Lattice-Based Zero-Knowledge Arguments for Integer Relations

Benoît Libert, San Ling, Khoa Nguyen, Huaxiong Wang

Lattice-Based Zero-Knowledge Arguments for Integer Relations

Benoît Libert[1,2], San Ling[3], 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. We provide lattice-based protocols allowing to prove relations among committed integers. While the most general zero-knowledge proof techniques can handle arithmetic circuits in the lattice setting, adapting them to prove statements over the integers is non-trivial, at least if we want to handle exponentially large integers while working with a polynomial-size modulus q. For a polynomial L, we provide zero-knowledge arguments allowing a prover to convince a verifier that committed L-bit bitstrings x, y and z are the binary representations of integers X, Y and Z satisfying $Z = X + Y$ over $\mathbb{Z}$. The complexity of our arguments is only linear in L. Using them, we construct arguments allowing to prove inequalities $X < Z$ among committed integers, as well as arguments showing that a committed X belongs to a public interval $[\alpha, \beta]$, where α and β can be arbitrarily large. Our range arguments have logarithmic cost (i.e., linear in L) in the maximal range magnitude. Using these tools, we obtain zero-knowledge arguments showing that a committed element X does *not* belong to a public set S using $\widetilde{\mathcal{O}}(n \cdot \log |S|)$ bits of communication, where n is the security parameter. We finally give a protocol allowing to argue that committed L-bit integers X, Y and Z satisfy multiplicative relations $Z = XY$ over the integers, with communication cost subquadratic in L. To this end, we use our protocol for integer addition to prove the correct recursive execution of Karatsuba's multiplication algorithm. The security of our protocols relies on standard lattice assumptions with polynomial modulus and polynomial approximation factor.

1 Introduction

Lattice-based cryptography has been an extremely active area since the celebrated results of Ajtai [3] and Regev [59]. In comparison with discrete-logarithm and factoring-based techniques, it indeed offers numerous advantages like simpler arithmetic operations, a better asymptotic efficiency, advanced functionalities or a conjectured resistance to quantum computing. Its development was further boosted by breakthrough results of [27,54] showing how to safely use lattice trapdoors, which have been the cornerstone of many advanced primitives.

While lattices enable powerful functionalities that have no counterpart using traditional number theoretic tools, they do not easily lend themselves to the

realization of certain fundamental tasks, like efficient zero-knowledge proofs. Zero-knowledge protocols [31] make it possible to prove properties about certain secret witnesses in order to have users demonstrate their correct behavior while protecting their privacy. For simple statements such as proving knowledge of a secret key, efficient solutions have been reported in [56,51,40,48]. In order to prove relations among committed values, the best known methods rely on the extra algebraic structure [61,8,5] offered by the ring-LWE or ring-SIS problems [52] and no truly efficient solution is known for standard (i.e., non-ideal) lattices.

In this paper, we investigate the problem of proving, under standard lattice assumptions, that large committed *integers* satisfy certain algebraic relations. Namely, if $\mathbf{c}_x$, $\mathbf{c}_y$ and $\mathbf{c}_z$ are commitments to integers X, Y, Z of arbitrary polynomial bit-size $L = \mathsf{poly}(n)$, where n is the security parameter, we consider the problem of proving statements of the form $Z = X + Y$ and $Z = X \cdot Y$ over $\mathbb{Z}$. Note that this problem is different from the case of arithmetic circuits addressed in [8]: here, we are interested in proving relations over the integers. Furthermore, we would like to design zero-knowledge arguments for various other relations among large committed integers. As specific applications, we consider the problems of: (i) Proving that a committed integer X belongs to a publicly known range $[\alpha, \beta]$; (ii) Proving order relations $Y < X < Z$ between committed integers Y, X, Z; (iii) Proving that a committed element X does not belong to a public set (which allows users to prove their non-blacklisting).

While these problems received much attention in the literature, the most efficient solutions [49,35,21] handling large integers appeal to integer commitments [26,22] based on hidden-order groups (e.g., RSA groups), which are vulnerable to quantum computing. In particular, designing a solution based on mild assumptions in standard lattices is a completely open problem to our knowledge. Even in ideal lattices, handling integers of polynomial length L requires to work with exponentially large moduli, which affects both the efficiency and the approximation factor of the lattice assumption. Here, our goal is to realize the aforementioned protocols using polynomial moduli and approximation factors.

If we were to use known zero-knowledge proof systems [61,8,5] in ideal lattices to handle additive relations over $\mathbb{Z}$, we would need (super-)exponentially large moduli. In particular, in order to prove that committed integers X, Y, Z of bit-size $L = \mathsf{poly}(n)$ satisfy $Z = X + Y$, these protocols would require to prove that $Z = X + Y \mod q$ for a large modulus $q = 2^{\mathsf{poly}(n)}$. With current techniques, this would imply to work with a commitment scheme over rings R_q, for the same modulus q. In terms of efficiency, a single ring element would cost thousand times L bits to represent since the modulus should contain more than L bits. When it comes to proving smallness of committed values (in order to prove $Z = X + Y$ over $\mathbb{Z}$ via $Z = X + Y \mod q$, the prover should guarantee that X and Y are small w.r.t. q) together with relations among them, the prover may need to send hundreds of ring elements. As a consequence, the communication cost could be as large as $k \cdot L$, where k is up to hundreds of thousands. In terms of security, we note that such approaches may require at least sub-exponential approximation

factors for the underlying ideal-lattice problems. Moreover, ensuring soundness may be non-trivial as the protocols of [8,5] only guarantee relaxed soundness.

Our Contributions. We provide statistical zero-knowledge arguments allowing to prove additive and multiplicative relations among committed integers of bit-size $L = \mathsf{poly}(n)$ under mild assumptions in standard (i.e., non-ideal) lattices. Our protocols can work with two flavors of the commitment scheme by Kawachi, Tanaka and Xagawa (KTX) [40]. If we commit to integers in a bit-by-bit fashion, the modulus q can be as small as $\widetilde{\mathcal{O}}(n)$ and the security of our protocols can rely on the worst-case hardness of SIVP_γ with $\gamma = \widetilde{\mathcal{O}}(n)$, which turns out to be one the weakest assumptions in the entire literature on lattice-based cryptography. On the other hand, if we rely on a stronger assumption with $\gamma = \widetilde{\mathcal{O}}(\sqrt{L} \cdot n)$ for a modulus $q = \widetilde{\mathcal{O}}(\sqrt{L} \cdot n)$, then we can commit to L bits at once and reduce the communication cost. For this all-at-once commitment variant, the complexities of our protocols are summarized as follows.

The protocol for integer additions has communication cost $(\zeta + 20L) \cdot \kappa$ bits, where $\zeta = \widetilde{\mathcal{O}}(n) + 6L \log q$ is the cost of proving knowledge of valid openings for the commitments to X, Y, Z and $\kappa = \omega(\log n)$ is the number of protocol repetitions to make the soundness error negligibly small. Thus, the actual cost for proving the additive relation is $20L \cdot \kappa$ bits. In terms of computation complexity, both the prover and the verifier only perform $\mathcal{O}(L)$ simple operations.

We offer two options for proving integer multiplications. For practically interesting values of L, e.g., $L \leq 8000$, we can emulate the schoolbook multiplication algorithm by proving L additive relations, and obtain communication cost $\widetilde{\mathcal{O}}(n + L^2) \cdot \kappa$ as well as computation costs $\mathcal{O}(L^2)$ for both parties. To our knowledge, all known methods for proving integer multiplications (sometimes implicitly) involve $\mathcal{O}(L^2)$ computation and/or communication complexities. Can we break this quadratic barrier?

As a theoretical contribution, we put forward the first protocol for multiplicative relations that does not incur any quadratic costs. Specifically, by proving in zero-knowledge the correct execution of a Karatsuba multiplication algorithm [39], we obtain both computation and communication complexities of order $\mathcal{O}(L^{\log_2 3})$.

Applications. While our protocol for additive relations only handles non-negative integers, it suffices for many applications, such as arguments of inequalities among committed integers, range membership for public/hidden ranges, and set non-membership. Moreover, it can also be used in higher-level protocols like zero-knowledge lists [28].[4] or privacy-preserving certificate transparency [25].

In particular, for a set of N elements with bit-size $\widetilde{\mathcal{O}}(n)$, our zero-knowledge protocol for proving non-membership of a committed value only cost $\widetilde{\mathcal{O}}(n \cdot \log N)$ bits. In the lattice setting, this is the first non-membership proof that achieves communication cost logarithmic in the cardinality of the set. Meanwhile, in our protocol for proving that a committed L-bit integer belongs to a given range $[\alpha, \beta]$, where $\beta - \alpha \approx 2^L$, besides the cost of proving knowledge of a valid opening

[4] These involve a prover wishing to convince a verifier that a committed list contains elements $\{a_i\}_i$ in a specific order without revealing anything else.

for the commitment, the prover only has to send $23L \cdot \kappa$ bits to the verifier. In Table 1, we provide the concrete cost of the protocol variant achieving soundness error 2^{-80}, for commonly used lattice parameters.

We remark that, if we only had to prove the correct evaluation of binary addition circuits, MPC-based techniques [37,29,20] could perform slightly better than our protocols. However, they become much less efficient for the algebraic parts of the statements we have to prove (in particular, we also need to prove knowledge of openings of SIS-based commitments). Indeed, the MPC-in-the head paradigm [37] and its follow-ups [29,20] have linear complexities in the size of the circuit, which is much larger than the witness size as the commitment relation entails $\Theta(n(L+m))$ additions and multiplications over $\mathbb{Z}_q$. In our protocols, proving knowledge of an opening takes $\Theta((L+m)\log q)$ bits of communication.

Range size	2^{1000}	2^{2000}	2^{4000}	2^{8000}
Proving knowledge of committed X	3.16	3.65	4.63	6.59
Proving range membership of X	0.38	0.75	1.5	3
Total communication cost	3.54 MB	4.4 MB	6.13 MB	9.59 MB

Table 1. Concrete communication cost of our lattice-based zero-knowledge argument (Section 5.1) for proving knowledge of committed integer X belonging to a given range, w.r.t. various range sizes. We work with lattice parameters $n = 256$, $q \approx 2^{15}$, $m = 4608$. To achieve soundness error 2^{-80}, we set $\kappa = 137$.

Our Techniques. We proceed by emulating integer commitments by means of bit commitments. To commit to an L-bit integer X in an all-in-one fashion, we generate a KTX commitment $\mathbf{c}_x = \sum_{i=0}^{L-1} \mathbf{a}_i \cdot x_i + \mathbf{B} \cdot \mathbf{r} \in \mathbb{Z}_q^n$ to its binary representation $(x_{L-1}, \ldots, x_0)_2$ using public matrices $\mathbf{A} = [\mathbf{a}_0 \mid \ldots \mid \mathbf{a}_{L-1}] \in \mathbb{Z}_q^{n \times L}$ and $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ and random coins $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$.

Integer Additions. To prove additive relations among committed integers, we come up with an idea that may sound natural for computer processors, but, to the best of our knowledge, has not been considered in the context of zero-knowledge proofs. The idea is to view integer additions as binary additions *with carries*. Suppose that we add two bits x and y with carry-in c_{in} to obtain a bit z and carry-out c_{out}. Then, the relations among these bits are captured by equations

$$z = x + y + c_{in} \bmod 2, \qquad c_{out} = x \cdot y + z \cdot c_{in} + c_{in} \bmod 2,$$

which is equivalent to a homogeneous system of two equations over $\mathbb{Z}_2$. Using the above adder, we consider the addition of L-bit integers $X = (x_{L-1}, ..., x_0)_2$ and $Y = (y_{L-1}, ..., y_0)_2$ assuming that the committed sum is of length $L+1$ and written as $Z = (z_L, z_{L-1}, ..., z_0)_2$. For each $i \in \{0, ..., L-1\}$, we denote by c_{i+1}

the carry-out of the i-th addition and define $c_L = z_L$. The equations become

$$\begin{aligned} z_0 + x_0 + y_0 &= 0 \bmod 2 \\ c_1 + x_0 \cdot y_0 &= 0 \bmod 2 \\ z_1 + x_1 + y_1 + c_1 &= 0 \bmod 2 \\ c_2 + x_1 \cdot y_1 + z_1 \cdot c_1 + c_1 &= 0 \bmod 2 \\ &\vdots \\ z_{L-1} + x_{L-1} + y_{L-1} + c_{L-1} &= 0 \bmod 2 \\ z_L + x_{L-1} \cdot y_{L-1} + z_{L-1} \cdot c_{L-1} + c_{L-1} &= 0 \bmod 2. \end{aligned}$$

We observe that all the terms in the above equations are either bits or products of two bits. By adapting the Stern-like [60] techniques for hiding secret bits [45] and handling quadratic relations [43], we manage to prove that the bits of X, Y, Z satisfy the above equations modulo 2, which is equivalent to $X + Y = Z$ over $\mathbb{Z}$. Meanwhile, to prove that those bits coincide with the values committed under the KTX commitment requires to additionally prove a linear equation modulo q.

Interestingly, we show that, not only the problem of proving additive relations among committed integers can be reduced to proving secret bits satisfying linear and quadratic equations modulo 2 and one linear equation modulo q, such type of reduction is doable for all subsequently considered relations (multiplications, range membership, set non-membership). To handle the reduced statements in a modular manner, we thus design (in Section 3) a general zero-knowledge protocol that subsumes all argument systems of this work. In comparison with previous protocols [40,48,44,46] built on Stern's framework [60], this general protocol introduces a technical novelty which allows to reduce the communication cost.

Range Membership and Set Non-Membership. Our techniques for additions of non-negative integers directly yield a method for proving inequalities of the form $X \leq Z$, where it suffices to show the existence of non-negative integer Y such that $X + Y = Z$. This method can be further adapted to handle strict inequalities. To prove that $X < Z$, we demonstrate the existence of non-negative Y such that $X + Y + 1 = Z$, for which only a small additional treatment for the least significant bits of X, Y, Z is needed. Then, by combining two sub-protocols for inequalities, we can obtain range arguments for the statements "$X \in [\alpha, \beta]$", "$X \in [\alpha, \beta)$", "$X \in (\alpha, \beta]$" and "$X \in (\alpha, \beta)$", where X is committed under the KTX commitment, and α, β can be hidden/committed or public.

Given the techniques for proving inequalities, we can further obtain arguments of non-membership. In order to prove that a committed string $X \in \{0,1\}^k$ does *not* belong to a public set $S = \{s_1, \ldots, s_N\}$, the prover generates a (publicly computable) Merkle tree [53] whose leaves are the elements of S arranged in lexicographical order. Then, the prover can use the technique of Libert *et al.* [45] – which allows arguing possession of a path in a lattice-based Merkle tree – to prove knowledge of two paths leading to adjacent leaves for which the corresponding set elements $Y, Z \in \{0,1\}^k$ satisfy $Y < X < Z$ in lexicographical order. Here, the adjacency of the leaves Y and Z is argued using our techniques for integers

additions, which allows proving that their labels (i.e., the binary encoding of the path that connects them to the root) encode integers V, W such that $W = V + 1$.

Subquadratic Integer Multiplications. Proving multiplicative relations among L-bit committed integers with subquadratic complexity requires some additional tricks. Karatsuba's technique [39] divides integers X, Y into equal halves $X = X_1|X_0$ and $Y = Y_1|Y_0$, each of which has length $L/2$. If the length is odd, the factors must be padded with zeroes in the left halves, which raises technical difficulties as will be explained below. We have $X = 2^{L/2} \cdot X_1 + X_0$ and $Y = 2^{L/2} \cdot Y_1 + Y_0$, so that $X \cdot Y$ can be written

$$X \cdot Y = (2^L - 2^{L/2})(X_1 Y_1) + (1 - 2^{L/2})(X_0 Y_0) + 2^{L/2}(X_1 + X_0)(Y_1 + Y_0). \quad (1)$$

To prove this equation, we first prove knowledge of 3 partial products and then prove their correct shifting w.r.t. multiplication by powers of 2 before proving the correctness of additions. Each of the factors $X_1, Y_1, X_0, Y_0, X_1 + X_0, Y_1 + Y_0$ of (1) is recursively broken into 3 smaller products until reaching an easy-to-prove "base multiplication". One difficulty is that the length of $X_1 + X_0$ and $Y_1 + Y_0$ are one bit longer than the length $L/2$ of X_0, X_1, Y_0, Y_1. Since $L/2 + 1$ is odd, we need to pad with a zero before dividing any further and the same issue arises when dividing X_1, Y_1, X_0, Y_0. In the context of zero-knowledge proofs, it makes it very complicated to keep track of the lengths of witnesses in the underlying equations and determine where the original bits of X and Y should be.

To address the problems caused by carry-on bits in additions, Knuth [41] suggested to use subtractions and re-write the product $X \cdot Y$ as

$$(2^L + 2^{L/2}) \cdot (X_1 \cdot Y_1) + (1 + 2^{L/2}) \cdot (X_0 \cdot Y_0) - 2^{L/2} \cdot (X_1 - X_0) \cdot (Y_1 - Y_0). \quad (2)$$

The difference $X_1 - X_0$ is now guaranteed to have length $L/2$, which allows using $L = 2^k$ and recursively come down to base multiplications of two-bit integers. However, this modification introduces another problem as $X_1 - X_0$ and $Y_1 - Y_0$ can now be negative integers, which are more difficult to handle in our setting. For this reason, we need to make sure that we always subtract a smaller integer from a larger one, while preserving the ability to prove correct computations.

