Zero-Knowledge Arguments for Circuits from Mild Lattice Assumptions, and Applications to Attribute-Based Signatures

San Ling[1], Khoa Nguyen[1], Duong Hieu Phan[2], and Huaxiong Wang[1]

[1] Nanyang Technological University, Singapore
{lingsan,khoantt,hxwang}@ntu.edu.sg
[2] XLIM - CNRS - University of Limoges, Limoges, France
duong-hieu.phan@unilim.fr

The design of zero-knowledge proofs and arguments for circuit satisfiability is a central and long studied problem in the theory of cryptography. There exist protocols from generic assumptions, but they are too expensive and should be viewed only as feasibility results. Thus, a large body of work has focused on finding solutions based on concrete computational assumptions. While the state-of-affair is quite well-settled for protocols from traditional number-theoretic assumptions, it remains somewhat unsatisfactory in the scope of lattice-based cryptography: Existing protocols for proving circuit-related statements have to resort to assumptions in ideal lattices, which might provide less security confidence than those in general lattices. In this work, we close this gap by providing zero-knowledge argument for boolean circuit satisfiability based on a mild assumption that the standard worst-case problem $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ is hard in general lattices, where n is the security parameter. Our argument system is built on a sub-protocol for proving arbitrary binary boolean relation among committed bits.

Our construction can be extended to additionally handle algebraic statements, which enables interesting applications. As an illustration, we put forward the first attribute-based signature scheme for circuits from lattice assumptions.

1 Introduction

Zero-knowledge (ZK) proofs, introduced by Goldwasser, Micali and Rackoff [26], enable a prover to convince a verifier of the truth of a statement without leaking any other information. In the last three decades, ZK proofs have become a fundamental notion in complexity theory and a core building block in the design of numerous cryptographic protocols. One can separate ZK *proofs*, which provide statistical soundness, and ZK *arguments*, which only guarantee computational soundness. Goldreich et al. [25] and Brassard et al. [13] showed that every NP language has a ZK proof system and an argument system, respectively. Gentry et al. [21] made use of fully homomorphic encryption to obtain ZK proofs for circuits, whose communication cost is related to the size of the witness. Kilian [38] relied on the PCP theorem to construct ZK arguments with communication cost

logarithmic in the size of the circuits. However, those generic results are too expensive and should be viewed only as feasibility results.

A large body of work has been focusing on finding solutions based on concrete computational assumptions. In the discrete logarithm setting, Cramer and Damgård [18] provided ZK arguments for arithmetic circuits with linear communication cost. Improvements over this result was subsequently suggested in [28,31,62], and a recent work by Bootle et al. [9] has achieved protocols with logarithmic cost. In the bilinear setting, the inventions of non-interactive ZK proofs and arguments [33,30,32] for circuits and algebraic relations gave rise to numerous applications. Another line of recent research [29,46,8] has developed succinct non-interactive arguments, but they have to rely on non-falsifiable knowledge assumptions [23].

In the area of ZK protocols based on post-quantum foundations, the state-of-affair is less satisfactory. There exist non-interactive ZK proofs for some specific lattice problems [57], but we still do not have post-quantum analogues of Groth-Sahai proofs [33]. When it comes to proving circuit-related statements, there have been proposed several interactive protocols. Jain et al. [35] introduced a commitment scheme based on the Learning Parity with Noise (LPN) problem from coding theory, and provided ZK proofs for arbitrary relations among committed bit strings. Xie et al. [67] adapted Jain et al.'s constructions into the ideal lattice setting, yielding a commitment scheme and its companion ZK proofs based on the Ring Learning with Errors (Ring-LWE) problem [50]. Benhamouda et al. [7] relied on the Ring-LWE assumption with respect to a specialized ring to build a commitment scheme allowing to commit to a vector over a finite field. They also constructed ZK proofs for proving linear and multiplicative relations among committed values. More recently, Baum et al. [4] presented an additively homomorphic commitment scheme based on the hardness of the Ring Short Integer Solution (Ring-SIS) problem [49], which allows to commit to one ring element. They also designed zero-knowledge protocols for proving linear relations among committed values. For the time being, the three lattice-based constructions [67,7,4] in this direction all exploit the rich algebraic structures of polynomial rings, and all have to resort to ideal-lattice assumptions. On the one hand, ideal lattices usually yield cryptographic schemes with efficiency advantage over their general-lattice counterparts. On the other hand, the additional algebraic structures they contain might provide less security confidence than those in general lattices. While the state-of-the-art algorithms for solving ideal-lattice problems [17,39] do not perform significantly better than those addressing general lattices, there have been concerns about the their hardness (see, e.g., [10] for a comprehensive discussion on this issue.)

The state-of-affair of lattice-based ZK protocols for circuit-related statements, as discussed above, inspires us to investigate the design of protocols for proving arbitrary relations and circuit satisfiability based on milder computational assumptions in general lattices. This is a non-trivial problem, since general lattices do not possess the rich algebraic structures that are typically useful for such type of protocols.

1.1 Our Contributions

Theoretical results. We provide statistical zero-knowledge argument for boolean circuit satisfiability based on a mild lattice assumption. While previous works in the field have to resort to ideal-lattice assumptions, our argument system has stronger security guarantee since we only have to rely on the worst-case hardness of standard lattice problem $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ in general lattices, where n is the security parameter.

