top \in \{0,1\}^\ell$, $\mathbf{v}_1, \mathbf{v}_2 \in [-\beta, \beta]^m$ such that the following equations hold:

$$\begin{cases} \mathbf{A}\cdot\mathbf{v}_1 + \mathbf{A}_0\cdot\mathbf{v}_2 + \sum_{j=1}^{\ell}\mathbf{A}_j\cdot(\tau[j]\cdot\mathbf{v}_2) - \mathbf{D}\cdot\mathfrak{m} = \mathbf{u} \bmod q; \\ \mathbf{H}_{n+t,q-1}\cdot\mathfrak{m} + \begin{pmatrix} \mathbf{F} \\ \mathbf{P}^\top \end{pmatrix}\cdot\mathbf{e} + \begin{pmatrix} \mathbf{0}^{n\times t} \\ \lfloor \frac{q}{2} \rfloor\cdot\mathbf{I}_t \end{pmatrix}\cdot\mu + \begin{pmatrix} \mathbf{0}^{n\times t} \\ \mathbf{I}_t \end{pmatrix}\cdot\nu = \begin{pmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{pmatrix} \bmod q. \end{cases} \tag{19}$$

For this purpose, we perform the following transformations on the witnesses.

Decompositions. Decompose vectors $\mathbf{v}_1, \mathbf{v}_2, \nu$ to vectors $\bar{\mathbf{v}}_1 \in \{-1,0,1\}^{m\delta_\beta}$, $\bar{\mathbf{v}}_2 \in \{-1,0,1\}^{m\delta_\beta}$, $\bar{\nu} \in \{-1,0,1\}^{t\delta_B}$, respectively.

Extensions/Combinations.

- Let $\mathbf{w}_1 = (\mathfrak{m}^\top \mid \mu^\top)^\top \in \{0,1\}^{m_d+t}$ and extend it into $\mathbf{w}_1^* \in \mathsf{B}^2_{m_d+t}$.
- Let $\mathbf{w}_2 = (\bar{\mathbf{v}}_1^\top \mid \bar{\nu}^\top \mid \mathbf{e}^\top)^\top \in \{-1,0,1\}^{m\delta_\beta+t\delta_B+t}$ and extend it into the vector $\mathbf{w}_2^* \in \mathsf{B}^3_{m\delta_\beta+t\delta_B+t}$.
- Extend $\bar{\mathbf{v}}_2$ into $\mathbf{s}_0 \in \mathsf{B}^3_{m\delta_\beta}$. Then, for each $j \in [\ell]$, define $\mathbf{s}_j = \mathsf{expand}(\tau[j], \mathbf{s}_0)$. (We refer to Table 1 for details about $\mathsf{expand}(\cdot,\cdot)$.)

Now, we form vector $\mathbf{w} = \big(\mathbf{w}_1^{*\top} \mid \mathbf{w}_2^{*\top} \mid \mathbf{s}_0^\top \mid \mathbf{s}_1^\top \mid \ldots \mid \mathbf{s}_\ell^\top\big)^\top \in \{-1,0,1\}^D$, where $D = (2\ell+2)3m\delta_\beta + 3t\delta_B + 3t + 2(m_d+t)$. At this point, we observe that the equations in (19) can be equivalently transformed into $\mathbf{M}\cdot\mathbf{w} = \mathbf{v} \bmod q$, where the matrix $\mathbf{M}$ and the vector $\mathbf{v}$ are built from the public input.

Having performed the above transformations, we now define VALID as the set of all vectors $\mathbf{t} = (\mathbf{t}_1^\top \mid \mathbf{t}_2^\top \mid \mathbf{t}_{3,0}^\top \mid \mathbf{t}_{3,1}^\top \mid \ldots \mid \mathbf{t}_{3,\ell}^\top)^\top \in \{-1,0,1\}^D$ for which there exists $\tau = (\tau[1],\ldots,\tau[\ell])^\top \in \{0,1\}^\ell$ such that:

$$\mathbf{t}_1 \in \mathsf{B}^2_{m_d+t};\ \mathbf{t}_2 \in \mathsf{B}^3_{m\delta_\beta+t\delta_B+t};\ \mathbf{t}_{3,0} \in \mathsf{B}^3_{m\delta_\beta};\ \forall j \in [\ell] : \mathbf{t}_{3,j} = \mathsf{expand}(\tau[j], \mathbf{t}_{3,0}).$$

It can be seen that $\mathbf{w}$ belongs to this tailored set. Now, let us specify the set $\mathcal{S}$ and permutations of D elements $\{\mathit{\Gamma}_\phi : \phi \in \mathcal{S}\}$ satisfying the conditions in (12).

- $\mathcal{S} := \mathcal{S}_{2(m_d+t)} \times \mathcal{S}_{3(m\delta_\beta+t\delta_B+t)} \times \mathcal{S}_{3m\delta_\beta} \times \{0,1\}^\ell$.
- For $\phi = \big(\phi_1, \phi_2, \phi_3, (b[1],\ldots,b[\ell])^\top\big) \in \mathcal{S}$, we define the permutation $\mathit{\Gamma}_\phi$ that transforms vector $\mathbf{t} = (\mathbf{t}_1^\top \mid \mathbf{t}_2^\top \mid \mathbf{t}_{3,0}^\top \mid \mathbf{t}_{3,1}^T \mid \ldots \mid \mathbf{t}_{3,\ell}^\top)^\top \in \mathbb{Z}^D$ as follows:

$$\mathit{\Gamma}_\phi(\mathbf{t}) = \big(\phi_1(\mathbf{t}_1)^\top \mid \phi_2(\mathbf{t}_2)^\top \mid \phi_3(\mathbf{t}_{3,0})^\top \mid \\ T_{\mathsf{exp}}[b[1],\phi_3](\mathbf{t}_{3,1})^\top \mid \ldots \mid T_{\mathsf{exp}}[b[\ell],\phi_3](\mathbf{t}_{3,\ell})^\top\big)^\top.$$

By inspection, it can be seen that the properties in (12) are indeed satisfied. As a result, we can obtain the required argument of knowledge by running the protocol from Section 6.1 with common input $(\mathbf{M},\mathbf{v})$ and prover's input $\mathbf{w}$. The protocol has communication cost $\mathcal{O}(D\log q) = \widetilde{\mathcal{O}}(\lambda)\cdot\mathcal{O}(\log N + t)$ bits.

6.6 Protocol 4: A Treatment of Hidden Branching Programs

We now present the proof system run by the user in the OT-AC system of Section 5. It allows arguing knowledge of an input $\mathbf{x} = (x_0,\ldots,x_{\kappa-1})^\top \in \{0,1\}^\kappa$ satisfying a hidden branching program $\mathsf{BP} = \{(\mathrm{var}(\theta), \pi_{\theta,0}, \pi_{\theta,1})\}_{\theta=1}^L$ of length for $L \in \mathsf{poly}(\lambda)$. The prover should additionally demonstrate that: (i) He has a valid credential for $\mathbf{x}$; (ii) The hashed encoding of BP is associated with some hidden ciphertext of the database (and he knows a signature guaranteeing this link); (iii) A given ciphertext is a re-randomization of that hidden ciphertext.

Recall that, at each step $\theta \in [L]$ of the evaluation of $\mathsf{BP}(\mathbf{x})$, we have to look up the value $x_{\mathrm{var}(\theta)}$ in $\mathbf{x} = (x_0,\ldots,x_{\kappa-1})^\top$ to compute the θ-th state η_θ as per

$$\eta_\theta = \pi_{\theta,x_{\mathrm{var}(\theta)}}(\eta_{\theta-1}) = \pi_{\theta,0}(\eta_{\theta-1})\cdot\bar{x}_{\mathrm{var}(\theta)} + \pi_{\theta,1}(\eta_{\theta-1})\cdot x_{\mathrm{var}(\theta)}. \tag{20}$$

To prove that each step is done correctly, it is necessary to provide evidence that the corresponding search is honestly carried out without revealing $x_{\mathrm{var}(\theta)}$,

$\mathrm{var}(\theta)$ nor $\{\pi_{\theta,b}\}_{b=0}^{1}$. To this end, a first idea is to perform a simple left-to-right search on $(x_0,\ldots,x_{\kappa-1})$: namely, (20) is expressed in terms of a matrix-vector relation where η_θ is encoded as a unit vector of dimension 5; $\{\pi_{\theta,b}\}_{b=0}^{1}$ are represented as permutation matrices; and $\mathbf{x}_{\mathrm{var}(\theta)} = \mathbf{M}_{\mathrm{var}(\theta)} \cdot \mathbf{x}$ is computed using a matrix $\mathbf{M}_{\mathrm{var}(\theta)} \in \{0,1\}^{\kappa\times\kappa}$ containing exactly one 1 per row. While this approach can be handled using proofs for matrix-vector relations using the techniques of [45], the expected complexity is $\mathcal{O}(\kappa)$ for each step, so that the total complexity becomes $\mathcal{O}(L\kappa)$. Fortunately, a better complexity can be achieved.

If we instead perform a dichotomic search on $\mathbf{x} = (x_0,\ldots,x_{\kappa-1})^\top$, we can reduce the complexity of each step to $\mathcal{O}(\log\kappa)$. To this end, we need to prove a statement "I performed a correct dichotomic search on my secret array $\mathbf{x}$".

In order to solve this problem, we will employ two existing lattice-based tools (which we recall in Section H.1 of Appendix, for the sake of completeness):

(i) A variant of the SIS-based computationally binding and statistically hiding commitment scheme from [41], which allows to commit to one-bit messages;

(ii) The SIS-based Merkle hash tree proposed in [46].

Let $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$ and $\mathbf{a}_{\mathsf{com}} \hookleftarrow U(\mathbb{Z}_q^n)$. For each $i \in [0,\kappa-1]$, we let the receiver commit to $x_i \in \{0,1\}$ as $\mathsf{com}_i = \mathbf{a}_{\mathsf{com}} \cdot x_i + \bar{\mathbf{A}} \cdot \mathbf{r}_{\mathsf{com},i}$, with $\mathbf{r}_{\mathsf{com},i} \hookleftarrow U(\{0,1\}^m)$, and reveal $\mathsf{com}_1,\ldots,\mathsf{com}_{\kappa-1}$ to the sender. We build a Merkle tree of depth $\delta_\kappa = \lceil\log\kappa\rceil$ on top of the leaves $\mathsf{com}_0,\ldots,\mathsf{com}_{\kappa-1}$ using the SIS-based hash function $h_{\bar{\mathbf{A}}} : \{0,1\}^{n\lceil\log q\rceil} \times \{0,1\}^{n\lceil\log q\rceil} \to \{0,1\}^{n\lceil\log q\rceil}$ of [46]. Our use of Merkle trees is reminiscent of [46] in that the content of the leaves is public. The Merkle tree will actually serve as a "bridge" ensuring that: (i) The same string $\mathbf{x}$ is used in all steps while enabling dichotomic searches; (ii) At each step, the prover indeed uses some coordinate of $\mathbf{x}$ (without revealing which one), the choice of which is dictated by a path in the tree determined by $\mathrm{var}(\theta)$.

Since $\{\mathsf{com}_i\}_{i=0}^{\kappa-1}$ are public, both parties can deterministically compute the root $\mathbf{u}_{\mathsf{tree}}$ of the Merkle tree. For each $\theta \in [L]$, we consider the binary representation $d_{\theta,1},\ldots,d_{\theta,\delta_\kappa}$ of $\mathrm{var}(\theta)$, which is part of the encoding of BP defined in (7). We then prove knowledge of a bit y_θ satisfying the statement "From the root $\mathbf{u}_{\mathsf{tree}} \in \{0,1\}^{n\lceil\log q\rceil}$ of the tree, the path determined by the bits $d_{\theta,1},\ldots,d_{\theta,\delta_\kappa}$ leads to the leaf associated with the commitment opened to y_θ." If the Merkle tree and the commitment scheme are both secure, it should hold that $y_\theta = x_{\mathrm{var}(\theta)}$. Said otherwise, we can provably perform a "dichotomic search" for $x_{\mathrm{var}(\theta)} = y_\theta$. Moreover, the techniques from [46] can be adapted to do this in zero-knowledge manner, i.e., without revealing the path nor the reached leaf.

Now, our task can be divided into 3 steps: (i) Proving that the searches on Merkle tree yield $y_1,\ldots,y_L$; (ii) Proving that the branching program evaluates to $\mathsf{BP}(\mathbf{x}) = 1$ if $y_1,\ldots,y_L$ are used in the evaluation; (iii) Proving all the other relations mentioned above, as well as the consistency of $\{\mathsf{com}_i\}_{i=0}^{\kappa-1}$ and the fact that they open to a certified $\mathbf{x} \in \{0,1\}^\kappa$.

Thanks to dichotomic searches, the communication cost drops to $\mathcal{O}(L\delta_\kappa + \kappa)$. These steps can be treated as explained below.

The Merkle Tree Step. At each step $\theta \in [L]$, the prover demonstrates knowledge of a path consisting of δ_κ nodes $\mathbf{g}_{\theta,1}, \ldots, \mathbf{g}_{\theta,\delta_\kappa} \in \{0,1\}^{n\lceil \log q \rceil}$ determined by $d_{\theta,1}, \ldots, d_{\theta,\delta_\kappa}$, as well as their sibling nodes $\mathbf{t}_{\theta,1}, \ldots, \mathbf{t}_{\theta,\delta_\kappa} \in \{0,1\}^{n\lceil \log q \rceil}$. Also, the prover argues knowledge of an opening $(y_\theta, \mathbf{r}_\theta) \in \{0,1\} \times \{0,1\}^m$ for the commitment of which $\mathbf{g}_{\theta,\delta_\kappa}$ is a binary decomposition. As shown in [46] (and recalled in Appendix H.1), it suffices to prove the following relations (mod q):

$$\forall \theta \in [L] \begin{cases} \bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,1}, \mathbf{g}_{\theta,1}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,1}, \mathbf{t}_{\theta,1}) = \mathbf{H}_{n,q-1} \cdot \mathbf{u}_{\mathsf{tree}}, \\ \bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,2}, \mathbf{g}_{\theta,2}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,2}, \mathbf{t}_{\theta,2}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad -\mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,1} = \mathbf{0}, \\ \qquad\qquad\qquad\qquad \vdots \\ \bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,\delta_\kappa}, \mathbf{g}_{\theta,\delta_\kappa}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,\kappa}, \mathbf{t}_{\theta,\kappa}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad -\mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,\delta_\kappa - 1} = \mathbf{0}, \\ \mathbf{a}_{\mathsf{com}} \cdot y_\theta + \bar{\mathbf{A}} \cdot \mathbf{r}_\theta - \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,\delta_\kappa} = \mathbf{0}, \end{cases} \tag{21}$$

where $\mathsf{expand}(\cdot,\cdot)$ is defined in Table 1.

Extending.

- For each $(\theta, i) \in [L] \times [\delta_\kappa]$: Extend $\mathbf{g}_{\theta,i}, \mathbf{t}_{\theta,i} \in \{0,1\}^{m/2}$ to $\widetilde{\mathbf{g}}_{\theta,i}, \widetilde{\mathbf{t}}_{\theta,i} \in \mathsf{B}^2_{m/2}$, respectively. Then, let $\widehat{\mathbf{g}}_{\theta,i} = \mathsf{expand}(d_{\theta,i}, \widetilde{\mathbf{g}}_{\theta,i})$ and $\widehat{\mathbf{t}}_{\theta,i} = \mathsf{expand}(\bar{d}_{\theta,i}, \widetilde{\mathbf{t}}_{\theta,i})$.
- For each $\theta \in [L]$, extend the bit y_θ into the vector $\mathbf{y}_\theta = \mathsf{ext}_2(y_\theta) \in \{0,1\}^2$.
- Let $\widetilde{\mathbf{r}} = (\mathbf{r}_1^\top \mid \ldots \mid \mathbf{r}_L^\top)^\top \in \{0,1\}^{mL}$, then extend it into the vector $\widehat{\mathbf{r}} \in \mathsf{B}^2_{mL}$.

Combining. Next, we let $D_{\mathsf{tree}} = 5mL\delta_\kappa + 2L + 2mL$ and define

$$\mathbf{w}_{\mathsf{tree}} = \big(\widetilde{\mathbf{g}}_{1,1}^\top \mid \widehat{\mathbf{g}}_{1,1}^\top \mid \widehat{\mathbf{t}}_{1,1}^\top \mid \ldots \mid \widetilde{\mathbf{g}}_{1,\delta_\kappa}^\top \mid \widehat{\mathbf{g}}_{1,\delta_\kappa}^\top \mid \widehat{\mathbf{t}}_{1,\delta_\kappa}^\top \mid \ldots \mid \widetilde{\mathbf{g}}_{L,1}^\top \mid \widehat{\mathbf{g}}_{L,1}^\top \mid \widehat{\mathbf{t}}_{L,1}^\top \\ \mid \ldots \mid \widetilde{\mathbf{g}}_{L,\delta_\kappa}^\top \mid \widehat{\mathbf{g}}_{L,\delta_\kappa}^\top \mid \widehat{\mathbf{t}}_{L,\delta_\kappa}^\top \mid \mathbf{y}_1^\top \mid \ldots \mid \mathbf{y}_L^\top \mid \widehat{\mathbf{r}}^\top\big)^\top \in \{0,1\}^{D_{\mathsf{tree}}}. \tag{22}$$

Then, observe that, the above $L(\delta_\kappa + 1)$ equations can be combined into one:

$$\mathbf{M}_{\mathsf{tree}} \cdot \mathbf{w}_{\mathsf{tree}} = \mathbf{v}_{\mathsf{tree}} \bmod q, \tag{23}$$

where matrix $\mathbf{M}_{\mathsf{tree}}$ and vector $\mathbf{v}_{\mathsf{tree}}$ are built from the public input.

The Branching Program Step. The last three parts of Table 1 describe the vector transformations that will be used to handle the secret vectors appearing in the evaluation of BP. The following equations emulate the evaluation process. In particular, for each $\theta \in [2, L]$, we introduce an extra vector $\mathbf{e}_\theta = (c_{\theta,0}, \ldots, c_{\theta,4}) \in$

$\{0,1\}^5$ to enable the extraction of the values $\pi_{\theta,0}(\eta_{\theta-1})$, and $\pi_{\theta,1}(\eta_{\theta-1})$.

$$\begin{cases} \pi_{1,0}(0)\cdot\bar{y}_1 + \pi_{1,1}(0)\cdot y_1 - \eta_1 = 0, & \text{// computing } \eta_1 \text{ with } \eta_0 = 0 \\ \mathbf{e}_2 - \sum_{i=0}^{4} \mathsf{unit}_i \cdot c_{2,i} = (0,0,0,0,0)^\top, & \text{// we will also prove } \mathbf{e}_2 = \mathsf{unit}_{\eta_1} \\ f_{2,0} - \sum_{i=0}^{4} \pi_{2,0}(i)\cdot c_{2,i} = 0, & \text{// meaning: } f_{2,0} = \pi_{2,0}(\eta_1) \\ f_{2,1} - \sum_{i=0}^{4} \pi_{2,1}(i)\cdot c_{2,i} = 0, & \text{// meaning: } f_{2,1} = \pi_{2,1}(\eta_1) \\ f_{2,0}\cdot\bar{y}_2 + f_{2,1}\cdot y_2 - \eta_2 = 0, & \text{// computing } \eta_2 \\ \qquad\qquad\vdots & \\ \mathbf{e}_L - \sum_{i=0}^{4} \mathsf{unit}_i \cdot c_{L,i} = (0,0,0,0,0)^\top, & \text{// we will also prove } \mathbf{e}_L = \mathsf{unit}_{\eta_{L-1}} \\ f_{L,0} - \sum_{i=0}^{4} \pi_{L,0}(i)\cdot c_{L,i} = 0, & \text{// meaning: } f_{L,0} = \pi_{L,0}(\eta_{L-1}) \\ f_{L,1} - \sum_{i=0}^{4} \pi_{L,1}(i)\cdot c_{L,i} = 0, & \text{// meaning: } f_{L,1} = \pi_{L,1}(\eta_{L-1}) \\ f_{L,0}\cdot\bar{y}_L + f_{L,1}\cdot y_L = 0. & \text{// final state } \eta_L = 0 \end{cases} \tag{24}$$

Extending.

- For each $\theta \in [L-1]$, extend $\eta_\theta \in [0,4]$ to 5-dimensional vector $\mathbf{s}_\theta = \mathsf{ext}_5(\eta_\theta)$.
- For each $(\theta, j) \in [2, L] \times \{0,1\}$, extend $f_{\theta,j} \in [0,4]$ to $\mathbf{f}_{\theta,j} = \mathsf{ext}_5(f_{\theta,j})$.
- For each $(\theta, i) \in [2, L] \times [0,4]$, extend $c_{\theta,i} \in \{0,1\}$ to $\mathbf{c}_{\theta,i} = \mathsf{ext}_2(c_{\theta,i})$.
- Extend the products $\pi_{1,0}(0)\cdot\bar{y}_1$ and $\pi_{1,1}(0)\cdot y_1$ into 10-dimensional vectors $\mathbf{h}_{1,0} = \mathsf{ext}_{5\times2}(\pi_{1,0}(0), \bar{y}_1)$ and $\mathbf{h}_{1,1} = \mathsf{ext}_{5\times2}(\pi_{1,1}(0), y_1)$, respectively.
- For each $\theta \in [2, L]$, extend the products $f_{\theta,0}\cdot\bar{y}_\theta$ and $f_{\theta,1}\cdot y_\theta$ into 10-dimensional vectors $\mathbf{h}_{\theta,0} = \mathsf{ext}_{5\times2}(f_{\theta,0}, \bar{y}_\theta)$ and $\mathbf{h}_{\theta,1} = \mathsf{ext}_{5\times2}(f_{\theta,1}, y_\theta)$.
- For $(\theta, i) \in [2, L] \times [0,4]$, extend the products $\pi_{\theta,0}(i)\cdot c_{\theta,i}$ and $\pi_{\theta,1}(i)\cdot c_{\theta,i}$ into $\mathbf{z}_{\theta,0,i} = \mathsf{ext}_{5\times2}(\pi_{\theta,0}(i), c_{\theta,i})$ and $\mathbf{z}_{\theta,1,i} = \mathsf{ext}_{5\times2}(\pi_{\theta,1}(i), c_{\theta,i})$, respectively.

Combining. Let $D_{\mathsf{BP}} = 150L - 130$, and form $\mathbf{w}_{\mathsf{BP}} \in [0,4]^{D_{\mathsf{BP}}}$ of the form:

$$\big(\mathbf{s}_1^\top \mid \ldots \mid \mathbf{s}_{L-1}^\top \mid \mathbf{e}_2^\top \mid \ldots \mid \mathbf{e}_L^\top \mid \mathbf{c}_{2,0}^\top \mid \ldots \mid \mathbf{c}_{L,4}^\top \mid \mathbf{z}_{2,0,0}^\top \mid \ldots \mid \mathbf{z}_{L,1,4}^\top \mid \\ \mathbf{f}_{2,0}^\top \mid \ldots \mid \mathbf{f}_{L,1}^\top \mid \mathbf{h}_{1,0}^\top \mid \mathbf{h}_{1,1}^\top \mid \mathbf{h}_{2,0}^\top \mid \mathbf{h}_{2,1}^\top \mid \ldots \mid \mathbf{h}_{L,0}^\top \mid \mathbf{h}_{L,1}^\top\big)^\top. \tag{25}$$

Then, observe that the vector $\mathbf{w}_{\mathsf{BP}}$ of (25) satisfies *one* equation of the form:

$$\mathbf{M}_{\mathsf{BP}} \cdot \mathbf{w}_{\mathsf{BP}} = \mathbf{v}_{\mathsf{BP}}, \tag{26}$$

where matrix $\mathbf{M}_{\mathsf{BP}}$ and vector $\mathbf{v}_{\mathsf{BP}}$ are obtained from the common input. Note that we work with integers in $[0,4]$, which are much smaller than q. As a result,

$$\mathbf{M}_{\mathsf{BP}} \cdot \mathbf{w}_{\mathsf{BP}} = \mathbf{v}_{\mathsf{BP}} \bmod q. \tag{27}$$

Conversely, if we can prove that (27) holds for a well-formed vector $\mathbf{w}_{\mathsf{BP}}$, then that vector should also satisfy (26).

THE THIRD STEP. In the third layer, we have to prove knowledge of:

$$\begin{cases} d_{1,1}, \ldots, d_{L,\delta_\kappa} \in \{0,1\}, \pi_{1,0}(0), \ldots, \pi_{L,1}(4) \in [0,4],\ \mathfrak{m} \in \{0,1\}^{m_d}, \\ \mathbf{x} = (x_0, \ldots, x_{\kappa-1})^\top \in \{0,1\}^\kappa,\ \mathfrak{m}_{\mathsf{U},\mathbf{x}} \in \{0,1\}^{\frac{m}{2}+\kappa},\ \widehat{\mathfrak{m}}_{\mathsf{U},\mathbf{x}} \in \{0,1\}^{\frac{m}{2}}, \\ \mathbf{e}_{\mathsf{U}} \in \{0,1\}^m,\ \mathbf{r}_{\mathsf{com},0}, \ldots, \mathbf{r}_{\mathsf{com},\kappa-1} \in \{0,1\}^m,\ \mu \in \{0,1\}^t,\ \tau \in \{0,1\}^\ell, \\ \tau_{\mathsf{U}} \in \{0,1\}^{\ell_I}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_{\mathsf{U},1}, \mathbf{v}_{\mathsf{U},2}, \mathbf{r}_{\mathsf{U}} \in [-\beta,\beta]^m, \mathbf{e} \in \{-1,0,1\}^t, \nu \in [-B,B]^t, \end{cases} \tag{28}$$

which satisfy the equations of (11) for $\mathbf{z}_{\mathsf{BP},\rho} = (d_{1,1}, \ldots, d_{L,\delta_\kappa}, \pi_{1,0}(0), \ldots, \pi_{L,1}(4))^\top$ and, $\forall i \in [0, \kappa-1]$, the bit x_i is committed in com_i with randomness $\mathbf{r}_{\mathsf{com},i}$:

$$\begin{bmatrix} \mathbf{a}_{\mathsf{com}} & & \\ & \ddots & \\ & & \mathbf{a}_{\mathsf{com}} \end{bmatrix} \cdot \mathbf{x} + \begin{bmatrix} \bar{\mathbf{A}} & & \\ & \ddots & \\ & & \bar{\mathbf{A}} \end{bmatrix} \cdot \begin{pmatrix} \mathbf{r}_{\mathsf{com},0} \\ \vdots \\ \mathbf{r}_{\mathsf{com},\kappa-1} \end{pmatrix} = \begin{pmatrix} \mathsf{com}_0 \\ \vdots \\ \mathsf{com}_{\kappa-1} \end{pmatrix} \bmod q.$$

Decomposing. We use $\mathsf{vdec}'_{m,\beta}(\cdot)$ to decompose $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_{\mathsf{U},1}, \mathbf{v}_{\mathsf{U},2}, \mathbf{r}_{\mathsf{U}} \in [-\beta,\beta]^m$ into $\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \bar{\mathbf{v}}_{\mathsf{U},1}, \bar{\mathbf{v}}_{\mathsf{U},2}, \bar{\mathbf{r}}_{\mathsf{U}} \in \{-1,0,1\}^{m\delta_\beta}$, respectively. Similarly, we decompose vector $\nu \in [-B,B]^t$ into vector $\bar{\nu} = \mathsf{vdec}'_{t,B}(\nu) \in \{-1,0,1\}^{t\delta_B}$.