To this end, our idea is to compare X_1 and X_0 and let the smaller one be subtracted from the larger one. To do this, we define auxiliary variables X'_1, X'_0 such that $X'_1 > X'_0$ and $\{X'_1, X'_0\} = \{X_1, X_0\}$. Letting b be the bit such that $b = 1$ if $X'_1 \geq X'_0$ and $b = 0$ otherwise, this can be expressed by the equation:

$$(X'_1 - X'_0) = b \cdot (X_1 - X_0) + (1 - b) \cdot (X_0 - X_1),$$

which is provable in zero-knowledge using our techniques for integer additions. If we repeat the above process and define variables Y'_1, Y'_0 such that $\{Y'_1, Y'_0\} = \{Y_1, Y_0\}$ and an order control bit $c \in \{0, 1\}$, if we define $d = b + c \bmod 2$, we have

$$\begin{aligned} (X_1 - X_0) \cdot (Y_1 - Y_0) &= (X'_1 - X'_0) \cdot (Y'_1 - Y'_0) \qquad \text{if} \quad d = 0 \\ (X_1 - X_0) \cdot (Y_1 - Y_0) &= -(X'_1 - X'_0) \cdot (Y'_1 - Y'_0) \qquad \text{if} \quad d = 1. \end{aligned}$$

The term $(X_1 - X_0) \cdot (Y_1 - Y_0)$ appearing in equation (2) can thus be written as

$$(X_1 - X_0) \cdot (Y_1 - Y_0) = (1 - d) \cdot (X'_1 - X'_0) \cdot (Y'_1 - Y'_0) - d \cdot (X'_1 - X'_0) \cdot (Y'_1 - Y'_0),$$

which yields an equation compatible our techniques while avoiding to handle negative integers. At each recursive step, we further divide the differences $X'_1 - X'_0$ and $Y'_1 - Y'_0$ and keep track of the control bits b, c, d which are part of the witnesses.

Related Work. The first integer commitment scheme was proposed by Fujisaki and Okamoto [26] who suggested to use it to prove relation over the integers. They underlined the importance of zero-knowledge arguments over the integers in order to be able to prove modular relations when the modulus is not known in advance, when the commitment key is generated. Damgård and Fujisaki [22] corrected a flaw in the Fujisaki-Okamoto commitment and generalized it to abelian groups satisfying specific properties.

Lipmaa [49] highlighted the cryptographic importance of the class **D** of Diophantine sets[5] [1] and gave improved constructions of zero-knowledge proofs for Diophantine equations. As special cases, he obtained efficient zero-knowledge arguments for intervals, unions of intervals, exponential relations and gcd relations. In [34], Groth suggested another integer commitment scheme based on the Strong RSA assumption [4] which, like [26,22], relies on groups of hidden order. Couteau, Peters and Pointcheval [21] recently suggested to combine integer commitments with a commitment scheme to field elements in order to improve the efficiency of zero-knowledge proofs over the integers. They also revisited the Damgård-Fujisaki commitment [22] and proved it the security of its companion argument system under the standard RSA assumption. While our results are not as general as those of [49,21] as we do not handle negative integers, they suffice for many applications of integer commitments, as we previously mentioned.

Range proofs were introduced by Brickell *et al.* [10] and received a permanent attention [19,12,9,49,36,18,21,32] since then. They served as a building block of countless cryptographic applications, which include anonymous credentials [14], anonymous e-cash [13], auction protocols [50], e-voting [35] privacy-preserving certificate transparency [25] and many more.

Currently known range proofs proceed via two distinct approaches. The first one proceeds by breaking integers into bits or small digits [10,7,23,12,36,32], which allows communicating a sub-logarithmic (in the range size) number of group elements in the best known constructions [12,36,32]. The second approach [9,49,35,21] appeals to integer commitments and groups of hidden order. This approach is usually preferred for very large ranges (which often arise in applications like anonymous credentials [14], where range elements are comprised of thousands of bits) where it tends to be more efficient and it does not require the maximal range length to be known when the commitment key is chosen.

Despite three decades of research, all known efficient range proofs (by "efficient", we mean that the communication complexity should be only logarithmic in the range size) build on quantum-vulnerable assumptions and the only candidates supporting very large integers rely on groups of hidden order. By proving knowledge of small secret vectors, lattice-based protocols [40,48] can be seen as

[5] For $k, \ell \in \mathbb{N}$, a Diophantine set is a set of the form $S = \{\boldsymbol{x} \in \mathbb{Z}^k \mid \exists \boldsymbol{w} \in \mathbb{Z}^\ell : P_S(\boldsymbol{x}, \boldsymbol{w}) = 0\}$, for some representing polynomial $P_S(\boldsymbol{X}, \boldsymbol{W})$ defined over integer vectors $\boldsymbol{X} \in \mathbb{Z}^k$, $\boldsymbol{W} \in \mathbb{Z}^\ell$. Any recursively enumerable set is [24] Diophantine.

providing a limited form of range proofs: if we can prove that a committed $\mathbf{x} \in \mathbb{Z}^m$ has infinity norm $\|\mathbf{x}\|_\infty < B$ for some basis $B < q$ of a B-ary representation, we can prove that $\mathbf{x}$ encodes an integer X in the range $[0, B^m - 1]$. However, it is not clear how to deal with arbitrary ranges. Using homomorphic integer commitments, any range $[\alpha, \beta]$ can be handled (see [17] and references therein) by exploiting the homomorphic properties of the commitment scheme and proving that $X - \alpha \in [0, \beta - \alpha]$. With homomorphic commitments used in the context of lattice-based cryptography, there is no obvious way to shift the committed value by an integer α when $\alpha > q$. Even with a sub-exponential modulus q, the size L of integers can be at most sub-linear in n. To our knowledge, no flexible solution has been proposed in the lattice setting, let alone under standard lattice assumptions with polynomial approximation factors and polynomial-size moduli. Our schemes thus provide a first answer to this question.

In the context of set non-membership, our construction bears resemblance with a technique used by Nakanishi *et al.* [57] to handle revocation in privacy-preserving protocols by proving inequalities over the integers. For a public set $S = \{s_1, \ldots, s_N\}$ arranged in lexicographical order, they rely on a trusted authority to create Camenisch-Lysyanskaya signatures [16] on all ordered pairs $\{\mathsf{Msg}_i = (s_i, s_{i+1})\}_{i=1}^{N-1}$ of adjacent set elements. To prove that a committed s is not in S, the prover proceeds with a proof of knowledge of two message-signature pairs (Msg_j, sig_j), $(\mathsf{Msg}_{j+1}, sig_{j+1})$ for which $\mathsf{Msg}_j = (s_j, s_{j+1})$ and $\mathsf{Msg}_{j+1} = (s_{j+1}, s_{j+2})$ contain elements s_j, s_{j+1} such that $s_j < s < s_{j+1}$. While this approach could be instantiated with our technique for proving integer inequalities, it would require proofs of knowledge of signatures and thus lattice trapdoors (indeed, all known lattice-based signatures compatible with proofs of knowledge rely on lattice trapdoors [27,54]). By using proofs of knowledge of a Merkle tree path [45] instead of signatures, our solution eliminates the need for lattice trapdoors, which allows for a better efficiency (note that proving inequalities $s_j < s < s_{j+1}$ incurs a complexity $\Omega(\log N)$ in both cases, so that using Merkle trees does not affect the asymptotic complexity). Moreover, the technique of Nakanishi *et al.* [57] involves a trusted entity to sign all pairs $(s_i, s_{i+1})\}_{i=1}^{N-1}$ in a setup phase whereas no trusted setup is required in our construction. Eskandarian *et al.* [25] recently used proofs of integer inequalities and hash trees in their proofs of non-membership. Still, they prove inequalities by using signatures issued by some TTP. In contrast, our approach does not require any TTP.

Other approaches to prove (non-)membership of a public set were suggested in [15,42,12,47]. However, they rely on a trusted entity to approve the sets of which (non-)membership must be proven during a setup phase. Setup-free accumulator-based set membership proofs were described in [11,45], but they are not known to support non-membership proofs.

In [6], Bayer and Groth cleverly used Σ protocols to handle proofs of non-membership without assuming a trusted setup. Their construction achieves logarithmic complexity in the cardinality of the set, but it crucially relies on commitment schemes, like Pedersen's discrete-log-based commitment [58], with homomorphic properties over the message space and the randomness space. For

lack of a lattice-based commitment scheme with similar properties, their approach does not seem readily instantiable under lattice assumptions.

2 Preliminaries

NOTATIONS. When working with an integer $X \in [0, 2^L - 1]$, we use the notation $X = (x_{L-1}, \ldots, x_0)_2$ to describe its bits, and use bold lower-case letter $\mathbf{x}$ to denote the representation of X as binary column vector $(x_{L-1}, \ldots, x_0) \in \{0,1\}^L$. The column concatenation of matrices $\mathbf{A} \in \mathbb{Z}^{n \times k}$ and $\mathbf{B} \in \mathbb{Z}^{n \times m}$ is denoted by $[\mathbf{A} \,|\, \mathbf{B}] \in \mathbb{Z}^{n \times (k+m)}$. When concatenating column vectors $\mathbf{x} \in \mathbb{Z}^k$ and $\mathbf{y} \in \mathbb{Z}^m$, for simplicity, we often use the notation $(\mathbf{x} \| \mathbf{y}) \in \mathbb{Z}^{k+m}$ (instead of $(\mathbf{x}^\top \| \mathbf{y}^\top)^\top$).

2.1 Lattice-Based Cryptographic Building Blocks

We first recall the average-case problem SIS and its hardness.

Definition 1 ($\mathsf{SIS}^\infty_{n,m,q,\beta}$ **[2][27]).** *Given uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_\infty \leq \beta$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q$.*

If $m, \beta = \mathsf{poly}(n)$, and $q > \beta \cdot \widetilde{\mathcal{O}}(\sqrt{n})$, the $\mathsf{SIS}^\infty_{n,m,q,\beta}$ problem is at least as hard as worst-case lattice problem SIVP_γ for some $\gamma = \beta \cdot \widetilde{\mathcal{O}}(\sqrt{nm})$ (see, e.g., [27][55]).

We will use two SIS-based cryptographic ingredients: the commitment scheme of Kawachi, Tanaka and Xagawa [40] (KTX) and the Merkle hash tree from [45].

The KTX commitment scheme. The scheme works with security parameter n, prime modulus $q = \mathcal{O}(\sqrt{L} \cdot n)$, and dimension $m = n(\lceil \log_2 q \rceil + 3)$. We will consider several flavours of the scheme.

In the variant that allows committing to $L \leq \mathsf{poly}(n)$ bits, the commitment key is $(\mathbf{a}_0, \ldots, \mathbf{a}_{L-1}, \mathbf{B}) \hookleftarrow U(\mathbb{Z}_q^{n \times (m+L)})$. To commit to a bitstring $x_0, \ldots, x_{L-1}$, one samples $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$, and outputs $\mathbf{c} = \sum_{i=0}^{L-1} \mathbf{a}_i \cdot x_i + \mathbf{B} \cdot \mathbf{r} \bmod q$. Then, to open the commitment, one simply reveals $x_0, \ldots, x_{L-1} \in \{0,1\}$ and $\mathbf{r} \in \{0,1\}^m$.

If one can compute two valid openings $(x'_0, \ldots, x'_{L-1}, \mathbf{r}')$ and $(x''_0, \ldots, x''_{L-1}, \mathbf{r}'')$ for the same commitment $\mathbf{c}$, where $(x'_0, \ldots, x'_{L-1}) \neq (x''_0, \ldots, x''_{L-1})$, then one can compute a solution to the $\mathsf{SIS}^\infty_{n,m+L,q,1}$ problem associated with the uniformly random matrix $[\mathbf{a}_0 \mid \ldots \mid \mathbf{B}] \in \mathbb{Z}_q^{n \times (m+L)}$. Thus, the scheme is computationally binding, assuming the worst-case hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(\sqrt{L} \cdot n)}$. On the other hand, by the Leftover Hash Lemma [30], the distribution of a commitment $\mathbf{c}$ is statistically close to uniform over $\mathbb{Z}_q^n$. This implies that the scheme is statistically hiding.

In the special case when $L = 1$, the scheme becomes a bit commitment scheme, in which case it can use a small modulus $q = \widetilde{\mathcal{O}}(n)$ and rely on a weak SIVP assumption with $\gamma = \widetilde{\mathcal{O}}(n)$.

Kawachi *et al.* [40] extended the above fixed-length commitment scheme to a string commitment scheme $\mathsf{COM} : \{0,1\}^* \times \{0,1\}^m \to \mathbb{Z}_q^n$. The obtained scheme is also statistically hiding for the given setting of parameters, and computationally binding assuming that $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ is hard.

Here, we will use the first commitment variant to commit to secret bits and the string commitment scheme COM as a building block for Stern-like protocols.

Lattice-based Merkle hash tree. The construction relies on the following collision-resistant hash function. Let n be the security parameter, $q = \widetilde{\mathcal{O}}(n)$, $k = n\lceil \log_2 q \rceil$ and $m = 2k$. Define the "powers-of-2" matrix

$$\mathbf{G} = \mathbf{I}_n \otimes [1\ 2\ 4\ \ldots\ 2^{\lceil \log_2 q \rceil - 1}] \in \mathbb{Z}_q^{n \times k}.$$

Note that for every $\mathbf{v} \in \mathbb{Z}_q^n$, we have $\mathbf{v} = \mathbf{G} \cdot \mathsf{bin}(\mathbf{v})$, where $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^k$ denotes the binary representation of $\mathbf{v}$.

For matrix $\mathbf{B} = [\mathbf{B}_0 \mid \mathbf{B}_1] \hookleftarrow U(\mathbb{Z}_q^{n \times m})$, where $\mathbf{B}_0, \mathbf{B}_1 \in \mathbb{Z}_q^{n \times k}$, define the function $h_{\mathbf{B}} : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}^k$ as follows:

$$(\mathbf{u}_0, \mathbf{u}_1) \ \mapsto \ h_{\mathbf{B}}(\mathbf{u}_0, \mathbf{u}_1) = \mathsf{bin}\big(\mathbf{B}_0 \cdot \mathbf{u}_0 + \mathbf{B}_1 \cdot \mathbf{u}_1 \bmod q\big).$$

Note that $h_{\mathbf{B}}(\mathbf{u}_0, \mathbf{u}_1) = \mathbf{u} \Leftrightarrow \mathbf{B}_0 \cdot \mathbf{u}_0 + \mathbf{B}_1 \cdot \mathbf{u}_1 = \mathbf{G} \cdot \mathbf{u} \bmod q$. This hash function was shown collision-resistant if $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ is hard [2,45]. It allows building Merkle trees to securely accumulate data. In particular, for an ordered set $S = \{\mathbf{d}_0, \ldots, \mathbf{d}_{2^\ell - 1}\}$ consisting of $2^\ell \in \mathsf{poly}(n)$ elements of bit-size k, one builds the binary tree of depth ℓ on top of elements of the set, as follows. First, associate the 2^ℓ leaf nodes with elements of the set, with respect to the order of these elements. Then, every non-leaf node of the tree is associated with the hash value of its two children. Finally, output the root of the tree $\mathbf{u} \in \{0,1\}^k$. Note that, the collision resistance of the hash function $h_{\mathbf{B}}$ guarantees that it is infeasible to find a tree path starting from the root $\mathbf{u}$ and ending with $\mathbf{d}' \notin S$.

2.2 Zero-Knowledge Argument Systems and Stern-like Protocols

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

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

An argument system is called statistical zero-knowledge if there exists a PPT simulator $\mathcal{S}(y)$ having oracle access to any $\widehat{\mathcal{V}}(y)$ and producing a simulated transcript that is statistically close to the one of the real interaction between $\mathcal{P}(y, w)$ and $\widehat{\mathcal{V}}(y)$. A related notion is argument of knowledge, which requires the witness-extended emulation property. For protocols consisting of 3 moves (*i.e.*, commitment-challenge-response), witness-extended emulation is implied by *special soundness* [33], where the latter assumes that there exists a PPT extractor which takes as input a set of valid transcripts with respect to all possible values of the "challenge" to the same "commitment", and outputs w' such that $(y, w') \in \mathrm{R}$.

The statistical zero-knowledge arguments of knowledge presented in this work are Stern-like [60] protocols. In particular, they are Σ-protocols in the generalized sense defined in [38] (where 3 valid transcripts are needed for extraction, instead of just 2). The basic protocol consists of 3 moves: commitment, challenge, response. If a statistically hiding and computationally binding string commitment scheme, such as the KTX scheme [40], is employed in the first move, then one obtains a statistical zero-knowledge argument of knowledge (ZKAoK) with perfect completeness, constant soundness error 2/3. In many applications, the protocol is repeated $\kappa = \omega(\log n)$ times to make the soundness error negligibly small in n.

3 A General Zero-Knowledge Argument of Knowledge

This section presents a general Stern-like zero-knowledge argument system that subsumes all the subsequent constructions in Sections 4, 5 and 6. Before describing the protocol, we first recall two previous Stern-like techniques that it will use.

3.1 Some Previous Extending-then-Permuting Techniques

Let us recall the techniques for proving knowledge of a single secret bit x, and for proving knowledge of bit product $x_1 \cdot x_2$, from [45] and [43], respectively. These techniques will be employed in the protocol presented in Section 3.2.

For any bit $b \in \{0,1\}$, denote by $\overline{b}$ the bit $\overline{b} = b + 1 \bmod 2$, and by $\mathsf{ext}_2(b)$ the 2-dimensional vector $(\overline{b}, b) \in \{0,1\}^2$.

For any bit $c \in \{0,1\}$, define P_c^2 as the permutation that transforms the integer vector $\mathbf{v} = (v_0, v_1) \in \mathbb{Z}^2$ into $P_c^2(\mathbf{v}) = (v_c, v_{\overline{c}})$. Namely, if $c = 0$ then P_c^2 keeps the arrangement the coordinates of $\mathbf{v}$; or swaps them if $c = 1$. Note that:

$$\mathbf{v} = \mathsf{ext}_2(b) \iff P_c^2(\mathbf{v}) = \mathsf{ext}_2(b + c \bmod 2). \tag{3}$$

As shown in [45], the equivalence (3) helps proving knowledge of a secret bit x that may appear in several correlated linear equations. To this end, one extends x to $\mathsf{ext}_2(x) \in \{0,1\}^2$, and permutes the latter using P_c^2, where c is a uniformly random bit. Seeing the permuted vector $\mathsf{ext}_2(x + c \bmod 2)$ convinces the verifier that the original vector $\mathsf{ext}_2(x)$ is well-formed – which in turn implies knowledge of some bit x – while c acts as a "one-time pad" that completely hides x.

To prove that a bit is the product $x_1 \cdot x_2$ of two secret bits, Libert *et al.* [43] introduced the following t echnique. For any two bits b_1, b_2, define

$$\mathsf{ext}_4(b_1, b_2) = (\ \overline{b}_1 \cdot \overline{b}_2,\ \overline{b}_1 \cdot b_2,\ b_1 \cdot \overline{b}_2,\ b_1 \cdot b_2\ ) \in \{0,1\}^4,$$

which is an extension of the bit product $b_1 \cdot b_2$. Next, define a specific type of permutation associated with two bits, as follows.

For any two bits $c_1, c_2 \in \{0,1\}$, define P_{c_1,c_2}^4 as the permutation that transforms the integer vector $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1}) \in \mathbb{Z}^4$ into

$$P_{c_1,c_2}^4(\mathbf{v}) = \left(v_{c_1,c_2}, v_{c_1,\overline{c}_2}, v_{\overline{c}_1,c_2}, v_{\overline{c}_1,\overline{c}_2}\right) \in \mathbb{Z}^4.$$

For any bits b_1, b_2, c_1, c_2 and any vector $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1}) \in \mathbb{Z}^4$, we have

$$\mathbf{v} = \mathsf{ext}_4(b_1, b_2) \iff P^4_{c_1,c_2}(\mathbf{v}) = \mathsf{ext}_4(b_1 + c_1 \bmod 2, b_2 + c_2 \bmod 2). \quad (4)$$

As a result, to prove the well-formedness of $x_1 \cdot x_2$, one can extend it to the vector $\mathsf{ext}_4(x_1, x_2)$, permute the latter using $P^4_{c_1,c_2}$, where c_1, c_2 are uniformly random bits, and send the permuted vector to the verifier who should be convinced that the original vector, i.e., $\mathsf{ext}_4(x_1, x_2)$, is well-formed, while learning nothing else about x_1 and x_2, thanks to the randomness of c_1 and c_2. Furthermore, this sub-protocol can be combined with other Stern-like protocols, where one has to additionally prove that x_1, x_2 satisfy other conditions. This is done by using the same "one-time pads" c_1, c_2 at all occurrences of x_1 and x_2, respectively.

3.2 Our General Protocol

Let $N, \mathfrak{m}_1, \mathfrak{m}_2$ be positive integers, where $\mathfrak{m}_1 \leq N$. Let $T = \{(i_1, j_1), \ldots, (i_{|T|}, j_{|T|})\}$ be a non-empty subset of $[N] \times [N]$. Define $d_1 = 2(\mathfrak{m}_1 + \mathfrak{m}_2)$, $d_2 = 2N + 4|T|$ and $d = d_1 + d_2$. Let $n_1 \leq d_1, n_2 \leq d_2$ and $q > 2$ be positive integers. The argument system we aim to construct can be summarized as follows.

Public input consists of $\mathbf{g}_1, \ldots, \mathbf{g}_{\mathfrak{m}_1}, \mathbf{b}_1, \ldots, \mathbf{b}_{\mathfrak{m}_2}, \mathbf{u}_1 \in \mathbb{Z}_q^{n_1}$ and

$$\{h_{\ell,k}\}_{(\ell,k) \in [n_2] \times [N]}; \quad \{f_{\ell,t}\}_{(\ell,t) \in [n_2] \times [\,|T|\,]}; \quad v_1, \ldots, v_{n_2} \in \mathbb{Z}_2.$$

Prover's witness is $(N + \mathfrak{m}_2)$-bit vector $\mathbf{s} = (s_1, \ldots, s_{\mathfrak{m}_1}, \ldots, s_N, \ldots, s_{N+\mathfrak{m}_2})$.

Prover's goal is to prove in zero-knowledge that:

1. The first $\mathfrak{m}_1$ bits $s_1, \ldots, s_{m_1}$ and the last $\mathfrak{m}_2$ bits $s_{N+1}, \ldots, s_{N+\mathfrak{m}_2}$ satisfy the following linear equation modulo q.

$$\sum_{i \in [\mathfrak{m}_1]} \mathbf{g}_i \cdot s_i + \sum_{j \in [\mathfrak{m}_2]} \mathbf{b}_j \cdot s_{N+j} = \mathbf{u}_1 \bmod q. \quad (5)$$

2. The first N bits $s_1, \ldots, s_{\mathfrak{m}_1}, \ldots, s_N$ satisfy the following n_2 equations modulo 2 that contain N linear terms and a total of $|T|$ quadratic terms $\{s_{i_t} \cdot s_{j_t}\}_{t=1}^{|T|}$.

$$\forall \ell \in [n_2] : \sum_{k=1}^{N} h_{\ell,k} \cdot s_k + \sum_{t=1}^{|T|} f_{\ell,t} \cdot (s_{i_t} \cdot s_{j_t}) = v_\ell \bmod 2. \quad (6)$$

Looking ahead, all the statements that we will consider in Sections 4, 5 and 6 can be handled as special cases of the above general protocol, which will serve as an "umbrella" for all of our subsequent constructions.

As a preparation for the protocol construction, let us first introduce a few notations and techniques.

Encoding vector $\mathsf{ENC}(\cdot)$. In the protocol, we will work with a binary vector of length $\mathbf{d}$ that has a very specific constraint determined by $N + \mathfrak{m}_2$ bits. For

any $\mathbf{b} = (b_1, \ldots, b_{\mathfrak{m}_1}, \ldots, b_N, \ldots, b_{N+\mathfrak{m}_2}) \in \{0,1\}^{N+\mathfrak{m}_2}$, we denote by $\mathsf{ENC}(\mathbf{b}) \in \{0,1\}^d$ the vector encoding $\mathbf{b}$ as follows:

$$\begin{aligned}\mathsf{ENC}(\mathbf{b}) = \big(\, &\mathsf{ext}_2(b_1) \;\|\; \ldots \| \,\mathsf{ext}_2(b_{\mathfrak{m}_1}) \,\|\, \mathsf{ext}_2(b_{N+1}) \,\|\, \ldots \| \,\mathsf{ext}_2(b_{N+\mathfrak{m}_2}) \\ &\| \,\mathsf{ext}_2(b_1) \;\|\; \ldots \| \,\mathsf{ext}_2(b_N) \,\|\, \mathsf{ext}_4(b_{i_1}, b_{j_1}) \,\|\, \ldots \| \,\mathsf{ext}_4(b_{i_{|T|}}, b_{j_{|T|}}) \,\big),\end{aligned}$$

where $\mathsf{ext}_2(\cdot)$ and $\mathsf{ext}_4(\cdot,\cdot)$ are as in Section 3.1.

Permutation Γ. To prove in zero-knowledge of a vector that has the form $\mathsf{ENC}(\cdot)$, we will need to a specific type of permutation. To this end, we associate each $\mathbf{c} = (c_1, \ldots, c_N, \ldots, c_{N+\mathfrak{m}_2}) \in \{0,1\}^{N+\mathfrak{m}_2}$ with a permutation $\Gamma_{\mathbf{c}}$ that acts as follows. When being applied to vector

$$\begin{aligned}\mathbf{v} = \big(\mathbf{v}_1 \| \ldots \| \mathbf{v}_{\mathfrak{m}_1} \;\|\; \mathbf{v}_{\mathfrak{m}_1+1} \| \ldots \| \mathbf{v}_{\mathfrak{m}_1+\mathfrak{m}_2} \| \mathbf{v}_{\mathfrak{m}_1+\mathfrak{m}_2+1} \| \ldots \| \mathbf{v}_{\mathfrak{m}_1+\mathfrak{m}_2+N} \| \\ \| \,\mathbf{v}_{\mathfrak{m}_1+\mathfrak{m}_2+N+1} \| \ldots \| \mathbf{v}_{\mathfrak{m}_1+\mathfrak{m}_2+N+|T|}\big) \in \mathbb{Z}^d,\end{aligned}$$

whose first $\mathfrak{m}_1 + \mathfrak{m}_2 + N$ blocks are of length 2 and last $|T|$ blocks are of length 4, it transforms these blocks as described below.

$$\mathbf{v}_i \mapsto P^2_{c_i}(\mathbf{v}_i),\ \forall i \in [\mathfrak{m}_1]; \qquad \mathbf{v}_{\mathfrak{m}_1+j} \mapsto P^2_{c_{N+j}}(\mathbf{v}_{\mathfrak{m}_1+j}),\ \forall j \in [\mathfrak{m}_2];$$
$$\mathbf{v}_{\mathfrak{m}_1+\mathfrak{m}_2+k} \mapsto P^2_{c_k}(\mathbf{v}_{\mathfrak{m}_1+\mathfrak{m}_2+k}),\ \forall k \in [N];$$
$$\mathbf{v}_{\mathfrak{m}_1+\mathfrak{m}_2+N+t} \mapsto P^4_{c_{i_t},c_{j_t}}(\mathbf{v}_{\mathfrak{m}_1+\mathfrak{m}_2+N+t}),\ \forall t \in [\,|T|\,].$$

Based on the equivalences observed in (3)-(4), it can be checked that the following holds. For all $\mathbf{b}, \mathbf{c} \in \{0,1\}^{N+\mathfrak{m}_2}$, all $\mathbf{v} \in \mathbb{Z}^d$,

$$\mathbf{v} = \mathsf{ENC}(\mathbf{b}) \iff \Gamma_{\mathbf{c}}(\mathbf{v}) = \mathsf{ENC}(\mathbf{b} + \mathbf{c} \bmod 2). \tag{7}$$

Let us now present the protocol, based on the above notations and techniques. First, we perform the following extensions for the secret objects:

$$\begin{cases}\forall k \in [N + \mathfrak{m}_2] : \mathbf{s}_k = \mathsf{ext}_2(s_k) \in \{0,1\}^2 \\ \forall (i_t, j_t) \in T : \mathbf{y}_{i_t,j_t} = \mathsf{ext}_4(s_{i_t}, s_{j_t}) \in \{0,1\}^4.\end{cases} \tag{8}$$

Now, we will perform some transformations regarding equation (5). Observe that, for each $i \in [\mathfrak{m}_1]$, if we form matrix $\mathbf{G}_i = [\mathbf{0}^{n_1} \mid \mathbf{g}_i] \in \mathbb{Z}_q^{n_1 \times 2}$, then we will have $\mathbf{G}_i \cdot \mathbf{s}_i = \mathbf{g}_i \cdot s_i \bmod q$. Similarly, for each $j \in [\mathfrak{m}_2]$, if we form $\mathbf{B}_j = [\mathbf{0}^{n_1} \mid \mathbf{b}_j] \in \mathbb{Z}_q^{n_1 \times 2}$, then we will have $\mathbf{B}_j \cdot \mathbf{s}_{N+j} = \mathbf{b}_j \cdot s_{N+j} \bmod q$.

Therefore, if we build matrix $\mathbf{M}_1 = [\mathbf{G}_1 \mid \ldots \mid \mathbf{G}_{\mathfrak{m}_1} \mid \mathbf{B}_1 \mid \ldots \mid \mathbf{B}_{\mathfrak{m}_2}] \in \mathbb{Z}_q^{n_1 \times d_1}$, equation (5) can be expressed as $\mathbf{M}_1 \cdot \mathbf{w}_1 = \mathbf{u}_1 \bmod q$, where $\mathbf{w}_1 = \big(\mathbf{s}_1 \| \ldots \| \mathbf{s}_{\mathfrak{m}_1} \| \mathbf{s}_{N+1} \| \ldots \| \mathbf{s}_{N+\mathfrak{m}_2}\big) \in \{0,1\}^{d_1}$.

Next, we will unify all the n_2 equations in (6) into just one equation modulo 2, in the following manner. We form matrices

$$\begin{cases}\mathbf{H}_{\ell,k} = \big[\,0 \mid h_{\ell,k}\,\big] \in \mathbb{Z}_2^{1\times 2},\ \forall (\ell,k) \in [n_2] \times [N]; \\ \mathbf{F}_{\ell,t} = \big[\,0 \mid 0 \mid 0 \mid f_{\ell,t}\,\big] \in \mathbb{Z}_2^{1\times 4},\ \forall (\ell,t) \in [n_2] \times [\,|T|\,],\end{cases}$$

and note that $\mathbf{H}_{\ell,k} \cdot \mathbf{s}_k = h_{\ell,k} \cdot s_k \bmod 2$ and $\mathbf{F}_{\ell,t} \cdot \mathbf{y}_{i_t,j_t} = f_{\ell,t} \cdot (s_{i_j} \cdot s_{i_t}) \bmod 2$. Thus, (6) can be rewritten as:

$$\begin{aligned}
&\mathbf{H}_{1,1} \cdot \mathbf{s}_1 + \ldots + \mathbf{H}_{1,N} \cdot \mathbf{s}_N + \mathbf{F}_{1,1} \cdot \mathbf{y}_{i_1,j_1} + \cdots + \mathbf{F}_{1,|T|} \cdot \mathbf{y}_{i_{|T|},j_{|T|}} = v_1 \bmod 2 \\
&\mathbf{H}_{2,1} \cdot \mathbf{s}_1 + \ldots + \mathbf{H}_{2,N} \cdot \mathbf{s}_N + \mathbf{F}_{2,1} \cdot \mathbf{y}_{i_1,j_1} + \cdots + \mathbf{F}_{2,|T|} \cdot \mathbf{y}_{i_{|T|},j_{|T|}} = v_2 \bmod 2 \\
&\qquad\qquad \vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
&\mathbf{H}_{n_2,1} \cdot \mathbf{s}_1 + \cdots + \mathbf{H}_{n_2,N} \cdot \mathbf{s}_N + \mathbf{F}_{n_2,1} \cdot \mathbf{y}_{i_1,j_1} + \cdots + \mathbf{F}_{n_2,|T|} \cdot \mathbf{y}_{i_{|T|},j_{|T|}} = v_{n_2} \bmod 2.
\end{aligned}$$

Letting $\mathbf{u}_2 = (v_1, \ldots, v_{n_2})^\top \in \mathbb{Z}_2^{n_2}$, the above equations can be unified into

$$\mathbf{M}_2 \cdot \mathbf{w}_2 = \mathbf{u}_2 \bmod 2, \tag{9}$$

where matrix $\mathbf{M}_2 \in \mathbb{Z}_2^{n_2 \times d_2}$ is built from $\mathbf{H}_{\ell,k}, \mathbf{F}_{\ell,t}$, and

$$\mathbf{w}_2 = \left(\mathbf{s}_1 \,\|\, \ldots \,\|\, \mathbf{s}_N \,\|\, \mathbf{y}_{i_1,j_1} \,\|\, \ldots \,\|\, \mathbf{y}_{i_{|T|},j_{|T|}} \right) \in \{0,1\}^{2N+4|T|}.$$

Now, let us construct the vector $\mathbf{w} = (\mathbf{w}_1 \| \mathbf{w}_2) \in \{0,1\}^d$, which has the form

$$\left(\mathbf{s}_1 \,\|\, \ldots \,\|\, \mathbf{s}_{\mathfrak{m}_1} \,\|\, \mathbf{s}_{N+1} \,\|\, \ldots \,\|\, \mathbf{s}_{N+\mathfrak{m}_2} \,\|\, \mathbf{s}_1 \,\|\, \ldots \,\|\, \mathbf{s}_N \,\|\, \mathbf{y}_{i_1,j_1} \,\|\, \ldots \,\|\, \mathbf{y}_{i_{|T|},j_{|T|}} \right),$$

where its components blocks are as described in (8). Then, by our above definition of encoding vectors, we have $\mathbf{w} = \mathsf{ENC}(\mathbf{s})$.