Extending and Combining. Next, we perform the following steps:

- For each $(\theta, i) \in [L] \times [\delta_\kappa]$, extend $d_{\theta,i}$ to $\mathbf{d}_{\theta,i} = \mathsf{ext}_2(d_{\theta,i})$.
- For each $(\theta, j, i) \in [L] \times \{0,1\} \times [0,4]$, extend $\pi_{\theta,j}(i)$ to $\mathit{\Pi}_{\theta,j,i} = \mathsf{ext}_5(\pi_{\theta,j}(i))$.
- Let $\overline{\mathbf{w}}_{3,1} = \big(\mathbf{x}^\top | \mathbf{r}^\top_{\mathsf{com},0} | \ldots | \mathbf{r}^\top_{\mathsf{com},\kappa-1} | \mathfrak{m}^\top_{\mathsf{U},\mathbf{x}} | \widehat{\mathfrak{m}}^\top_{\mathsf{U},\mathbf{x}} | \mathfrak{m}^\top \mid \mathbf{e}^\top_{\mathsf{U}} | \mu^\top\big)^\top \in \{0,1\}^{D_{3,1}}$, where $D_{3,1} = \kappa(m+2) + 2m + m_d + t$. Then extend $\overline{\mathbf{w}}_{3,1}$ to $\mathbf{w}_{3,1} \in \mathsf{B}^2_{D_{3,1}}$.
- Define the vector $\overline{\mathbf{w}}_{3,2} = (\bar{\mathbf{v}}_1^\top | \bar{\mathbf{v}}^\top_{\mathsf{U},1} | \bar{\mathbf{r}}^\top_{\mathsf{U}} | \bar{\nu}^\top | \mathbf{e}^\top)^\top \in \{-1,0,1\}^{D_{3,2}}$ of dimension $D_{3,2} = 3m\delta_\beta + t(\delta_B + 1)$ and extend it into $\mathbf{w}_{3,2} \in \mathsf{B}^3_{D_{3,2}}$.
- Extend $\bar{\mathbf{v}}_2$ to $\mathbf{s}_0 \in \mathsf{B}^3_{m\delta_\beta}$. Then for $j \in [\ell]$, form vector $\mathsf{s}_j = \mathsf{expand}\big(\tau[j], \mathbf{s}_0\big)$.
- Extend $\bar{\mathbf{v}}_{\mathsf{U},2}$ to $\mathbf{s}_{\mathsf{U},0} \in \mathsf{B}^3_{m\delta_\beta}$. Then for $j \in [\ell_I]$, form $\mathsf{s}_{\mathsf{U},j} = \mathsf{expand}\big(\tau_{\mathsf{U}}[j], \mathbf{s}_{\mathsf{U},0}\big)$.

Given the above transformations, let $D_3 = 2L(\delta_\kappa + 25) + 2D_{3,1} + 3D_{3,2} + 3m\delta_\beta(2\ell+1) + 3m\delta_\beta(2\ell_I + 1)$ and construct vector $\mathbf{w}_3 \in [-1,4]^{D_3}$ of the form:

$$\begin{aligned} \big(\mathbf{d}^\top_{1,1} \mid \ldots \mid \mathbf{d}^\top_{L,\delta_\kappa} \mid \mathit{\Pi}^\top_{1,0,0} \mid \ldots \mid \mathit{\Pi}^\top_{L,1,4} \mid \mathbf{w}^\top_{3,1} \mid \mathbf{w}^\top_{3,2} \mid \\ \mathbf{s}^\top_0 \mid \mathbf{s}^\top_1 \mid \ldots \mid \mathbf{s}^\top_\ell \mid \mathbf{s}^\top_{\mathsf{U},0} \mid \mathbf{s}^\top_{\mathsf{U},1} \mid \ldots \mid \mathbf{s}^\top_{\mathsf{U},\ell_I} \mid \big)^\top. \end{aligned} \tag{29}$$

Observe that the given five equations can be combined into one of the form:

$$\mathbf{M}_3 \cdot \mathbf{w}_3 = \mathbf{v}_3 \bmod q, \tag{30}$$

where matrix $\mathbf{M}_3$ and vector $\mathbf{v}_3$ can be built from the public input.

PUTTING PIECES ALTOGETHER. At the final stage of the process, we connect the three aforementioned steps. Indeed, all the equations involved in our process are captured by (23), (27), and (30) - which in turn can be combined into:

$$\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q, \tag{31}$$

where $\mathbf{w} = (\mathbf{w}_{\mathsf{tree}}^\top \mid \mathbf{w}_{\mathsf{BP}}^\top \mid \mathbf{w}_3^\top)^\top \in [-1,4]^D$, for

$$D = D_{\mathsf{tree}} + D_{\mathsf{BP}} + D_3 = \widetilde{\mathcal{O}}(\lambda) \cdot (L \cdot \log \kappa + \kappa) + \widetilde{\mathcal{O}}(\lambda) \cdot (\log N + \lambda) + \widetilde{\mathcal{O}}(1) \cdot t.$$

The components of $\mathbf{w}$ all have constraints listed in Table 1. By construction, these blocks either belong to the special sets $\mathsf{B}_{\mathsf{m}}^2$, $\mathsf{B}_{\mathsf{m}}^3$ or they have the special forms $\mathsf{expand}(\cdot,\cdot)$, $\mathsf{ext}_2(\cdot)$, $\mathsf{ext}_5(\cdot)$, $\mathsf{ext}_{5\times 2}(\cdot,\cdot)$, which are invariant under the permutations defined in Table 1. As a result, we can specify suitable sets VALID, $\mathcal{S}$ and permutations of D elements $\{\varGamma_\phi : \phi \in \mathcal{S}\}$, for which the conditions of (12) are satisfied. The description of VALID, $\mathcal{S}$ and $\varGamma_\phi$ is detailed in Appendix H.2.

Our desired argument system then works as follows. At the beginning of the interaction, the prover computes commitments $\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1} \in \mathbb{Z}_q^n$ and send them once to the verifier. Both parties construct matrix $\mathbf{M}$ and vector $\mathbf{v}$ based on the public input as well as $\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1}$, while the prover prepares vector $\mathbf{w}$, as described. Finally, they run the protocol of Section 6.1, which has communication cost $\mathcal{O}(D \log q) = \mathcal{O}(L \cdot \log \kappa + \kappa)$.

Acknowledgements

Part of this research was funded by Singapore Ministry of Education under Research Grant MOE2016-T2-2-014(S) and by the French ANR ALAMBIC project (ANR-16-CE39-0006). Another part was funded by BPI-France in the context of the national project RISQ (P141580).

References

1. M. Abe, J. Camenisch, M. Dubovitskaya, and R. Nishimaki. Universally composable adaptive oblivious transfer (with access control) from standard assumptions. *ACM Workshop on Digital Identity Management*, 2013.
2. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. *Eurocrypt*, 2010.
3. W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. *Eurocrypt*, 2001.
4. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *STACS 2009*, 2009.
5. G. Asharov, A. Jain, A. Lopez-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. *Eurocrypt 2012*, 2012.
6. D. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc1. *STOC'86*, 1986.
7. F. Böhl, D. Hofheinz, T. Jager, J. Koch, and C. Striecks. Confined guessing: New signatures from standard assumptions. *Journal of Cryptology*, 28(1):176–208, 2015.
8. D. Boneh and X. Boyen. Efficient selective-ID secure Identity-Based Encryption without random oracles. *Eurocrypt*, 2004.
9. F. Bourse, R. Del Pino, M. Minelli, and H. Wee. FHE circuit privacy almost for free. *Crypto 2016*, 2016.
10. X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. *PKC 2010*, 2010.

11. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. On the classical hardness of learning with errors. *STOC*, 2013.
12. E. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung. Design validations for discrete logarithm based signature schemes. *PKC 2000*, 2000.
13. J. Camenisch, M. Dubovitskaya, R. Enderlein, and G. Neven. Oblivious transfer with hidden access control from attribute-based encryption. *SCN 2012*, 2012.
14. J. Camenisch, M. Dubovitskaya, and G. Neven. Oblivious transfer with access control. *ACM-CCS 2009*, 2009.
15. J. Camenisch, M. Dubovitskaya, G. Neven, and G. Zaverucha. Oblivious transfer with hidden access control policies. *PKC'11*, 2011.
16. J. Camenisch, G. Neven, and a. shelat. Simulatable adaptive oblivious transfer. *Eurocrypt 2007*, 2007.
17. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. *FOCS 2001*, 2001.
18. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. *Eurocrypt*, 2010.
19. B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. *FOCS 1995*, 1995.
20. S. Coull, M. Green, and S. Hohenberger. Controlling access to an oblivious database using stateful anonymous credentials. *PKC 2009*, 2009.
21. I. Damgård and J.-B. Nielsen. Universally composable efficient multiparty computation from threshold homomorphic encryption. *Crypto 2003*, 2003.
22. G. Di Crescenzo, R. Ostrovsky, and S. Rajagopalan. Conditional oblivious transfer and timed-release encryption. *Eurocrypt'99*, 1999.
23. N. Döttling, N. Fleischhacker, J. Krupp, and D. Schröder. Two-message, oblivious evaluation of cryptographic functionalities. *Crypto 2016*, 2016.
24. L. Ducas and D. Stehlé. Sanitization of FHE ciphertexts. *Eurocrypt 2016*, 2016.
25. S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.
26. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. *Crypto'86*. Springer, 1987.
27. M. Freedman, Y. Ishai, B. Pinkas, and O. Reingold. Keyword search and oblivious pseudorandom functions. *TCC 2005*, 2005.
28. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. *STOC*, 2008.
29. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. *Crypto*, 2013.
30. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. *STOC*, 1987.
31. S. Gorbunov and D. Vinayagamurthy. Riding on asymmetry: Efficient abe for branching programs. *Asiacrypt 2015*, 2015.
32. S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. *Asiacrypt 2010*, 2010.
33. M. Green and S. Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. *Asiacrypt 2007*, 2007.
34. M. Green and S. Hohenberger. Universally composable adaptive oblivious transfer. *Asiacrypt 2008*, 2008.
35. M. Green and S. Hohenberger. Practical adaptive oblivious transfer from simple assumptions. *TCC 2011*, 2011.
36. J. Herranz. Restricted adaptive oblivious transfer. *Theoretical Computer Science*, 412(46):6498–6506, 2011.

37. R. Hiromasa, M. Abe, and T. Okamoto. Packing messages and optimizing bootstrapping in GSW-FHE. *PKC 2015*, 2015.
38. S. Hohenberger and B. Waters. Short and stateless signatures from the RSA assumption. *Crypto*, 2009.
39. Y. Ishai and A. Paskin. Evaluating branching programs on encrypted data. *TCC*, 2007.
40. S. Jarecki and X. Liu. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. *TCC 2009*, 2009.
41. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. *Asiacrypt'08*, 2008.
42. K. Kurosawa, L. Phong, and R. Nojima. Efficiency-improved fully simulatable adaptive OT under the DDH assumption. *SCN 2010*, 2010.
43. K. Kurosawa, L. Phong, and R. Nojima. Generic fully simulatable adaptive oblivious transfer. *ACNS 2011*, 2011.
44. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. *Asiacrypt 2016*, 2016.
45. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. *Asiacrypt 2016*, 2016.
46. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. *Eurocrypt 2016*, 2016.
47. A. Y. Lindell. Efficient fully-simulatable oblivious transfer. *CT-RSA*, 2008.
48. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the isis problem, and applications. *PKC 2013*, 2013.
49. S. Ling, K. Nguyen, and H. Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based. *PKC 2015*, 2015.
50. H. Lipmaa. Private branching programs: On communication-efficient cryptocomputing. *IACR Cryptology ePrint Archive*, 2008:107, 2008.
51. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. *Eurocrypt*, 2012.
52. M. Naor and B. Pinkas. Oblivious transfer with adaptive queries. *Crypto*, 1999.
53. M. Naor and B. Pinkas. Efficient oblivious transfer protocols. *SODA*, 2001.
54. M. Naor and B. Pinkas. Computationally secure oblivious transfer. *Journal of Cryptology*, 18(1):1–35, 2005.
55. T. Nishide, K. Yoneyama, and K. Ohta. Attribute-based encryption with partially hidden encryptor-specified access structures. *ACNS'08*, 2008.
56. C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. *Crypto 2008*, 2008.
57. M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.
58. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *STOC*, 2005.
59. A. Sahai and B. Waters. Fuzzy identity-based encryption. *Eurocrypt 2005*, 2005.
60. J. Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.
61. Y. Tauman-Kalai. Smooth projective hashing and two-message oblivious transfer. *Eurocrypt*, 2005.
62. Y. Zhang, M.-H. Au, D. Wong, Q. Huang, N. Mamoulis, D. Cheung, and S.-M. Yiu. Oblivious transfer with access control: Realizing disjunction without duplication. *Pairing*, 2010.

A Comparisons Among Adaptive OT and OT-AC Protocols

Protocol	Initialization Cost	Transfer Cost	Assumptions	Security
Folklore	$\cdot$	$\mathcal{O}(\lambda N)$	general	Full Sim
NP [52]	$\cdot$	$\mathcal{O}(\lambda \cdot \log(N))$	DDH + $\mathcal{OT}_1^2$	Half Sim
KPN [42]	$\mathcal{O}(\lambda(N \cdot U))$	$\mathcal{O}(\lambda)$	DDH	Full Sim
CNS [16]	$\mathcal{O}(\lambda(N + U))$	$\mathcal{O}(\lambda)$	q-type	Full Sim
GH08 [34]	$\mathcal{O}(\lambda(N + U))$	$\mathcal{O}(\lambda)$	DLIN + q-type	UC
JL [40]	$\mathcal{O}(\lambda(N + U))$	$\mathcal{O}(\lambda)$	Comp. Dec. Residuosity + q-type	Full Sim
GH11 [35]	$\mathcal{O}(\lambda(N + U))$	$\mathcal{O}(\lambda)$	Decision 3-Party DH	Full Sim
GH11 [35]	$\mathcal{O}(\lambda N)$	$\mathcal{O}(\lambda)$	3-Party DDH + DLIN	Full Sim
Ours, §2.3	$\mathcal{O}(\lambda(N \cdot U))$	$\mathcal{O}(\lambda \cdot \log N)$	LWE + SIS	Full Sim
Ours, App D	$\mathcal{O}(\lambda N)$	$\mathcal{O}(\lambda \cdot \log N)$	LWE + SIS	Full Sim (ROM)

Table 2. This table gives an overview of adaptive OT protocols (without access control). In this table, λ stands for the security parameter; N denotes the size of the database and U is the number of receivers. The horizontal lines classify the schemes into categories based on their communication complexities. We note that, like the protocols of [43], the one of [42] is secure in a strictly weaker model than ours. In particular, the sender detects if the same record is obtained twice, as pointed out in [35].

In this section, Table 2 and Table 3 give comparisons between existing adaptive oblivious transfer protocols and ours. These results are to be taken carefully, as earlier constructions are mostly designed in the pairing-based cryptography setting. The communication complexities thus take into account the number of underlying mathematical objects exchanged during each interactive protocols, which are group elements in the previous constructions, and vectors in our case.

Table 3 considers protocols that provide access control. They manage access policies in the fashion of Camenisch *et al.* [14] in that they model *access policies* as access categories bound to users (like their role, or their permission), which are certified by the issuer. In [14], a given message in the database is accessible to receivers holding a credential for a *conjunction* of access categories: namely, to access a specific file, a user has to be in *all* the categories the file is associated with. To handle disjunctions, [14] suggests to simply duplicate the file. However, this requires the database to provide evidence that encryptions of a duplicated file are consistent and indeed encrypt the same message. Moreover, this approach does not efficiently extend to, e.g., threshold policies.

By handling access control via branching programs, we directly enable access control for policies in $\mathsf{NC}1$.

Protocol	Initialization Cost	Transfer Cost	Assumptions	Policies	Private Policies	Security
CDN [14]	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda) \cdot \mathsf{Poly}\,(\lambda)$	q-type	Conj.	✗	Full Sim
CDNZ [15]	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda) \cdot \mathsf{Poly}\,(\lambda)$	q-type + XDDH	Conj.	✓	Full Sim
ACDN [1]	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda) \cdot \mathsf{Poly}\,(\lambda)$	DLIN + SXDH	Conj.	✗	UC
ZAW+ [62]	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda)$	CP-ABE + q-type	$\mathsf{NC1}$	✗	Full-Sim
CDEN [13]	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda \log N) + \mathsf{Poly}\,(\lambda)$	CP-ABE + GGM	CNF$^-$	✓	Full-Sim
Ours, §5	$\mathcal{O}(\lambda \cdot N)$	$\widetilde{\mathcal{O}}(\lambda \log N) + \mathsf{Poly}\,(\lambda)$	LWE + SIS	$\mathsf{NC1}$	✗	Full Sim

Table 3. Overview of the different adaptive OT-AC protocols secure in the standard model. Here N denotes the size of the database. The polynomial $\mathsf{Poly}\,(\lambda)$ in transfer costs captures the complexity incurred by the treatment of access policies. In [13], GGM stands for "generic group model" while CNF$^-$ refers to restricted CNF formulas: each clause is restricted to have its literals from the same category (e.g., if categories are `Job Title` = {Professor, Post-Doc, Ph.D. Student} and `Gender` = {Male, Female}, a policy can be "(Professor ∨ Post-doc) ∧ (Male ∨ Female)", but cannot contain a clause of the form (Professor ∨ Female) or two clauses containing litterals from the `Job Title` category). Finally, "Conj." means "Conjunctions", meaning that the user has to possess all the credentials for a given message, and disjunctions can be achieved by duplicating database entries.

B Security Definitions for Adaptive OT and OT-AC

B.1 Security Definitions for Adaptive k-out-of-N Oblivious Transfer

Security is defined via the "real-world/ideal-world" paradigm which was first introduced in the Universal Composability (UC) framework [17]. Like [16,14], however, we do not incorporate all the formalities of the UC framework. We define two experiments: the **Real** experiment, where the two parties run the actual protocol, and the **Ideal** experiment wherein a *trusted third party* assumes the role of the functionality.

The model of [16] formalizes two security notions called *sender security* and *receiver security*. The former considers the security of honest senders against cheating senders whereas the latter considers the security of honest receivers interacting with malicious senders.

For an adaptive OT protocol $\mathcal{OT}_k^N$ comprised of algorithms $(\mathsf{S_I}, \mathsf{S_T}, \mathsf{R_I}, \mathsf{R_T})$, we denote define the honest sender S as the algorithm that runs $\mathsf{S_I}(M_1, \ldots, M_N)$ during the initialization phase, runs $\mathsf{S_T}$ at each transfer and eventually returns $S_k = \epsilon$ as its final output. Similarly, the honest receiver R is the algorithm that runs $\mathsf{R_I}$ in the initialization phase, runs $\mathsf{R_T}(R_{i-1}, \rho_i)$ during the i-th transfer and eventually returns $R_k = (M'_{\rho_1}, \ldots, M'_{\rho_k})$ as its final output.

Real Experiment. Here, a sender $\widehat{\mathsf{S}}$ and a receiver $\widehat{\mathsf{R}}$ which proceed as follows for experiment $\mathbf{Real}_{\widehat{\mathsf{S}},\widehat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$. The sender $\widehat{\mathsf{S}}$ is given messages $M_1, \ldots, M_N$ and interacts with $\widehat{\mathsf{R}}$ which does not have any input in the initialization phase. At end of the latter, $\widehat{\mathsf{S}}$ and $\widehat{\mathsf{R}}$ output their initial states S_0 and R_0 respectively. Then, $\widehat{\mathsf{S}}$ and $\widehat{\mathsf{R}}$ start k sequential interactions: for $i \in [k]$, in the i-th transfer, the sender $\widehat{\mathsf{S}}$ and the receiver $\widehat{\mathsf{R}}$ run $S_i \leftarrow \widehat{\mathsf{S}}(S_{i-1})$

and $(R_i, M'_{\rho_i}) \leftarrow \widehat{\mathsf{R}}(R_{i-1}, \rho_i)$, where $\rho_i \in [N]$ is a message index and (S_i, R_i) denote updated states for $\widehat{\mathsf{S}}$ and $\widehat{\mathsf{R}}$, respectively. Note that M'_{ρ_i} may be different from M_{ρ_i} if one of the participant deviates from the protocol. At the end of the k-th interaction, $\widehat{\mathsf{S}}$ and $\widehat{\mathsf{R}}$ output strings S_k and R_k respectively. The output of $\mathbf{Real}_{\widehat{\mathsf{S}},\widehat{\mathsf{R}}}$ is the pair (S_k, R_k).

The honest sender S is the algorithm that runs $\mathsf{S}(M_1, \ldots, M_N)$ as in the initialization phase, runs S_T in all subsequent interactions and always outputs $S_k = \varepsilon$. The honest receiver R is the algorithm that runs R_I in the initialization phase, runs $\mathsf{R}_\mathsf{T}(\mathsf{R}_{i-1}, \rho_i)$ at the i-th transfer and returns the list of received messages $R_k = (M'_{\rho_1}, \ldots, M'_{\rho_k})$ as its final output.

Ideal Experiment. The experiment $\mathbf{Ideal}_{\widehat{\mathsf{S}}',\widehat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$ is defined as follows. The (possibly malicious) algorithm $\widehat{\mathsf{S}}'(M_1, \ldots, M_N)$ generates messages $M'_1, \ldots, M'_N$ which are given to the trusted party T. In each of the k transfers, T obtains a bit b_i from the sender $\widehat{\mathsf{S}}'$ and an index ρ'_i from the (possibly malicious) receiver $\widehat{\mathsf{R}}'(\rho_i)$. If $b_i = 1$, and $\rho'_i \in [N]$, then T reveals M'_{ρ_i} to the receiver $\widehat{\mathsf{R}}'$. Otherwise, $\widehat{\mathsf{R}}'$ receives $\perp$ from T. At the end of the k-th transfer, $\widehat{\mathsf{S}}'$ and $\widehat{\mathsf{R}}'$ output a string S_k and R_k and the output of the experiment is the pair (S_k, R_k).

The ideal sender $\mathsf{S}'(M_1, \ldots, M_N)$ is defined the be the sender that sends $(M_1, \ldots, M_N)$ which sends the messages $(M_1, \ldots, M_N)$ to T in the initialization phase, sends $b_i = 1$ in each transfer and outputs the final state $S_k = \varepsilon$. The honest ideal receiver R' is defined to be the algorithm that sends T the real selection index ρ_i at each transfer and eventually outputs the list of all received messages $R_k = (M'_{\rho_1}, \ldots, M'_{\rho_k})$ as its final state.

The bit b_i sent by $\widehat{\mathsf{S}}'$ at each transfer models its capability of making the transfer fail. By forcing $\widehat{\mathsf{S}}'$ to choose b_i without seeing ρ_i, the definition prevents the cheating sender $\widehat{\mathsf{S}}'$ from deciding to cause a failure of the transfer for specific values of ρ_i.

Definition 4 (Sender Security). *An $\mathcal{OT}_k^N$ protocol is sender-secure if, for any PPT real-world cheating receiver $\widehat{\mathsf{R}}$, there exists a PPT ideal-world receiver $\widehat{\mathsf{R}}'$ such that, for any polynomial $N_m(\lambda)$, any $N \in [N_m(\lambda)]$, any $k \in [N]$, any messages $M_1, \ldots, M_N$, and any indices $\rho_1, \ldots, \rho_k \in [N]$, no PPT distinguisher can separate the two following distributions with noticeable advantage:*

$$\mathbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$$

and

$$\mathbf{Ideal}_{\mathsf{S}',\widehat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k).$$

Definition 5 (Receiver Security). *An $\mathcal{OT}_k^N$ protocol is receiver-secure if, for any PPT real-world cheating sender $\widehat{\mathsf{S}}$, there exists a PPT ideal-world sender $\widehat{\mathsf{S}}'$ such that, for any polynomial $N_m(\lambda)$, any $N \in [N_m(\lambda)]$, any $k \in [N]$, any* *messages $M_1, \ldots, M_N$, and any indices $\rho_1, \ldots, \rho_k \in [N]$, no PPT distinguisher can tell apart the two following distributions with non-negligible advantage:*

$$\mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$$

and

$$\mathbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k).$$

B.2 Security Definition for Adaptive Oblivious Transfer with Access Control

Similarly to the $\mathcal{OT}^N_{k\times 1}$ case, security is defined by requiring that any PPT real-world adversary $\mathcal{A}$ and any environment $\mathcal{E}$, there exists a PPT adversary $\mathcal{A}'$ which controls the same parties and such that no environment $\mathcal{E}$ can tell if it is running in the real world interacting with the real $\mathcal{A}$ or in the ideal-world interacting with $\mathcal{A}'$. The distribution of outputs of the environment in the different settings is denoted by $\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)$ and $\mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}(\lambda)$ for real-world adversary $\mathcal{A}$ and ideal-world adversary $\mathcal{A}'$, respectively.

Definition 6. *An OT-AC protocol is said to securely implement the functionality if for any real-world adversary $\mathcal{A}$ and any real world environment $\mathcal{E}$, there exists an ideal-world simulator $\mathcal{A}'$ controlling the same parties in the ideal-world as $\mathcal{A}$ does in the real-world, such that*

$$|\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda) - \mathbf{Ideal}_{\mathcal{E},\mathcal{A}}(\lambda)| = \mathsf{negl}(\lambda).$$

Real World. We describe the way that real-world algorithms interact when all participants (i.e., the real-world users $\mathsf{U}_1, \ldots, \mathsf{U}_U$, the database DB and the issuer I) are honest. The issuer starts by generating a key pair $(PK_I, SK_I) \leftarrow \mathsf{ISetup}(\mathsf{pp})$, and sends PK_I to all users $\{\mathsf{U}_i\}_{i=1}^U$ and the database DB.