The transformations we have done so far allow us to reduce the original statement to proving knowledge of vector $\mathbf{s} \in \{0,1\}^{N+\mathfrak{m}_2}$, such that the component vectors $\mathbf{w}_1 \in \{0,1\}^{d_1}$, $\mathbf{w}_2 \in \{0,1\}^{d_2}$ of $\mathbf{w} = \mathsf{ENC}(\mathbf{s})$ satisfy the equations $\mathbf{M}_1 \cdot \mathbf{w}_1 = \mathbf{u}_1 \bmod q$ and $\mathbf{M}_2 \cdot \mathbf{w}_2 = \mathbf{u}_2 \bmod 2$. The derived statement can be handled in Stern's framework, based on the following main ideas.

- To prove that $\mathbf{w} = \mathsf{ENC}(\mathbf{s})$, we will use the equivalence (7). To this end, we sample a uniformly random $\mathbf{c} \in \{0,1\}^{N+\mathfrak{m}_2}$ and prove instead that $\Gamma_{\mathbf{c}}(\mathbf{w}) = \mathsf{ENC}(\mathbf{s} + \mathbf{c} \bmod 2)$. Seeing this, the verifier is convinced in ZK that $\mathbf{w}$ indeed satisfies the required constraint, thanks to the randomness of $\mathbf{c}$.
- To prove that equations $\mathbf{M}_1 \cdot \mathbf{w}_1 = \mathbf{u}_1 \bmod q$ and $\mathbf{M}_2 \cdot \mathbf{w}_2 = \mathbf{u}_2 \bmod 2$ hold, we sample uniformly random $\mathbf{r}_1 \in \mathbb{Z}_q^{d_1}$, $\mathbf{r}_2 \in \mathbb{Z}_2^{d_2}$, and demonstrate that

$$\mathbf{M}_1 \cdot (\mathbf{w}_1 + \mathbf{r}_1) = \mathbf{u}_1 + \mathbf{M}_1 \cdot \mathbf{r}_1 \bmod q; \quad \mathbf{M}_2 \cdot (\mathbf{w}_2 + \mathbf{r}_2) = \mathbf{u}_2 + \mathbf{M}_2 \cdot \mathbf{r}_2 \bmod 2.$$

The interactive protocol. Our interactive protocol goes as follows.

- The public input consists of matrices $\mathbf{M}_1, \mathbf{M}_2$ and vectors $\mathbf{u}_1, \mathbf{u}_2$, which are constructed from the original public input, as discussed above.
- The prover's witness consists of the original secret vector $\mathbf{s} \in \{0,1\}^{N+\mathfrak{m}_2}$ and vector $\mathbf{w} = (\mathbf{w}_1 \| \mathbf{w}_2) = \mathsf{ENC}(\mathbf{s})$ derived from $\mathbf{s}$, as described above.

The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ interact as described in Figure 1. The protocol uses the KTX string commitment scheme COM, which is statistically hiding

and computationally binding. For simplicity of presentation, for vectors $\mathbf{w} = \big(\mathbf{w}_1 \| \mathbf{w}_2\big) \in \mathbb{Z}^d$ and $\mathbf{r} = \big(\mathbf{r}_1 \| \mathbf{r}_2\big) \in \mathbb{Z}^d$, we denote by $\mathbf{w} \boxplus \mathbf{r}$ the operation that computes $\mathbf{z}_1 = \mathbf{w}_1 + \mathbf{r}_1 \bmod q$, $\mathbf{z}_2 = \mathbf{w}_2 + \mathbf{r}_2 \bmod 2$, and outputs d-dimensional integer vector $\mathbf{z} = \big(\mathbf{z}_1 \| \mathbf{z}_2\big)$. We note that, for all $\mathbf{c} \in \{0,1\}^{N+\mathfrak{m}_2}$, if $\mathbf{t} = \Gamma_{\mathbf{c}}(\mathbf{w})$ and $\mathbf{s} = \Gamma_{\mathbf{c}}(\mathbf{r})$, then we have $\Gamma_{\mathbf{c}}(\mathbf{w} \boxplus \mathbf{r}) = \mathbf{t} \boxplus \mathbf{s}$.

The described protocol can be seen as an improved version of a Stern-like protocol presented in [46], in the following aspect. In the case $Ch = 1$, instead of sending $\Gamma_{\mathbf{c}}(\mathbf{w}) = \mathsf{ENC}(\mathbf{c}^\star)$ - which costs $d = 2(\mathfrak{m}_1 + \mathfrak{m}_2) + 2N + 4|T|$ bits, we let the prover send $\mathbf{c}^\star$ which enables the verifier to compute the value $\mathsf{ENC}(\mathbf{c}^\star)$ and which costs only $N + \mathfrak{m}_2$ bits. Due to this modification, the results from [46] are not directly applicable to our protocol, and thus, in the proof of Theorem 1, we will analyze the protocol from scratch.

1. **Commitment:** $\mathcal{P}$ samples $\mathbf{c} \leftarrow U(\{0,1\}^{N+\mathfrak{m}_2})$, $\mathbf{r}_1 \leftarrow U(\mathbb{Z}_q^{d_1}), \mathbf{r}_2 \leftarrow U(\mathbb{Z}_2^{d_2})$, and computes $\mathbf{r} = (\mathbf{r}_1 \| \mathbf{r}_2)$, $\mathbf{z} = \mathbf{w} \boxplus \mathbf{r}$.
   Then $\mathcal{P}$ samples randomness ρ_1, ρ_2, ρ_3 for COM, and sends $\mathrm{CMT} = \big(C_1, C_2, C_3\big)$ to $\mathcal{V}$, where $C_1 = \mathsf{COM}(\mathbf{c}, \mathbf{M}_1 \cdot \mathbf{r}_1 \bmod q, \mathbf{M}_2 \cdot \mathbf{r}_2 \bmod 2;\ \rho_1)$, and
   $$C_2 = \mathsf{COM}(\Gamma_{\mathbf{c}}(\mathbf{r});\ \rho_2), \quad C_3 = \mathsf{COM}(\Gamma_{\mathbf{c}}(\mathbf{z});\ \rho_3).$$
2. **Challenge:** $\mathcal{V}$ sends a challenge $Ch \leftarrow U(\{1,2,3\})$ to $\mathcal{P}$.
3. **Response:** $\mathcal{P}$ sends RSP computed according to Ch, as follows:
   - $Ch = 1$: $\mathrm{RSP} = (\mathbf{c}^\star, \mathbf{v}, \rho_2, \rho_3)$, where $\mathbf{c}^\star = \mathbf{s} + \mathbf{c} \bmod 2$ and $\mathbf{v} = \Gamma_{\mathbf{c}}(\mathbf{r})$.
   - $Ch = 2$: $\mathrm{RSP} = (\mathbf{b}, \mathbf{x}, \rho_1, \rho_3)$, where $\mathbf{b} = \mathbf{c}$ and $\mathbf{x} = \mathbf{z}$.
   - $Ch = 3$: $\mathrm{RSP} = (\mathbf{e}, \mathbf{y}, \rho_1, \rho_2)$, where $\mathbf{e} = \mathbf{c}$ and $\mathbf{y} = \mathbf{r}$.

Verification: Receiving RSP, $\mathcal{V}$ proceeds as follows:

- $Ch = 1$: Let $\mathbf{t} = \mathsf{ENC}(\mathbf{c}^\star)$. Check that $C_2 = \mathsf{COM}(\mathbf{v}; \rho_2)$, $C_3 = \mathsf{COM}(\mathbf{t} \boxplus \mathbf{v}; \rho_3)$.
- $Ch = 2$: Parse $\mathbf{x} = (\mathbf{x}_1 \| \mathbf{x}_2)$, where $\mathbf{x}_1 \in \mathbb{Z}_q^{d_1}$ and $\mathbf{x}_2 \in \mathbb{Z}_2^{d_2}$, and check that
  $$C_1 = \mathsf{COM}(\mathbf{b}, \mathbf{M}_1 \cdot \mathbf{x}_1 - \mathbf{u}_1 \bmod q, \mathbf{M}_2 \cdot \mathbf{x}_2 - \mathbf{u}_2 \bmod 2; \rho_1), \quad C_3 = \mathsf{COM}(\Gamma_{\mathbf{b}}(\mathbf{x}); \rho_3).$$
- $Ch = 3$: Parse $\mathbf{y} = (\mathbf{y}_1 \| \mathbf{y}_2)$, where $\mathbf{y}_1 \in \mathbb{Z}_q^{d_1}$ and $\mathbf{y}_2 \in \mathbb{Z}_2^{d_2}$, and check that
  $$C_1 = \mathsf{COM}(\mathbf{e}, \mathbf{M}_1 \cdot \mathbf{y}_1 \bmod q, \mathbf{M}_2 \cdot \mathbf{y}_2 \bmod 2; \rho_1), \quad C_2 = \mathsf{COM}(\Gamma_{\mathbf{e}}(\mathbf{y}); \rho_2).$$

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

Fig. 1: The interactive protocol.

Theorem 1. *Suppose that* COM *is a statistically hiding and computationally binding string commitment. Then, the protocol described above is a statistical* ZKAoK *for the considered relation, with perfect completeness, soundness error* $2/3$ *and communication cost* $\zeta + 2 + N + \mathfrak{m}_2 + 2(\mathfrak{m}_1 + \mathfrak{m}_2)\lceil \log_2 q \rceil + 2N + 4|T|$*, where* $\zeta = \mathcal{O}(n \log n)$ *is the total bit-size of* CMT *and two commitment randomness.*

Proof. We first analyze the completeness and efficiency of the protocol. Then we prove that it is a zero-knowledge argument of knowledge.

Completeness. Suppose that the prover is honest and follows the protocol. Then, observe that the verifier outputs 1 under the following conditions.

1. $\mathbf{t} \boxplus \mathbf{v} = \Gamma_{\mathbf{c}}(\mathbf{z})$. This conditions holds, since $\mathbf{w} = \mathsf{ENC}(\mathbf{s})$, and by equivalence (7), we have $\mathbf{t} = \mathsf{ENC}(\mathbf{c}^\star) = \mathsf{ENC}(\mathbf{s} + \mathbf{c} \bmod 2) = \Gamma_{\mathbf{c}}(\mathsf{ENC}(\mathbf{s})) = \Gamma_{\mathbf{c}}(\mathbf{w})$. Hence, $\mathbf{t} \boxplus \mathbf{v} = \Gamma_{\mathbf{c}}(\mathbf{w}) \boxplus \Gamma_{\mathbf{c}}(\mathbf{r}) = \Gamma_{\mathbf{c}}(\mathbf{w} \boxplus \mathbf{r}) = \Gamma_{\mathbf{c}}(\mathbf{z})$.
2. $\mathbf{M}_1 \cdot \mathbf{x}_1 - \mathbf{u}_1 = \mathbf{M}_1 \cdot \mathbf{r}_1 \bmod q$ and $\mathbf{M}_2 \cdot \mathbf{x}_2 - \mathbf{u}_2 = \mathbf{M}_2 \cdot \mathbf{r}_2 \bmod 2$. These two equations hold, because $\mathbf{x}_1 = \mathbf{w}_1 + \mathbf{r}_1 \bmod q$, $\mathbf{x}_2 = \mathbf{w}_2 + \mathbf{r}_2 \bmod 2$ and $\mathbf{M}_1 \cdot \mathbf{w}_1 = \mathbf{u}_1 \bmod q$, $\mathbf{M}_2 \cdot \mathbf{w}_2 = \mathbf{u}_2 \bmod 2$.

Therefore, the protocol has perfect completeness.

Efficiency. Both prover and verifier only have to carry out $\mathcal{O}(d)$ simple operations modulo q and modulo 2. In terms of communication cost, apart from ζ bits needed for transferring CMT and two commitment randomness, the prover has to send a vector in $\{0,1\}^{N+\mathfrak{m}_2}$, a vector in $\mathbb{Z}_q^{d_1}$ and a vector in $\mathbb{Z}_2^{d_2}$, while the verifier only has to send 2 bits. Thus, the total cost is $\zeta + 2 + N + \mathfrak{m}_2 + 2(\mathfrak{m}_1 + \mathfrak{m}_2)\lceil \log_2 q \rceil + 2N + 4|T|$ bits. (When COM is the KTX string commitment scheme, we have $\zeta = 3n\lceil \log_2 q \rceil + 2m$.)

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, it 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 $\overline{Ch} = 1$: The simulator uses linear algebra over $\mathbb{Z}_q$ and $\mathbb{Z}_2$ to compute vectors $\mathbf{w}_1' \in \mathbb{Z}_q^{d_1}$ and $\mathbf{w}_2' \in \mathbb{Z}_2^{d_2}$ such that $\mathbf{M}_1 \cdot \mathbf{w}_1' = \mathbf{u}_1 \bmod q$ and $\mathbf{M}_2 \cdot \mathbf{w}_2' = \mathbf{u}_2 \bmod 2$. Let $\mathbf{w}' = (\mathbf{w}_1' \| \mathbf{w}_2')$.

Next, it samples $\mathbf{c} \leftarrow U(\{0,1\}^{N+\mathfrak{m}_2})$, $\mathbf{r}_1 \leftarrow U(\mathbb{Z}_q^{d_1}), \mathbf{r}_2 \leftarrow U(\mathbb{Z}_2^{d_2})$, and computes $\mathbf{r} = (\mathbf{r}_1 \| \mathbf{r}_2)$, $\mathbf{z}' = \mathbf{w}' \boxplus \mathbf{r}$. Then, it samples randomness ρ_1, ρ_2, ρ_3 for COM and sends the commitment $\mathrm{CMT} = \big(C_1', C_2', C_3'\big)$ to $\widehat{\mathcal{V}}$, where

$$C_1' = \mathsf{COM}(\mathbf{c}, \mathbf{M}_1 \cdot \mathbf{r}_1 \bmod q, \mathbf{M}_2 \cdot \mathbf{r}_2 \bmod 2;\, \rho_1),$$

$$C_2' = \mathsf{COM}(\Gamma_{\mathbf{c}}(\mathbf{r});\, \rho_2), \quad C_3' = \mathsf{COM}(\Gamma_{\mathbf{c}}(\mathbf{z}');\, \rho_3).$$

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

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

Case $\overline{Ch} = 2$: SIM samples $\mathbf{s}' \leftarrow U(\{0,1\}^{N+\mathfrak{m}_2})$ and computes $\mathbf{w}' = \mathsf{ENC}(\mathbf{s}')$. Next, it picks $\mathbf{c} \leftarrow U(\{0,1\}^{N+\mathfrak{m}_2})$, and $\mathbf{r}_1 \leftarrow U(\mathbb{Z}_q^{d_1}), \mathbf{r}_2 \leftarrow U(\mathbb{Z}_2^{d_2})$, and computes

$\mathbf{r} = (\mathbf{r}_1 \| \mathbf{r}_2)$, $\mathbf{z}' = \mathbf{w}' \boxplus \mathbf{r}$. Then, it samples randomness ρ_1, ρ_2, ρ_3 for COM and sends the commitment $\mathrm{CMT} = \left(C'_1, C'_2, C'_3\right)$ to $\widehat{\mathcal{V}}$, where

$$C'_1 = \mathsf{COM}(\mathbf{c}, \mathbf{M}_1 \cdot \mathbf{r}_1 \bmod q, \mathbf{M}_2 \cdot \mathbf{r}_2 \bmod 2;\, \rho_1),$$

$$C'_2 = \mathsf{COM}(\Gamma_{\mathbf{c}}(\mathbf{r});\, \rho_2), \quad C'_3 = \mathsf{COM}(\Gamma_{\mathbf{c}}(\mathbf{z}');\, \rho_3).$$

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

- If $Ch = 1$: Send $\mathrm{RSP} = \left(\mathbf{s}' + \mathbf{c} \bmod 2, \Gamma_{\mathbf{c}}(\mathbf{r}), \rho_2, \rho_3\right)$.
- If $Ch = 2$: Output $\perp$ and abort.
- If $Ch = 3$: Send $\mathrm{RSP} = \left(\mathbf{c}, \mathbf{r}, \rho_1, \rho_2\right)$.

Case $\overline{Ch} = 3$: SIM prepares $\mathrm{CMT} = \left(C'_1, C'_2, C'_3\right)$ as in the case $\overline{Ch} = 2$ above, except that C'_1 is computed as

$$C'_1 = \mathsf{COM}(\mathbf{c}, \mathbf{M}_1 \cdot (\mathbf{w}'_1 + \mathbf{r}_1) - \mathbf{u}_1 \bmod q, \mathbf{M}_2 \cdot (\mathbf{w}'_2 + \mathbf{r}_2) - \mathbf{u}_2 \bmod 2;\, \rho_1).$$

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.

In all the above cases, since COM is statistically hiding, the distribution of the commitment CMT and that of the challenge Ch from $\widehat{\mathcal{V}}$ are statistically close to those of the real interaction. Hence, the probability that the simulator outputs $\perp$ is negligibly far from $1/3$. Moreover, whenever the simulator does not halt, it provides an accepting transcript, of which the distribution is statistically close to that of the prover in a real interaction. We thus described a simulator that can successfully emulate the honest prover with probability negligibly close to $2/3$.