When $\mathcal{E}$ sends a message $\big(\mathtt{initdb}, \mathrm{DB} = (M_i, \mathsf{AP}_i)_{i=1}^N\big)$ to the database DB, the latter encrypts the database DB by running $\mathsf{DBSetup}$ and sends the encrypted records to all users.

When $\mathcal{E}$ sends a message $(\mathtt{issue}, x)$ to user U_i, this user starts an Issue protocol with the issuer on common input x, at the end of which it returns 1 to the environment if the protocol succeeded or 0 otherwise.

When $\mathcal{E}$ sends a message $(\mathtt{transfer}, \rho)$ to user U_i, this user first checks if its credentials Cred_U are sufficient to access the record M_ρ. If it is the case, it engages in a $\mathsf{Transfer}$ protocol with the database DB, at the end of which it receives either the message M_ρ, or an error symbol $\perp$. If it failed at any steps, the user returns 0 to $\mathcal{E}$, or 1 if it succeeded.

Notice that in this setting, neither the database nor the issuer return any outputs to the environment.

Ideal World. In the ideal world, participants only communicate via the trusted party T which implements the functionality of the protocol. We describe how

T proceeds when receiving inputs from the ideal-world users $\{\mathsf{U}'_i\}_{i=1}^{U}$, issuer I' and database DB'. T maintains an initially empty set C_i for each user U'_i and sets $\mathrm{DB} \leftarrow \perp$. It handles the queries of the different parties as follows:

- When receiving a message $(\mathtt{initdb}, \mathrm{DB} = (M_i, \mathsf{AP}_i)_{i=1}^{N})$ from DB', T sets $\mathrm{DB} = (M_i, \mathsf{AP}_i)_{i=1}^{N}$.
- When receiving $(\mathtt{issue}, x)$ from U'_i, T sends $(\mathtt{issue}, \mathsf{U}'_i, x)$ to I' which replies with a bit b. If $b = 1$, then T adds x to C_i. In any cases, T sends b to U'_i.
- When receiving $(\mathtt{transfer}, \rho)$ from U'_i, the trusted party T acts as follows. If U'_i previously sent a message of the form $(\mathtt{transfer}, .)$, T defines $f_{\mathsf{U}',DB} = 1$. Otherwise, it sets $f_{\mathsf{U}',DB} = 0$. If $\mathrm{DB} \neq \perp$, it sends $(\mathtt{transfer}, f_{\mathsf{U}',DB})$ to DB', who sends a bit b. If $b = 1$ and if st_i contains a vector $\mathbf{x}$ such that $\mathsf{AP}_i(x) = 1$, then it sends the record to U'_i. In any other cases, it sends $\perp$ to U'_i.

In other words, the ideal-world users, database and issuer relay inputs and outputs between the environment $\mathcal{E}$ and the trusted party T.

Note that, like [14], the ideal functionality allows the database to learn whether a given user interacts with the database for the first time or not. The reason is that, like the protocol of [14], our basic OT-AC scheme requires the database to provide a particular interactive zero-knowledge proof at the very first time each user queries the database. In protocols where the database generates such an interactive proof, it is inevitable for U to reveal his state bit f_{DB} to DB. In constructions where the zero-knowledge proof is made non-interactive and made publicly available at the same time as the database itself, this can be avoided and we can prevent DB from learning the state bit f_{DB}. In this case, T does not send $f_{\mathsf{U}',DB}$ to DB' in the ideal-world experiment.

The ideal world thus implies the following security properties.

User Anonymity. The database cannot tell which user a given query comes from and neither can it tell which record is being accessed. It only learns whether the user previously queried the database or not. Otherwise, two transfers involving the same users are unlinkable.

Database Security. A single cheating user cannot access a record for which he does not have a certified authorized attribute string. Colluding users cannot pool their credentials to gain access to a record which none of them can individually access. Moreover, if the issuer colludes with some users, the protocol still provides the equivalent of sender security in the $\mathcal{OT}_{k\times 1}^{N}$ functionality.

C Deferred Proofs for the Modified Signature Scheme

C.1 Proof of Theorem 1

Proof. We show that the scheme presented in Section 3.2 is secure against non-adaptive chosen-message attacks (na-CMA) under the SIS assumption. The shape of the proof is similar to the security proof of the signature scheme of [44]. Namely, to prove the security, we distinguish two kinds of attacks:

Type I attacks, where in the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star)$, $\tau^\star$ did not appear in any outputs of the signing oracle.

Type II attacks, where in the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star)$, $\tau^\star$ has been recycled from an output $sig^{(i^\star)} = \big(\tau^{(i^\star)}, \mathbf{v}^{(i^\star)}\big)$ of the signing oracle for some query $i^\star \in \{1, \ldots, Q\}$.

Lemma 5 states that the signature scheme is secure against Type I forgery using the same technique as is [2,10,51]. Lemma 6 claims that the signature scheme resists Type II attacks, with a proof that is very similar to the one of Lemma 5. Both security proofs assume the computational hardness of the SIS problem. □

Lemma 5. *The signature scheme of Section 3.2 is secure against Type I attacks if the* $\mathsf{SIS}_{n,m,q,\beta'}$ *assumption holds, with* $\beta' = \sigma^2 m^{3/2}(\ell+2) + \sigma m^{1/2}$.

Proof. Let $\mathcal{A}$ be a PPT adversary against the $\mathsf{na\text{-}CMA}$ security of our scheme that mounts Type I attacks with non negligible success probability ϵ. We construct a PPT algorithm $\mathcal{B}$ using $\mathcal{A}$ to break the $\mathsf{SIS}_{n,m,q,\beta'}$ assumption. Our reduction $\mathcal{B}$ takes as input a target matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times m}$ and computes $\mathbf{v} \in \Lambda_q^\perp(\bar{\mathbf{A}})$ satisfying $0 < \|\mathbf{v}\| \le \beta'$.

At first, $\mathcal{B}$ calls $\mathcal{A}$ to obtain the messages to be queried: $\mathfrak{m}^{(1)}, \ldots, \mathfrak{m}^{(Q)}$. For the sake of readability, let us define $\tau^{(i)} = i$, viewed as a bit-string, to be the tag corresponding to the i-th signature in our scheme.

Setup. As in [38], the reduction guesses the shortest prefix such that the string $\tau^\star$ embedded in $\mathcal{A}$'s forgery differs from all prefixes to $\{\tau^{(1)}, \ldots, \tau^{(Q)}\}$. To achieve this, $\mathcal{B}$ chooses at random $i^\dagger \hookleftarrow U(\{1, \ldots, Q\})$ and $t^\dagger \hookleftarrow U(\{1, \ldots, \ell\})$. Then, with probability $1/(Q\cdot\ell)$, the longest common prefix between $\tau^\star$ and one of the tags $\{\tau^{(i)}\}_{i=1}^Q$ is the string $\tau^\star[1]\cdots\tau^\star[t^\dagger - 1] \in \{0,1\}^{t^\dagger - 1}$: the first $(t^\dagger-1)$-th bits of $\tau^\star$. Let us define $\tau^\dagger = \tau^\star_{|t^\dagger}$, where $s_{|i}$ denotes the i-th prefix for a string s. By construction $\tau^\dagger \notin \{\tau^{(1)}_{|t^\dagger}, \ldots, \tau^{(Q)}_{|t^\dagger}\}$ with probability $1/(Q\cdot\ell)$.

Next, the reduction $\mathcal{B}$ runs $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain matrices $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ and a short basis $\mathbf{T_C} \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^\perp(\mathbf{C})$, which will be useful to answer the following opening oracle queries. The reduction $\mathcal{B}$ continues by picking $\ell+1$ matrices $\mathbf{Q}_0, \ldots, \mathbf{Q}_\ell \in \mathbb{Z}^{m\times m}$ where each matrix $\mathbf{Q}_i$ has its column independently sampled from $D_{\mathbb{Z}^m,\sigma}$, and $\mathcal{B}$ defines the matrices $\mathbf{A} = \bar{\mathbf{A}}$ and $\{\mathbf{A}_j\}_{j=0}^{\ell}$ as follows

$$\begin{cases} \mathbf{A}_0 = \bar{\mathbf{A}}\cdot\mathbf{Q}_0 + \left(\sum_{j=1}^{t^\dagger} \tau^\star[j]\right)\cdot\mathbf{C} & \\ \mathbf{A}_j = \bar{\mathbf{A}}\cdot\mathbf{Q}_j + (-1)^{\tau^\star[j]}\cdot\mathbf{C} & \text{for } j \in [1, t^\dagger] \\ \mathbf{A}_j = \bar{\mathbf{A}}\cdot\mathbf{Q}_j & \text{for } j \in [t^\dagger+1, \ell] \end{cases} .$$

We can notice that

$$
\begin{aligned}
\mathbf{A}_{\tau^{(i)}} &= \Big[\mathbf{A} \;\Big|\; \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j]\mathbf{A}_j\Big] \\
&= \Big[\bar{\mathbf{A}} \;\Big|\; \bar{\mathbf{A}} \cdot \big(\mathbf{Q}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot \mathbf{Q}_j\big) + \big(\sum_{j=1}^{t^\dagger} \tau^\star[j] + (-1)^{\tau^\star[j]} \cdot \tau^{(i)}[j]\big) \cdot \mathbf{C}\Big] \\
&= \Big[\bar{\mathbf{A}} \;\Big|\; \bar{\mathbf{A}} \cdot \big(\mathbf{Q}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot \mathbf{Q}_j\big) + h_{\tau^{(i)}} \cdot \mathbf{C}\Big],
\end{aligned}
$$

where $h_{\tau^{(i)}}$ denotes the hamming distance between $\tau^{(i)}_{|t^\dagger}$ and $\tau^\dagger$. With probability $1/(Q \cdot \ell)$, and as $\ell > q$, it holds that $h_{\tau^{(i)}} \neq 0 \bmod q$ whenever $\tau^{(i)}_{|t^\dagger} \neq \tau^\star_{|t^\dagger}$.

The reduction then picks a random short matrix $\mathbf{R} \hookleftarrow \mathbb{Z}^{m \times m_d}$ which has its m_d columns independently sampled from $D_{\mathbb{Z}^m,\sigma}$, and $\mathcal{B}$ computes

$$\mathbf{D} = \bar{\mathbf{A}} \cdot \mathbf{R} \in \mathbb{Z}_q^{n \times m_d}.$$

To finish, $\mathcal{B}$ samples a short vector $\mathbf{e}_u \in D_{\mathbb{Z}^m,\sigma}$ and computes the vector $\mathbf{u} = \bar{\mathbf{A}} \cdot \mathbf{e}_u$. The following public key is finally given to $\mathcal{A}$:

$$PK := (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{u}).$$

Signing queries. To handle signature queries, the reduction $\mathcal{B}$ uses the trapdoor $\mathbf{T}_{\mathbf{C}} \in \mathbb{Z}^{m \times m}$ to generate a signature. To this end, $\mathcal{B}$ starts by computing the vector $\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathfrak{m}^{(i)}$. Then $\mathcal{B}$ can use $\mathbf{T}_{\mathbf{C}}$ with the algorithm $\mathsf{SampleRight}$ from Lemma 4 to compute a short vector $\mathbf{v}^{(i)}$ in $D^{\mathbf{u}_M}_{\Lambda^\perp(\mathbf{A}_{\tau^{(i)}}),\sigma}$, distributed like a valid signature and satisfying the verification equation (2).

Output. At some point, the attacker $\mathcal{A}$ halts and outputs a *valid* signature $sig^\star = (\tau^\star, \mathbf{v}^\star)$ for a message $\mathfrak{m}^\star \notin \{\mathfrak{m}^{(1)}, \ldots, \mathfrak{m}^{(Q)}\}$. Since the signature is valid, it satisfies $\|\mathbf{v}^\star\| \leq \sigma\sqrt{2m}$.
Parsing $\mathbf{v}^\star$ as $[\mathbf{v}_1^\star \mid \mathbf{v}_2^\star]$ with $\mathbf{v}_1^\star, \mathbf{v}_2^\star \in \mathbb{Z}^m$ and injecting it in (2) give:

$$
\begin{aligned}
\Big[\bar{\mathbf{A}} \;\Big|\; \bar{\mathbf{A}} \cdot \big(\mathbf{Q}_0 + \sum_{j=1}^{\ell} \tau^\star[j] \cdot \mathbf{Q}_j\big)\Big] \cdot \begin{bmatrix} \mathbf{v}_1^\star \\ \mathbf{v}_2^\star \end{bmatrix} &= \mathbf{u} + \mathbf{D} \cdot \mathfrak{m}^\star \mod q \\
&= \bar{\mathbf{A}} \cdot \big(\mathbf{e}_u + \mathbf{R} \cdot \mathfrak{m}^\star\big) \mod q
\end{aligned}
$$

Thus, the vector

$$\mathbf{v}' = \mathbf{v}_1^\star + \big(\mathbf{Q}_0 + \sum_{j=1}^{\ell} \tau^\star[j] \cdot \mathbf{Q}_j\big) \cdot \mathbf{v}_2^\star - \mathbf{e}_u - \mathbf{R} \cdot \mathfrak{m}^\star$$

is in $\Lambda^\perp(\bar{\mathbf{A}})$, and $\mathbf{v}'$ is non-zero with overwhelming probabilities, since in $\mathcal{A}$'s view, the distribution of $\mathbf{e}_u$ is $D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$, which guarantees that $\mathbf{e}_u$ is statistically hidden by the syndrome $\mathbf{u} = \bar{\mathbf{A}} \cdot \mathbf{e}_u$. Finally, the norm of $\mathbf{v}'$ is upper bounded by $\beta' = \sigma^2 m^{3/2}(\ell + 2) + 2\sigma m^{1/2}$. □

Lemma 6. *The signature scheme of Section 3.2 is secure against Type II attacks if* $\mathsf{SIS}_{n,m,q,\beta''}$ *holds, with* $\beta'' = \sqrt{2}(\ell+2)\sigma m^{3/2} + m^{1/2}$.

Proof. We will prove this result using techniques analogous to the previous proof. We show that given an adversary $\mathcal{A}$ that comes out with a Type II signature in the na-CMA game with non negligible probability ε, we can construct a PPT $\mathcal{B}$ that breaks the SIS assumption with advantage ε/Q using $\mathcal{A}$.

Firstly, the reduction $\mathcal{B}$ is given a matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m_d}$ as input and has to output an integer vector $\mathbf{v} \in \mathbb{Z}^{m_d}$ in $\Lambda_q^{\perp}(\bar{\mathbf{A}})$ such that $0 < \|\mathbf{v}\| \leq \beta''$. Next, $\mathcal{B}$ receives from $\mathcal{A}$ the messages $\mathfrak{m}^{(1)}, \ldots, \mathfrak{m}^{(Q)}$ for which $\mathcal{A}$ will further ask signature queries.

To compute the public key, at the outset of the game, the reduction $\mathcal{B}$ starts by sampling $i^{\dagger} \hookleftarrow U(\{1, \ldots, Q\})$ corresponding to the guess that $\mathcal{A}$'s forgery will recycle $\tau^{(i\dagger)}$. This is independent of $\mathcal{A}$'s view, and the guess will be correct with probability $1/Q$. Using this guess to compute PK, the reduction $\mathcal{B}$ picks $h_0, \ldots, h_\ell \in \mathbb{Z}_q$ subject to the constraints

$$\begin{cases} h_0 + \sum_{j=1}^{\ell} \tau^{(i^{\dagger})}[j] \cdot h_j = 0 \mod q \\ h_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot h_j \neq 0 \mod q \quad \forall i \in \{1, \ldots, Q\} \backslash \{i^{\dagger}\} \end{cases} \tag{32}$$

$\mathcal{B}$ then runs $(\mathbf{C}, \mathbf{T_C}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. The resulting matrix $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ is statistically random, and the trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ is a short basis of $\Lambda_q^{\perp}(\mathbf{C})$. Next $\mathcal{B}$ re-randomize $\bar{\mathbf{A}}$ using short matrices $\mathbf{S}, \mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_\ell \in \mathbb{Z}^{m_d \times m}$ which are obtained by sampling their columns from the distribution $D_{\mathbb{Z}^{m_d}, \sigma}$. The challenger $\mathcal{B}$ then uses these matrices to define:

$$\begin{aligned} \mathbf{A} &= \bar{\mathbf{A}} \cdot \mathbf{S} \\ \mathbf{A}_0 &= \bar{\mathbf{A}} \cdot \mathbf{S}_0 + h_0 \cdot \mathbf{C} \\ \mathbf{A}_j &= \bar{\mathbf{A}} \cdot \mathbf{S}_j + h_j \cdot \mathbf{C} \qquad\qquad j \in \{1, \ldots, \ell\} \end{aligned}$$

and sets $\mathbf{D} = \bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m_d}$. Observe that matrices $\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}$ are all statistically uniform over $\mathbb{Z}_q^{n \times m}$. Then, $\mathcal{B}$ samples short vectors $\mathbf{v}_1^{\dagger}, \mathbf{v}_2^{\dagger} \hookleftarrow D_{\mathbb{Z}^m, \sigma}$ and computes $\mathbf{u} \in \mathbb{Z}_q^n$ as

$$\mathbf{u} = \mathbf{A}_{\tau^{(i^{\dagger})}} \cdot \begin{bmatrix} \mathbf{v}_1^{\dagger} \\ \hline \mathbf{v}_2^{\dagger} \end{bmatrix} - \bar{\mathbf{A}} \cdot \mathfrak{m}^{(i^{\dagger})} \mod q. \tag{33}$$

Finally, $\mathcal{B}$ sends to $\mathcal{A}$ the public key

$$PK := \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{u}\big)$$

which is distributed as the PK of the real scheme.

To answer signing queries, the challenger $\mathcal{B}$ do as follows.

– If the query is not the $i^\dagger$-th, we have:

$$\begin{aligned}\mathbf{A}_{\tau^{(i)}} &= \Big[\mathbf{A} \;\Big|\; \mathbf{A}_0 + \sum_{j=0}^{\ell} \tau^{(i)}[j]\cdot \mathbf{A}_j\Big] \\ &= \Big[\bar{\mathbf{A}}\cdot \mathbf{S} \;\Big|\; \bar{\mathbf{A}}\cdot(\mathbf{S}_0 + \sum_{j=0}^{\ell}\tau^{(i)}[j]\cdot \mathbf{S}_j) + h_{\tau^{(i)}}\cdot \mathbf{C}\Big],\end{aligned}$$

with $h_{\tau^{(i)}} = h_0 + \sum \tau^{(i)}[j]\cdot h_j \neq 0$ due to the first constraint of (32). Thus, using the same technique as in the previous proof from [51], the challenger $\mathcal{B}$ can use the trapdoor $\mathbf{T_C}$ along with $\mathsf{SampleRight}$ algorithm to sample a short vector in $\mathit{\Lambda}_q^{\mathbf{u}_M}(\mathbf{A}_{\tau^{(i)}})$ satisfying (2).

– At the $i^\dagger$-th query, thanks to the second constraint of (32), we have:

$$\begin{aligned}\mathbf{A}_{\tau^{(i^\dagger)}} &= \Big[\mathbf{A} \;\Big|\; \mathbf{A}_0 + \sum_{j=0}^{\ell} \tau^{(i^\dagger)}[j]\cdot \mathbf{A}_j\Big] \\ &= \Big[\bar{\mathbf{A}}\cdot \mathbf{S} \;\Big|\; \bar{\mathbf{A}}\cdot(\mathbf{S}_0 + \sum_{j=0}^{\ell}\tau^{(i^\dagger)}[j]\cdot \mathbf{S}_j)\Big].\end{aligned}$$

To answer this specific query, the challenger $\mathcal{B}$ returns $sig^{(i^\dagger)} = (\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)})$ where $\mathbf{v}^{(i^\dagger)} = (\mathbf{v}_1^{\dagger T} \mid \mathbf{v}_2^{\dagger T})^\top$ verifying (33), which furthermore implies that $sig^{(i^\dagger)}$ verifies (2).

Thus we claim that $\mathcal{B}$ can solve the SIS problem using the Type II forgery provided by $\mathcal{A}$. At the end of the game, the adversary outputs a valid signature $sig^\star = (\tau^{(i^\star)}, \mathbf{v}^\star)$ on a message $\mathfrak{m}^\star$ with $\|\mathbf{v}^\star\| \leq \sigma\sqrt{2m}$. In the event that $\tau^{(i^\star)} \neq \tau^{i^\dagger}$, the reduction aborts. The latter event happens with probability $1 - 1/Q$. If we parse $\mathbf{v}^\star$ as $(\mathbf{v}_1^{\star,T} \mid \mathbf{v}_2^{\star T})^\top \in \mathbb{Z}^{2m}$, with $\mathbf{v}_1^\star, \mathbf{v}_2^\star \in \mathbb{Z}^m$, it holds that:

$$\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \left[\begin{array}{c}\mathbf{v}_1^\star \\ \hline \mathbf{v}_2^\star\end{array}\right] = \mathbf{u} + \bar{\mathbf{A}}\cdot \mathfrak{m}^\star \mod q. \tag{34}$$

According to the way $\mathbf{u}$ was defined at the beginning of the game, we also have a vector $\mathbf{v}^\dagger = (\mathbf{v}_1^{\dagger T} \mid \mathbf{v}_2^{\dagger T})^\top$ such that

$$\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \left[\begin{array}{c}\mathbf{v}_1^\dagger \\ \hline \mathbf{v}_2^\dagger\end{array}\right] = \mathbf{u} + \bar{\mathbf{A}}\cdot \mathfrak{m}^\dagger \mod q. \tag{35}$$

As $sig^\star$ is a valid forgery for the dn-CMA game, it follows that $m^\dagger \neq m^\star$. And we get by subtracting (34) and (35)

$$\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \left[\begin{array}{c}\mathbf{v}_1^\star - \mathbf{v}_1^\dagger \\ \hline \mathbf{v}_2^\star - \mathbf{v}_2^\dagger\end{array}\right] = \bar{\mathbf{A}}\cdot\big(\mathfrak{m}^\star - \mathfrak{m}^\dagger\big) \mod q,$$

$$\Big[\bar{\mathbf{A}}\cdot \mathbf{S} \;\Big|\; \bar{\mathbf{A}}\cdot(\mathbf{S}_0 + \sum_{j=0}^{\ell}\tau^{(i^\dagger)}[j]\cdot \mathbf{S}_j)\Big] \cdot \left[\begin{array}{c}\mathbf{v}_1^\star - \mathbf{v}_1^\dagger \\ \hline \mathbf{v}_2^\star - \mathbf{v}_2^\dagger\end{array}\right] = \bar{\mathbf{A}}\cdot\big(\mathfrak{m}^\star - \mathfrak{m}^\dagger\big) \mod q.$$

Leading us to the fact that

$$\mathbf{v}' = \underbrace{\mathbf{S} \cdot (\mathbf{v}_1^{\star} - \mathbf{v}_2^{\dagger}) + \left(\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i^{\dagger})}[j] \cdot \mathbf{S}_j\right) \cdot (\mathbf{v}_2^{\star} - \mathbf{v}_2^{\dagger})}_{(a)} + \underbrace{\mathfrak{m}^{\dagger} - \mathfrak{m}^{\star}}_{-(b)} \qquad (36)$$

is an integer vector of $\Lambda_q^{\perp}(\bar{\mathbf{A}})$, with norm bounded by $\|\mathbf{v}'\| \leq \sqrt{2}(\ell + 2)\sigma m^{3/2} + m^{1/2} = \beta''$. Furthermore, if $\mathbf{v}'$ was zero, it implies that $(a) = (b)$ in Equation (36). And as $sig^{\star} \neq sig^{\dagger}$, we have that either $\mathbf{v}_1^{\star} \neq \mathbf{v}_1^{\dagger}$ or $\mathbf{v}_2^{\star} \neq \mathbf{v}_2^{\dagger}$. As a consequence, (a) is information theoretically unpredictable for $\mathcal{A}$ since the columns of $\mathbf{S}, \mathbf{S}_0, \ldots \mathbf{S}_{\ell}$ are statistically hidden from $\mathcal{A}$, as shown in [51] for instance: conditionally on the public key, each column of $\mathbf{S}$ and $\{\mathbf{S}_j\}_{j=0}^{\ell}$ has at least n bits of min-entropy. □

D Reducing the Communication Complexity in the Random Oracle Model

One limitation of our basic adaptive OT protocol is that it requires the sender to repeat the zero-knowledge proofs of the initialization phase for each user. In total, the communication cost of the initialization phase thus amounts to $\Omega(\lambda N U)$, which is even more expensive than the $O(\lambda(N + U))$ complexities of [16,33,14,40]. As pointed out by Green and Hohenberger [35], decreasing the cost of the initialization phase to be independent of the number of users is highly desirable: ideally, one would certainly prefer a non-interactive initialization phase where the Sender can publicize a $O(\lambda N)$-size commitment to the database, which can subsequently be used by arbitrarily many receivers.

In the random oracle model, we show that our protocols can both be modified to obtain this optimized communication complexity. This can be achieve by the simple expedient of making the sender's zero-knowledge proofs non-interactive via the Fiat-Shamir heuristic.

By removing interaction from *all* the sender's proof (i.e., even those of the transfer phase), we also minimize the number of communication rounds since we only need the verifier's arguments to be witness indistinguishable and we can thus safely repeat them in parallel. Relying on the random oracle model thus allows the sender to publicize the entire database and proofs on a public repository so as to avoid repeating these proofs for each receiver.

D.1 Description

The description hereunder relies on the same parameters as in sections 4 and 5. Namely, we use $m = 2n\lceil \log q \rceil$, a modulus q for which the noise distribution χ is αq-bounded, for some $0 < \alpha < 1$, and also define an integer B as a randomization parameter such that $(m + 1)\alpha q/B$ is negligible and choosing α such that $(m + 1)\alpha \leq 1/5$ ensures decryption correctness.

We assume two random oracles $H_F : \{0,1\}^* \to \mathbb{Z}_q^{n\times m}$ and $H_{\mathsf{FS}} : \{0,1\}^* \to \{1,2,3\}^\varsigma$, for some $\varsigma \in \omega(\log n)$. The former will be used to derive the sender's public matrix $\mathbf{F} \in \mathbb{Z}_q^{n\times m}$ while the latter will provide the verifier's challenges when we apply the Fiat-Shamir heuristic.

Initialization$\big(\mathsf{S_I}(1^\lambda, \mathrm{DB}), \mathsf{R_I}(1^\lambda)\big)$**:** In this protocol, the sender $\mathsf{S_I}$ has a database $\mathrm{DB} = (M_1, \ldots, M_N)$ of N messages, where $M_i \in \{0,1\}^t$ for each $i \in [N]$, for some $t \in \mathsf{poly}(\lambda)$. It interacts with the receiver $\mathsf{R_I}$ as follows.

1. Generate a key pair for the signature scheme of Section 3.2 in order to sign $Q = N$ messages of length $m_d = (n+t)\cdot\lceil\log q\rceil$ each. This key pair consists of $SK_{sig} = \mathbf{T_A} \in \mathbb{Z}^{m\times m}$ and
$$PK_{sig} := \big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{u}\big),$$
where $\ell = \log N$ and $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell \in U(\mathbb{Z}_q^{n\times m})$, $\mathbf{D} \in U(\mathbb{Z}_q^{n\times m_d})$ with $m = 2n\lceil\log q\rceil$, $m_d = (n+t)\lceil\log q\rceil$. The counter is initialized to $\tau = 0$.
2. Choose a matrix $\mathbf{S} \hookleftarrow \chi^{n\times t}$ that will serve as a secret key for an LWE-based encryption scheme. Then, define the matrix $\mathbf{F} = H_F(\varepsilon) \in \mathbb{Z}_q^{n\times m}$ and sample a matrix $\mathbf{E} \hookleftarrow \chi^{m\times t}$ to compute
$$\mathbf{P} = [\mathbf{p}_1 \mid \ldots \mid \mathbf{p}_t] = \mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E} \ \in \mathbb{Z}_q^{m\times t} \tag{37}$$
so that $(\mathbf{F}, \mathbf{P}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{m\times t}$ forms a public key for a t-bit variant of Regev's encryption scheme [58] (or, equivalently, a set of m encryptions of the all-zeroes t-bit string).
3. Sample vectors $\mathbf{a}_1, \ldots, \mathbf{a}_N \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{x}_1, \ldots, \mathbf{x}_N \hookleftarrow \chi^t$ to compute
$$(\mathbf{a}_i, \mathbf{b}_i) = \big(\mathbf{a}_i,\ \mathbf{S}^\top \cdot \mathbf{a}_i + \mathbf{x}_i + M_i \cdot \lfloor q/2 \rfloor\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t \qquad \forall i \in [N] \tag{38}$$
4. For each $i = 1$ to N, generate a signature $(\tau_i, \mathbf{v}_i) \leftarrow \mathsf{Sign}(SK_{sig}, \tau, \mathfrak{m}_i)$ on the message $\mathfrak{m}_i = \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_i|\mathbf{b}_i) \in \{0,1\}^{m_d}$ obtained by decomposing $(\mathbf{a}_i^\top|\mathbf{b}_i^\top)^\top \in \mathbb{Z}_q^{n+t}$.
5. $\mathsf{S_I}$ sends $\mathsf{R_I}$ the initialization data
$$R_0 = \big(PK_{sig},\ (\mathbf{F}, \mathbf{P}),\ \{(\mathbf{a}_i, \mathbf{b}_i), (\tau_i, \mathbf{v}_i)\}_{i=1}^N,\ \pi_K\big), \tag{39}$$
which includes a NIZK argument of knowledge π_K of small-norm matrices $\mathbf{S} \in \mathbb{Z}^{n\times t}$ and $\mathbf{E} \in \chi^{m\times t}$ and t-bit messages $\{M_i\}_{i=1}^N$ that are consistent with (37)-(38). The argument π_K is built by taking the following steps:

   a. Define $\mathbf{A}_{\mathrm{DB}} = [\mathbf{a}_1|\ldots|\mathbf{a}_N] \in \mathbb{Z}_q^{n\times N}$, $\mathbf{B}_{\mathrm{DB}} = [\mathbf{b}_1|\ldots|\mathbf{b}_N] \in \mathbb{Z}_q^{t\times N}$, $\mathbf{M} = [M_1|\ldots|M_N] \in \{0,1\}^{t\times N}$, $\mathbf{X} = [\mathbf{x}_1|\ldots|\mathbf{x}_N] \in \chi^{t\times N}$ and parse $\mathbf{S}$ and $\mathbf{E}$ as $\mathbf{S} = [\mathbf{s}_1|\ldots|\mathbf{s}_t] \in \chi^{n\times t}$, $\mathbf{E} = [\mathbf{e}_1|\ldots|\mathbf{e}_t] \in \chi^{m\times t}$.

   b. For each $j \in [t]$, define $\bar{M}_j \in \{0,1\}^N$ to be the j-th column of $\mathbf{M}^\top = [\bar{M}_1|\ldots|\bar{M}_t]$. Likewise, let $\bar{\mathbf{b}}_j \in \mathbb{Z}_q^N$ (resp. $\bar{\mathbf{x}}_j \in \chi^N$) be the j-th column of $\mathbf{B}_{\mathrm{DB}}^\top = [\bar{\mathbf{b}}_1|\ldots|\bar{\mathbf{b}}_t] \in \mathbb{Z}_q^{N\times t}$ (resp. $\mathbf{X}^\top = [\bar{\mathbf{x}}_1|\ldots|\bar{\mathbf{x}}_t]$)

and generate a signature of knowledge of $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$, $\bar{\mathbf{x}}_j \in \chi^N$, $\bar{M}_j \in \{0,1\}^N$, for $j \in [t]$, such that

$$\left[\begin{array}{c|c|c|c} \mathbf{F}^\top & \mathbf{I}_m & & \\ \hline \mathbf{A}_{\mathrm{DB}}^\top & & \mathbf{I}_N & \lfloor q/2 \rfloor \cdot \mathbf{I}_N \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{s}_j \\ \hline \mathbf{e}_j \\ \hline \bar{\mathbf{x}}_j \\ \hline \bar{M}_j \end{array}\right] = \left[\begin{array}{c} \mathbf{p}_j \\ \hline \mathbf{b}_j \end{array}\right] \tag{40}$$

Let the NIZK proof be $\pi_K = (\{\mathsf{Comm}_{K,j}\}_{j=1}^{\varsigma}, \mathsf{Chall}_K, \{\mathsf{Resp}_{K,j}\}_{j=1}^{\varsigma})$, where $\mathsf{Chall}_K = H_{\mathsf{FS}}\big((\mathbf{F}, \mathbf{P}, \mathbf{A}_{\mathrm{DB}}, \mathbf{B}_{\mathrm{DB}}), \{\mathsf{Comm}_{K,j}\}_{j=1}^{\varsigma}\big) \in \{1,2,3\}^{\varsigma}$.

c. If the proof of knowledge π_K does not verify or if there exists $i \in [N]$ such that $(\tau_i, \mathbf{v}_i)$ is an invalid signature on $\mathsf{vdec}_{n+t,q-1}\big((\mathbf{a}_i^\top | \mathbf{b}_i^\top)^\top\big)^\top$, then R_I aborts.

6. Finally S_I defines $S_0 = \big((\mathbf{S}, \mathbf{E}), (\mathbf{F}, \mathbf{P}), PK_{sig}\big)$, which it keeps to itself.

Transfer$\big(\mathsf{S}_\mathsf{T}(S_{i-1}), \mathsf{R}_\mathsf{T}(R_{i-1}, \rho_i)\big)$**:** At the i-th transfer, the receiver R_T has state R_{i-1} and an index $\rho_i \in [1, N]$. It interacts as follows with the sender S_T that has state S_{i-1} in order to obtain M_{ρ_i} from DB.

1. R_T samples vectors $\mathbf{e} \hookleftarrow U(\{-1,0,1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and a random $\nu \hookleftarrow U([-B,B]^t)$ to compute

$$(\mathbf{c}_0, \mathbf{c}_1) = \big(\mathbf{a}_{\rho_i} + \mathbf{F} \cdot \mathbf{e},\ \mathbf{b}_{\rho_i} + \mathbf{P}^\top \cdot \mathbf{e} + \mu \cdot \lfloor q/2 \rfloor + \nu\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t, \tag{41}$$

which is a re-randomization of $(\mathbf{a}_{\rho_i}, \mathbf{b}_{\rho_i} + \mu \cdot \lfloor q/2 \rfloor)$. The resulting ciphertext $(\mathbf{c}_0, \mathbf{c}_1)$ is sent to S_T. In addition, R_T provides an interactive WI argument that $(\mathbf{c}_0, \mathbf{c}_1)$ is indeed a re-randomization of $(\mathbf{a}_{\rho_i}, \mathbf{b}_{\rho_i})$ for some index $\rho_i \in [N]$. To this end, R_T argues knowledge of short vectors $\mathfrak{m} = \mathsf{vdec}_{n+1,q-1}(\mathbf{a}_i | \mathbf{b}_i) \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B,B]^t$, $\tau \in \{0,1\}^\ell$ and $\mathbf{v} = (\mathbf{v}_1^\top | \mathbf{v}_2^\top)^\top \in \mathbb{Z}^{2m}$ such that

$$\left[\begin{array}{c|c|c|c} \mathbf{H}_{n,q-1} \qquad\qquad & \mathbf{F} & & \\ \hline \qquad\qquad \mathbf{H}_{t,q-1} & \mathbf{P}^\top & \mathbf{I}_t \cdot \lfloor q/2 \rfloor & \mathbf{I}_t \end{array}\right] \cdot \left[\begin{array}{c} \mathfrak{m} \\ \hline \mathbf{e} \\ \hline \mu \\ \hline \nu \end{array}\right] = \left[\begin{array}{c} \mathbf{c}_0 \\ \hline \mathbf{c}_1 \end{array}\right] \tag{42}$$

and

$$\left[\, \mathbf{A} \mid \mathbf{A}_0 \mid \cdots \mid \mathbf{A}_\ell \,\right] \cdot \left[\begin{array}{c} \mathbf{v}_1 \\ \hline \mathbf{v}_2 \\ \hline \tau[1] \cdot \mathbf{v}_2 \\ \hline \vdots \\ \hline \tau[\ell] \cdot \mathbf{v}_2 \end{array}\right] = \mathbf{u} + \mathbf{D} \cdot \mathfrak{m} \mod q \tag{43}$$

2. If the WI argument of step 1 verifies, S_T uses $\mathbf{S} \in \chi^{n \times t}$ to decrypt $(\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ and obtain

$$M' = \lfloor (\mathbf{c}_1 - \mathbf{S}^\top \cdot \mathbf{c}_0)/(q/2) \rceil \in \{0,1\}^t,$$

which is sent back to $\mathsf{R_T}$. In addition, $\mathsf{S_T}$ provides a NIZK argument π_T of knowledge of $\mathbf{y} = \mathbf{c}_1 - \mathbf{S}^\top \cdot \mathbf{c}_0 - M' \cdot \lfloor q/2 \rfloor \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \leq q/5$ and $\mathbf{E} = [\mathbf{e}_1 | \ldots | \mathbf{e}_t] \in \chi^{m \times t}$ satisfying (modulo q)

$$\mathbf{P} = \mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E} \ , \qquad \mathbf{c}_0^\top \cdot \mathbf{S} + \mathbf{y}^\top = \mathbf{c}_1^\top - M'^\top \cdot \lfloor q/2 \rfloor . \tag{44}$$

Given $\mathbf{y} = (\mathbf{y}[1], \ldots, \mathbf{y}[t])^\top \in \mathbb{Z}^t$ and $\mathbf{S} = [\mathbf{s}_1 | \ldots | \mathbf{s}_t]$, this amounts to proving, for each $j \in [t]$, knowledge of $\mathbf{s}_j \in \chi^n$, $\mathbf{y}[j] \in \mathbb{Z}$ such that $|\mathbf{y}[j]| < q/4$ and $\mathbf{e}_j \in \chi^m$, such that

$$\left[\begin{array}{c|c|c} \mathbf{F}^\top & \mathbf{I}_m & \\ \hline \mathbf{c}_0^\top & & 1 \end{array} \right] \cdot \left(\begin{array}{c} \mathbf{s}_j \\ \hline \mathbf{e}_j \\ \hline \mathbf{y}[j] \end{array} \right) = \left(\begin{array}{c} \mathbf{p}_j \\ \hline \mathbf{c}_1[j] - M'[j] \cdot \lfloor q/2 \rfloor \end{array} \right) \qquad \forall j \in [t], \tag{45}$$

where $\mathbf{c}_1 = (\mathbf{c}_1[1], \ldots, \mathbf{c}_1[t])^\top$ and $M' = (M'[1], \ldots, M'[t])^\top$. Let the NIZK argument be $\pi_T = (\{\mathsf{Comm}_{T,j}\}_{j=1}^{\varsigma}, \mathsf{Chall}_T, \{\mathsf{Resp}_{T,j}\}_{j=1}^{\varsigma})$, where $\mathsf{Chall}_T = H_{\mathsf{FS}}\big((\mathbf{F}, \mathbf{P}, \mathbf{c}_0, \mathbf{c}_1), \{\mathsf{Comm}_{T,j}\}_{j=1}^{\varsigma}\big) \in \{1,2,3\}^{\varsigma}$.

3. If th argument π_T produced by $\mathsf{S_T}$ does not properly verify, $\mathsf{R_T}$ halts and outputs $\perp$. Otherwise, $\mathsf{R_T}$ recalls the random string $\mu \in \{0,1\}^t$ that was chosen at step 1 and computes $M_{\rho_i} = M' \oplus \mu$. The transfer ends with $\mathsf{S_T}$ and $\mathsf{R_T}$ outputting $S_i = S_{i-1}$ and $R_i = R_{i-1}$, respectively.

D.2 Security

For simplicity, our proofs are given in the single-receiver setting but they readily carry over to the multi-receiver setting, as defined in [35, Appendix B].

Theorem 5. *The above $\mathcal{OT}_{k \times 1}^N$ protocol provides receiver security under the* SIS *assumption in the random oracle model.*

Proof. We show how to map any real-world cheating sender $\hat{\mathsf{S}}$ to an ideal-world cheating sender $\hat{\mathsf{S}}'$ such that, under the SIS assumption, the distributions $\mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$ and $\mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$ are computationally indistinguishable.

We consider a sequence of hybrid experiments with binary outputs. In each experiment Exp_i, a distinguisher D inputs the states (S_k, R_k) produced by $\hat{\mathsf{S}}$ and R' at the end of Exp_i and outputs a bit. We define W_i to be the event that the output of Exp_i is 1. The first experiment outputs whatever the distinguisher D outputs and corresponds to the real interaction between the cheating sender $\hat{\mathsf{S}}$ and the receiver R.

Exp_0: This experiment involves a real execution of $\hat{\mathsf{S}}$ in interaction with a honest receiver R which queries the index $\rho_i \in [N]$ at the i-th transfer for each $i \in [k]$. The output of Exp_0 is exactly the output of the distinguisher D on input of $X = (S_k, R_k) \leftarrow \mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}$, so that we have

$$\Pr[W_0] = \Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}].$$

Exp_1: This experiment is like Exp_0 except that R' programs the random oracle $H_F : \{0,1\}^* \to \mathbb{Z}_q^{n \times m}$ in the following way. It runs the trapdoor generation algorithm $(\mathbf{F}, \mathbf{T_F}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ of [4] so as to obtain a statistically uniform matrix $\mathbf{F} \in \mathbb{Z}_q^{n \times m}$ and a small-norm $\mathbf{T_F} \in \mathbb{Z}^{m \times m}$ basis of $\Lambda_q^{\perp}(\mathbf{F})$. It then programs the random oracle so as to have $H_F(\varepsilon) = \mathbf{F} \in \mathbb{Z}_q^{n \times m}$. Clearly, this change leaves the adversary's view statistically unchanged: we have $|\Pr[W_1] - \Pr[W_0]| \in \mathsf{negl}(\lambda)$.

Exp_2: is as Exp_1 but, at step 5 of the initialization phase, R' uses the short basis $\mathbf{T_F} \in \mathbb{Z}^{m \times m}$ of $\Lambda_q^{\perp}(\mathbf{F})$ (which satisfies $\mathbf{F} \cdot \mathbf{T_F} = \mathbf{0}^n \bmod q$) to extract witnesses $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$ from the columns $\mathbf{p}_j = \mathbf{F}^\top \cdot \mathbf{s}_j + \mathbf{e}_j \in \mathbb{Z}^m$ of the matrix $\mathbf{P} = [\mathbf{p}_1 \mid \ldots \mid \mathbf{p}_t] \in \mathbb{Z}_q^{m \times t}$ for each $j \in [t]$. Note that this can be done by inverting the LWE function (see, e.g., [32, Section 2.3]). At this point, R' aborts the interaction in the event that one of the following conditions holds:

E.1: The LWE-inversion algorithm fails to compute small-norm vectors $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$ such that $\mathbf{p}_j = \mathbf{F}^\top \cdot \mathbf{s}_j + \mathbf{e}_j \in \mathbb{Z}_q^m$ for some $j \in [t]$.

E.2: The columns of $\mathbf{S} = [\mathbf{s}_1 \mid \ldots \mid \mathbf{s}_t] \in \chi^{n \times t}$ are successfully extracted but there exists $i \in [N]$ such that one of the coordinates of $\mathbf{b}_i - \mathbf{S}^\top \cdot \mathbf{a}_i \bmod q$ is neither close to 0 nor $\lfloor q/2 \rfloor$ (i.e., the inequalities $|\mathbf{b}_i - \mathbf{S}^\top \cdot \mathbf{a}_i \bmod q| > \alpha q$ and $|(\mathbf{b}_i - \mathbf{S}^\top \cdot \mathbf{a}_i \bmod q) - \lfloor q/2 \rfloor| > \alpha q$ are both satisfied).

In either of the above situations, R' infers that $\hat{\mathsf{S}}$ managed to create a convincing argument for a false statement and aborts the interaction. In such a situation, however, R' can be turned into an algorithm that breaks the binding property of the commitment scheme used in the ZK argument (which contradicts the SIS assumption if the statistically hiding commitment of [41] is used) by replaying the adversary with the same random tape but a different random oracle H_{FS}. According to the General Forking Lemma of [12], replaying $\hat{\mathsf{S}}$ up to $32 \cdot Q_H/(\epsilon - 3^{-t})$ times (where Q_H is the number of queries to H_{FS} is sufficient to extract a breach in the binding property of the commitment). Otherwise (i.e., if R' does not fail), the matrix $\mathbf{S} \in \chi^{n \times t}$ allows R' to decode the messages $M_1, \ldots, M_N \in \{0,1\}^t$ from the encrypted database $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$. Under the SIS assumption, it follows that Exp_1 returns 1 with about the same probability as Exp_0. In the random oracle model, the SIS assumption thus implies that $|\Pr[W_2] - \Pr[W_1]| \in \mathsf{negl}(\lambda)$.

Exp_3: is identical to Exp_2 except that the receiver R' makes use of the matrix $\mathbf{S} \in \chi^{n \times t}$, which was extracted at step 5 of the initialization phase. At step 2 of each transfer, R' uses $\mathbf{S}$ to determine if the NIZK argument π_T really proves a true statement or if $\hat{\mathsf{S}}$ managed to break its soundness. Namely, upon receiving $\hat{\mathsf{S}}$'s response $M' \in \{0,1\}^t$ at step 2, R' uses the previously extracted $\mathbf{S} \in \chi^{n \times t}$ to determine if there exists $\mathbf{y} \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \leq q/5$ such that

$$\mathbf{c}_0^\top \cdot \mathbf{S} + \mathbf{y}^\top = \mathbf{c}_1^\top - {M'}^\top \cdot \lfloor q/2 \rfloor. \tag{46}$$

If such vector $\mathbf{y}$ turns out not to exist, R' deduces R' that $\hat{\mathsf{S}}$ was able to fake a convincing argument for a false statement and aborts the interaction.

However, R' can then be turned into a PPT adversary against the binding property of the commitment scheme used in the ZK argument (and thus the SIS assumption if the commitment of [41] is used) by replaying the adversary according to the General Forking technique [12]. The result of [12] tells us that replaying $\hat{\mathsf{S}}$ up to $32 \cdot Q_H/(\epsilon - 3^{-t})$ times (where Q_H is the number of queries to H_{FS}) suffices to break the binding property of the commitment. Under the SIS assumption, we have $|\Pr[W_3] - \Pr[W_2]| \in \mathbf{negl}(\lambda)$.

Exp_4: This experiment is like Exp_3 with the difference that, at each transfer, the receiver R' chooses the index $\rho_i = 1$ and thus always requests the first message of the encrypted database. In more details, at each transfer, R' samples vectors $\mathbf{e} \hookleftarrow U(\{-1,0,1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and $\nu \hookleftarrow U([-B,B]^t)$ to compute and send

$$(\mathbf{c}_0, \mathbf{c}_1) = \big(\mathbf{a}_1 + \mathbf{F} \cdot \mathbf{e},\ \mathbf{b}_1 + \mathbf{P}^\top \cdot \mathbf{e} + \mu \cdot \lfloor q/2 \rfloor + \nu\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t,$$

which is a re-randomization of $(\mathbf{a}_1, \mathbf{b}_1 + \mu \cdot \lfloor q/2 \rfloor)$. Moreover, R'_T uses the witness $\rho_i = 1$ to faithfully generate an interactive WI argument that $(\mathbf{c}_0, \mathbf{c}_1)$ is a re-randomization of $(\mathbf{a}_{\rho_i}, \mathbf{b}_{\rho_i})$. It thus generates a WI argument of knowledge of vectors $\mathfrak{m} = \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_1|\mathbf{b}_1) \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B,B]^t$, $\tau \in \{0,1\}^\ell$ and $(\mathbf{v}_1^\top|\mathbf{v}_2^\top)^\top \in \mathbb{Z}^{2m}$ satisfying relations (19). By the statistically WI of the interactive argument system, this modification has no noticeable impact on the output distribution of a cheating sender $\hat{\mathsf{S}}$ whatsoever. We have $|\Pr[W_4] - \Pr[W_3]| \in \mathsf{negl}(\lambda)$.

In Exp_4, we define the ideal-world cheating sender $\hat{\mathsf{S}}'$ in the following way. It programs the random oracle $H_F : \{0,1\}^* \to \mathbb{Z}_q^{n \times m}$ in such a way that $H_F(\varepsilon) = \mathbf{F} \in \mathbb{Z}_q^{n \times m}$ for some statistically random matrix produced as $(\mathbf{F}, \mathbf{T_F}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. At the initialization phase, $\hat{\mathsf{S}}'$ uses the small-norm basis $\mathbf{T_F}$ of $\Lambda_q^\perp(\mathbf{F})$ to extract the small-norm matrices $\mathbf{S} = [\mathbf{s}_1|\ldots|\mathbf{s}_t] \in \chi^{n \times t}$ and $\mathbf{E} = [\mathbf{e}_1|\ldots|\mathbf{e}_t] \in \chi^{m \times t}$ satisfying (37) and decrypt $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$ into messages $M_1, \ldots, M_N \in \{0,1\}^N$. If the extraction fails because one of the events E.1 and E.2 (as defined in Exp_2) comes about, $\hat{\mathsf{S}}'$ aborts. Otherwise, it then submits $M_1, \ldots, M_N \in \{0,1\}^N$ to the trusted party T. As in Exp_2, during each transfer phase, $\hat{\mathsf{S}}'$ computes $(\mathbf{c}_0, \mathbf{c}_1)$ as a re-randomization of $(\mathbf{a}_1, \mathbf{b}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ and faithfully generates the receiver's argument of knowledge using the witness $\rho_i = 1$ at step 1. At step 2 of each transfer, $\hat{\mathsf{S}}'$ aborts if it realizes that $\hat{\mathsf{S}}$ created a convincing NIZK argument π_T for a false statement. If π_T correctly verifies and indeed relates to a true statement (which $\hat{\mathsf{S}}'$ can detect by applying the test (46) using the matrix $\mathbf{S} \in \chi^{n \times t}$ extracted in the initialization phase), $\hat{\mathsf{S}}'$ sends 1 to the trusted party T so as to authorize the transfer in the ideal world. Otherwise, $\hat{\mathsf{S}}'$ sends 0 to T. At the end of the k-th transfer phase, $\hat{\mathsf{S}}'$ outputs whatever $\hat{\mathsf{S}}$ outputs as its final state S_k.

In Exp_4, it is easy to see that

$$\Pr[W_4] = \Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}].$$

Putting the above altogether, we find that the SIS assumption implies such that

$$|\Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}] - \Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}]| \in \mathsf{negl}(\lambda).$$

□

The proof of security against a dishonest receiver is almost identical to the proof of Theorem 3. The only difference is that, from experience Exp_3 onwards, the sender's ZK arguments are non-interactive and can be simulated by programming the random oracle $H_{\mathsf{FS}} : \{0,1\}^* \to \{1,2,3\}^{\varsigma}$ in the standard way. The detailed proof of the following theorem is thus omitted.

Theorem 6. *The above $\mathcal{OT}^N_{k\times 1}$ protocol provides sender security under the* SIS *assumption in the random oracle model.*

As mentioned in [35], extending the simulation-based definitions to the multi-receiver setting is rather straightforward (see [35, Appendix B] for details).

Analogously to the Green-Hohenberger protocol [35, Section 4], our proof of sender security goes through in the multi-receiver setting as long as the receivers interact with the sender in a sequential manner. This restriction is important since the simulator has to rewind the receiver's zero-knowledge arguments at step 1 of each transfer, which would not be possible in concurrent sessions.

E Security Proofs for the $\mathcal{OT}^N_{k\times 1}$ Protocol of Section 4

E.1 Proof of Theorem 2

Let us first recall Theorem 2:

Theorem 7. *The $\mathcal{OT}^N_{k\times 1}$ protocol provides receiver security under the* SIS *assumption.*

Proof. We prove that any real-world cheating sender $\hat{\mathsf{S}}$ implies an ideal-world cheating sender $\hat{\mathsf{S}}'$ such that, under the SIS assumption, the two distributions $\mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$ and $\mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$ are indistinguishable to any PPT distinguisher D.

To this end, we consider a sequence of hybrid experiments with binary outputs. In each experiment Exp_i, a distinguisher D takes as input the states (S_k, R_k) produced by $\hat{\mathsf{S}}$ and R' at the end of the experiment and outputs a bit. We define W_i as the event that the output of experiment Exp_i is 1. The first experiment outputs whatever the distinguisher D outputs and corresponds to the real interaction between the cheating sender $\hat{\mathsf{S}}$ and the receiver R.