Argument of Knowledge. Suppose that we have $\mathrm{RSP}_1 = (\mathbf{c}^\star, \mathbf{v}, \rho_2^{(1)}, \rho_3^{(1)})$, $\mathrm{RSP}_2 = (\mathbf{b}, \mathbf{x}, \rho_1^{(2)}, \rho_3^{(2)})$, and $\mathrm{RSP}_3 = (\mathbf{e}, \mathbf{y}, \rho_1^{(3)}, \rho_2^{(3)})$, which are accepting transcripts for the three possible values of the challenge and the same commitment $\mathrm{CMT} = (C_1, C_2, C_3)$. Let us parse $\mathbf{x}$ and $\mathbf{y}$ as $\mathbf{x} = (\mathbf{x}_1 \| \mathbf{x}_2)$, $\mathbf{y} = (\mathbf{y}_1 \| \mathbf{y}_2)$, where $\mathbf{x}_1, \mathbf{y}_1 \in \mathbb{Z}_q^{d_1}$ and $\mathbf{x}_2, \mathbf{y}_2 \in \mathbb{Z}_2^{d_2}$.

The validity of the given responses implies that:

$$\begin{cases} C_1 = \mathsf{COM}(\mathbf{b}, \mathbf{M}_1 \cdot \mathbf{x}_1 - \mathbf{u}_1 \bmod q, \mathbf{M}_2 \cdot \mathbf{x}_2 - \mathbf{u}_2 \bmod 2; \rho_1^{(2)}); \\ C_1 = \mathsf{COM}(\mathbf{e}, \mathbf{M}_1 \cdot \mathbf{y}_1 \bmod q, \mathbf{M}_2 \cdot \mathbf{y}_2 \bmod 2; \rho_1^{(3)}); \\ C_2 = \mathsf{COM}(\mathbf{v}; \rho_2^{(1)}) = \mathsf{COM}(\Gamma_{\mathbf{e}}(\mathbf{y}); \rho_2^3); \\ C_3 = \mathsf{COM}(\mathbf{t} \boxplus \mathbf{v}; \rho_3^{(1)}) = \mathsf{COM}(\Gamma_{\mathbf{b}}(\mathbf{x}); \rho_3^{(2)}), \end{cases}$$

where $\mathbf{t} = \mathsf{ENC}(\mathbf{c}^\star)$. Since COM is computationally binding, we can deduce that:

$$\mathbf{b} = \mathbf{e}; \quad \mathbf{v} = \Gamma_{\mathbf{e}}(\mathbf{y}); \quad \mathbf{t} \boxplus \mathbf{v} = \Gamma_{\mathbf{b}}(\mathbf{x});$$

$$\mathbf{M}_1 \cdot \mathbf{x}_1 - \mathbf{u}_1 = \mathbf{M}_1 \cdot \mathbf{y}_1 \bmod q; \quad \mathbf{M}_2 \cdot \mathbf{x}_2 - \mathbf{u}_2 = \mathbf{M}_2 \cdot \mathbf{y}_2 \bmod 2.$$

Let $\mathbf{s}' = \mathbf{c}^\star + \mathbf{e} \bmod 2$ and $\mathbf{w}' = [\Gamma_{\mathbf{e}}]^{-1}(\mathbf{t})$. Since $\mathbf{t} = \mathsf{ENC}(\mathbf{c}^\star)$, by equivalence (7), we have that $\mathbf{w}' = \mathsf{ENC}(\mathbf{s}')$. Furthermore, note that $\Gamma_{\mathbf{e}}(\mathbf{w}') \boxplus \Gamma_{\mathbf{e}}(\mathbf{y}) = \Gamma_{\mathbf{e}}(\mathbf{x})$, which implies that $\mathbf{w}' \boxplus \mathbf{y} = \mathbf{x}$.

Now, parse $\mathbf{w}'$ as $\mathbf{w}' = (\mathbf{w}'_1 \| \mathbf{w}'_2)$, where $\mathbf{w}'_1 \in \{0,1\}^{d_1}$ and $\mathbf{w}'_2 \in \{0,1\}^{d_2}$. Then, we have $\mathbf{w}'_1 + \mathbf{y}_1 = \mathbf{x}_1 \bmod q$, $\mathbf{w}'_2 + \mathbf{y}_2 = \mathbf{x}_2 \bmod 2$, and

$$\begin{aligned} \mathbf{M}_1 \cdot \mathbf{w}'_1 &= \mathbf{M}_1 \cdot \mathbf{x}_1 - \mathbf{M}_1 \cdot \mathbf{y}_1 = \mathbf{u}_1 \bmod q; \\ \mathbf{M}_2 \cdot \mathbf{w}'_2 &= \mathbf{M}_2 \cdot \mathbf{x}_2 - \mathbf{M}_2 \cdot \mathbf{y}_2 = \mathbf{u}_2 \bmod 2. \end{aligned}$$

This implies $\mathbf{w}' = (\mathbf{w}'_1 \| \mathbf{w}'_2) = \mathsf{ENC}(\mathbf{s}')$, as well as $\mathbf{M}_1 \cdot \mathbf{w}'_1 = \mathbf{u}_1 \bmod q$ and $\mathbf{M}_2 \cdot \mathbf{w}'_2 = \mathbf{u}_2 \bmod 2$. Let $\mathbf{s}' = (s'_1, \ldots, s'_{\mathfrak{m}_1}, \ldots, s'_N, \ldots, s'_{N+\mathfrak{m}_2}) \in \{0,1\}^{N+\mathfrak{m}_2}$. By reversing the transformations, it can be seen that the bits of $\mathbf{s}'$ satisfy

$$\begin{aligned} &\sum_{i \in [\mathfrak{m}_1]} \mathbf{g}_i \cdot s'_i + \sum_{j \in [\mathfrak{m}_2]} \mathbf{b}_j \cdot s'_{N+j} = \mathbf{u}_1 \bmod q; \\ \forall \ell \in [n_2] : \ &\sum_{k=1}^{N} h_{\ell,k} \cdot s'_k + \sum_{t=1}^{|T|} f_{\ell,t} \cdot (s'_{i_t} \cdot s'_{j_t}) = v_\ell \bmod 2. \end{aligned}$$

Hence, we have extracted $\mathbf{s}' = (s'_1, \ldots, s'_{\mathfrak{m}_1}, \ldots, s'_N, \ldots, s'_{N+\mathfrak{m}_2})$, which is a valid witness for the considered relation. □

As we mentioned earlier, all the statements we will consider in the next sections will be reduced into instances of the presented general protocol. For each of them, we will employ the same strategy. First, we demonstrate that the considered statement can be expressed as an equation modulo q of the form (5) and equations modulo 2 of the form (6). This implies that we can run the general protocol to handle the statement, and obtain a statistical ZKAoK via Theorem 1. Next, as the complexity of the protocol depends on $\mathfrak{m}_1 + \mathfrak{m}_2, N, |T|$, we count these respective numbers in order to evaluate its communication cost.

4 Zero-Knowledge Arguments for Integer Additions

This section presents our lattice-based ZK argument system for additive relation among committed integers. Let n be the security parameter, and let $L = \mathsf{poly}(n)$. Given KTX commitments to L-bit integers $X = (x_{L-1}, \ldots, x_0)_2$, $Y = (y_{L-1}, \ldots, y_0)_2$ and $(L+1)$-bit integer $Z = (z_L, z_{L-1}, \ldots, z_0)_2$, the protocol allows the prover to convince the verifier in ZK that $X + Y = Z$ over $\mathbb{Z}$.

As discussed in Section 1 and Section 2.1, using different flavors of the KTX commitment scheme, we can commit to all the bits of X, Y, Z at once or a bit-by-bit fashion. Both approaches are both compatible with (and independent of) our ZK techniques. Depending on which commitments we use, we obtain different give trade-offs in terms of parameters, key sizes, security assumptions and communication costs. In the following, we will use the former variant, which yields communication complexity $\widetilde{\mathcal{O}}(L + n)$. Our protocol can be easily adjusted

to handle the bit-wise commitment variant, which yields complexity $\widetilde{\mathcal{O}}(L \cdot n)$, but allows smaller parameters, smaller keys and weaker lattice assumption.

Commitments. Let a prime $q = \widetilde{\mathcal{O}}(\sqrt{L} \cdot n)$ and $m = n(\lceil \log_2 q \rceil + 3)$. Choose a commitment key $(\mathbf{a}_0, \ldots, \mathbf{a}_{L-1}, \mathbf{a}_L, \mathbf{b}_1, \ldots, \mathbf{b}_m) \hookleftarrow U(\mathbb{Z}_q^{n \times (L+m+1)})$. To commit to X, Y, Z, sample $r_{i,1}, \ldots, r_{i,m}, \hookleftarrow U(\{0,1\})$, for $i \in \{1,2,3\}$, and compute

$$\begin{cases} \sum_{i=0}^{L-1} \mathbf{a}_i \cdot x_i + \sum_{j=1}^{m} \mathbf{b}_j \cdot r_{1,j} = \mathbf{c}_x \bmod q; \\ \sum_{i=0}^{L-1} \mathbf{a}_i \cdot y_i + \sum_{j=1}^{m} \mathbf{b}_j \cdot r_{2,j} = \mathbf{c}_y \bmod q; \\ \sum_{i=0}^{L} \mathbf{a}_i \cdot z_i + \sum_{j=1}^{m} \mathbf{b}_j \cdot r_{3,j} = \mathbf{c}_z \bmod q, \end{cases} \tag{10}$$

and output commitments $\mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z \in \mathbb{Z}_q^n$. The scheme relies on the worst-case hardness of SIVP_γ, for $\gamma = \widetilde{\mathcal{O}}(\sqrt{L} \cdot n)$.

Before presenting our protocol, we note that the three equations (10) can be unified into one equation of the form

$$\sum_{i=0}^{L-1} \mathbf{a}_i^{(1)} \cdot x_i + \sum_{i=0}^{L-1} \mathbf{a}_i^{(2)} \cdot y_i + \sum_{i=0}^{L} \mathbf{a}_i^{(3)} \cdot z_i + \sum_{(i,j) \in [3] \times [m]} \mathbf{b}_j^{(i)} \cdot r_{i,j} = \mathbf{c} \bmod q, \tag{11}$$

where $\mathbf{a}_i^{(1)}, \mathbf{a}_i^{(2)}, \mathbf{a}_i^{(3)} \in \mathbb{Z}_q^{3n}$ are extensions of $\mathbf{a}_i$; $\mathbf{b}_j^{(1)}, \mathbf{b}_j^{(2)}, \mathbf{b}_j^{(3)} \in \mathbb{Z}_q^{3n}$ are extensions of $\mathbf{b}_j$; and $\mathbf{c} = (\mathbf{c}_x \| \mathbf{c}_y \| \mathbf{c}_z) \in \mathbb{Z}_q^{3n}$. Having done this simple transformation, we observe that equation (11) does have the form captured by equation (5) in the protocol we put forward in Section 3. Here, the secret bits contained in the equations are the bits of X, Y, Z and those of the commitment randomness.

Proving integer additions. At a high level, our main idea consists in translating the addition operation $X + Y$ over the integers into the binary addition operation *with carries* of $(x_{L-1}, \ldots, x_0)_2$ and $(y_{L-1}, \ldots, y_0)_2$ and proving that this process indeed yields result $(z_L, z_{L-1}, \ldots, z_0)_2$. For the latter statement, we capture the whole process as equations modulo 2 that contain linear and quadratic terms, and show how this statement, when combined with the commitment equations (11), reduces to an instance of the protocol of Section 3.

Let us first consider the addition of two bits x, y with carry-in bit c_{in}. Let the output be bit z and the carry-out bit be c_{out}. Then, observe that the relation among $x, y, z, c_{\mathsf{in}}, c_{\mathsf{out}} \in \{0,1\}$ is captured by equations

$$\begin{cases} z = x + y + c_{\mathsf{in}} \bmod 2 \\ c_{\mathsf{out}} = x \cdot y + z \cdot c_{\mathsf{in}} + c_{\mathsf{in}} \bmod 2 \end{cases} \iff \begin{cases} z + x + y + c_{\mathsf{in}} = 0 \bmod 2 \\ c_{\mathsf{out}} + x \cdot y + z \cdot c_{\mathsf{in}} + c_{\mathsf{in}} = 0 \bmod 2. \end{cases}$$

Therefore, the addition with carries of $(x_{L-1}, \ldots, x_0)_2$ and $(y_{L-1}, \ldots, y_0)_2$ results in $(z_L, z_{L-1}, \ldots, z_0)_2$ if and only if the following equations hold:

$$\begin{cases} z_0 + x_0 + y_0 = 0 \bmod 2; \\ c_1 + x_0 \cdot y_0 = 0 \bmod 2; \\ z_1 + x_1 + y_1 + c_1 = 0 \bmod 2; \\ c_2 + x_1 \cdot y_1 + z_1 \cdot c_1 + c_1 = 0 \bmod 2; \\ \quad \vdots \\ z_{L-1} + x_{L-1} + y_{L-1} + c_{L-1} = 0 \bmod 2; \\ z_L + x_{L-1} \cdot y_{L-1} + z_{L-1} \cdot c_{L-1} + c_{L-1} = 0 \bmod 2. \end{cases} \tag{12}$$

Here, for each $i \in \{1, \ldots, L-1\}$, c_i denotes the carry-out bit at the i-th step which is also the carry-in bit at the $(i+1)$-th step. (The last carry-out bit is z_L.)

Now, observe that, together with equation (11), the $2L$ equations in (12) lead us to an instance of the protocol of Section 3. It indeed fits the pattern if we let $N := 4L$, $\mathfrak{m}_1 := 3L + 1$, $\mathfrak{m}_2 := 3m$ and denote the ordered tuple of $N + \mathfrak{m}_2$ secret bits $\big(x_0, \ldots, x_{L-1}, y_0, \ldots, y_{L-1}, z_0, \ldots, z_L, c_1, \ldots, c_{L-1}, r_{1,1}, \ldots, r_{3,m}\big)$ by $\big(s_1, \ldots, s_{N+\mathfrak{m}_2}\big)$. Then, note that the first $\mathfrak{m}_1$ bits $s_1, \ldots, s_{\mathfrak{m}_1}$ and the last $\mathfrak{m}_2$ bits $s_{N+1}, \ldots, s_{N+\mathfrak{m}_2}$ satisfy the linear equation modulo q from (11), while the first N bits $s_1, \ldots, s_N$ satisfy the equations modulo 2 in (12), which contain N linear terms and a total of $|T| := 2L - 1$ quadratic terms, i.e.:

$$x_0 \cdot y_0, \;\; x_1 \cdot y_1, \;\; z_1 \cdot c_1, \;\; \ldots, \;\; x_{L-1} \cdot y_{L-1}, \;\; z_{L-1} \cdot c_{L-1}.$$

As a result, our ZK argument system can be obtained from the protocol constructed in Section 3. The protocol is a statistical ZKAoK assuming the security of two variants of the KTX commitment scheme: the variant used to commit to X, Y, Z - which relies on the hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(\sqrt{L} \cdot n)}$, and the commitment COM used in the interaction between two parties - which relies on the hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$. By Theorem 1, each execution of the protocol has perfect completeness, soundness error $2/3$ and communication cost

$$\mathcal{O}(n \log n) + 3m + 2(3L + 1 + 3m)\lceil \log_2 q \rceil + 20L$$

bits, where $\mathcal{O}(n \log n)$ is the total bit-size of 3 KTX commitments (sent by the prover in the first move) and 2 commitment randomness. Here, it is important to note that the cost of proving knowledge of valid openings for $\mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z$ is $\mathcal{O}(n \log n) + 3m + 2(3L + 1 + 3m)\lceil \log_2 q \rceil$ bits. Thus, the actual cost for proving the addition relation is $20L$ bits.

We further remark that the protocol can easily be adapted to less challenging situations such as: (i) The bit-size of the sum Z is public known to be exactly L (instead of $L+1$); (ii) Not all elements X, Y, Z need to be hidden and committed. Indeed, in those scenarios, our strategy of expressing the considered relations as equations modulo q and modulo 2 easily goes through. Moreover, it even simplifies the resulting protocols and reduces their complexity because the number of secret bits to deal with is smaller than in the above protocol.

5 Logarithmic-Size Arguments for Range Membership and Set Non-Membership

We present two applications of our zero-knowledge protocol for integer additions from Section 4: range membership and set non-membership arguments.

5.1 Range Membership Arguments

Our range arguments build on the integer addition protocol of Section 4. We consider the problem of proving in ZK that a committed integer X satisfies $X \in [\alpha, \beta]$, i.e., $\alpha \leq X \leq \beta$, for publicly known integers α, β.

Let $L = \mathsf{poly}(n)$, $q = \widetilde{\mathcal{O}}(\sqrt{L} \cdot n)$ and $m = n(\lceil \log_2 q \rceil + 3)$. Suppose that L-bit integer $X = (x_{L-1}, \ldots, x_0)_2$ is committed via the KTX commitment scheme, using a public commitment key $\mathbf{a}_0, \ldots, \mathbf{a}_{L-1}, \mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{Z}_q^n$ and randomness $r_1, \ldots, r_m \in \{0, 1\}$. Namely, the commitment $\mathbf{c} \in \mathbb{Z}_q^n$ is computed as

$$\sum_{i=0}^{L-1} \mathbf{a}_i \cdot x_i + \sum_{j=1}^{m} \mathbf{b}_j \cdot r_j = \mathbf{c} \bmod q. \tag{13}$$

Our goal is to prove in ZK that $X \in [\alpha, \beta]$, for publicly given L-bit integers $\alpha = (\alpha_{L-1}, \ldots, \alpha_0)_2$ and $\beta = (\beta_{L-1}, \ldots, \beta_0)_2$.

The main idea. We observe that X satisfies $\alpha \leq X \leq \beta$ if and only if there exist non-negative L-bit integers Y, Z such that

$$\alpha + Y = X \quad \text{and} \quad X + Z = \beta. \tag{14}$$

We thus reduce the task of proving $X \in [\alpha, \beta[$ to proving two addition relations among integers, which can be achieved using the techniques of Section 4. To this end, it suffices to demonstrate that the relations among the secret bits of X, Y, Z and public bits of α, β can be expressed as equations modulo 2 of the form (6).

The underlying equations modulo 2. Let the bits of integers Y, Z be $(y_{L-1}, \ldots, y_0)_2$ and $(z_{L-1}, \ldots, z_0)_2$, respectively. The addition $\alpha + Y = X$ over $\mathbb{Z}$, when viewed as a binary addition with carries, can be expressed as the following $2L$ equations modulo 2 which contain $L-1$ quadratic terms $x_1 \cdot c_1, \ldots, x_{L-1} \cdot c_{L-1}$.

$$\begin{cases} x_0 + y_0 = \alpha_0 \bmod 2; \\ c_1 + \alpha_0 \cdot y_0 = 0 \bmod 2; & // \text{ First carry-bit} \\ x_1 + y_1 + c_1 = \alpha_1 \bmod 2; \\ c_2 + \alpha_1 \cdot y_1 + x_1 \cdot c_1 + c_1 = 0 \bmod 2; & // \text{ Second carry-bit} \\ \quad \vdots \\ c_{L-1} + \alpha_{L-2} \cdot y_{L-2} + x_{L-2} \cdot c_{L-2} + c_{L-2} = 0 \bmod 2; \\ x_{L-1} + y_{L-1} + c_{L-1} = \alpha_{L-1} \bmod 2; \\ \alpha_{L-1} \cdot y_{L-1} + x_{L-1} \cdot c_{L-1} + c_{L-1} = 0 \bmod 2. & // \text{ Last carry-bit is } 0. \end{cases} \tag{15}$$

The relation $X + Z = \beta$ is handled similarly. We obtain the following $2L$ equations modulo 2, which contain L quadratic terms $x_0 \cdot z_0, x_1 \cdot z_1, \ldots, x_{L-1} \cdot z_{L-1}$.

$$\begin{cases} x_0 + z_0 = \beta_0 \bmod 2; \\ e_1 + x_0 \cdot z_0 = 0 \bmod 2; & // \text{ First carry-bit} \\ x_1 + z_1 + e_1 = \beta_1 \bmod 2; \\ e_2 + x_1 \cdot z_1 + \beta_1 \cdot e_1 + e_1 = 0 \bmod 2; & // \text{ Second carry-bit} \\ \quad \vdots \\ e_{L-1} + x_{L-2} \cdot z_{L-2} + \beta_{L-2} \cdot e_{L-2} + e_{L-2} = 0 \bmod 2; \\ x_{L-1} + z_{L-1} + e_{L-1} = \beta_{L-1} \bmod 2; \\ x_{L-1} \cdot z_{L-1} + \beta_{L-1} \cdot e_{L-1} + e_{L-1} = 0 \bmod 2. & // \text{ Last carry-bit is } 0. \end{cases} \tag{16}$$

Combining (15) and (16), we obtain a system of $4L$ equations modulo 2, which contain $N := 5L - 2$ linear terms

$$x_0, \ldots, x_{L-1}, y_0, \ldots, y_{L-1}, z_0, \ldots, z_{L-1}, c_1, \ldots, c_{L-1}, e_1, \ldots, e_{L-1},$$

and a total of $|T| = 2L - 1$ quadratic terms

$$x_1 \cdot c_1, \ldots, x_{L-1} \cdot c_{L-1}, x_0 \cdot z_0, x_1 \cdot z_1, \ldots, x_{L-1} \cdot z_{L-1}.$$

Putting it altogether. Based on the above transformations, we have translated the task of proving that committed integer X satisfies $X \in [\alpha, \beta]$ to proving knowledge of $N + \mathfrak{m}_2 = 5L - 2 + m$ secret bits

$$x_0, \ldots, x_{L-1}, y_0, \ldots, y_{L-1}, z_0, \ldots, z_{L-1}, c_1, \ldots, c_{L-1}, e_1, \ldots, e_{L-1}, r_1, \ldots, r_m, \tag{17}$$

where the first $\mathfrak{m}_1 = L$ bits and the last $\mathfrak{m}_2 = m$ bits satisfy equation (13) modulo q, while the first $N = 5L - 2$ bits satisfy a system of equations modulo 2 containing N linear terms and $|T| = 2L - 1$ quadratic terms. In other words, we have reduced the considered statement to an instance of the general protocol of Section 3.2. By running the latter with the witness described in (17), we obtain a statistical ZKAoK hardness of based on the hardness of SIVP_γ with factor $\gamma \leq \widetilde{\mathcal{O}}(\sqrt{L} \cdot n)$. Each execution of the protocol has perfect completeness, soundness error 2/3 and communication cost

$$\mathcal{O}(n \log n) + m + 2(L + m)\lceil \log_2 q \rceil + 23L$$

bits, where $\mathcal{O}(n \log n)$ is the total bit-size of 3 KTX commitments (sent by the prover in the first move) and 2 commitment randomness. Here, the cost of proving knowledge of a valid opening for $\mathbf{c}$ is $\mathcal{O}(n \log n) + m + 2(L + m)\lceil \log_2 q \rceil$ bits. The actual cost for proving the range membership thus amounts to $23L$ bits.

Variants. Our techniques can be easily adapted to handle other variants of range membership arguments. To prove a strict inequality, e.g., $X < \beta$ for a given β, we can simply prove that $X \leq \beta - 1$ using the above approach. In the

case of hidden ranges, e.g., when we need prove that $Y < X < Z$ where X, Y, Z are all committed, then we proceed by proving the existence of non-negative L-bit integers Y_1, Z_1 such that $Y + Y_1 + 1 = X$ and $X + Z_1 + 1 = Z$. This can be done by executing two instances of the protocol for addition relation among committed integers from Section 4.

5.2 Set Non-Membership Arguments

In this section, we construct a protocol allowing to prove that a committed element is not in a public set Set. The goal is to do this without relying on a trusted third party to approve the description of Set by signing its elements or any other means. To this end, we combine our protocols for integer addition and inequalities with arguments of knowledge of a path in a Merkle tree [45]. While Merkle trees were introduced for proving set membership, we (somewhat counter-intuitively) use them for dual purposes.

For security parameter n, choose $q = \widetilde{\mathcal{O}}(n)$, $k = n\lceil \log_2 q \rceil$ and $m = 2k$. Sample uniformly random matrices $\mathbf{A}, \mathbf{B}_0, \mathbf{B}_1 \in \mathbb{Z}_q^{n \times k}$, and denote their columns as $\mathbf{a}_0, \ldots, \mathbf{a}_{k-1}, \mathbf{b}_{0,0}, \ldots, \mathbf{b}_{0,k-1}, \mathbf{b}_{1,0}, \ldots, \mathbf{b}_{1,k-1} \in \mathbb{Z}_q^n$. These vectors will serve as public key for the KTX commitment scheme with k-bit committed values, while matrix $\mathbf{B} = [\mathbf{B}_0 \mid \mathbf{B}_1] \in \mathbb{Z}_q^{n \times 2k}$ will also serve as the public key for the Merkle tree from [44]. Let $\mathbf{G} \in \mathbb{Z}_q^{n \times k}$ be the "powers-of-2" matrix of Section 2.1.

Let $X = (x_{k-1}, \ldots, x_0)_2$ be a k-bit integer, and let $\mathbf{c} \in \mathbb{Z}_q^n$ be a KTX commitment to X, i.e., we have the following equation modulo q:

$$\sum_{i=0}^{k-1} \mathbf{a}_i \cdot x_i + \sum_{(i,j) \in \{0,1\} \times k} \mathbf{b}_{i,j} \cdot r_{i,j} = \mathbf{c} \bmod q, \tag{18}$$

where bits $r_{0,1}, \ldots, r_{1,k} \in \{0,1\}$ are the commitment randomness.

Let $\mathsf{Set} = \{S_1, \ldots, S_M\}$ be a public set containing $M = \mathsf{poly}(n)$ integers of bit-size k, where $S_1 < S_2 < \ldots < S_M$. We wish to prove in ZK that an integer X, which has been committed to via $\mathbf{c} \in \mathbb{Z}_q^n$, does not belong to Set. We aim at communication complexity $\mathcal{O}(\log M)$, so that the protocol scales well for large sets. To this end, we will use the lattice-based Merkle hash tree from [45].

Without loss of generality, assuming that $M = 2^\ell - 2$ for some positive integer ℓ.[6] For each $i = 0, \ldots, M$, let $\mathbf{s}_i \in \{0,1\}^k$ be the binary-vector representation of S_i. Let $\mathbf{s}_0 = (0, \ldots, 0)$ and $\mathbf{s}_{M+1} = (1, \ldots, 1)$ be the all-zero and all-one vectors of length k, which represent 0 and $2^k - 1$, the smallest and the largest non-negative integers of bit-size k, respectively. Using the SIS-based hash function $h_{\mathbf{B}}$ (see Section 2.1), we build a Merkle tree of depth ℓ on top of 2^ℓ vectors $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_M, \mathbf{s}_{M+1}$ and obtain the root $\mathbf{u} \in \{0,1\}^k$. For each $i \in [0, M+1]$, the tree path from leaf $\mathbf{s}_i$ to root $\mathbf{u}$ is determined by the ℓ bits representing integer i.

We prove knowledge of two *consecutive* paths from leaves $\mathbf{y} \in \{0,1\}^k$ and $\mathbf{z} \in \{0,1\}^k$ to the public root $\mathbf{u}$ such that the k-bit integers Y and Z corresponding

[6] If M does not have this form, one can duplicate S_1 sufficiently many times until the cardinality of the set has this property. Our protocol remains the same in this case.

to $\mathbf{y}$ and $\mathbf{z}$ satisfy $Y < X < Z$, where X is the integer committed in $\mathbf{c}$.

Let $v_{\ell-1}, \ldots, v_0$ and $w_{\ell-1}, \ldots, w_0$ be the bits determining the paths from the leaves $\mathbf{y}$ and $\mathbf{z}$, respectively, to root $\mathbf{u}$. Then, by "consecutive", we mean that the ℓ-bit integers $V = (v_{\ell-1}, \ldots, v_0)_2$ and $W = (w_{\ell-1}, \ldots, w_0)_2$ satisfy $V + 1 = W$.

We remark that the truth of the statement – which is ensured by the soundness of the argument – implies that the integer committed in $\mathbf{c}$ does not belong to Set, assuming the collision-resistance of the Merkle hash tree and the security of the commitment scheme. This is because: (i) The existence of the two tree paths guarantees that $\mathbf{y}, \mathbf{z} \in \mathsf{Set}$; (ii) The fact that they are consecutive further ensures that $(\mathbf{y}, \mathbf{z}) = (\mathbf{s}_i, \mathbf{s}_{i+1})$, for some $i \in [0, M]$; (iii) The inequalities $Y < X < Z$ then implies that either $X < S_1$ or $S_M < X$ or $S_j < X < S_{j+1}$, for some $j \in [1, M-1]$. In either case, it must be true that $X \notin \mathsf{Set}$.

The considered statement can be divided into 4 steps: (1) Proving knowledge of X committed in $\mathbf{c}$; (2) Proving knowledge of the tree paths from $\mathbf{y}$ and $\mathbf{z}$; (3) Proving the range membership $Y < Z < X$; (4) Proving the addition relation $V + 1 = W$. We show that the entire statement can be expressed as one linear equation modulo q together with linear and quadratic equations modulo 2, which allows reducing it to an instance of the general protocol from Section 3.2. Regarding (1), we have obtained equation (18). As for (2), we use the techniques from [45] to translate Merkle tree inclusions into a set of provable equations modulo q and modulo 2. The sub-statement (3) can be handled as in Section 5.1. Finally, (4) can easily be expressed as $2\ell - 1$ simple equations modulo 2.

The details of these steps are provided in Appendix A. We finally remark that set elements can have a longer representation than $k = n\lceil \log q \rceil$ bits if we hash them into k-bit string before building the Merkle tree. For this purpose, a SIS-based hash function $H_{\mathsf{SIS}} : \{0,1\}^m \to \mathbb{Z}_q^n$ like [2] should be used to preserve the compatibility with zero-knowledge proofs.

6 Subquadratic Arguments for Integer Multiplications

For $L = \mathsf{poly}(n)$, we consider the problem of proving that committed integers $X = (x_{L-1}, \ldots, x_0)_2$, $Y = (y_{L-1}, \ldots, y_0)_2$, $Z = (z_{2L-1}, \ldots, z_0)_2$ satisfy the multiplicative relation $Z = XY$. This task can be realized by running L instances of the protocol for integer additions from Section 4, but this naive method would yield complexity at least $\mathcal{O}(L^2)$. Our target here is to design an asymptotically more efficient protocol with computation/communication cost subquadratic in L. From a theoretical point of view, such a protocol is particularly interesting, because its execution must somehow employ a subquadratic multiplication algorithm. This inspires us to consider for the first time in the context of ZK proofs the Karatsuba multiplication algorithm [39] that achieves subquadratic complexity $\mathcal{O}(L^{\log_2 3})$. Specifically, we will prove that the result of applying the Karatsuba algorithm to committed integers X, Y is exactly the committed integer Z.

Commitments. Choose a prime $q = \widetilde{\mathcal{O}}(\sqrt{L} \cdot n)$ and let $m = n(\lceil \log_2 q \rceil + 3)$. We use the KTX commitment scheme with public key $(\mathbf{a}_0, \ldots, \mathbf{a}_{2L-1}, \mathbf{b}_1, \ldots, \mathbf{b}_m) \hookleftarrow$

$U(\mathbb{Z}_q^{n\times(2L+m)})$. Let $\mathbf{c}_x, \mathbf{c}_y, \mathbf{c}_z \in \mathbb{Z}_q^n$ be commitments to X, Y, Z, where

$$\begin{cases} \sum_{i=0}^{L-1} \mathbf{a}_i \cdot x_i + \sum_{j=1}^{m} \mathbf{b}_j \cdot r_{1,j} = \mathbf{c}_x \bmod q; \\ \sum_{i=0}^{L-1} \mathbf{a}_i \cdot y_i + \sum_{j=1}^{m} \mathbf{b}_j \cdot r_{2,j} = \mathbf{c}_y \bmod q; \\ \sum_{i=0}^{2L-1} \mathbf{a}_i \cdot z_i + \sum_{j=1}^{m} \mathbf{b}_j \cdot r_{3,j} = \mathbf{c}_z \bmod q, \end{cases}$$

where bits $\{r_{i,j}\}_{(i,j)\in[3]\times[m]}$ are the commitment randomness. Then, as in Section 4, we can unify the 3 equations into one linear equation modulo q:

$$\sum_{i=0}^{L-1} \mathbf{a}_i^{(1)} \cdot x_i + \sum_{i=0}^{L-1} \mathbf{a}_i^{(2)} \cdot y_i + \sum_{i=0}^{2L-1} \mathbf{a}_i^{(3)} \cdot z_i + \sum_{(i,j)\in[3]\times[m]} \mathbf{b}_j^{(i)} \cdot r_{i,j} = \mathbf{c} \bmod q. \quad (19)$$

6.1 An Interpretation of the Karatsuba Algorithm

Let $L = 2^k$ for some positive integer k. We will employ a variant of the Karatsuba algorithm, suggested by Knuth [41, Section 4.3.3]. First, we need to interpret the execution of the algorithm in a fashion compatible with our ZK technique.

The First Iteration. For the first application of Karatsuba algorithm, we break X and Y into their "most significant" and "least significant" halves:

$$X = [X^{(1)}, X^{(0)}] \text{ and } Y = [Y^{(1)}, Y^{(0)}], \quad (20)$$

where $X^{(1)}, X^{(0)}, Y^{(1)}, Y^{(0)}$ are $L/2$-bit integers. Then, as suggested by Knuth, the product Z can be written as:

$$\begin{aligned} Z = XY = (2^L + 2^{L/2}) \ \cdot \ X^{(1)}Y^{(1)} + (2^{L/2}+1) \cdot X^{(0)}Y^{(0)} \\ - 2^{L/2} \cdot (X^{(1)} - X^{(0)})(Y^{(1)} - Y^{(0)}). \end{aligned} \quad (21)$$

The advantage of Knuth's approach over Karatsuba's is that it allows working with the differences $(X^{(1)} - X^{(0)})$, $(Y^{(1)} - Y^{(0)})$ that guarantee to have bit-size $L/2$, rather than working with the sums $(X^{(1)} + X^{(0)})$, $(Y^{(1)} + Y^{(0)})$ that cause a burden of carry-on bits. However, this modification introduces a new issue as these differences may be negative, which are more difficult to handle in our setting. For this reason, we need to make sure that we always subtract a smaller integer from a larger one, while preserving the ability to prove correct computations.