Exp_0: This experiment involves a real execution of $\hat{\mathsf{S}}$ in interaction with a honest receiver R which queries the index $\rho_i \in [N]$ at the i-th transfer for each $i \in [k]$. The output of Exp_0 is exactly the output of the distinguisher D on input of $X = (S_k, R_k) \leftarrow \mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}$, so that we have

$$\Pr[W_0] = \Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}].$$

Exp_1**:** This experiment is like Exp_0 except that, at step 5 of the initialization phase, the knowledge extractor of the argument system is used to extract the witnesses $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$, $\bar{\mathbf{x}}_j \in \chi^N$, $\bar{M}_j \in \{0,1\}^N$, for each $j \in [t]$, from the sender's argument. In the event that the knowledge extractor fails to extract valid witnesses, the experiment aborts and outputs $\perp$. We know that the zero-knowledge argument system is computationally sound as long as the underlying commitment is computationally binding. If the perfectly hiding commitment of [41] is used, the binding property is in turn implied by the SIS assumption. Under the SIS assumption, it follows that Exp_1 returns 1 with about the same probability as Exp_0. Specifically, there exists a SIS solver $\mathcal{B}$ such that $|\Pr[W_1] - \Pr[W_0]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$.

Exp_2**:** This experiment is identical to Exp_1 except that the receiver R' makes use of the matrix $\mathbf{S} \in \chi^{n \times t}$, which underlies $\mathbf{P} \in \mathbb{Z}_q^{m \times t}$ in (3) and was extracted at step 5 of the initialization phase. Namely, at step 2 of each transfer, R' uses $\mathbf{S}$ to determine if the ZK argument sent by $\hat{\mathsf{S}}$ is really an argument for a true statement or if $\hat{\mathsf{S}}$ somehow managed to break the soundness of the argument system. Namely, upon receiving the response $M' \in \{0,1\}^t$ of $\hat{\mathsf{S}}$ at step 2, R' uses the previously extracted $\mathbf{S} \in \chi^{n \times t}$ to determine whether there exists a vector $\mathbf{y} \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \leq q/5$ such that

$$\mathbf{c}_0^\top \cdot \mathbf{S} + \mathbf{y}^\top = \mathbf{c}_1^\top - {M'}^\top \cdot \lfloor q/2 \rfloor. \qquad (47)$$

If no such vector $\mathbf{y}$ exists, R' infers that $\hat{\mathsf{S}}$ broke the soundness of the argument system. In this case, $\hat{\mathsf{S}}$ can be rewound so as to break the binding property of the statistically hiding commitment scheme used by the ZK argument system, which in turn contradicts the SIS assumption. We thus have $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$ for some efficient algorithm $\mathcal{B}$ which is given rewinding access to $\hat{\mathsf{S}}$.

Exp_3**:** This experiment is like Exp_2 with the difference that, at each transfer, the receiver R' chooses the index $\rho_i = 1$ and thus always requests the first message of the encrypted database. In more details, at each transfer, R' samples vectors $\mathbf{e} \hookleftarrow U(\{-1,0,1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and $\nu \hookleftarrow U([-B,B]^t)$ to compute and send

$$(\mathbf{c}_0, \mathbf{c}_1) = \big(\mathbf{a}_1 + \mathbf{F} \cdot \mathbf{e},\ \mathbf{b}_1 + \mathbf{P}^\top \cdot \mathbf{e} + \mu \cdot \lfloor q/2 \rfloor + \nu\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t,$$

which is a re-randomization of $(\mathbf{a}_1, \mathbf{b}_1 + \mu \cdot \lfloor q/2 \rfloor)$. Moreover, R'_{T} uses the witness $\rho_i = 1$ to faithfully generate an interactive WI argument that $(\mathbf{c}_0, \mathbf{c}_1)$ is a re-randomization of $(\mathbf{a}_{\rho_i}, \mathbf{b}_{\rho_i})$. It thus generates a WI argument of knowledge of vectors $\mathfrak{m} = \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_1|\mathbf{b}_1) \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B,B]^t$, $\tau \in \{0,1\}^\ell$ and $(\mathbf{v}_1^\top|\mathbf{v}_2^\top)^\top \in \mathbb{Z}^{2m}$ satisfying relations (19). By the statistically WI of the interactive argument system, this modification has no noticeable impact on the output distribution of a cheating sender $\hat{\mathsf{S}}$. Indeed, since we chose B as a randomization parameter such that $(m+1)\alpha q/B$ is negligible, the result of [24, Section 4.1] implies that always re-randomizing $(\mathbf{a}_1, \mathbf{b}_1 + \mu \cdot \lfloor q/2 \rfloor)$ leaves the view of $\hat{\mathsf{S}}$ statistically unchanged. We have $|\Pr[W_2] - \Pr[W_1]| \leq \mathsf{negl}(\lambda)$.

In Exp_3, we can define the ideal-world cheating sender $\hat{\mathsf{S}}'$ which emulates the honest receiver R' interacting with $\hat{\mathsf{S}}$. At the initialization phase, $\hat{\mathsf{S}}'$ appeals to the knowledge extractor of the argument system so as to extract the small-norm matrices $\mathbf{S} = [\mathbf{s}_1|\ldots|\mathbf{s}_t] \in \chi^{n\times t}$ and $\mathbf{E} = [\mathbf{e}_1|\ldots|\mathbf{e}_t] \in \chi^{m\times t}$ satisfying (3). Armed with the decryption key $\mathbf{E} \in \chi^{m\times t}$ of the cryptosystem, $\hat{\mathsf{S}}'$ can decrypt $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$ and obtain the messages $M_1, \ldots, M_N \in \{0,1\}^N$ that were encrypted in (4) by $\hat{\mathsf{S}}$. It then submits $M_1, \ldots, M_N \in \{0,1\}^N$ to the trusted party T. As in Exp_2, during each transfer phase, $\hat{\mathsf{S}}'$ computes $(\mathbf{c}_0, \mathbf{c}_1)$ as a re-randomization of $(\mathbf{a}_1, \mathbf{b}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ and faithfully generates the receiver's argument of knowledge using the witness $\rho_i = 1$ at step 1. At step 2 of each transfer, $\hat{\mathsf{S}}'$ plays the role of the verifier on behalf of R' in the interactive zero-knowledge argument generated by $\hat{\mathsf{S}}$. If $\hat{\mathsf{S}}'$ detects that $\hat{\mathsf{S}}$ creates a verifying argument for a false statement (which $\hat{\mathsf{S}}'$ can detect using the extracted matrix $\mathbf{S} \in \mathbb{Z}^{n\times t}$, by applying the test (47)), it aborts the interaction as in Exp_3. If the ZK argument involves a true statement, $\hat{\mathsf{S}}'$ sends 1 to the trusted party T so as to authorize the transfer in the ideal world. Otherwise, $\hat{\mathsf{S}}'$ sends 0 to T. At the end of the k-th transfer phase, $\hat{\mathsf{S}}'$ outputs whatever $\hat{\mathsf{S}}$ outputs as its final state S_k.

In Exp_3, it is easy to see that

$$\Pr[W_3] = \Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}].$$

When putting the above altogether, we find that there exists a PPT SIS solver $\mathcal{B}$ such that

$$\begin{aligned} &|\Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}] \\ &\qquad - \Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}]| \leq 2\cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda) + \mathsf{negl}(\lambda), \end{aligned}$$

which proves the result. □

E.2 Proof of Theorem 3

Let us restate Theorem 3.

Theorem 8. *The $\mathcal{OT}_{k\times 1}^N$ protocol provides sender security under the* SIS *and* LWE *assumptions.*

Proof. Given a real malicious receiver $\hat{\mathsf{R}}$, we construct a cheating receiver $\hat{\mathsf{R}}'$ in the ideal world such that, under the SIS and LWE assumption, no PPT distinguisher D can tell apart the distributions $\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$ and $\mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$.

To do this, we proceed again via a sequence of hybrid experiments with binary outputs. For each i, we consider the probability that a distinguisher D outputs 1 on input of the states (S_k, R_k) that constitute the outcome of experiment Exp_i. We also define W_i to be the event that experiment Exp_i outputs 1.

Exp_0**:** This experiment corresponds to a real execution of $\hat{\mathsf{R}}$ in interaction with a honest sender $\mathsf{S}(M_1,\ldots,M_N)$. The output of the experiment is identical to that of the distinguisher D on input of $X=(S_k,R_k)\leftarrow \mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}$. We have

$$\Pr[W_0]=\Pr[\mathsf{D}(X)=1 \mid X\leftarrow \mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}].$$

Exp_1**:** This experiment departs from Exp_0 in that, when the dishonest receiver $\hat{\mathsf{R}}_{\mathsf{T}}$ sends the ciphertext $(\mathbf{c}_0,\mathbf{c}_1)\in\mathbb{Z}_q^n\times\mathbb{Z}_q^t$ at step 1 of each transfer, the knowledge extractor of the argument system is used to extract the witnesses $\mathfrak{m}\in\{0,1\}^{m_d}$, $\mathbf{e}\in\{-1,0,1\}^t$, $\mu\in\{0,1\}^t$, $\nu\in[-B,B]^t$, $\tau\in\{0,1\}^\ell$ and $\mathbf{v}=(\mathbf{v}_1^\top|\mathbf{v}_2^\top)^\top\in\mathbb{Z}^{2m}$ which satisfy (19). If the knowledge extractor fails to produce valid witnesses at some transfer, the experiment aborts and outputs $\perp$. Recall that the zero-knowledge argument system is computationally sound if the underlying commitment is binding, which is equivalent to the SIS assumption if the perfectly hiding commitment of [41] is used. Under the SIS assumption, experiment Exp_1 returns 1 with about the same probability as Exp_0. There thus exists a SIS solver $\mathcal{B}$ such that $|\Pr[W_1]-\Pr[W_0]|\le k\cdot\mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$, where k is the number of transfers.

Exp_2**:** This experiment is identical to Exp_1 except that, at step 1 of each transfer, the experiment aborts if the extracted witnesses $\mathfrak{m}\in\{0,1\}^{m_d}$, $\mathbf{e}\in\{-1,0,1\}^t$, $\mu\in\{0,1\}^t$, $\nu\in[-B,B]^t$, $\tau\in\{0,1\}^\ell$ and $\mathbf{v}=(\mathbf{v}_1^\top|\mathbf{v}_2^\top)^\top\in\mathbb{Z}^{2m}$ are such that the product

$$\begin{bmatrix}\mathbf{a}_{\mathfrak{m}}\\ \hline \mathbf{b}_{\mathfrak{m}}\end{bmatrix}=\begin{bmatrix}\mathbf{H}_{n,q-1} & \\ \hline & \mathbf{H}_{t,q-1}\end{bmatrix}\cdot\mathfrak{m}\ \in\mathbb{Z}_q^{n+t}$$

does not match any ciphertext $\{(\mathbf{a}_i,\mathbf{b}_i)\}_{i=1}^N$ appearing in R_0 (namely, we have $(\mathbf{a}_{\mathfrak{m}},\mathbf{b}_{\mathfrak{m}})\neq(\mathbf{a}_i,\mathbf{b}_i)$ for each $i\in[N]$). We claim that such an event implies a breach in the bounded message security of the signature scheme:

Lemma 7. *Under the* SIS *assumption, experiments* Exp_2 *and* Exp_1 *are computationally indistinguishable: there exists a PPT algorithm* $\mathcal{B}$ *such that* $|\Pr[W_2]-\Pr[W_1]|\le N\cdot\mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$.

Exp_3**:** This experiment is like Exp_2 except that, at step 5 of the initialization phase, the zero-knowledge argument of knowledge of $\mathbf{s}_j\in\chi^n$, $\mathbf{e}_j\in\chi^m$, $\bar{\mathbf{x}}_j\in\chi^N$, $\bar{M}_j\in\{0,1\}^N$ such that

$$\left[\begin{array}{c|c|c|c}\mathbf{F}^\top & \mathbf{I}_m & & \\ \hline \mathbf{A}_{\mathrm{DB}}^\top & & \mathbf{I}_N & \lfloor q/2\rfloor\cdot\mathbf{I}_N\end{array}\right]\cdot\begin{bmatrix}\mathbf{s}_j\\ \hline \mathbf{e}_j\\ \hline \bar{\mathbf{x}}_j\\ \hline \bar{M}_j\end{bmatrix}=\begin{bmatrix}\mathbf{p}_j\\ \hline \mathbf{b}_j\end{bmatrix}\qquad\forall j\in[t]$$

is replaced by a simulated interactive argument and so is the ZK argument of knowledge of $\{(\mathbf{s}_j,\mathbf{e}_j,\mathbf{y}[j])\}_{j=1}^t$ satisfying (17) at step 2 of each transfer protocol. From this experiment on, we notice that the small-norm matrices

$\mathbf{S} = [\mathbf{s}_1|\dots|\mathbf{s}_t] \in \mathbb{Z}^{n\times t}$, $\mathbf{E} = [\mathbf{e}_1|\dots|\mathbf{e}_t] \in \chi^{m\times t}$ satisfying (3) are no longer used by the sender S. Yet, the statistical ZK property of the zero-knowledge argument system ensures that $|\Pr[W_3] - \Pr[W_2]| \leq \mathsf{negl}(\lambda)$.

Exp_4: This experiment is like Exp_3 with the difference that, at step 2 of the initialization phase, each column $\mathbf{p}_i$ of the Regev's encryption public key matrix $\mathbf{P} = [\mathbf{p}_1 \mid \dots \mid \mathbf{p}_t] = \mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E} \in \mathbb{Z}_q^{m\times t}$ is traded for a uniformly random vector $\mathbf{p}_i \leftarrow U(\mathbb{Z}_q^m)$. At the same time, each $\mathbf{b}_i = \mathbf{S}^\top \cdot \mathbf{a}_i + \mathbf{x}_i + M_i\lfloor\frac{q}{2}\rfloor \in \mathbb{Z}_q^t$ is replaced by a truly uniform random vector in $\mathbb{Z}_q^t$. Therefore, $\mathbf{P}$ is a uniformly distributed matrix in $\mathbb{Z}_q^{m\times t}$, and the $(\mathbf{b}_i)_{i=1}^N$ are distributed as uniform vectors in $(\mathbb{Z}_q^t)^N$. Now, at step 5 of the initialization phase and step 2 of each transfer, the sender's zero-knowledge arguments are simulated arguments for false statements. However, a straightforward reduction shows that, under the LWE assumption over $t\cdot(m+N)$ samples, these changes should remain unnoticed to the malicious receiver $\hat{\mathsf{R}}$ and have no impact on the distinguisher's output: we have $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(\lambda)$.

The ideal-world receiver $\hat{\mathsf{R}}'$ is defined as follows. It assumes the role of the sender S' in interaction with the real-world receiver $\hat{\mathsf{R}}$ in Exp_4. This implies that, in the initialization phase, the matrices $(\mathbf{F},\mathbf{P})$ are chosen as uniformly random matrices $(\mathbf{F},\mathbf{P}) \leftarrow U(\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{m\times t})$ and while, at step 3, $(\mathbf{a}_i,\mathbf{b}_i) \leftarrow U(\mathbb{Z}_q^n \times \mathbb{Z}_q^t)$ is chosen at random for each $i\in[N]$. The randomly generated pairs $\{(\mathbf{a}_i,\mathbf{b}_i)\}_{i=1}^N$ are faithfully signed using $SK_{sig} = \mathbf{T}_{\mathbf{A}}$ at step 4. In step 5 of the initialization phase, $\hat{\mathsf{R}}'$ appeals to the simulator of the ZK argument. At the i-th transfer, when $\hat{\mathsf{R}}$ sends $(\mathbf{c}_0,\mathbf{c}_1)$ and argues knowledge of $(\mathfrak{m},\mathbf{e},\mu,\nu,\tau,\mathbf{v}_1,\mathbf{v}_2)$ at step 1, $\hat{\mathsf{R}}'$ uses the knowledge extractor of the argument system to extract the witnesses $(\mathfrak{m},\mathbf{e},\mu,\nu,\tau,\mathbf{v}_1,\mathbf{v}_2) \in \{0,1\}^{m_d} \times \{-1,0,1\}^t \times \{0,1\}^t \times [-B,B]^t \times \{0,1\}^\ell$ and determine the index $\rho_i \in [N]$ such that

$$\begin{bmatrix}\mathbf{a}_{\rho_i}\\ \hline \mathbf{b}_{\rho_i}\end{bmatrix} = \left[\begin{array}{cc}\mathbf{H}_{n,q-1} & \\ \hline & \mathbf{H}_{t,q-1}\end{array}\right]\cdot \mathfrak{m} \ \in \mathbb{Z}_q^{n+t}.$$

Note that, by Lemma 7, such an index must exist unless $\hat{\mathsf{R}}$ can forge a signature. Having determined the index $\rho_i \in [N]$ of the queried database entry, $\hat{\mathsf{R}}'$ sends ρ_i to the trusted party T which returns the message $M_{\rho_i} \in \{0,1\}^t$. The latter is used together with the extracted witness $\mu \in \{0,1\}^t$ to define the response $M' = M_{\rho_i} \oplus \mu \in \{0,1\}^t$ that $\hat{\mathsf{R}}'$ generates on behalf of the sender $\hat{\mathsf{S}}'$ at step 2 of the transfer. In addition, the ideal-world dishonest receiver $\hat{\mathsf{R}}'$ appeals to the simulator of the zero-knowledge argument system to simulate an argument of knowledge of $\{(\mathbf{s}_j,\mathbf{e}_j,\mathbf{y}[j])\}_{j=1}^t$ for the statement (17).

It is easy to see that, when $\hat{\mathsf{R}}$ interacts with the simulator $\hat{\mathsf{R}}'$ that emulates the real-world sender S', its view is identical to that of Exp_4: we have

$$\Pr[W_4] = \Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}].$$

When combining the above, we conclude that there exist PPT algorithms $\mathcal{B}$ and $\mathcal{B}'$ such that

$$\begin{aligned}&|\Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}]\\ &- \Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}]| \leq 2 \cdot \mathbf{Adv}^{\mathsf{SIS}}_{\mathcal{B}}(\lambda) + \mathbf{Adv}^{\mathsf{LWE}}_{\mathcal{B}'}(\lambda) + \mathsf{negl}(\lambda).\end{aligned}$$

This proves the sender security under the SIS and LWE assumptions. □

Proof (of Lemma 7). Let us assume that, at some transfer, the knowledge extractor reveals witnesses that contain a m_d-bit string $\mathfrak{m}^\star$ which is not the decomposition of any $(\mathbf{a}_i|\mathbf{b}_i) \in \mathbb{Z}_q^{n+t}$ appearing in R_0. We construct an algorithm $\mathcal{B}$ that breaks the random-message security of the stateful signature scheme in Section 3.2 after N non-adaptive signing queries. In turn, this will contradict the SIS assumption.

Algorithm $\mathcal{B}$ begins by faithfully running steps 2 and 3 of the real initialization algorithm. Having generated $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$, it computes their m_d-bit decompositions $\mathfrak{m}_i \leftarrow \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_i|\mathbf{b}_i) \in \{0,1\}^{m_d}$ for each $i \in [N]$. Then, it submits $\{\mathfrak{m}_i\}_{i=1}^N$ to its own challenger for the bounded-message security of the signature scheme. The latter challenger replies by returning a public key PK_{sig} and a set of signatures $\{(\tau_i, \mathbf{v}_i)\}_{i=1}^N$, which $\mathcal{B}$ uses to define

$$R_0 = \big(PK_{sig},\ (\mathbf{F}, \mathbf{P}),\ \{(\mathbf{a}_i, \mathbf{b}_i), (\tau_i, \mathbf{v}_i)\}_{i=1}^N\big).$$

Since $\mathcal{B}$ knows the messages $\{M_i\}_{i=1}^N$ and $(\mathbf{S}, \mathbf{E}) \in \chi^{n\times t} \times \chi^{m\times t}$ for having generated the matrix $\mathbf{P} = \mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E}$ at step 2, it can faithfully generate the sender's argument of knowledge at step 5 of the initialization algorithm.

By hypothesis, we know that, in some transfer, the knowledge extractor will extract witnesses $\mathfrak{m} \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B,B]^t$, $\tau \in \{0,1\}^\ell$ and $\mathbf{v} = (\mathbf{v}_1^\top|\mathbf{v}_2^\top)^\top \in \mathbb{Z}^{2m}$ such that $\mathfrak{m} \notin \{\mathfrak{m}_1, \ldots, \mathfrak{m}_N\}$. Moreover, we know that $(\tau, \mathbf{v}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m}$ is a valid signature on $\mathfrak{m}$ unless the failure event introduced in Exp_1 (i.e., a failure of the knowledge extractor) occurs. This means that $\mathcal{B}$ can break the non-adaptive chosen-message security of the stateful signature scheme by outputting $\big(\mathfrak{m}, (\tau, \mathbf{v})\big)$. □

F Security Proofs for the OT-AC Protocol

F.1 Security of Honest Users Against Corrupted Issuer and Database

Lemma 8. *For any environment $\mathcal{E}$ and any adversary $\mathcal{A}$ controlling the issuer and the database, there exists an ideal-world adversary $\mathcal{A}'$ such that the distributions $\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)$ and $\mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}(\lambda)$ are computationally indistinguishable under the SIS assumption.*

Proof. As in [14], we assume that there is only one honest user U since the adversary can simulate the remaining users on its own. We consider a sequence of hybrid experiments Exp_i where, for each i, a simulator Sim_i interacts with $\mathcal{A}$ on behalf of U. For each i, we define by W_i the event that the environment $\mathcal{E}$ outputs 1 in Exp_i.

Exp_0: This experiment involves a real execution of the adversary controlling DB and I in interaction with a honest user U which obtains credentials from I and queries database entries $\rho_i \in [N]$ as dictated by the environment $\mathcal{E}$. The output of Exp_0 is the output of the environment $\mathcal{E}$, so that we have

$$\Pr[W_0] = \Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)].$$

Exp_1: This experiment is identical to Exp_0 with the difference that, at the first transfer dictated by $\mathcal{E}$ (when its state bit is $f_{DB} = 0$), the simulator Sim_1 appeals to the knowledge extractor of the argument system is used to extract the witnesses $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$, $\bar{\mathbf{x}}_j \in \chi^N$, $\bar{M}_j \in \{0,1\}^N$, for each $j \in [t]$, at step 1 of the $\mathsf{Transfer}$ protocol. Should the knowledge extractor fail to extract valid witnesses, Sim_1 aborts and outputs $\perp$. However, we know that the argument system is computationally sound whenever the underlying commitment is computationally binding, which boils down to the SIS assumption if the underlying commitment is the Kawachi *et al.* commitment [41]. Under the SIS assumption, $\mathcal{E}$ returns 1 with essentially the same probability as Exp_0 as there exists a SIS solver $\mathcal{B}$ such that $|\Pr[W_1] - \Pr[W_0]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$.

Exp_2: This experiment is like Exp_1 with the difference that the receiver Sim_2 makes use of the matrix $\mathbf{S} \in \chi^{n \times t}$, which was extracted at the first transfer ordered by $\mathcal{E}$. Specifically, at step 3 of each transfer, Sim_2 uses $\mathbf{S}$ to decide if the ZK argument sent by DB is an argument for a true statement or if the dishonest DB was able to break the soundness of the argument system. Upon receiving the response $M' \in \{0,1\}^t$ of DB at step 3, Sim_2 uses $\mathbf{S} \in \chi^{n \times t}$ to test whether there exists a vector $\mathbf{y} \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \leq q/5$ such that

$$\mathbf{c}_0^\top \cdot \mathbf{S} + \mathbf{y}^\top = \mathbf{c}_1^\top - {M'}^\top \cdot \lfloor q/2 \rfloor. \tag{48}$$

If no such vector $\mathbf{y}$ exists, Sim_2 concludes that $\mathcal{A}$ broke the soundness of the argument and aborts its interaction with $\mathcal{A}$. In this case, however, Sim_2 can be turned into an algorithm $\mathcal{B}$ that breaks the binding property of the statistically hiding commitment used by the ZK argument system using rewinding access to $\mathcal{A}$. This contradicts the SIS assumption and we have $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$.

Exp_3: This experiment is like Exp_2 except that, instead of faithfully querying the database entry ρ_i dictated by $\mathcal{E}$ at each transfer, Sim_3 chooses a random index $\rho \in [N]$ among those for which, on behalf of U, it obtained a valid credential for an attribute string $\mathbf{x}$ satisfying $\mathsf{BP}_\rho(\mathbf{x}) = 1$. That is, at each transfer, Sim_3 samples vectors $\mathbf{e} \hookleftarrow U(\{-1,0,1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and $\nu \hookleftarrow U([-B,B]^t)$ to compute

$$(\mathbf{c}_0, \mathbf{c}_1) = \big(\mathbf{a}_\rho + \mathbf{F} \cdot \mathbf{e},\ \mathbf{b}_\rho + \mathbf{P}^\top \cdot \mathbf{e} + \mu \cdot \lfloor q/2 \rfloor + \nu\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t,$$

which is a re-randomization of $(\mathbf{a}_\rho, \mathbf{b}_\rho + \mu \cdot \lfloor q/2 \rfloor)$. Moreover, Sim_2 uses the witness ρ to faithfully generate an interactive WI argument that $(\mathbf{c}_0, \mathbf{c}_1)$ is a re-randomization of $(\mathbf{a}_\rho, \mathbf{b}_\rho)$. It thus generates a WI argument of knowledge of vectors $\mathbf{z}_{\mathsf{BP},\rho}$, $\mathfrak{m} = \mathsf{vdec}_{2n+t,q-1}\big((\mathbf{a}_\rho^\top | \mathbf{b}_\rho^\top | \mathbf{h}_{\mathsf{BP},\rho}^\top)^\top\big) \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B,B]^t$, $\tau \in \{0,1\}^\ell$ and $(\mathbf{v}_1^\top | \mathbf{v}_2^\top)^\top \in \mathbb{Z}^{2m}$ satisfying relations (19) as well as the verification equations of BP_ρ. The statistical WI property of the argument system ensures that this modification leaves the output distribution of $\mathcal{E}$ statistically unchanged. Indeed, since the randomization parameter B was chosen to make $(m+1)\alpha q/B$ negligible, the distribution of $(\mathbf{c}_0, \mathbf{c}_1)$ is not affected (by the result of [24, Section 4.1]) if Sim_3 trades the real index ρ_i for a random one. We have $|\Pr[W_3] - \Pr[W_2]| \leq \mathsf{negl}(\lambda)$.

From the adversary $\mathcal{A}$ of Exp_3, we construct an ideal-world adversary $\mathcal{A}'$ which assumes the roles of the database and the dishonest issuer. This ideal-world adversary $\mathcal{A}'$ relays all the messages between $\mathcal{E}$ and the real-world $\mathcal{A}$. It runs $\mathcal{A}$ so as to obtain the issuer's public key $PK_I = (\mathbf{A}_I, \{\mathbf{A}_{I,j}\}_{j=0}^{\ell_I}, \mathbf{D}_I, \{\mathbf{D}_{I,0}, \mathbf{D}_{I,1}\}, \mathbf{u}_I)$ and the encrypted database $\big(PK_{\mathrm{DB}}, \{ER_i = \big(\mathbf{a}_i, \mathbf{b}_i, (\tau_i, \mathbf{v}_i)\big), \mathsf{BP}_i\}_{i=1}^N\big)$.