Let $\widehat{X}^{(1)}, \widehat{X}^{(0)}$ such that $\widehat{X}^{(1)} \geq \widehat{X}^{(0)}$ and $\{\widehat{X}^{(1)}, \widehat{X}^{(0)}\} = \{X^{(1)}, X^{(0)}\}$. If we use an order control bit b that is assigned value 1 if $X^{(1)} \geq X^{(0)}$, or value 0 otherwise, and let $X^{(2)} = \widehat{X}^{(1)} - \widehat{X}^{(0)} \geq 0$, then we have the relations

$$\widehat{X}^{(1)} = b \cdot X^{(1)} + \overline{b} \cdot X^{(0)}; \quad \widehat{X}^{(0)} = \overline{b} \cdot X^{(1)} + b \cdot X^{(0)}; \quad X^{(2)} + \widehat{X}^{(0)} = \widehat{X}^{(1)}. \quad (22)$$

Conversely, if non-negative integers $X^{(1)}, X^{(0)}, \widehat{X}^{(1)}, \widehat{X}^{(0)}, X^{(2)}$ and bit b satisfy (22), then it holds that $\{\widehat{X}^{(1)}, \widehat{X}^{(0)}\} = \{X^{(1)}, X^{(0)}\}$ and $\widehat{X}^{(1)} \geq \widehat{X}^{(0)}$ and $X^{(2)} = \widehat{X}^{(1)} - \widehat{X}^{(0)}$.

Similarly, we can obtain $\widehat{Y}^{(1)}, \widehat{Y}^{(0)}$ such that $\widehat{Y}^{(1)} \geq \widehat{Y}^{(0)}$, non-negative $Y^{(2)}$ such that $Y^{(2)} = \widehat{Y}^{(1)} - \widehat{Y}^{(0)}$, as well as a control bit d satisfying

$$\widehat{Y}^{(1)} = d \cdot Y^{(1)} + \overline{d} \cdot Y^{(0)}; \quad \widehat{Y}^{(0)} = \overline{d} \cdot Y^{(1)} + d \cdot Y^{(0)}; \quad Y^{(2)} + \widehat{Y}^{(0)} = \widehat{Y}^{(1)}. \quad (23)$$

Relations (22)-(23) essentially establish a "bridge" that allows us to work (in the subtractions $X^{(1)} - X^{(0)}$ and $Y^{(1)} - Y^{(0)}$ incurring in (21)) with non-negative integers $X^{(2)}$ and $Y^{(2)}$ instead of possibly negative integers. Indeed, letting $s = b + d \bmod 2$, we have

$$(X^{(1)} - X^{(0)})(Y^{(1)} - Y^{(0)}) = \overline{s} \cdot X^{(2)}Y^{(2)} - s \cdot X^{(2)}Y^{(2)}.$$

Then, equation (21) can be expressed as

$$Z = XY = (2^L + 2^{L/2})Z^{(1)} + (2^{L/2} + 1)Z^{(0)} + 2^{L/2}(s \cdot Z^{(2)}) - 2^{L/2}(\overline{s} \cdot Z^{(2)}), \quad (24)$$

where $Z^{(1)} = X^{(1)}Y^{(1)}$, $Z^{(0)} = X^{(0)}Y^{(0)}$ and $Z^{(2)} = X^{(2)}Y^{(2)}$ are L-bit integers. These values are computed based on recursive applications of the Karatsuba algorithm until we reach integers of bit-size $L/2^{k-1} = 2$, as described below.

The Recursion. For $t = 1$ to $k-2$, and for string $\alpha \in \{0,1,2\}^t$, on input of $L/2^t$-bit integers $X^{(\alpha)}$ and $Y^{(\alpha)}$, we recursively obtain $L/2^{t+1}$-bit integers

$$X^{(\alpha 1)};\ X^{(\alpha 0)};\ \widehat{X}^{(\alpha 1)};\ \widehat{X}^{(\alpha 0)};\ X^{(\alpha 2)};\ \ Y^{(\alpha 1)};\ Y^{(\alpha 0)};\ \widehat{Y}^{(\alpha 1)};\ \widehat{Y}^{(\alpha 0)};\ Y^{(\alpha 2)},$$

and bits $b^{(\alpha)}, d^{(\alpha)}, s^{(\alpha)}$ satisfying the following relations.

$$\begin{cases} X^{(\alpha)} = [X^{(\alpha 1)}, X^{(\alpha 0)}]; \\ \widehat{X}^{(\alpha 1)} = b^{(\alpha)} \cdot X^{(\alpha 1)} + \overline{b}^{(\alpha)} \cdot X^{(\alpha 0)}; \quad \widehat{X}^{(\alpha 0)} = \overline{b}^{(\alpha)} \cdot X^{(\alpha 1)} + b^{(\alpha)} \cdot X^{(\alpha 0)}; \\ X^{(\alpha 2)} + \widehat{X}^{(\alpha 0)} = \widehat{X}^{(\alpha 1)}; \\ Y^{(\alpha)} = [Y^{(\alpha 1)}, Y^{(\alpha 0)}]; \\ \widehat{Y}^{(\alpha 1)} = d^{(\alpha)} \cdot Y^{(\alpha 1)} + \overline{d}^{(\alpha)} \cdot Y^{(\alpha 0)}; \quad \widehat{Y}^{(\alpha 0)} = \overline{d}^{(\alpha)} \cdot Y^{(\alpha 1)} + d^{(\alpha)} \cdot Y^{(\alpha 0)}; \\ Y^{(\alpha 2)} + \widehat{Y}^{(\alpha 0)} = \widehat{Y}^{(\alpha 1)}; \\ s^{(\alpha)} = b^{(\alpha)} + d^{(\alpha)} \bmod 2. \end{cases} \quad (25)$$

Let $Z^{(\alpha 1)} = X^{(\alpha 1)}Y^{(\alpha 1)}$, $Z^{(\alpha 0)} = X^{(\alpha 0)}Y^{(\alpha 0)}$, $Z^{(\alpha 2)} = X^{(\alpha 2)}Y^{(\alpha 2)}$. Note that these $L/2^t$-bit integers satisfy the equation:

$$\begin{aligned} Z^{(\alpha)} := X^{(\alpha)}Y^{(\alpha)} = \left(2^{L/2^t} + 2^{L/2^{t+1}}\right) \cdot Z^{(\alpha 1)} + (2^{L/2^{t+1}} + 1) \cdot Z^{(\alpha 0)} \\ + 2^{L/2^{t+1}} \cdot (s^{(\alpha)} \cdot Z^{(\alpha 2)}) - 2^{L/2^{t+1}} \cdot (\overline{s}^{(\alpha)} \cdot Z^{(\alpha 2)}). \end{aligned} \quad (26)$$

We remark that the number of secret bits contained in the integers

$$\{X^{(\alpha 1)};\ X^{(\alpha 0)};\ \widehat{X}^{(\alpha 1)};\ \widehat{X}^{(\alpha 0)};\ X^{(\alpha 2)}\},\ \text{where } \alpha \in \{0,1,2\}^t, \forall t = 0, \ldots, k-2,$$

derived from X in the above process is

$$5 \cdot \sum_{t=0}^{k-2} (3^t \cdot \frac{L}{2^{t+1}}) = \frac{5L}{3} \cdot \sum_{t=0}^{k-2} \left(\frac{3}{2}\right)^{t+1} = \frac{10L}{3} \cdot \left(\frac{3}{2}\right)^{k} - 5L = \frac{10}{3} \cdot 3^{\log_2 L} - 5L.$$

That is also the number of secret bits in the integers derived from Y. Meanwhile, the number of control bits $b^{(\alpha)}, d^{(\alpha)}, s^{(\alpha)}$ is $3 \cdot \sum_{t=0}^{k-2} 3^t = (3^{\log_2 L} - 3)/2$. In total, the process gives us $\mathcal{O}(3^{\log_2 L}) = \mathcal{O}(L^{\log_2 3})$ secret bits.

6.2 Representing All Relations as Equations Modulo 2

As shown in Sections 4 and 5, to prove that committed integers satisfy some statement, it suffices to demonstrate that the statement can be expressed as one linear equation modulo q together with linear and quadratic equations modulo 2, which effectively reduces it to an instance of the general protocol of Section 3.2. We have already obtained the linear equation modulo q from 19. Our main task is now to show that all the relations among $\mathcal{O}(L^{\log_2 3})$ secret bits obtained in Section 6.1 can be expressed in terms of linear and quadratic equations modulo 2.

We observe that, apart from the linear equations $s^{(\alpha)} = b^{(\alpha)} + d^{(\alpha)} \bmod 2$, there are several common types of relations among the secret objects derived in Section 6.1, for which we handle as follows.

The first type is relation of the form $X^{(\alpha)} = [X^{(\alpha 1)}, X^{(\alpha 0)}]$, between an $L/2^t$-bit integer $X^{(\alpha)}$ and its halves $X^{(\alpha 1)}$ and $X^{(\alpha 0)}$. Let $X^{(\alpha)} = (x^{(\alpha)}_{\frac{L}{2^t}-1}, \ldots, x^{(\alpha)}_0)_2$ and $X^{(\alpha 1)} = (x^{(\alpha 1)}_{\frac{L}{2^{t+1}}-1}, \ldots, x^{(\alpha 1)}_0)_2$, $X^{(\alpha 0)} = (x^{(\alpha 0)}_{\frac{L}{2^{t+1}}-1}, \ldots, x^{(\alpha 0)}_0)_2$. This type of relation can be expressed as the following linear equations modulo 2:

$$\forall i = 0, \ldots, \frac{L}{2^{t+1}} - 1 : x^{(\alpha 0)}_i + x^{(\alpha)}_i = 0 \bmod 2; \quad x^{(\alpha 1)}_i + x^{(\alpha)}_{i+\frac{L}{2^{t+1}}} = 0 \bmod 2.$$

The second type is relation of the form

$$\widehat{X}^{(\alpha 1)} = b^{(\alpha)} \cdot X^{(\alpha 1)} + \overline{b}^{(\alpha)} \cdot X^{(\alpha 0)}; \quad \widehat{X}^{(\alpha 0)} = \overline{b}^{(\alpha)} \cdot X^{(\alpha 1)} + b^{(\alpha)} \cdot X^{(\alpha 0)},$$

reflecting how $L/2^{t+1}$-bit integers $\widehat{X}^{(\alpha 1)}, \widehat{X}^{(\alpha 0)}$ are computed from $X^{(\alpha 1)}, X^{(\alpha 0)}$ based on a control bit $b^{(\alpha)}$. This type of relation can be translated into the following equations modulo 2, with respect to the bits of those integers

$$\forall i = 0, \ldots, L/2^{t+1} - 1 : \ \hat{x}^{(\alpha 1)}_i + b^{(\alpha)} \cdot x^{(\alpha 1)}_i + \overline{b}^{(\alpha)} \cdot x^{(\alpha 0)}_i = 0 \bmod 2;$$
$$\forall i = 0, \ldots, L/2^{t+1} - 1 : \ \hat{x}^{(\alpha 0)}_i + \overline{b}^{(\alpha)} \cdot x^{(\alpha 1)}_i + b^{(\alpha)} \cdot x^{(\alpha 0)}_i = 0 \bmod 2,$$

that contains $4 \cdot \frac{L}{2^{t+1}}$ quadratic terms.

The third type is the addition relation $X^{(\alpha 2)} + \widehat{X}^{(\alpha 0)} = \widehat{X}^{(\alpha 1)}$ among $L/2^{t+1}$-bit integers. This can be handled using our techniques from Section 4, resulting in equations modulo 2 with less than $2 \cdot \frac{L}{2^{t+1}}$ quadratic terms in total.

The fourth type of relations appears when we reach the base multiplication

of 2-bit integers: e.g., $Z^{(\alpha 1)} = X^{(\alpha 1)}Y^{(\alpha 1)}$, where $\alpha \in \{0,1,2\}^{k-2}$. Let $X^{(\alpha 1)} = (x_1^{(\alpha 1)}, x_0^{(\alpha 1)})_2$, $Y^{(\alpha 1)} = (y_1^{(\alpha 1)}, y_0^{(\alpha 1)})_2$ and $Z^{(\alpha 1)} = (z_3^{(\alpha 1)}, z_2^{(\alpha 1)}, z_1^{(\alpha 1)}, z_0^{(\alpha 1)})_2$. This relation can then be expressed by the following equations modulo 2, which contain 6 quadratic terms.

$$\begin{cases} z_0^{(\alpha 1)} + x_0^{(\alpha 1)} \cdot y_0^{(\alpha 1)} = 0 \bmod 2; \\ t_{1,0}^{(\alpha 1)} + x_1^{(\alpha 1)} \cdot y_0^{(\alpha 1)} = 0 \bmod 2; \quad \text{// assign value } x_1^{(\alpha 1)} \cdot y_0^{(\alpha 1)} \text{ to } t_{1,0}^{(\alpha 1)} \\ t_{0,1}^{(\alpha 1)} + x_0^{(\alpha 1)} \cdot y_1^{(\alpha 1)} = 0 \bmod 2; \quad \text{// assign value } x_0^{(\alpha 1)} \cdot y_1^{(\alpha 1)} \text{ to } t_{0,1}^{(\alpha 1)} \\ z_1^{(\alpha 1)} + t_{1,0}^{(\alpha 1)} + t_{0,1}^{(\alpha 1)} = 0 \bmod 2; \\ c_1^{(\alpha 1)} + t_{1,0}^{(\alpha 1)} \cdot t_{0,1}^{(\alpha 1)} = 0 \bmod 2; \quad \text{// carry bit} \\ t_{1,1}^{(\alpha 1)} + x_1^{(\alpha 1)} \cdot y_1^{(\alpha 1)} = 0 \bmod 2; \quad \text{// assign value } x_1^{(\alpha 1)} \cdot y_1^{(\alpha 1)} \text{ to } t_{1,1}^{(\alpha 1)} \\ z_2^{(\alpha 1)} + t_{1,1}^{(\alpha 1)} + c_1^{(\alpha 1)} = 0 \bmod 2; \\ z_3^{(\alpha 1)} + t_{1,1}^{(\alpha 1)} \cdot c_1^{(\alpha 1)} = 0 \bmod 2, \end{cases}$$

The other types of relations come into the scene when we add up partial products and their shifts to compute the $Z^{(\alpha)}$'s and finally reach Z, which are reflected by equations (26) and (24). To handle the shifts, e.g., left-shifting integer $Z^{(\alpha 1)}$ by $L/2^{t+1}$ positions, we assign an auxiliary variable $\widetilde{Z}^{(\alpha 1)} := 2^{L/2^{t+1}} \cdot Z^{(\alpha 1)}$ and express the relations between bits of $\widetilde{Z}^{(\alpha 1)}$ and $Z^{(\alpha 1)}$ as linear equations modulo 2, as is done for the first type of relation considered above. After performing all the shifts, we will need to handle a few additions of integers to compute a partial product such as $Z^{(\alpha)}$ in (26). There, the subtraction by $2^{L/2^{t+1}} \cdot (\overline{s}^{(\alpha)} \cdot Z^{(\alpha 2)})$ can be transformed into an equivalent addition relation. Then, we can represent each of the addition operations in (26) as linear and quadratic equations modulo 2.

Based on the above discussion, it can be seen that the whole execution of the Karatsuba algorithm can be expressed as linear and quadratic equations modulo 2. Combining with the linear equation modulo q from 19, we thus obtain an instance of the general protocol from Section 3.2. As a result, we achieve a statistical ZKAoK of committed integers X, Y, Z satisfying $XY = Z$. The security of the argument system relies on the binding of the COM used in the interaction and the binding of the commitment variant used for committing to X, Y, Z. Overall, the protocol is secure assuming the hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(\sqrt{L} \cdot n)}$.

We remark that, in our process of translating the relations in Section 6.1 into equations modulo 2, for each type of relations, the number of secret bits and the number of quadratic terms we need to handle are only a constant times larger than those before translating. Thus, the final numbers N and $|T|$ are of order $\mathcal{O}(L^{\log_2 3})$. Meanwhile, from equation (19), we obtain that $\mathfrak{m}_1 + \mathfrak{m}_2 = 4L + 3m$. Therefore, when repeating the protocol $\kappa = \omega(\log n)$ times to achieve negligible soundness error, the total communication cost is of order $\big(\mathcal{O}\big(L + m\big) \log q\big) + \mathcal{O}(L^{\log_2 3})\big) \cdot \kappa$. In terms of computation cost, the total number of bit operations performed by the prover and the verifier is of order $\mathcal{O}(L^{\log_2 3})$, i.e., subquadratic in L.

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). This work was also supported in part by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701).