At each instruction $(\mathsf{issue}, \mathsf{U}', \mathbf{x})$ received from T, it starts an execution of the Issue protocol on behalf of U interacting with the issuer-executing $\mathcal{A}$. At the first instruction $\mathsf{transfer}$ received from T, $\mathcal{A}'$ runs to the knowledge extractor of the argument system at step 1 of $\mathsf{Transfer}$ so as to extract the small-norm $\mathbf{S} = [\mathbf{s}_1 | \ldots | \mathbf{s}_t] \in \chi^{n \times t}$ and $\mathbf{E} = [\mathbf{e}_1 | \ldots | \mathbf{e}_t] \in \chi^{m \times t}$ satisfying (8). Having obtained $\mathbf{S} \in \chi^{n \times t}$, $\mathcal{A}'$ can decrypt $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$ and obtain the messages $M_1, \ldots, M_N \in \{0,1\}^N$ that were encrypted as per (9). It then sends the message $(\mathsf{initdb}, \{(M_i, \mathsf{BP}_i)\}_{i=1}^N)$ to the trusted party T. Then, it simulates step 2 of $\mathsf{Transfer}$ on behalf of the real-world U and queries a random record $\rho \in [N]$ among those for which it has a credential for an input $\mathbf{x} \in \{0,1\}^\kappa$ such that $\mathsf{BP}_\rho(\mathbf{x}) = 1$. At step 3 of the transfer, it aborts the interaction in the event that $\mathcal{A}$ provides a verifying ZK argument whereas condition (48) is not satisfied. If the real-world transfer is successful (resp. fails), $\mathcal{A}'s$ sends $b = 1$ (resp. $b = 0$) to the trusted party T. Subsequent $\mathsf{transfer}$ instructions from T are treated in the same way, but without the first step of running the knowledge extractor.

We observe that $\mathcal{A}'$ provides the real-world database/issuer $\mathcal{A}$ with the same view as Sim_3 does in Exp_2. We have

$$\Pr[W_3] = \Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}(\lambda)].$$

When combining the above altogether, we obtained the claimed statement as there exists a PPT SIS solver $\mathcal{B}$ such that

$$\begin{aligned} |\Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\mathcal{E},\mathcal{A}}] \\ - \Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}]| \leq 2 \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda) + \mathsf{negl}(\lambda). \end{aligned}$$

□

F.2 Security Against a Corrupted Database

Lemma 9. *For any environment $\mathcal{E}$ and any adversary $\mathcal{A}$ controlling only the database, there exists an ideal-world adversary $\mathcal{A}'$ such that the distributions $\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)$ and $\mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}(\lambda)$ are computationally indistinguishable under the* SIS *assumption.*

Proof. Again, we assume that there is only one honest user U as the adversary can simulate the remaining users on its own. The proof is almost identical to the proof of Lemma 8: the sequence of experiments Exp_i is the same and only the simulation changes. We thus only describe the simulation.

From the adversary $\mathcal{A}$ of Exp_2, we construct an ideal-world adversary $\mathcal{A}'$ which assumes the roles of the database and the dishonest issuer. This ideal-world adversary $\mathcal{A}'$ relays all the messages between $\mathcal{E}$ and the real-world $\mathcal{A}$. It runs $\mathcal{A}$ so as to obtain the issuer's public key $PK_I = (\mathbf{A}_I, \{\mathbf{A}_{I,j}\}_{j=0}^{\ell_I}, \mathbf{D}_I, \{\mathbf{D}_{I,0}, \mathbf{D}_{I,1}\}, \mathbf{u}_I)$ and the encrypted database $\big(PK_{\mathrm{DB}}, \{ER_i = \big(\mathbf{a}_i, \mathbf{b}_i, (\tau_i, \mathbf{v}_i)\big), \mathsf{BP}_i\}_{i=1}^{N}\big)$.

At the first instruction $\mathsf{transfer}$ received from trusted party T, $\mathcal{A}'$ runs to the knowledge extractor of the argument system at step 1 of $\mathsf{Transfer}$ so as to extract the small-norm $\mathbf{S} = [\mathbf{s}_1|\ldots|\mathbf{s}_t] \in \chi^{n\times t}$ and $\mathbf{E} = [\mathbf{e}_1|\ldots|\mathbf{e}_t] \in \chi^{m\times t}$ satisfying relations (8). Using $\mathbf{S} \in \chi^{n\times t}$, $\mathcal{A}'$ is able to decrypt $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^{N}$ and obtain $M_1, \ldots, M_N \in \{0,1\}^N$ which were encrypted in (9). It then sends the message $(\mathsf{initdb}, \{(M_i, \mathsf{BP}_i)\}_{i=1}^{N})$ to T. Next, it simulates step 2 of $\mathsf{Transfer}$ on behalf of U and queries a random record $\rho \in [N]$ among those for which it has a credential for an $\mathbf{x} \in \{0,1\}^\kappa$ satisfying $\mathsf{BP}_\rho(\mathbf{x}) = 1$. At step 3 of the transfer, it aborts the interaction with $\mathcal{A}$ if $\mathcal{A}$ provides a verifying ZK argument but condition (48) does not hold. If the transfer succeeds (resp. fails), $\mathcal{A}'s$ sends $b = 1$ (resp. $b = 0$) to the trusted party T. Subsequent $\mathsf{transfer}$ instructions from T are handled in the same way, but without running the knowledge extractor to extract $\mathbf{S}$ and $\mathbf{E}$.

We remark that, when emulating the honest user U, the ideal-world adversary $\mathcal{A}'$ provides the real-world database/issuer $\mathcal{A}$ with the same view as Sim_3 does in Exp_3. We thus find that $\Pr[W_2] = \Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}(\lambda)]$. The above implies the existence of a PPT SIS solver $\mathcal{B}$ such that

$$\begin{aligned} |\Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\mathcal{E},\mathcal{A}}] \\ - \Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}]| \leq 2 \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda) + \mathsf{negl}(\lambda). \end{aligned}$$

□

F.3 Security Against Corrupted Users

Lemma 10. *For any environment $\mathcal{E}$ and any adversary $\mathcal{A}$ controlling only some of the users, there exists an ideal-world adversary $\mathcal{A}'$ such that the distributions $\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)$ and $\mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}(\lambda)$ are computationally indistinguishable under the* SIS *and* LWE *assumptions.*

Proof. We consider again a sequence of hybrid experiments Exp_i where, for each i, a simulator Sim_i interacts with $\mathcal{A}$ on behalf of the database and the issuer. For each i, W_i stands for the event that the environment $\mathcal{E}$ outputs 1 in Exp_i.

Exp_0: This experiment corresponds to a real execution of the adversary $\mathcal{A}$ in interaction with a honest sender and a honest issuer. The output of the experiment is identical to the output of the environment $\mathcal{E}$. We have

$$\Pr[W_0] = \Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)].$$

Exp_1: This experiment is like Exp_0 except that, when the user $\mathcal{U}$ sends the ciphertext $(\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ at step 2 of each transfer, Sim_1 appeals to the knowledge extractor of the argument system so as to extract the witnesses $\mathbf{x} \in \{0,1\}^\kappa$, $\mathbf{r}_{\mathsf{com},0}, \ldots, \mathbf{r}_{\mathsf{com},\kappa-1} \in \{0,1\}^m$, $d_{1,1}, \ldots, d_{L,\delta_\kappa} \in \{0,1\}$, $\mathbf{g}_{1,0}, \ldots, \mathbf{g}_{L,\delta_\kappa} \in \{0,1\}^{n\lceil \log q \rceil}$, $(y_1, \mathbf{r}_1), \ldots, (y_L, \mathbf{r}_L) \in \{0,1\} \times \{0,1\}^m$, $\mathfrak{m} \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B, B]^t$, $\tau \in \{0,1\}^\ell$, $\mathbf{v} \in \mathbb{Z}^{2m}$, $\mathfrak{m}_{\mathsf{U},\boldsymbol{x}} \in \{0,1\}^{m_I}$, $\mathsf{cert}_{\mathsf{U},\mathbf{x}}$ which satisfy (11) and (19), as well as the relations (21)-(28) of Section 6.6. If the knowledge extraction fails at some transfer, Sim_1 aborts and outputs $\perp$. Since the zero-knowledge argument system is computationally sound as long as the underlying commitment is binding, which boils down to the SIS assumption if the commitment of [41] is used. Under the SIS assumption, $\mathcal{E}$ returns 1 with nearly the same probability in Exp_1 as in Exp_0: concretely, there exists a SIS-solver $\mathcal{B}$ such that $|\Pr[W_1] - \Pr[W_0]| \leq Q \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$, where Q is the total number of transfers.

Exp_2: This experiment is like Exp_1 with one modification. At each step $\theta \in [L]$ of each transfer, Sim_2 considers the extracted witnesses $\mathbf{g}_{1,0}, \ldots, \mathbf{g}_{L,\delta_\kappa} \in \{0,1\}^{n\lceil \log q \rceil}$ and $\mathrm{var}(\theta) = d_{\theta,1} \ldots d_{\theta,\delta_\kappa}$. It computes

$$\mathsf{com}'_{\mathrm{var}(\theta)} = \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,\delta_\kappa} \ \in \mathbb{Z}_q^n \tag{49}$$

and aborts in the event that $\mathsf{com}'_{\mathrm{var}(\theta)} \neq \mathsf{com}_{\mathrm{var}(\theta)}$. Owing to relations (21), the soundness of the argument system ensures that the knowledge extractor reveals a path connecting $\mathsf{com}'_{\mathrm{var}(\theta)}$ to the root $\mathbf{u}_{\mathsf{tree}}$ of the Merkle tree, which is deterministically computed by hashing $\{\mathsf{com}_i\}_{i=0}^{\kappa-1}$. Together with (49), the inequality $\mathsf{com}'_{\mathrm{var}(\theta)} \neq \mathsf{com}_{\mathrm{var}(\theta)}$ implies that Sim_2 obtains a collision on the hash chain from $\mathsf{com}'_{\mathrm{var}(\theta)} \neq \mathsf{com}_{\mathrm{var}(\theta)}$ to the root $\mathbf{u}_{\mathsf{tree}}$. This contradicts the SIS assumption when the Merkle tree is instantiated with the same SIS-based hash function as in [46]. We thus have $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$.

Exp_3: This experiment is identical to Exp_2 except for the following difference. At each step $\theta \in [L]$ of each transfer, Sim_3 considers $\mathrm{var}(\theta) = d_{\theta,1} \ldots d_{\theta,\delta_\kappa}$ and checks if the witness $x_{\mathrm{var}(\theta)} \in \{0,1\}$ (i.e., the $\mathrm{var}(\theta)$-th coordinate of the extracted witness $\mathbf{x}$ of relation (28)) disagrees with the extracted bit $y_\theta \in \{0,1\}$ for which relation (21) ensures that $(y_\theta, \mathbf{r}_\theta) \in \{0,1\} \times \{0,1\}^m$ forms an opening of $\mathsf{com}_{\mathrm{var}(\theta)} = \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,\delta_\kappa}$. In this case, the knowledge extractor outputs two distinct openings $(x_{\mathrm{var}(\theta)}, \mathbf{r}_{\mathsf{com},\mathrm{var}(\theta)})$ and $(y_\theta, \mathbf{r}_\theta)$ of $\mathsf{com}_{\mathrm{var}(\theta)}$, which breaks the binding property of the commitment. Since the latter relies on the SIS assumption, we have

$$|\Pr[W_3] - \Pr[W_2]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda).$$

The changes introduced in Exp_2 and Exp_3 rule out the event that the dishonest user manages to cheat by breaking the binding property of the Merkle tree in an attempt to use an attribute string $\mathbf{x} \in \{0,1\}^{\kappa}$ that is accepted by BP_{ρ}, but for which he has no credential. We now turn to the case where the adversary rather uses the strategy of breaking the signature scheme used by the issuer.

Exp_4: This experiment is like Exp_3 with the difference that, at each transfer, Sim_4 aborts and outputs $\perp$ in the event that the extracted pair $(\mathfrak{m}_{\mathsf{U},\boldsymbol{x}}, \mathsf{cert}_{\mathsf{U},\mathbf{x}})$ is such that was never $\mathfrak{m}_{\mathsf{U},\boldsymbol{x}}$ certified by the honest issuer in any execution of Issue. This event clearly implies that $\mathcal{A}$ was able to forge a valid credential $\mathsf{cert}_{\mathsf{U},\mathbf{x}}$ for a message $\mathfrak{m}_{\mathsf{U},\boldsymbol{x}} = \mathfrak{m}_P || \mathbf{x} \in \{0,1\}^{m_I}$. Note that this covers the following cases: (i) A given user with pseudonym $P_{\mathsf{U}} = \mathbf{H}_{n,q-1} \cdot \mathfrak{m}_P \in \mathbb{Z}_q^n$ who pools his legally obtained credentials for distinct attribute strings $\mathbf{x}_1, \ldots \mathbf{x}_Q$ to create one for a new $\mathbf{x} \notin \{\mathbf{x}_1, \ldots \mathbf{x}_Q\}$; (ii) Distinct users with different pseudonyms $P_{\mathsf{U},1}, \ldots, P_{\mathsf{U},Q} \in \mathbb{Z}_q^n$ who pool their credentials to create one for $P_{\mathsf{U}} \notin \{P_{\mathsf{U},1}, \ldots, P_{\mathsf{U},Q}\}$. The security of the signature scheme in Section 3.1 ensures that this event only occurs with negligible probability under the SIS assumption. Specifically, the result of [44, Theorem 1] implies the existence of a SIS algorithm such that $|\Pr[W_4] - \Pr[W_3]| \leq Q_I \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$, where Q_I is the number of queries made by $\mathcal{A}$ to the honest issuer controlled by Sim_4.

Exp_5: This experiment is the same as Exp_4 with the difference that, after each knowledge extraction incurred by step 2 of $\mathsf{Transfer}$, Exp_5 aborts if the extracted witnesses $\mathfrak{m} \in \{0,1\}^{m_d}$, $\tau \in \{0,1\}^{\ell}$ and $\mathbf{v} = (\mathbf{v}_1^\top | \mathbf{v}_2^\top)^\top \in \mathbb{Z}^{2m}$ satisfy the equality

$$\begin{bmatrix} \mathbf{A} \mid \mathbf{A}_0 \mid \ldots \mid \mathbf{A}_{\ell} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_1 \\ \hline \mathbf{v}_2 \\ \hline \tau[1] \cdot \mathbf{v}_2 \\ \hline \cdots \\ \hline \tau[\ell] \cdot \mathbf{v}_2 \end{bmatrix} = \mathbf{u} + \mathbf{D} \cdot \mathfrak{m},$$

whereas the left-hand-side member of

$$\begin{bmatrix} \mathbf{a} \\ \hline \mathbf{b} \\ \hline \mathbf{h}_{\mathsf{BP}} \end{bmatrix} = \mathbf{H}_{2n+t,q-1} \cdot \mathfrak{m} \in \mathbb{Z}_q^{2n+t} \qquad (50)$$

does not coincide with any triple $\{(\mathbf{a}_i, \mathbf{b}_i, \mathbf{h}_{\mathsf{BP},i})\}_{i=1}^{N}$ appearing in DB. This event clearly implies a breach in the bounded message security of the signature scheme in Section 3.2. Under the SIS assumption, experiments Exp_5 and Exp_4 are thus computationally indistinguishable as there exists a PPT algorithm $\mathcal{B}$ such that $|\Pr[W_5] - \Pr[W_4]| \leq Q \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$.

Exp_6: This experiment is the same as Exp_5 but that, at the first $\mathsf{Transfer}$ query dictated by the environment $\mathcal{E}$, the zero-knowledge argument of knowledge of $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$, $\bar{\mathbf{x}}_j \in \chi^N$, $\bar{M}_j \in \{0,1\}^N$ such that

$$\left[\begin{array}{c|c|c|c} \mathbf{F}^\top & \mathbf{I}_m & & \\ \hline \mathbf{A}_{\mathrm{DB}}^\top & & \mathbf{I}_N & \lfloor q/2 \rfloor \cdot \mathbf{I}_N \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{s}_j \\ \hline \mathbf{e}_j \\ \hline \bar{\mathbf{x}}_j \\ \hline \bar{M}_j \end{array}\right] = \left[\begin{array}{c} \mathbf{p}_j \\ \hline \mathbf{b}_j \end{array}\right] \qquad \forall j \in [t]$$

(which is generated at step 1 of $\mathsf{Transfer}$) is replaced by a simulated interactive argument (which the ZK simulator generates without knowing the witnesses) and so is the ZK argument of knowledge of $\{(\mathbf{s}_j, \mathbf{e}_j, \mathbf{y}[j])\}_{j=1}^t$ satisfying (17) at step 3 of $\mathsf{Transfer}$. Henceforth, the small-norm matrices $\mathbf{S} = [\mathbf{s}_1 | \ldots | \mathbf{s}_t] \in \mathbb{Z}^{n \times t}$, $\mathbf{E} = [\mathbf{e}_1 | \ldots | \mathbf{e}_t] \in \chi^{m \times t}$ satisfying (8) are no longer used by Sim_4. The statistical ZK property of the argument system guarantees that $|\Pr[W_6] - \Pr[W_5]| \leq \mathsf{negl}(\lambda)$.

Exp_7: This experiment is like Exp_6 with the difference that, at step 3 of $\mathsf{DBSetup}$, Sim_7 replaces the columns of the matrix $\mathbf{P} = [\mathbf{p}_1 \mid \ldots \mid \mathbf{p}_t] = \mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E} \in \mathbb{Z}_q^{m \times t}$ by uniformly random vectors $\mathbf{p}_i \leftarrow U(\mathbb{Z}_q^m)$. Meaning that the matrix $\mathbf{P}$ is now distributed uniformly in $\mathbb{Z}_q^{m \times t}$. Sim_5 also replaces the Regev ciphertexts $\mathbf{b}_i = \mathbf{S}^\top \cdot \mathbf{a}_i + \mathbf{x}_i + M_i \lfloor \frac{q}{2} \rfloor$ by uniformly random vectors $\mathbf{b}_i \leftarrow U(\mathbb{Z}_q^t)$. Hence, the sender's zero-knowledge arguments are now simulated arguments for false statements at steps 1 and 3 of $\mathsf{Transfer}$. Still, under the LWE assumption over $t \cdot (m + N)$ samples, this modified distribution of $(\mathbf{P}, \{\mathbf{b}_i\}_{i=1}^N)$ is not noticeable to $\mathcal{A}$ and has no significant impact on $\mathcal{E}$'s output distribution: we have $|\Pr[W_7] - \Pr[W_6]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(\lambda)$.

The ideal-world adversary $\mathcal{A}'$ is designed as follows. It plays the role of DB and I in interaction with the real-world receivers U in Exp_7. In the $\mathsf{DBSetup}$ phase, $(\mathbf{F}, \mathbf{P})$ are chosen as truly random matrices $(\mathbf{F}, \mathbf{P}) \leftarrow U(\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{m \times t})$ and while each pair $(\mathbf{a}_i, \mathbf{b}_i) \leftarrow U(\mathbb{Z}_q^n \times \mathbb{Z}_q^t)$ is sampled at random for all $i \in [N]$. The tuples $\{(\mathbf{a}_i, \mathbf{b}_i, \mathbf{h}_{\mathsf{BP},i})\}_{i=1}^N$ are faithfully signed using $SK_{sig} = \mathbf{T}_{\mathbf{A}}$ at step 5.b of $\mathsf{DBSetup}$. At the first $\mathsf{Transfer}$ message sent by T, $\mathcal{A}'$ runs the simulator of the ZK argument. When $\mathcal{A}$ sends $(\mathbf{c}_0, \mathbf{c}_1)$ and argues knowledge of $(\mathfrak{m}, \mathbf{e}, \mu, \nu, \tau, \mathbf{v}_1, \mathbf{v}_2)$ at step 2 of the transfer phase, $\mathcal{A}'$ uses the knowledge extractor of the argument system to extract $(\mathfrak{m}, \mathbf{e}, \mu, \nu, \tau, \mathbf{v}_1, \mathbf{v}_2) \in \{0,1\}^{m_d} \times \{-1,0,1\}^t \times \{0,1\}^t \times [-B,B]^t \times \{0,1\}^\ell$ and determine the index $\rho_i \in [N]$ such that

$$\left[\begin{array}{c} \mathbf{a}_{\rho_i} \\ \hline \mathbf{b}_{\rho_i} \\ \hline \mathbf{h}_{\mathsf{BP},i} \end{array}\right] = \mathbf{H}_{2n+t,q-1} \cdot \mathfrak{m} \in \mathbb{Z}_q^{2n+t}.$$

Note that, due to the change introduced in Exp_4, such an index must exist unless $\mathcal{A}$ can forge a signature for the key PK_{sig}. Having decoded the queried index $\rho_i \in [N]$, $\mathcal{A}'$ queries the trusted party T to obtain credentials for an $\mathbf{x} \in \{0,1\}^\kappa$ such that $\mathsf{BP}_{\rho_i}(\mathbf{x}) = 1$, which subsequently allows $\mathcal{A}'$ to query T so as to obtain $M_{\rho_i} \in \{0,1\}^t$. The latter is used together with the extracted witness $\mu \in \{0,1\}^t$ to reverse-engineer the correct response $M' = M_{\rho_i} \oplus \mu \in \{0,1\}^t$ that $\mathcal{A}'$ should return on behalf of the sender DB at step 3 of the transfer. In addition, the ideal-world $\mathcal{A}'$ runs simulator of the argument system so as to simulate an argument of knowledge of $\{(\mathbf{s}_j, \mathbf{e}_j, \mathbf{y}[j])\}_{j=1}^t$ for the statement (17).

We remark that, when $\mathcal{A}$ and $\mathcal{E}$ interact with the simulator $\mathcal{A}'$ which emulates the real-world DB and I while playing the ideal-world receiver, their view is identical to that of Exp_6: we have

$$\Pr[W_5] = \Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathcal{E}',\mathcal{A}'}].$$

We find that there exists a PPT algorithms $\mathcal{B}$ and $\mathcal{B}'$ such that

$$|\Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\mathcal{E},\hat{\mathcal{A}}}] - \Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}]| \\ \leq (Q_I + 2Q) \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda) + \mathbf{Adv}_{\mathcal{B}'}^{\mathsf{LWE}}(\lambda) + \mathsf{negl}(\lambda),$$

which proves the lemma's statement. □

F.4 Security of the Database Against Colluding Issuer and Users

Lemma 11. *For any environment $\mathcal{E}$ and any adversary $\mathcal{A}$ controlling the issuer and some of the users, there is an ideal-world adversary $\mathcal{A}'$ such that the distributions $\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)$ and $\mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}(\lambda)$ are computationally indistinguishable under the* SIS *and* LWE *assumptions.*

Proof. The proof is very similar to the proof of Lemma 10 except that, since the adversary controls the issuer, it can obtain any credential of its choice and we do not have to rule out credential forgeries any longer. Again, for each i, W_i stands for the event that the environment $\mathcal{E}$ outputs 1 in Exp_i.

Exp_0: This corresponds to a real execution of the adversary $\mathcal{A}$ interacting with a honest sender and a honest issuer. The output of the experiment is the output of the environment $\mathcal{E}$. We have $\Pr[W_0] = \Pr[\mathsf{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)]$.

Exp_1: This experiment is like Exp_0 except that, when the user $\mathcal{U}$ sends the ciphertext $(\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ at step 2 of each transfer, Sim_1 appeals to the knowledge extractor of the argument system to extract witnesses $\mathfrak{m} \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B, B]^t$, $\tau \in \{0,1\}^\ell$, $\mathbf{v} \in \mathbb{Z}^{2m}$, $\mathfrak{m}_{\mathsf{U},\boldsymbol{x}} \in \{0,1\}^{m_I}$, $\mathsf{cert}_{\mathsf{U},\mathbf{x}}$ satisfying (19) at step 2 of $\mathsf{Transfer}$. Should the knowledge extraction fail at some transfer, Sim_1 aborts and outputs $\perp$. Given that the zero-knowledge argument system is computationally sound so long as the underlying commitment is binding. If the commitment scheme of [41] is used for this purpose, the SIS assumption reduces to the binding property. Under the SIS assumption, $\mathcal{E}$ returns 1 with very close probabilities in Exp_1 as in Exp_0: if Q denotes the total number of transfers, we have $|\Pr[W_1] - \Pr[W_0]| \leq Q \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$, for some SIS solver $\mathcal{B}$.

Exp_2: This experiment is the same as Exp_1 except that, after each knowledge extraction incurred by step 2 of $\mathsf{Transfer}$, Exp_2 aborts if the obtained witnesses

$\mathfrak{m} \in \{0,1\}^{m_d}$, $\tau \in \{0,1\}^{\ell}$ and $\mathbf{v} = (\mathbf{v}_1^\top | \mathbf{v}_2^\top)^\top \in \mathbb{Z}^{2m}$ satisfy the equality

$$\big[\mathbf{A} \mid \mathbf{A}_0 \mid \ldots \mid \mathbf{A}_\ell\big] \cdot \begin{bmatrix} \mathbf{v}_1 \\ \hline \mathbf{v}_2 \\ \hline \tau[1] \cdot \mathbf{v}_2 \\ \hline \cdots \\ \hline \tau[\ell] \cdot \mathbf{v}_2 \end{bmatrix} = \mathbf{u} + \mathbf{D} \cdot \mathfrak{m},$$

but the left hand side member of

$$\begin{bmatrix} \mathbf{a} \\ \hline \mathbf{b} \\ \hline \mathbf{h}_{\mathsf{BP}} \end{bmatrix} = \mathbf{H}_{2n+t,q-1} \cdot \mathfrak{m} \in \mathbb{Z}_q^{2n+t}$$

differs from all tuples $\{(\mathbf{a}_i, \mathbf{b}_i, \mathbf{h}_{\mathsf{BP},i})\}_{i=1}^N$ in DB. This contradicts the bounded message security of the signature scheme in Section 3.2. Under the SIS assumption, Exp_2 and Exp_1 are thus computationally indistinguishable as there is a PPT algorithm $\mathcal{B}$ such that $|\Pr[W_2] - \Pr[W_1]| \leq Q \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$.