References

1. L. Adleman and K. Mander. Diophantine complexity. In *SFCS*, pages 81–88. IEEE Computer Society, 1976.
2. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC 1996*, 1996.
3. M. Ajtai. Generating hard instances of the short basis problem. In *ICALP*, 1999.
4. N. Baric and B. Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees. In *Eurocrypt 1997*, 1997.
5. C. Baum, I. Damgård, S. Oechsner, and C. Peikert. Efficient commitments and zero-knowledge protocols from ring-sis with applications to lattice-based threshold cryptosystems. *IACR Cryptology ePrint Archive*, 2016:997, 2016.
6. S. Bayer and J. Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. In *Eurocrypt*, 2013.
7. M. Bellare and S. Goldwasser. Verifiable partial key escrow. In *ACM-CCS*, 1997.
8. F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient Zero-Knowledge Proofs for Commitments from Learning With Errors over Rings. In *ESORICS*, 2015.
9. F. Boudot. Efficient proofs that a committed number lies in an interval. In *Eurocrypt*, 2000.
10. E. Brickell, D. Chaum, I. Damgård, and J. van de Graaf. Gradual and verifiable release of a secret. In *Crypto*. Springer, 1988.
11. P. Camacho, A. Hevia, M. A. Kiwi, and R. Opazo. Strong accumulators from collision-resistant hashing. *Int. J. Inf. Sec.*, 11(5):349–363, 2012.
12. J. Camenisch, R. Chaabouni, and a. shelat. Efficient protocols for set membership and range proofs. In *Asiacrypt*, 2008.
13. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *Eurocrypt 2005*.
14. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Eurocrypt*, 2001.
15. J. Camenisch and A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Crypto 2002*, 2002.
16. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Crypto*, 2004.
17. R. Chaabouni. Enhancing privacy protection: Set membership, range proofs, and the extended access control. PhD Thesis, EPFL, Lausanne, 2017.
18. R. Chaabouni, H. Lipmaa, and B. Zhang. A non-interactive range proof with constant communication. In *Financial Cryptography*, 2012.
19. A. Chan, Y. Frankel, and Y. Tsiounis. Easy come - easy go divisible cash. In *Eurocrypt*, 1998.
20. M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, and G. Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *ACM-CCS*, 2017.
21. G. Couteau, T. Peters, and D. Pointcheval. Removing the strong RSA assumption from arguments over the integers. In *Eurocrypt*, 2017.

22. I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *Asiacrypt*, 2002.
23. I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *PKC*, 2001.
24. M. Davis, H. Putnam, and J. Robinson. The decision problem for exponential diophantine equations. *Annals of Mathematics*, pages 425–436, 1961.
25. S. Eskandarian, E. Messeri, J. Bonneau, and D. Boneh. Certificate transparency with privacy. In *Privacy Enhancing Technologies*, 2017.
26. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Crypto*, 1997.
27. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.
28. E. Ghosh, O. Ohrimenko, and R. Tamassia. Zero-knowledge authenticated order queries and order statistics on a list. In *ACNS*, 2015.
29. I. Giacomelli, J. Madsen, and C. Orlandi. ZKBoo: faster zero-knowledge for boolean circuits. In *USENIX Security Symposium*, 2016.
30. S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS 2010*, pages 230–240, 2010.
31. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC*, 1985.
32. A. Gonzalez and C. Ràfols. New techniques for non-interactive shuffle and range arguments. In *ACNS*, 2017.
33. J. Groth. Evaluating security of voting schemes in the universal composability framework. In *ACNS*, 2004.
34. J. Groth. Cryptography in subgroups of $\mathbb{Z}_n^\star$. In *TCC*, 2005.
35. J. Groth. Non-interactive zero-knowledge arguments for voting. In *ACNS*, 2005.
36. J. Groth. Efficient zero-knowledge arguments from two-tiered homomorphic commitments. In *Asiacrypt*, 2011.
37. Y. Ishai, E. Kushilevitz, R. Ostrovksy, and A. Sahai. Zero-knowledge from secure multiparty computation. In *STOC*, 2007.
38. A. Jain, S. Krenn, K. Pietrzak, and A. Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In *Asiacrypt*, 2012.
39. A. Karatsuba and Y. Ofman. Multiplication of many-digital numbers by automatic computers. *Physics-Doklady 7*, 7:595–596, 1963.
40. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Asiacrypt*, 2008.
41. D. E. Knuth. *The art of computer programming, Volume II: Seminumerical Algorithms, 3rd Edition*. Addison-Wesley, 1998.
42. J. Li, N. Li, and R. Xue. Universal accumulators with efficient nonmembership proofs. In *ACNS*, 2007.
43. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In *Asiacrypt 2016*.
44. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *Asiacrypt*, 2016.
45. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *Eurocrypt*, 2016.
46. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based PRFs and applications to e-cash. In *Asiacrypt*, 2017. To appear.

47. B. Libert, T. Peters, and M. Yung. Scalable group signatures with revocation. In *Eurocrypt*, 2012.
48. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC 2013*, 2013.
49. H. Lipmaa. On Diophantine complexity and statistical zero-knowledge arguments. In *Asiacrypt*, 2003.
50. H. Lipmaa, N. Asokan, and V. Niemi. Secure vickrey auctions without threshold trust. In *Financial Cryptography*, 2002.
51. V. Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *PKC*, 2008.
52. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *Eurocrypt*, 2010.
53. R. C. Merkle. A Certified Digital Signature. In *Crypto*, 1989.
54. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt*, 2012.
55. D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. In *Crypto 2013*, 2013.
56. D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *Crypto*, 2003.
57. T. Nakanishi, H. Fujii, Y. Hira, and N. Funabiki. Revocable group signature schemes with constant costs for signing and verifying. In *PKC*, 2009.
58. T. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Crypto*, volume 576 of *LNCS*, pages 129–140. Springer, 1991.
59. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.
60. J. Stern. A new paradigm for public key identification. *Information Theory, IEEE Transactions on*, 42(6), 1996.
61. X. Xie, R. Xue, and M. Wang. Zero knowledge proofs from ring-LWE. In *CANS*, 2013.

A Detailed Description of the Protocol of Section 5.2

The tree paths. We walk the paths from $\mathbf{y}$ and $\mathbf{z}$ to the root $\mathbf{u}$ and capture all the relations as equations modulo q and modulo 2. Let use denote the nodes on the path from $\mathbf{y}$ by $\mathbf{d}_\ell = \mathbf{y}, \mathbf{d}_{\ell-1}, \ldots, \mathbf{d}_1, \mathbf{d}_0 = \mathbf{u} \in \{0,1\}^k$ and their siblings by $\mathbf{e}_\ell, \mathbf{e}_{\ell-1}, \ldots, \mathbf{e}_1 \in \{0,1\}^k$, respectively. Recall that the bits determining this path are denoted by $v_{\ell-1}, \ldots, v_0$.

Following the construction of the tree from Section 2.1, $\forall i \in \{\ell-1, \ldots, 1, 0\}$, we have:

$$\mathbf{d}_i = \begin{cases} h_{\mathbf{B}}(\mathbf{d}_{i+1}, \mathbf{e}_{i+1}), \text{ if } v_i = 0; \\ h_{\mathbf{B}}(\mathbf{e}_{i+1}, \mathbf{d}_{i+1}), \text{ if } v_i = 1. \end{cases} \Leftrightarrow \mathbf{d}_i = \overline{v}_i \cdot h_{\mathbf{B}}(\mathbf{d}_{i+1}, \mathbf{e}_{i+1}) + v_i \cdot h_{\mathbf{B}}(\mathbf{e}_{i+1}, \mathbf{d}_i),$$

which can be interpreted as:

$$\mathbf{B}_0 \cdot (\overline{v}_i \cdot \mathbf{d}_{i+1}) + \mathbf{B}_1 \cdot (v_i \cdot \mathbf{d}_{i+1}) + \mathbf{B}_0 \cdot (v_i \cdot \mathbf{e}_{i+1}) + \mathbf{B}_1 \cdot (\overline{v}_i \cdot \mathbf{e}_{i+1}) = \mathbf{G} \cdot \mathbf{d}_i \mod q.$$

For each $i = \ell - 1, \ldots, 0$, let

– $\mathbf{d}_{i+1} = \mathbf{d}_i^\star + \mathbf{d}'_i$, where $\mathbf{d}_i^\star = \overline{v}_i \cdot \mathbf{d}_{i+1}$ and $\mathbf{d}'_i = v_i \cdot \mathbf{d}_{i+1}$.
– $\mathbf{e}_{i+1} = \mathbf{e}_i^\star + \mathbf{e}'_i$, where $\mathbf{e}_i^\star = \overline{v}_i \cdot \mathbf{e}_{i+1}$ and $\mathbf{e}'_i = v_i \cdot \mathbf{e}_{i+1}$.

Then, we have the following equations modulo q:

$$\begin{cases} \mathbf{B}_0 \cdot \mathbf{d}^\star_{\ell-1} + \mathbf{B}_1 \cdot \mathbf{d}'_{\ell-1} + \mathbf{B}_0 \cdot \mathbf{e}'_{\ell-1} + \mathbf{B}_1 \cdot \mathbf{e}^\star_{\ell-1} - \mathbf{G} \cdot \mathbf{d}_{\ell-1} = \mathbf{0} \bmod q; \\ \mathbf{B}_0 \cdot \mathbf{d}^\star_{\ell-2} + \mathbf{B}_1 \cdot \mathbf{d}'_{\ell-2} + \mathbf{B}_0 \cdot \mathbf{e}'_{\ell-2} + \mathbf{B}_1 \cdot \mathbf{e}^\star_{\ell-2} - \mathbf{G} \cdot \mathbf{d}_{\ell-2} = \mathbf{0} \bmod q; \\ \qquad\qquad \vdots \\ \mathbf{B}_0 \cdot \mathbf{d}^\star_0 + \mathbf{B}_1 \cdot \mathbf{d}'_0 + \mathbf{B}_0 \cdot \mathbf{e}'_0 + \mathbf{B}_1 \cdot \mathbf{e}^\star_0 = \mathbf{G} \cdot \mathbf{u} \bmod q. \end{cases} \tag{27}$$

By re-writing the matrix-vector products in 27 as sums of vector-bit products, and then combining the equations together, we obtain one linear equation modulo q that contains a total of $5k\ell - k$ secret bits.

Meanwhile, we also have the following $(5k+1)\ell$ linear and quadratic equations modulo 2, for all $i = \ell - 1, \ldots, 0$:

$$\begin{cases} \overline{v}_i + v_i = 1 \bmod 2; & // \text{ The relation between } v_i \text{ and } \overline{v}_i \\ d_{i+1,j} + d^\star_{i,j} + d'_{i,j} = 0 \bmod 2, \forall j \in [0, k-1] & // \text{ The bits of } \mathbf{d}_{i+1}, \mathbf{d}_i^\star \text{ and } \mathbf{d}_{i+1} \\ d^\star_{i,j} + \overline{v}_i \cdot d_{i+1,j} = 0 \bmod 2, \forall j \in [0, k-1] & // \text{ The bits of } \mathbf{d}_i^\star \text{ and } \mathbf{d}_{i+1} \\ d'_{i,j} + v_i \cdot d_{i+1,j} = 0 \bmod 2, \forall j \in [0, k-1] & // \text{ The bits of } \mathbf{d}'_i \text{ and } \mathbf{d}_{i+1} \\ e^\star_{i,j} + \overline{v}_i \cdot e_{i+1,j} = 0 \bmod 2, \forall j \in [0, k-1] & // \text{ The bits of } \mathbf{e}_i^\star \text{ and } \mathbf{e}_{i+1} \\ e'_{i,j} + v_i \cdot e_{i+1,j} = 0 \bmod 2, \forall j \in [0, k-1] & // \text{ The bits of } \mathbf{e}'_i \text{ and } \mathbf{e}_{i+1}. \end{cases}$$

Note that these equations contain $(7k+2)\ell$ secret bits and $4k\ell$ quadratic terms.

Next, we proceed analogously for the path from $\mathbf{z}$ to $\mathbf{u}$, which is determined by the bits $w_{\ell-1}, \ldots, w_0$, and also obtain a linear equation involving $5k\ell - k$ secret bits over $\mathbb{Z}_q$, as well as $(5k+1)\ell$ equations over $\mathbb{Z}_2$, which contain $(7k+2)\ell$ secret bits and $4k\ell$ quadratic terms.

The range membership $Y < X < Z$. To prove this range membership among k-bit integers, we use the techniques of Section Section 4 and Section 5.1 to express the relation as equations modulo 2. Then, we obtain $4k$ equations containing $7k-2$ linear terms and $4k-2$ quadratic terms. We note that, among $7k-2$ linear terms appearing in these equations, $2k$ of them have previously appeared in the Merkle tree step: they are exactly the bits of Y and Z.

The addition relation $V + 1 = W$. This simple relation is captured by the followings $2\ell - 1$ equations modulo 2.

$$\begin{cases} w_0 + v_0 = 1 \bmod 2; \quad c_1 + v_0 = 0 \bmod 2 \\ w_1 + v_1 + c_1 = 0 \bmod 2; \quad c_2 + v_1 \cdot c_1 = 0; \\ \qquad \vdots \\ w_{\ell-1} + v_{\ell-1} + c_{\ell-1} = 0 \bmod 2, \end{cases}$$

where $c_1, \ldots, c_{\ell-1}$ are the carry-bits incurring in the additions. These equations contain $3\ell - 1$ linear terms and $\ell - 2$ quadratic terms. We note that, among the

$3\ell - 1$ linear terms considered here, 2ℓ of them, i.e., the bits of V and W, have previously appeared in the Merkle tree step.

Putting the above altogether. In the whole process, we obtain 3 equations modulo q: equation (18) in the commitment layer and two equations in the Merkle tree step. Using linear algebra, we combine them together and obtain one linear equation over $\mathbb{Z}_q$ that contains a total of $\mathfrak{m}_1 + \mathfrak{m}_2 := 3k + 2(5k\ell - k) = 10k\ell + k$ secret bits.

We also obtain $2(5k+1)\ell + 4k + (2\ell - 1) = 10k\ell + 4k + 4\ell - 1$ equations over $\mathbb{Z}_2$ which contain $N := 2(7k+2)\ell + (7k-2) + (3\ell - 1) - 2k - 2\ell = 14k\ell + 5k + 5\ell - 3$ linear terms and a total of $|T| := 8k\ell + (4k-2) + (\ell - 2) = 8k\ell + 4k + \ell - 4$ quadratic terms.

We have thus reduced the set non-membership statement to an instance of the general protocol of Section 3.2. As a result, we obtain a statistical ZKAoK for the former. In the simulation, we simply run the simulator of Theorem 1. For extraction, we invoke the extractor of Theorem 1 and then "backtrack" the transformations described above in order to reconstruct k-bit integers X', Y', Z', bits $r'_{0,1}, \ldots, r'_{1,m}$, and ℓ-bit integers V', W' such that:

- The bits of X', together with bits $r'_{0,1}, \ldots, r'_{1,m}$, form a valid opening of $\mathbf{c}$.
- $V' + 1 = W'$ and the two paths determined by the bits of V', W' have the binary-vector representations of Y', Z' at the leaf level.
- $Y' < X' < Z'$.

We show that the above facts imply that the k-bit integer committed in $\mathbf{c}$ does not belong to Set. For the sake of contradiction, let us assume that a PPT adversary can produce X'' and $r''_{0,1}, \ldots, r''_{1,m}$ which form a valid opening of $\mathbf{c}$, but $X'' \in \mathsf{Set}$. It follows from the binding property of the KTX commitment that $X'' = X'$. Also, the collision-resistance of the Merkle tree implies that $Y', Z' \in \mathsf{Set}$. Moreover, since the tree paths associated with these two set elements are determined by the bits representing V' and $W' = V' + 1$, respectively, it comes that $Y' = S_{V'}$ and $Z' = S_{V'+1}$. If $V' = 0$, then we will have $X' < S_1$. If $V' = M$, then we will have $X' > S_M$. If $V' \in [1, M-1]$, then $S_{V'}$ and $S_{V'+1}$ are consecutive elements of Set and $S_{V'} < X' < S_{V'+1}$. In either case, we obtain that $X'' = X' \notin \mathsf{Set}$, which yields a contradiction.

The security of our protocol thus relies on the binding property of the commitment COM used in the interaction, the binding property of the commitment used to commit X, and the security of the Merkle hash tree being used. These ingredients all rely on the assumption that $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ is hard.

If we repeat the protocol $\kappa = \omega(\log n)$ times to achieve negligible soundness error, then the total communication cost will be of order

$$\Big(\mathcal{O}\big((\mathfrak{m}_1 + \mathfrak{m}_2)\log q\big) + \mathcal{O}\big(N + |T|\big)\Big) \cdot \kappa = \mathcal{O}(n \log^2 q \cdot \log M) \cdot \kappa = \widetilde{\mathcal{O}}(n \cdot \log M).$$

The above description of the protocol assumes that the bit-size k of all elements $\{S_i\}_{i=1}^M$ equals $k = n\lceil \log q \rceil$, which is necessary to build the Merkle tree in a bottom-up fashion by recursively hashing strings using the same SIS-based

hash function as in [45]. If set elements $\{S_i\}_{i=1}^{M}$ cost less than $n\lceil \log q \rceil$ bits to represent, we can simply pad them with zeroes until they reach the desired length. In applications where their binary representation exceeds $n\lceil \log q \rceil$ bits, we can first apply a collision-resistant hash function in order to have strings of length $k = n\lceil \log q \rceil$. In this case, the leaves of the Merkle tree should be sorted in the lexicographical order of their k-bit hash values. In order to preserve the compatibility of the whole protocol with zero-knowledge arguments, we can use a SIS-based hash function $H_{\mathsf{SIS}} : \{0,1\}^m \rightarrow \mathbb{Z}_q^n$ like Ajtai's function [2] as it allows proving that a committed m-bit element hashes into some k-bit string which is *not* a leaf of the Merkle tree.