Exp_3: This experiment is the same as Exp_2 but that, at the first $\mathsf{Transfer}$ query dictated by the environment $\mathcal{E}$, the zero-knowledge argument of knowledge of $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$, $\bar{\mathbf{x}}_j \in \chi^N$, $\bar{M}_j \in \{0,1\}^N$ such that

$$\left[\begin{array}{c|c|c|c} \mathbf{F}^\top & \mathbf{I}_m & & \\ \hline \mathbf{A}_{\mathrm{DB}}^\top & & \mathbf{I}_N & \lfloor q/2 \rfloor \cdot \mathbf{I}_N \end{array}\right] \cdot \begin{bmatrix} \mathbf{s}_j \\ \hline \mathbf{e}_j \\ \hline \bar{\mathbf{x}}_j \\ \hline \bar{M}_j \end{bmatrix} = \begin{bmatrix} \mathbf{p}_j \\ \hline \bar{\mathbf{b}}_j \end{bmatrix} \qquad \forall j \in [t]$$

(which appears at step 1 of $\mathsf{Transfer}$) is simulated (i.e., the ZK simulator generates it without knowing the witnesses) and so is the ZK argument of knowledge of $\{(\mathbf{s}_j, \mathbf{e}_j, \mathbf{y}[j])\}_{j=1}^t$ satisfying (17) at step 3 of $\mathsf{Transfer}$. The small-norm matrices $\mathbf{S} = [\mathbf{s}_1 | \ldots | \mathbf{s}_t] \in \mathbb{Z}^{n \times t}$, $\mathbf{E} = [\mathbf{e}_1 | \ldots | \mathbf{e}_t] \in \chi^{m \times t}$ satisfying (8) are not used any longer by Sim_3. The statistical ZK property of the argument system guarantees that $|\Pr[W_3] - \Pr[W_2]| \leq \mathsf{negl}(\lambda)$.

Exp_4: Here, at step 3 of $\mathsf{DBSetup}$, Sim_4 replaces the columns $\mathbf{p}_i$ of the matrix $\mathbf{P} = [\mathbf{p}_1 \mid \ldots \mid p_t] = \mathbf{F}^\top \cdot \mathbf{S} + \mathbf{E} \in \mathbb{Z}_q^{m \times t}$ by truly random vectors $\mathbf{p}_i \leftarrow U(\mathbb{Z}_q^m)$. Which results in matrix $\mathbf{P}$ being uniform in $\mathbb{Z}_q^{m \times t}$. Simultaneously, Sim_4 trades the vectors $\mathbf{b}_i = \mathbf{S}^\top \cdot \mathbf{a}_i + \mathbf{x}_i + M_i \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t$ for uniformly random vectors in $\mathbb{Z}_q^t$. At this point, the sender's zero-knowledge arguments become simulated arguments for false statements at steps 1 and 3 of $\mathsf{Transfer}$. However, under the LWE assumption, these changes are not noticeable to $\mathcal{A}$ and should leave $\mathcal{E}$'s output distribution essentially unchanged: we have $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(\lambda)$.

From the real-world $\mathcal{A}$, we build an ideal-world $\mathcal{A}'$ that controls the issuer I and the users. In its interaction with the real-world $\mathcal{A}$ (which controls I and U

in Exp_4), it plays the role of the honest DB. In the $\mathsf{DBSetup}$ phase, $(\mathbf{F}, \mathbf{P})$ are chosen as truly random matrices $(\mathbf{F}, \mathbf{P}) \leftarrow U(\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{m \times t})$ and while each pair $(\mathbf{a}_i, \mathbf{b}_i) \leftarrow U(\mathbb{Z}_q^n \times \mathbb{Z}_q^t)$ is sampled at random for all $i \in [N]$. The triple pairs $\{(\mathbf{a}_i, \mathbf{b}_i, \mathbf{h}_{\mathsf{B}_{2m\delta_{p-1}},i})\}_{i=1}^N$ are faithfully signed using $SK_{sig} = \mathbf{T_A}$ at step 5.b of $\mathsf{DBSetup}$. At the first $\mathsf{Transfer}$ message sent by T, $\mathcal{A}'$ runs the simulator of the ZK argument. When $\mathcal{A}$ sends $(\mathbf{c}_0, \mathbf{c}_1)$ and argues knowledge of $(\mathfrak{m}, \mathbf{e}, \mu, \nu, \tau, \mathbf{v}_1, \mathbf{v}_2)$ at step 2 of the transfer phase, $\mathcal{A}'$ uses the knowledge extractor of the argument system to extract $(\mathfrak{m}, \mathbf{e}, \mu, \nu, \tau, \mathbf{v}_1, \mathbf{v}_2) \in \{0,1\}^{m_d} \times \{-1,0,1\}^t \times \{0,1\}^t \times [-B, B]^t \times \{0,1\}^\ell$ and determine the index $\rho_i \in [N]$ such that

$$\left[\begin{array}{c} \mathbf{a}_{\rho_i} \\ \hline \mathbf{b}_{\rho_i} \\ \hline \mathbf{h}_{\mathsf{BP},i} \end{array}\right] = \mathbf{H}_{2n+t,q-1} \cdot \mathfrak{m} \in \mathbb{Z}_q^{2n+t}.$$

Note that, due to the change introduced in Exp_3, such an index must exist. Having extracted the queried index $\rho_i \in [N]$, $\mathcal{A}'$ queries the trusted party T to obtain credentials for an $\mathbf{x} \in \{0,1\}^\kappa$ such that $\mathsf{BP}_{\rho_i}(\mathbf{x}) = 1$. Since $\mathcal{A}'$ controls the ideal-world I, it sends the bit 1 to T so as to deliver all the necessary credentials to the user. Then, $\mathcal{A}'$ queries T to obtain $M_{\rho_i} \in \{0,1\}^t$. The latter is used together with the extracted witness $\mu \in \{0,1\}^t$ to reverse-engineer the correct response $M' = M_{\rho_i} \oplus \mu \in \{0,1\}^t$ that $\mathcal{A}'$ should return on behalf of DB at step 3 of the transfer. In addition, the ideal-world $\mathcal{A}'$ runs simulator of the argument system so as to simulate an argument of knowledge of $\{(\mathbf{s}_j, \mathbf{e}_j, \mathbf{y}[j])\}_{j=1}^t$ for the statement 17.

When $\mathcal{A}$ and $\mathcal{E}$ interact with the simulator $\mathcal{A}'$ emulating the real-world DB while playing the ideal-world U and I, their view is identical to that of Exp_4: we have $\Pr[W_4] = \Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathcal{E}',\mathcal{A}'}]$. Putting the above altogether, we find that there exists a PPT algorithms $\mathcal{B}$ and $\mathcal{B}'$ such that

$$\begin{aligned} &|\Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\mathcal{E},\hat{\mathcal{A}}}] - \Pr[\mathcal{E}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}]| \\ &\qquad \leq 2Q \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda) + \mathbf{Adv}_{\mathcal{B}'}^{\mathsf{LWE}}(\lambda) + \mathsf{negl}(\lambda). \end{aligned}$$

□

G Analysis of the Abstract Protocol

We first restate Theorem 4.

Theorem 9. *The protocol in Figure 1 is a statistical* ZKAoK *with perfect completeness, soundness error* $2/3$*, and communication cost* $\mathcal{O}(D \log q)$*. Namely:*

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

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

We now prove that the protocol is a statistical zero-knowledge argument of knowledge.

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

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

Case $\overline{Ch} = 1$: Using basic linear algebra over $\mathbb{Z}_q$, SIM computes a vector $\mathbf{w}' \in \mathbb{Z}_q^D$ such that $\mathbf{M} \cdot \mathbf{w}' = \mathbf{v} \bmod q$. Next, it samples $\mathbf{r}_w \leftarrow U(\mathbb{Z}_q^D)$, $\phi \leftarrow U(\mathcal{S})$, and randomness ρ_1, ρ_2, ρ_3 for COM. Then, it sends the commitment $\mathrm{CMT} = \big(C_1', C_2', C_3'\big)$ to $\widehat{\mathcal{V}}$, where

$$\begin{gathered} C_1' = \mathsf{COM}(\phi, \mathbf{M} \cdot \mathbf{r}_w; \rho_1), \\ C_2' = \mathsf{COM}(\mathit{\Gamma}_\phi(\mathbf{r}_w); \rho_2), \quad C_3' = \mathsf{COM}(\mathit{\Gamma}_\phi(\mathbf{w}' + \mathbf{r}_w); \rho_3). \end{gathered}$$

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

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

Case $\overline{Ch} = 2$: SIM samples $\mathbf{w}' \leftarrow U(\mathsf{VALID})$, $\mathbf{r}_w \leftarrow U(\mathbb{Z}_q^D)$, $\phi \leftarrow U(\mathcal{S})$, and randomness ρ_1, ρ_2, ρ_3 for COM. Then it sends the commitment $\mathrm{CMT} = \big(C_1', C_2', C_3'\big)$ to $\widehat{\mathcal{V}}$, where

$$\begin{gathered} C_1' = \mathsf{COM}(\phi, \mathbf{M} \cdot \mathbf{r}_w; \rho_1), \\ C_2' = \mathsf{COM}(\mathit{\Gamma}_\phi(\mathbf{r}_w); \rho_2), \quad C_3' = \mathsf{COM}(\mathit{\Gamma}_\phi(\mathbf{w}' + \mathbf{r}_w); \rho_3). \end{gathered}$$

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

- If $Ch = 1$: Send $\mathrm{RSP} = \big(\mathit{\Gamma}_\phi(\mathbf{w}'), \mathit{\Gamma}_\phi(\mathbf{r}_w), \rho_2, \rho_3\big)$.
- If $Ch = 2$: Output $\perp$ and abort.
- If $Ch = 3$: Send $\mathrm{RSP} = \big(\phi, \mathbf{r}_w, \rho_1, \rho_2\big)$.

Case $\overline{Ch} = 3$: SIM samples $\mathbf{w}' \leftarrow U(\mathsf{VALID})$, $\mathbf{r}_w \leftarrow U(\mathbb{Z}_q^D)$, $\phi \leftarrow U(\mathcal{S})$, and randomness ρ_1, ρ_2, ρ_3 for COM. Then it sends the commitment $\mathrm{CMT} = \big(C_1', C_2', C_3'\big)$ to $\widehat{\mathcal{V}}$, where $C_2' = \mathsf{COM}(\mathit{\Gamma}_\phi(\mathbf{r}_w); \rho_2)$, $C_3' = \mathsf{COM}(\mathit{\Gamma}_\phi(\mathbf{w}' + \mathbf{r}_w); \rho_3)$ as in the previous two cases, while

$$C_1' = \mathsf{COM}(\phi, \mathbf{M} \cdot (\mathbf{w}' + \mathbf{r}_w) - \mathbf{v}; \rho_1).$$

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

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

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

Argument of Knowledge. Suppose that $\mathrm{RSP}_1 = (\mathbf{t}_w, \mathbf{t}_r, \rho_2, \rho_3)$, $\mathrm{RSP}_2 = (\phi_2, \mathbf{w}_2, \rho_1, \rho_3)$, $\mathrm{RSP}_3 = (\phi_3, \mathbf{w}_3, \rho_1, \rho_2)$ are 3 valid responses to the same commitment $\mathrm{CMT} = (C_1, C_2, C_3)$, with respect to all 3 possible values of the challenge. The validity of these responses implies that:

$$\begin{cases} \mathbf{t}_w \in \mathsf{VALID}; \\ C_1 = \mathsf{COM}(\phi_2, \mathbf{M} \cdot \mathbf{w}_2 - \mathbf{v} \bmod q; \rho_1) = \mathsf{COM}(\phi_3, \mathbf{M} \cdot \mathbf{w}_3; \rho_1); \\ C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2) = \mathsf{COM}(\varGamma_{\phi_3}(\mathbf{w}_3); \rho_2); \\ C_3 = \mathsf{COM}(\mathbf{t}_w + \mathbf{t}_r \bmod q; \rho_3) = \mathsf{COM}(\varGamma_{\phi_2}(\mathbf{w}_2); \rho_3). \end{cases}$$

Since COM is computationally binding, we can deduce that

$$\begin{cases} \mathbf{t}_w \in \mathsf{VALID};\ \phi_2 = \phi_3;\ \mathbf{t}_r = \varGamma_{\phi_3}(\mathbf{w}_3);\ \mathbf{t}_w + \mathbf{t}_r = \varGamma_{\phi_2}(\mathbf{w}_2) \bmod q; \\ \mathbf{M} \cdot \mathbf{w}_2 - \mathbf{v} = \mathbf{M} \cdot \mathbf{w}_3 \bmod q. \end{cases} \tag{51}$$

Since $\mathbf{t}_w \in \mathsf{VALID}$, if we let $\mathbf{w}' = [\varGamma_{\phi_2}]^{-1}(\mathbf{t}_w)$, then $\mathbf{w}' \in \mathsf{VALID}$. Furthermore, we have

$$\varGamma_{\phi_2}(\mathbf{w}') + \varGamma_{\phi_2}(\mathbf{w}_3) = \varGamma_{\phi_2}(\mathbf{w}_2) \bmod q,$$

which implies that $\mathbf{w}' + \mathbf{w}_3 = \mathbf{w}_2 \bmod q$, and that $\mathbf{M} \cdot \mathbf{w}' + \mathbf{M} \cdot \mathbf{w}_3 = \mathbf{M} \cdot \mathbf{w}_2 \bmod q$. As a result, we have $\mathbf{M} \cdot \mathbf{w}' = \mathbf{v} \bmod q$. This concludes the proof. □

H More Details of Protocol 4

H.1 The Underlying Commitment and Merkle Hash Tree

Here, we recall the bit commitment and the Merkle hash tree used in Protocol 4 (Section 6.6). Both schemes work with security parameter λ, lattice parameter $n = \mathcal{O}(\lambda)$, modulus q and dimension $m = 2n\lceil \log q \rceil$.

Commitment. The bit commitment we use is a variant of the scheme proposed by Kawachi et al. [41]. It is statistically hiding (based on the left-over hash lemma [58]) and computationally binding (based on the SIS assumption).

The commitment key consists of vector $\mathbf{a}_{\mathsf{com}} \hookleftarrow U(\mathbb{Z}_q^n)$ and matrix $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$. To commit to a bit $x \in \{0,1\}$, one samples $\mathbf{r}_{\mathsf{com}} \hookleftarrow U(\{0,1\}^m)$, and outputs the commitment

$$\mathsf{com} = \mathbf{a}_{\mathsf{com}} \cdot x + \bar{\mathbf{A}} \cdot \mathbf{r}_{\mathsf{com}} \in \mathbb{Z}_q^n.$$

For uniformly random $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times m}$ and uniformly random $\mathbf{r}_{\mathsf{com}} \in \{0,1\}^m$, the distribution of vector $\bar{\mathbf{A}} \cdot \mathbf{r}_{\mathsf{com}} \in \mathbb{Z}_q^n$ is statistically close to the uniform distribution over $\mathbb{Z}_q^n$ (by the left-over hash lemma). Thus, the commitment com statistically hides the committed bit x.

Meanwhile, if one can find $x', x'' \in \{0,1\}$ and $\mathbf{r}', \mathbf{r}'' \in \{0,1\}^m$, such that $x' \neq x''$ and

$$\mathbf{a}_{\mathsf{com}} \cdot x' + \bar{\mathbf{A}} \cdot \mathbf{r}' = \mathbf{a}_{\mathsf{com}} \cdot x'' + \bar{\mathbf{A}} \cdot \mathbf{r}'' \bmod q,$$

then the non-zero vector $\begin{pmatrix} x' - x'' \\ \mathbf{r}' - \mathbf{r}'' \end{pmatrix} \in \{-1,0,1\}^{m+1}$ yields a valid solution to the $\mathsf{SIS}_{n,m+1,q,\sqrt{m+1}}$ problem associated with matrix $\left[\mathbf{a}_{\mathsf{com}} \mid \bar{\mathbf{A}}\right] \in \mathbb{Z}_q^{n\times(m+1)}$. In other words, an adversary breaking the computational binding property of the scheme leads to an adversary breaking the SIS assumption.

Merkle hash tree. We now recall SIS-based Merkle tree proposed by Libert et al. [44]. The construction relies on the following collision-resistant hash function.

For matrix $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$, expressed as $\bar{\mathbf{A}} = [\mathbf{A}_0 \mid \mathbf{A}_1]$, where $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_q^{n\times n\lceil \log q\rceil}$, define the function $h_{\bar{\mathbf{A}}}$ as follows:

$$\begin{aligned} h_{\bar{\mathbf{A}}} : \{0,1\}^{n\lceil \log q\rceil} \times \{0,1\}^{n\lceil \log q\rceil} &\to \{0,1\}^{n\lceil \log q\rceil} \\ (\mathbf{u}_0, \mathbf{u}_1) &\mapsto \mathsf{vdec}_{n,q-1}\big(\mathbf{A}_0 \cdot \mathbf{u}_0 + \mathbf{A}_1 \cdot \mathbf{u}_1 \bmod q\big). \end{aligned}$$

Note that $h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1) = \mathbf{u} \Leftrightarrow \mathbf{A}_0 \cdot \mathbf{u}_0 + \mathbf{A}_1 \cdot \mathbf{u}_1 = \mathbf{H}_{n,q-1} \cdot \mathbf{u} \bmod q$. This hash function is collision-resistant under the SIS assumption. Indeed, if two pairs $(\mathbf{u}_0', \mathbf{u}_1')$ and $(\mathbf{u}_0'', \mathbf{u}_1'')$ cause a collision, then one has:

$$\begin{aligned} &\mathbf{A}_0 \cdot \mathbf{u}_0' + \mathbf{A}_1 \cdot \mathbf{u}_1' = \mathbf{A}_0 \cdot \mathbf{u}_0'' + \mathbf{A}_1 \cdot \mathbf{u}_1'' \bmod q \\ \Leftrightarrow\ &\mathbf{A} \cdot \mathbf{z} = \mathbf{0} \bmod q, \end{aligned}$$

where $\mathbf{z} = \begin{pmatrix} \mathbf{u}_0' - \mathbf{u}_0'' \\ \mathbf{u}_1' - \mathbf{u}_1'' \end{pmatrix} \in \{-1,0,1\}^m$ is a non-zero vector. Thus, $\mathbf{z}$ yields a valid solution to the $\mathsf{SIS}_{n,m,q,\sqrt{m}}$ problem associated with matrix $\bar{\mathbf{A}}$.

Using the hash function described above, one can build Merkle hash trees to securely accumulate data. In particular, for any $\kappa \in \mathsf{poly}(\lambda)$, and for any data values $\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1} \in \mathbb{Z}_q^n$, one builds the tree of depth $\delta_\kappa = \lceil \log \kappa \rceil$ on top of these values, as follows.

Leaf nodes: Since 2^{δ_κ} leaf nodes are needed to build a complete binary tree of depth δ_κ, one appends $2^{\delta_\kappa} - \kappa$ duplications of the value $\mathsf{com}_{\kappa-1}$ to the sequence $\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1}$ so that the appended sequence has length 2^{δ_κ}.

Applying the function $\mathsf{vdec}_{n,q-1}(\cdot)$ to elements of the appended sequence, one gets 2^{δ_κ} values denoted as $\mathbf{u}_{0,0,\ldots,0}, \ldots, \mathbf{u}_{1,1,\ldots,1} \in \{0,1\}^{n\lceil \log q\rceil}$. These values are associated with the 2^{δ_κ} leaves of the tree.

Non-leaf nodes: Every non-leaf node of the tree is associated with the hash value of its two children. Specifically:

1. At depth $i \in [\delta_\kappa]$, the node $\mathbf{u}_{b_1,\ldots,b_i} \in \{0,1\}^{n\lceil \log q \rceil}$, for all $(b_1,\ldots,b_i) \in \{0,1\}^i$, is defined as $h_{\bar{\mathbf{A}}}(\mathbf{u}_{b_1,\ldots,b_i,0}, \mathbf{u}_{b_1,\ldots,b_i,1})$.
2. At depth 0: The root $\mathbf{u}_{\mathsf{tree}} \in \{0,1\}^{n\lceil \log q \rceil}$ is defined as $h_{\bar{\mathbf{A}}}(\mathbf{u}_0, \mathbf{u}_1)$.

Note that, the collision resistance of the hash function $h_{\bar{\mathbf{A}}}$ guarantees that it is infeasible to find a tree path starting from $\mathbf{u}_{\mathsf{tree}}$ and ending with two distinct leaf nodes.

Now, for any integer $\mathsf{var}(\theta) \in [0, \kappa - 1]$, the witness for the fact that $\mathsf{com}_{\mathsf{var}(\theta)}$ was properly accumulated into $\mathbf{u}_{\mathsf{tree}}$ consists of:

- The binary representation $(d_{\theta,1}, \ldots, d_{\theta,\delta_\kappa})$ of $\mathsf{var}(\theta)$;
- The nodes $\mathbf{g}_{\theta,1}, \ldots, \mathbf{g}_{\theta,\delta_\kappa} \in \{0,1\}^{n\lceil \log q \rceil}$ on the tree path determined by $(d_{\theta,1}, \ldots, d_{\theta,\delta_\kappa})$, as well as their sibling nodes $\mathbf{t}_{\theta,1}, \ldots, \mathbf{t}_{\theta,\delta_\kappa} \in \{0,1\}^{n\lceil \log q \rceil}$. (Note that $\mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,\delta_\kappa} = \mathsf{com}_{\mathsf{var}(\theta)} \bmod q$.)

The witness satisfies the relations described below.

At the root, if $d_{\theta,1} = 0$ then one has $\mathbf{u}_{\mathsf{tree}} = h_{\bar{\mathbf{A}}}(\mathbf{g}_{\theta,1}, \mathbf{t}_{\theta,1})$; if $d_{\theta,1} = 1$, then $\mathbf{u}_{\mathsf{tree}} = h_{\bar{\mathbf{A}}}(\mathbf{t}_{\theta,1}, \mathbf{g}_{\theta,1})$. In other words, one has:

$$\begin{aligned}
&\bar{d}_{\theta,1} \cdot h_{\bar{\mathbf{A}}}(\mathbf{g}_{\theta,1}, \mathbf{t}_{\theta,1}) + d_{\theta,1} \cdot h_{\bar{\mathbf{A}}}(\mathbf{t}_{\theta,1}, \mathbf{g}_{\theta,1}) = \mathbf{u}_{\mathsf{tree}} \\
\Leftrightarrow\; &\bar{d}_{\theta,1} \cdot \big(\mathbf{A}_0 \cdot \mathbf{g}_{\theta,1} + \mathbf{A}_1 \cdot \mathbf{t}_{\theta,1}\big) + d_{\theta,1} \cdot \big(\mathbf{A}_0 \cdot \mathbf{t}_{\theta,1} + \mathbf{A}_1 \cdot \mathbf{g}_{\theta,1}\big) = \mathbf{H}_{n,q-1} \cdot \mathbf{u}_{\mathsf{tree}} \bmod q \\
\Leftrightarrow\; &\big[\mathbf{A}_0 \mid \mathbf{A}_1\big] \cdot \begin{pmatrix} \bar{d}_{\theta,1} \cdot \mathbf{g}_{\theta,1} \\ d_{\theta,1} \cdot \mathbf{g}_{\theta,1} \end{pmatrix} + \big[\mathbf{A}_0 \mid \mathbf{A}_1\big] \cdot \begin{pmatrix} d_{\theta,1} \cdot \mathbf{t}_{\theta,1} \\ \bar{d}_{\theta,1} \cdot \mathbf{t}_{\theta,1} \end{pmatrix} = \mathbf{H}_{n,q-1} \cdot \mathbf{u}_{\mathsf{tree}} \bmod q \\
\Leftrightarrow\; &\bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,1}, \mathbf{g}_{\theta,1}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,1}, \mathbf{t}_{\theta,1}) = \mathbf{H}_{n,q-1} \cdot \mathbf{u}_{\mathsf{tree}} \bmod q,
\end{aligned}$$

where $\mathsf{expand}(\cdot,\cdot)$ is defined in Table 1. Similarly, going along the path, one has equations:

$$\begin{cases}
\bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,2}, \mathbf{g}_{\theta,2}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,2}, \mathbf{t}_{\theta,2}) - \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,1} = \mathbf{0} \bmod q, \\
\vdots \\
\bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,\delta_\kappa}, \mathbf{g}_{\theta,\delta_\kappa}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,\kappa}, \mathbf{t}_{\theta,\kappa}) - \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,\delta_\kappa - 1} = \mathbf{0} \bmod q.
\end{cases}$$

Additionally, if $\mathbf{g}_{\theta,\delta_\kappa}$ is the binary representation of a commitment to a bit y_θ with randomness $\mathbf{r}_\theta$ (where the underlying commitment scheme is the one discussed above), then one eventually has:

$$\begin{cases}
\bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,1}, \mathbf{g}_{\theta,1}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,1}, \mathbf{t}_{\theta,1}) = \mathbf{H}_{n,q-1} \cdot \mathbf{u}_{\mathsf{tree}} \bmod q, \\
\bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,2}, \mathbf{g}_{\theta,2}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,2}, \mathbf{t}_{\theta,2}) - \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,1} = \mathbf{0} \bmod q, \\
\vdots \\
\bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,\delta_\kappa}, \mathbf{g}_{\theta,\delta_\kappa}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,\kappa}, \mathbf{t}_{\theta,\kappa}) - \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,\delta_\kappa - 1} = \mathbf{0} \bmod q, \\
\mathbf{a}_{\mathsf{com}} \cdot y_\theta + \bar{\mathbf{A}} \cdot \mathbf{r}_\theta - \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,\delta_\kappa} = \mathbf{0} \bmod q.
\end{cases}$$

This explains the equations considered in the "Merkle Tree Step" of Protocol 4 (Section 6.6).

H.2 Deferred Description Details of Protocol 4: VALID, $\mathcal{S}$ and $\mathit{\Gamma}_\phi$

Let VALID be the set of all vectors in $[-1,4]^D$ having the form $(\mathbf{w}_{\mathsf{tree}}^T \| \mathbf{w}_{\mathsf{BP}}^T \| \mathbf{w}_3^T)^T$, where $\mathbf{w}_{\mathsf{tree}}, \mathbf{w}_{\mathsf{BP}}, \mathbf{w}_3$ have the form (22), (25), and (29), respectively, and the following conditions hold:

- For each $(\theta, i) \in [L] \times [\delta_\kappa]$: There exists $d_{\theta,i} \in \{0,1\}$ and $\widetilde{\mathbf{g}}_{\theta,i}, \widetilde{\mathbf{t}}_{\theta,i} \in \mathsf{B}^2_{m/2}$ such that $\mathbf{d}_{\theta,i} = \mathsf{ext}_2(d_{\theta,i})$ and
$$\widehat{\mathbf{g}}_{\theta,i} = \mathsf{expand}(d_{\theta,i}, \widetilde{\mathbf{g}}_{\theta,i}); \quad \widehat{\mathbf{t}}_{\theta,i} = \mathsf{expand}(\bar{d}_{\theta,i}, \widetilde{\mathbf{t}}_{\theta,i}).$$
- There exist $y_1 \in \{0,1\}$ and $\pi_{1,0}(0), \pi_{1,1}(0) \in [0,4]$ such that
$$\begin{cases} \mathbf{y}_1 = \mathsf{ext}_2(y_1); \quad \mathbf{h}_{1,0} = \mathsf{ext}_{5\times 2}(\pi_{1,0}(0), \bar{y}_1); \quad \mathbf{h}_{1,1} = \mathsf{ext}_{5\times 2}(\pi_{1,1}(0), y_1); \\ \mathit{\Pi}_{1,0,0} = \mathsf{ext}_5(\pi_{1,0}(0)); \quad \mathit{\Pi}_{1,1,0} = \mathsf{ext}_5(\pi_{1,1}(0)). \end{cases}$$
- For all $(j,i) \in \{0,1\} \times [1,4]$: $\quad \mathit{\Pi}_{1,j,i} = \mathsf{ext}_5(\pi_{1,j}(i))$, for *some* $\pi_{1,j}(i) \in [0,4]$. (Note that these $\pi_{1,j}(i)$ do *not* participate in the evaluation of BP.)
- For $\theta \in [2, L]$: There exist $y_\theta \in \{0,1\}$, $f_{\theta,0}, f_{\theta,1} \in [0,4]$ such that $\mathbf{y}_\theta = \mathsf{ext}_2(y_\theta)$ and
$$\begin{cases} \mathbf{f}_{\theta,0} = \mathsf{ext}_5(f_{\theta,0}); \quad \mathbf{f}_{\theta,1} = \mathsf{ext}_5(f_{\theta,1}); \\ \mathbf{h}_{\theta,0} = \mathsf{ext}_{5\times 2}(f_{\theta,0}, \bar{y}_\theta); \quad \mathbf{h}_{\theta,1} = \mathsf{ext}_{5\times 2}(f_{\theta,1}, y_\theta). \end{cases}$$
- For $(\theta, j, i) \in [2, L] \times \{0,1\} \times [0,4]$: there exist $\pi_{\theta,j}(i) \in [0,4]$, $c_{\theta,i} \in \{0,1\}$ such that
$$\mathit{\Pi}_{\theta,j,i} = \mathsf{ext}_5(\pi_{\theta,j}(i)); \quad \mathbf{c}_{\theta,i} = \mathsf{ext}_2(c_{\theta,i}); \quad \mathbf{z}_{\theta,j,i} = \mathsf{ext}_{5\times 2}(\pi_{\theta,j}(i), c_{\theta,i}).$$
- There exist $\eta_1, \ldots, \eta_{L-1} \in [0,4]$ such that the following hold.
  1. For all $\theta = 1, \ldots, L-1$: $\mathbf{s}_\theta = \mathsf{ext}_5(\eta_\theta)$.
  2. For all $\theta = 2, \ldots, L$: $\mathbf{e}_\theta = \mathsf{unit}_{\eta_{\theta-1}}$.
- $\widehat{\mathbf{r}} \in \mathsf{B}^2_{mL}$; $\quad \mathbf{w}_{3,1} \in \mathsf{B}^2_{D_{3,1}}$; $\quad \mathbf{w}_{3,2} \in \mathsf{B}^3_{D_{3,2}}$;
- $\mathbf{s}_0 \in \mathsf{B}^3_{m\delta_\beta}$, and there exists $\tau = (\tau[1], \ldots, \tau[\ell])^T \in \{0,1\}^\ell$ such that for all $j \in [\ell]$: $\mathsf{s}_j = \mathsf{expand}\big(\tau[j], \mathsf{s}_0\big)$.
- $\mathbf{s}_{\mathsf{U},0} \in \mathsf{B}^3_{m\delta_\beta}$, and there exists $\tau_{\mathsf{U}} = (\tau_{\mathsf{U}}[1], \ldots, \tau_{\mathsf{U}}[\ell_I])^T \in \{0,1\}^{\ell_I}$ such that for all $j \in [\ell_I]$: $\mathsf{s}_{\mathsf{U},j} = \mathsf{expand}\big(\tau_{\mathsf{U}}[j], \mathsf{s}_{\mathsf{U},0}\big)$.

By construction, we have $\mathbf{w} \in \mathsf{VALID}$. Let us now specify the set $\mathcal{S}$ and permutations of D elements $\{\mathit{\Gamma}_\phi : \phi \in \mathcal{S}\}$, for which the conditions in (12) hold. Again, we refer to the notations and techniques from Table 1, which we will apply here. Let

$$\mathcal{S} = \{0,1\}^{L\delta_\kappa} \times \{0,1\}^L \times [0,4]^{10L} \times [0,4]^{L-1} \times \{0,1\}^{5(L-1)} \times [0,4]^{2(L-1)} \times \big((\mathcal{S}_m)^{2L\delta_\kappa} \times \mathcal{S}_{2mL}\big) \times (\mathcal{S}_{2D_{3,1}} \times \mathcal{S}_{3D_{3,2}} \times (\mathcal{S}_{3m\delta_\beta} \times \{0,1\}^\ell) \times (\mathcal{S}_{3m\delta_\beta} \times \{0,1\}^{\ell_I})).$$

For each $\phi = (\mathbf{b}_d, \mathbf{b}_y, \mathbf{b}_p, \mathbf{b}_c, \mathbf{b}_f, \Sigma_{\mathsf{tree}}, \Sigma_3) \in \mathcal{S}$, where:

$$\begin{cases} \mathbf{b}_d = (b_{d,1,1}, \ldots, b_{d,L,\delta_\kappa})^T, \quad \mathbf{b}_y = (b_{y,1}, \ldots, b_{y,L})^T, \\ \mathbf{b}_\eta = (b_{\eta,1}, \ldots, b_{\eta,L-1})^T, \;\; \mathbf{b}_c = (b_{c,2,0}, \ldots, b_{c,L,4})^T, \;\; \mathbf{b}_f = (b_{f,2,0}, \ldots, b_{f,L,1})^T, \\ \mathbf{b}_p = (b_{\pi,1,0,0}, \ldots, b_{\pi,L,1,4})^T, \;\; \Sigma_{\mathsf{tree}} = (\sigma_{g,1,1}, \ldots, \sigma_{g,\theta,\delta_\kappa}, \sigma_{t,1,1}, \ldots, \sigma_{t,\theta,\delta_\kappa}, \sigma_r), \\ \Sigma_3 = (\sigma_{3,1},\, \sigma_{3,2},\, \sigma_{3,3,1},\, b_\tau[1] \ldots b_\tau[\ell],\, \sigma_{3,3,2},\, b_{\tau,\mathsf{U}}[1] \ldots b_{\tau,\mathsf{U}}[\ell_I]), \end{cases}$$

let Γ_ϕ be the permutation that, when applying to vector $\mathbf{s}$ having the form $(\mathbf{w}_{\mathsf{tree}}^T \| \mathbf{w}_{\mathsf{BP}}^T \| \mathbf{w}_3^T)^T \in \mathbb{Z}^D$, where $\mathbf{w}_{\mathsf{tree}}, \mathbf{w}_{\mathsf{BP}}, \mathbf{w}_3$ have the form (22), (25), and (29), respectively, it transforms the block-vectors as follows:

- For each $(\theta, i) \in [L] \times [\delta_\kappa]$: $\widetilde{\mathbf{g}}_{\theta,i} \mapsto \sigma_{g,\theta,i}(\widetilde{\mathbf{g}}_{\theta,i})$ and
$$\widehat{\mathbf{g}}_{\theta,i} \mapsto T_{\mathsf{exp}}[b_{d,\theta,i}, \sigma_{g,\theta,i}](\widehat{\mathbf{g}}_{\theta,i}); \;\; \widehat{\mathbf{t}}_{\theta,i} \mapsto T_{\mathsf{exp}}[b_{d,\theta,i}, \sigma_{t,\theta,i}](\widehat{\mathbf{t}}_{\theta,i}).$$
- For each $\theta \in [L]$: $\mathbf{y}_\theta \mapsto T_2[b_{y,\theta}](\mathbf{y}_\theta)$; $\widehat{\mathbf{r}} \mapsto \sigma_r(\widehat{\mathbf{r}})$.
- For each $\theta = 1, \ldots, L-1$: $\mathbf{s}_\theta \mapsto T_5[b_{\eta,\theta}](\mathbf{s}_\theta)$.
- For each $\theta = 2, \ldots, L$: $\mathbf{e}_\theta \mapsto T_5[b_{\eta,\theta-1}](\mathbf{e}_\theta)$.
- For each $(\theta, i) \in [2, L] \times [0, 4]$: $\mathbf{c}_{\theta,i} \mapsto T_2[b_{c,\theta,i}](\mathsf{c}_{\theta,i})$.
- For each $(\theta, j, i) \in [2, L] \times \{0, 1\} \times [0, 4]$: $\mathbf{z}_{\theta,j,i} \mapsto T_{5\times 2}[b_{\pi,\theta,j,i}, b_{c,\theta,i}](\mathbf{z}_{\theta,j,i})$.
- For each $(\theta, j) \in [2, L] \times \{0, 1\}$: $\mathbf{f}_{\theta,j} \mapsto T_5[b_{f,\theta,j}](\mathbf{f}_{\theta,j})$.
- $\mathbf{h}_{1,0} \mapsto T_{5\times 2}[b_{\pi,1,0,0}, b_{y,1}](\mathbf{h}_{1,0})$; $\mathbf{h}_{1,1} \mapsto T_{5\times 2}[b_{\pi,1,1,0}, b_{y,1}](\mathbf{h}_{1,1})$.
- For each $(\theta, j) \in [2, L] \times \{0, 1\}$: $\mathbf{h}_{\theta,j} \mapsto T_{5\times 2}[b_{f,\theta,j}, b_{y,\theta}](\mathbf{h}_{\theta,j})$.
- For each $(\theta, i) \in [L] \times [\delta_\kappa]$: $\mathbf{d}_{\theta,i} \mapsto T_2[b_{d,\theta,i}](\mathbf{d}_{\theta,i})$.
- For each $(\theta, j, i) \in [L] \times \{0, 1\} \times [0, 4]$: $\Pi_{\theta,j,i} \mapsto T_5[b_{\pi,\theta,j,i}](\Pi_{\theta,j,i})$.
- For each $i \in [1, 2]$: $\mathbf{w}_{3,i} \mapsto \sigma_{3,i}(\mathbf{w}_{3,i})$.
- $\mathsf{s}_0 \mapsto \sigma_{3,3,1}(\mathsf{s}_0)$. For each $j \in [\ell]$: $\mathbf{s}_j \mapsto T_{\mathsf{exp}}[b_\tau[j], \sigma_{3,3,1}](\mathbf{s}_j)$.
- $\mathsf{s}_{\mathsf{U},0} \mapsto \sigma_{3,3,2}(\mathsf{s}_{\mathsf{U},0})$. For each $j \in [\ell_I]$: $\mathsf{s}_{\mathsf{U},j} \mapsto T_{\mathsf{exp}}[b_{\tau,\mathsf{U}}[j], \sigma_{3,3,2}](\mathsf{s}_{\mathsf{U},j})$.

Based on the equivalences observed in Table 1, it can be checked that the conditions of (12) hold, namely:

$$\begin{cases} \mathbf{w} \in \mathsf{VALID} \iff \Gamma_\phi(\mathbf{w}) \in \mathsf{VALID}, \\ \text{If } \mathbf{w} \in \mathsf{VALID} \text{ and } \phi \text{ is uniform in } \mathcal{S}, \text{ then } \Gamma_\phi(\mathbf{w}) \text{ is uniform in } \mathsf{VALID}. \end{cases}$$

H.3 Notes on the Simulator and Knowledge Extractor of Protocol 4

We recall that Protocol 4 invokes the Stern-like protocol from Section 6.1. Here, we give some remarks on how the transcript simulator $\overline{\mathsf{SIM}}$ and knowledge extractor $\overline{\mathsf{KE}}$ of the former protocol work, based on those of the latter protocol (see Theorem 4) and the facts that the bit commitment scheme and the Merkle hash tree being used are secure. In a nutshell:

- $\overline{\mathsf{SIM}}$ relies on the simulator SIM of the abstract protocol and the statistically hiding property of the bit commitment scheme.
- $\overline{\mathsf{KE}}$ works by invoking the extractor KE of the abstract protocol, the computationally binding property of the bit commitment scheme and the collision resistance of the Merkle hash tree.

Transcript Simulation. In the real protocol, at the beginning of the interaction, the prover sends commitments $\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1} \in \mathbb{Z}_q^n$ to the verifier for once, then both parties construct matrix $\mathbf{M}$ and vector $\mathbf{v}$, and they run the protocol from Section 6.1 on common input $(\mathbf{M}, \mathbf{v})$. As noted in Section H.1 of this Supplementary Material, for each $i \in [0, \kappa - 1]$ the distribution of com_i is statistically close to $U(\mathbb{Z}_q^n)$.

The simulator $\overline{\mathsf{SIM}}$ interacts with a (possibly dishonest) verifier $\widehat{\mathcal{V}}$ as follows.

- At the beginning of the interaction, it samples $\mathsf{com}'_0, \ldots, \mathsf{com}'_{\kappa-1} \hookleftarrow U(\mathbb{Z}_q^n)$, and sends these vectors to $\widehat{\mathcal{V}}$. The distribution of $(\mathsf{com}'_0, \ldots, \mathsf{com}'_{\kappa-1})$ is thus statistically close to the distribution of $(\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1})$ in the real protocol. Both parties then construct the common input as in the real protocol.

- Next, $\overline{\mathsf{SIM}}$ proceeds exactly as the simulator SIM of the abstract protocol (see Section G of this Supplementary Material). Note that the latter produces an accepted transcript statistically close to that produced by the real prover with success probability negligibly close to 2/3.

It can be seen from the above construction that $\overline{\mathsf{SIM}}$ successfully simulates the honest prover of Protocol 4 with probability negligibly close to 2/3.

Knowledge Extraction. The knowledge extractor $\overline{\mathsf{KE}}$ takes as input the following:

- Matrix $\mathbf{M}$ and vector $\mathbf{v}$, which are built from $\mathbf{F}$, $\mathbf{P}$, $\mathbf{A}_{\mathrm{HBP}}$, $\mathbf{c}_0$, $\mathbf{c}_1$, $\mathbf{A}$, $\mathbf{A}_0$, $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$, $\mathbf{D}$, $\mathbf{u}$, $\mathbf{A}_I$, $\mathbf{A}_{I,0}$, $\mathbf{A}_{I,1}, \ldots, \mathbf{A}_{I,\ell_I}$, $\mathbf{D}_I$, $\mathbf{u}_I$, $\mathbf{D}_{I,0}$, $\mathbf{D}_{I,1}$, $\mathbf{a}_{\mathsf{com}}$, $\bar{\mathbf{A}}$, $\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1}$, as described in Section 6.6.
- A commitment CMT and 3 valid responses $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ to all 3 possible values of the challenge Ch, with respect to the abstract protocol.

By invoking the knowledge extractor KE of the abstract protocol, $\overline{\mathsf{KE}}$ obtains vector $\mathbf{w}' \in \mathsf{VALID}$ such that $\mathbf{M} \cdot \mathbf{w}' = \mathbf{v} \bmod q$. Next, it parses $\mathbf{w}'$ as

$$\mathbf{w} = \big((\mathbf{w}')_{\mathsf{tree}}^\top \mid (\mathbf{w}')_{\mathsf{BP}}^\top \mid (\mathbf{w}')_3^\top\big)^\top,$$

where $\mathbf{w}'_{\mathsf{tree}} \in \{0,1\}^{D_{\mathsf{tree}}}$, $\mathbf{w}'_{\mathsf{BP}} \in [0,4]^{D_{\mathsf{BP}}}$ and $\mathbf{w}'_3 \in [-1,4]^{D_3}$. Now, $\overline{\mathsf{KE}}$ backtracks the transformations of Section 6.6, as follows.

Backtracking the third step. Write $\mathbf{w}'_3 \in [-1,4]^{D_3}$ in the form of (29) and backtracks the transformations of the third steps, $\overline{\mathsf{KE}}$ obtains

$$\begin{cases} d'_{1,1},\ldots,d'_{L,\delta_\kappa} \in \{0,1\}, \pi'_{1,0}(0),\ldots,\pi'_{L,1}(4) \in [0,4], \\ \mathbf{x}' = (x'_0,\ldots,x'_{\kappa-1})^\top \in \{0,1\}^\kappa, \\ \mathfrak{m}'_{\mathsf{U},\mathbf{x}} \in \{0,1\}^{\frac{m}{2}+\kappa},\ \widehat{\mathfrak{m}}'_{\mathsf{U},\mathbf{x}} \in \{0,1\}^{\frac{m}{2}},\ \mathfrak{m}' \in \{0,1\}^{m_d},\ \mathbf{e}'_{\mathsf{U}} \in \{0,1\}^m, \\ \mathbf{r}'_{\mathsf{com},0},\ldots,\mathbf{r}'_{\mathsf{com},\kappa-1} \in \{0,1\}^m,\ \mu' \in \{0,1\}^t,\ \tau' \in \{0,1\}^\ell,\ \tau'_{\mathsf{U}} \in \{0,1\}^{\ell_I}, \\ \mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_{\mathsf{U},1}, \mathbf{v}'_{\mathsf{U},2}, \mathbf{r}'_{\mathsf{U}} \in [-\beta,\beta]^m,\ \mathbf{e}' \in \{-1,0,1\}^t,\ \nu' \in [-B,B]^t, \end{cases}$$

and $\mathbf{z}'_{\mathsf{BP},\rho} = (d'_{1,1},\ldots,d'_{L,\delta_\kappa},\pi'_{1,0}(0),\ldots,\pi'_{L,1}(4))^\top$, satisfying the relations (modulo q)

$$\begin{cases} \mathbf{H}_{2n+t,q-1} \cdot \mathfrak{m}' + \left[\begin{array}{c|c|c|c} \mathbf{F} & & & \\ \hline \mathbf{P}^\top & \mathbf{I}_t \cdot \lfloor q/2 \rfloor & \mathbf{I}_t & \\ \hline & & & -\mathbf{A}_{\mathrm{HBP}} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{e}' \\ \hline \mu' \\ \hline \nu' \\ \hline \mathbf{z}'_{\mathsf{BP},\rho} \end{array}\right] = \left[\begin{array}{c} \mathbf{c}_0 \\ \hline \mathbf{c}_1 \\ \hline \mathbf{0}^n \end{array}\right] \\ \mathbf{A} \cdot \mathbf{v}'_1 + \mathbf{A}_0 \cdot \mathbf{v}'_2 + \sum_{j=1}^{\ell} \mathbf{A}_j \cdot (\tau'[j] \cdot \mathbf{v}'_2) - \mathbf{D} \cdot \mathfrak{m}' = \mathbf{u} \\ \mathbf{A}_I \cdot \mathbf{v}'_{\mathsf{U},1} + \mathbf{A}_{I,0} \cdot \mathbf{v}'_{\mathsf{U},2} + \sum_{j=1}^{\ell_I} \mathbf{A}_{I,j} \cdot (\tau'_{\mathsf{U}}[j] \cdot \mathbf{v}'_{\mathsf{U},2}) - \mathbf{D}_I \cdot \widehat{\mathfrak{m}}'_{\mathsf{U},\mathbf{x}} = \mathbf{u}_I \\ \mathbf{D}_{I,0} \cdot \mathbf{r}'_{\mathsf{U}} + \mathbf{D}_{I,1} \cdot \mathfrak{m}'_{\mathsf{U},\mathbf{x}} - \mathbf{H}_{n,q-1} \cdot \widehat{\mathfrak{m}}'_{\mathsf{U},\mathbf{x}} = \mathbf{0} \\ \left[\begin{array}{c|c} \mathbf{H}_{n,q-1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_\kappa \end{array}\right] \cdot \mathfrak{m}'_{\mathsf{U},\mathbf{x}} + \begin{bmatrix} -\bar{\mathbf{A}} \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{e}'_{\mathsf{U}} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{I}_\kappa \end{bmatrix} \cdot \mathbf{x}' = \mathbf{0}, \end{cases} \tag{52}$$

and

$$\forall i \in [0,\kappa-1] :\ \mathbf{a}_{\mathsf{com}} \cdot x'_i + \bar{\mathbf{A}} \cdot \mathbf{r}'_{\mathsf{com},i} = \mathsf{com}_i \bmod q. \tag{53}$$

Backtracking the Merkle tree step. Write $\mathbf{w}'_{\mathsf{tree}}$ in the form of (22) and backtracks the transformations done in the Merkle tree step, the extractor $\overline{\mathsf{KE}}$ obtains $\{\mathbf{g}'_{\theta,\delta_\kappa} \in \{0,1\}^{n\lceil \log q \rceil},\ (y'_\theta, \mathbf{r}'_\theta) \in \{0,1\} \times \{0,1\}^m\}_{\theta \in [L]}$ such that, for each $\theta \in [L]$:

1. The Merkle tree path determined by bits $d'_{\theta,1},\ldots,d'_{\theta,\delta_\kappa}$ (which were obtained above) leads to leaf node $\mathbf{g}'_{\theta,\delta_\kappa}$.
2. Equation $\mathbf{a}_{\mathsf{com}} \cdot y'_\theta + \bar{\mathbf{A}} \cdot \mathbf{r}'_\theta = \mathbf{H}_{n,q-1} \cdot \mathbf{g}'_{\theta,\delta_\kappa}$ mod q holds.

On the other hand, since the Merkle hash tree was built on top of public values $\mathsf{com}_0,\ldots,\mathsf{com}_{\kappa-1}$, there must exist an integer $\mathrm{var}'(\theta) \in [0,\kappa-1]$, such that the path determined by the same bits $d'_{\theta,1},\ldots,d'_{\theta,\delta_\kappa}$ leads to a leaf node denoted by $\mathbf{c}'_\theta \in \{0,1\}^{n\lceil \log q \rceil}$ such that $\mathbf{H}_{n,q-1} \cdot \mathbf{c}'_\theta = \mathsf{com}_{\mathrm{var}'(\theta)}$ mod q. Note that, if $\mathbf{g}'_{\theta,\delta_\kappa} \neq \mathbf{c}'_\theta$, then this yields a collision for the hash function $h_{\bar{\mathbf{A}}}$. Thus, under the SIS assumption, one can deduce that $\mathbf{g}'_{\theta,\delta_\kappa} = \mathbf{c}'_\theta$. This further implies that:

$$\begin{aligned}\mathbf{a}_{\mathsf{com}} \cdot y'_\theta + \bar{\mathbf{A}} \cdot \mathbf{r}'_\theta = \mathbf{H}_{n,q-1} \cdot \mathbf{g}'_{\theta,\delta_\kappa} &= \mathbf{H}_{n,q-1} \cdot \mathbf{c}'_\theta \\ &= \mathsf{com}_{\mathrm{var}'(\theta)} \bmod q.\end{aligned}$$

Meanwhile, from (53), one also has

$$\mathbf{a}_{\mathsf{com}} \cdot x'_{\mathrm{var}'(\theta)} + \bar{\mathbf{A}} \cdot \mathbf{r}'_{\mathsf{com},\mathrm{var}'(\theta)} = \mathsf{com}_{\mathrm{var}'(\theta)} \bmod q.$$

In other words, the same commitment $\mathsf{com}_{\mathrm{var}'(\theta)}$ opens to bits y'_θ and $x'_{\mathrm{var}'(\theta)}$. If these bits are distinct then the binding property of the commitment scheme is broken. Thus, under the SIS assumption, one can deduce that $y'_\theta = x'_{\mathrm{var}'(\theta)}$, for all $\theta \in [L]$.

Backtracking the branching program step. Write $\mathbf{w}'_{\mathsf{BP}}$ in the form of (25) and backtracks the transformations done in the branching program step, the knowledge extractor $\overline{\mathsf{KE}}$ obtains $\eta'_1, \ldots, \eta'_{L-1} \in [0,4]$, $f'_{2,0}, f'_{2,1}, \ldots, f'_{L,0}, f'_{L,1} \in [0,4]$, $c'_{2,0}, \ldots, c'_{L,4} \in \{0,1\}$, satisfying the relations

$$\begin{cases} \eta'_1 = \pi'_{1,0}(0) \cdot \bar{y}'_1 + \pi'_{1,1}(0) \cdot y'_1, \\ \mathsf{unit}_{\eta'_1} = (c'_{2,0}, c'_{2,1}, c'_{2,2}, c'_{2,3}, c'_{2,4})^\top, \\ f'_{2,0} = \sum_{i=0}^4 \pi'_{2,0}(i) \cdot c'_{2,i} = \pi'_{2,0}(\eta'_1), \\ f'_{2,1} = \sum_{i=0}^4 \pi'_{2,1}(i) \cdot c'_{2,i} = \pi'_{2,1}(\eta'_1), \\ \eta'_2 = f'_{2,0} \cdot \bar{y}'_2 + f'_{2,1} \cdot y'_2 = \pi'_{2,0}(\eta'_1) \cdot \bar{y}'_2 + \pi'_{2,1}(\eta'_1) \cdot y'_2 \\ \qquad\qquad\qquad\qquad \vdots \\ \mathsf{unit}_{\eta'_{L-1}} = (c'_{L,0}, c'_{L,1}, c'_{L,2}, c'_{L,3}, c'_{L,4})^\top \\ f'_{L,0} = \sum_{i=0}^4 \pi'_{L,0}(i) \cdot c'_{L,i} = \pi'_{L,0}(\eta'_{L-1}), \\ f'_{L,1} = \sum_{i=0}^4 \pi'_{L,1}(i) \cdot c'_{L,i} = \pi'_{L,1}(\eta'_{L-1}), \\ \eta'_L = f'_{L,0} \cdot \bar{y}'_L + f'_{L,1} \cdot y'_L = 0. \end{cases}$$

Taking into account the above deductions, these relations can be rewritten as

$$\begin{aligned} &\eta'_0 = 0; \\ &\forall \theta \in [L] : \eta'_\theta = \pi'_{\theta, x'_{\mathrm{var}'(\theta)}}(\eta'_{\theta-1}); \\ &\eta'_L = 0. \end{aligned} \tag{54}$$

In other words, one has $\mathsf{BP}'(\mathbf{x}') = 1$, where BP' is the branching program determined by $\mathrm{var}'(1), \ldots, \mathrm{var}'(\theta), \pi'_{1,0}(0), \ldots, \pi'_{L,1}(4)$.

To summarize, under the SIS assumption, the knowledge extractor $\overline{\mathsf{KE}}$ can extract witnesses satisfying the statement considered by Protocol 4.