At the heart of our construction is a sub-protocol for proving arbitrary binary boolean relation among committed bits. Note that there are 16 such operations in total, including AND, OR, XOR, NAND and NOR. Furthermore, this sub-protocol can also be modified in order to handle unary boolean operations (e.g., NOT). Therefore, although our argument system has communication cost linear in the circuit size, we have the flexibility in choosing the more convenient one to work with between two equivalent circuits. That is, in some cases, a circuit based on $\{\mathsf{AND}, \mathsf{OR}, \mathsf{NOT}\}$ could have smaller size than its equivalent representation consisting entirely of NAND gates. (For example, a typical construction of an OR gate would require 3 NAND gates.)

Applications. A notable feature of our argument system for circuit is that it interacts smoothly with existing Stern-like protocols [40,45,43,41,42] for proving algebraic statements underlying lattice-based signatures and lattice-based encryption schemes. This feature is very useful in the context of advanced privacy-preserving cryptographic protocols involving circuit-like statements, such as secure two-party and multi-party computations [68], anonymous credentials [16], functional signatures [12], policy-based signatures [5] and attribute-based signatures [52].

As an application, we put forward the first attribute-based signature scheme for circuits from lattice assumptions. This is done by combining our protocol for circuits with the protocol from [45,41] for the Boyen signature scheme [11,53]. In the random oracle model, the scheme is statistically private, and adaptively unforgeable if the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ problem is hard and the Boyen signature scheme is secure. The scheme produces signature of bit-size $\widetilde{\mathcal{O}}(n \cdot (\ell + N))$, where ℓ and N are the input length and the size of the circuit, respectively. This level of asymptotic efficiency is comparable to that of the pairing-based instantiation from [61].

1.2 Our Techniques

We first aim to design a mechanism that can be used to prove knowledge of secret bits x_1, x_2, x_3 such that $x_3 = x_1 \circ x_2$, where $\circ$ could be an arbitrary binary boolean operation. Our starting point is a recent technique for handling the bit product relation, suggested by Libert et al. [42] in the context of group encryption [37].

In order to prove knowledge of secret bits x_1, x_2, x_3 such that $x_3 = x_1 \cdot x_2$, Libert et al. introduce the following extending-then-permuting technique. First,

they extend the bit $x_3 = x_1 \cdot x_2$ into vector

$$\mathsf{ext}(x_1, x_2) \stackrel{\mathsf{def}}{=} (\ \overline{x}_1 \cdot \overline{x}_2,\ \overline{x}_1 \cdot x_2,\ x_1 \cdot \overline{x}_2,\ x_1 \cdot x_2\ ) \in \{0,1\}^4,$$

where $\bar{b}$ denotes the bit $1 - b$. Then, they define a specific type of permutation, which is associated with two bits c_1, c_2, such that, when acting on vector $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1}) \in \mathbb{Z}^4$, it transforms the latter into vector $T_{c_1,c_2}(\mathbf{v}) = \big(\ v_{c_1,c_2},\ v_{c_1,\bar{c}_2},\ v_{\bar{c}_1,c_2},\ v_{\bar{c}_1,\bar{c}_2}\ \big)$. The goal is to enable the following equivalence:

$$\mathbf{v} = \mathsf{ext}(x_1, x_2) \iff T_{c_1,c_2}(\mathbf{v}) = \mathsf{ext}(x_1 \oplus c_1, x_2 \oplus c_2). \tag{1}$$

In the framework of Stern's protocol [63], the technique works as follows. The prover prepares vector $\mathbf{v} = \mathsf{ext}(x_1, x_2)$, samples uniformly random bits c_1, c_2, then sends $\mathbf{y} = T_{c_1,c_2}(\mathbf{v})$ to the verifier. The latter, seeing that $\mathbf{y}$ is the correct extension of 2 bits $x_1 \oplus c_1, x_2 \oplus c_2$, should be convinced that the left-hand side of (1) holds and that the original bit x_3 indeed has the form $x_1 \cdot x_2$. Meanwhile, he cannot learn any additional information about x_1, x_2, thanks to the "one-time pads" c_1, c_2.

We first note that the above technique already yields a method for proving $x_3 = x_1\ \mathsf{AND}\ x_2$. When investigating the core ideas underlying the technique, we observe that, the bit-wise operation in question does not necessarily have to be the AND operation. Indeed, somewhat interestingly, we find that the technique can be generalized to be applicable to arbitrary binary boolean operation $\circ$.

Specifically, for any operation $\circ : \{0,1\} \times \{0,1\} \to \{0,1\}$ and any two bits x_1, x_2, we define the vector

$$\mathsf{ext}_\circ(x_1, x_2) \stackrel{\mathsf{def}}{=} \big(\overline{x}_1 \circ \overline{x}_2,\ \overline{x}_1 \circ x_2,\ x_1 \circ \overline{x}_2,\ x_1 \circ x_2\big) \in \{0,1\}^4.$$

Then we manage to prove that the generalized version of (1) also holds, i.e.,

$$\mathbf{v} = \mathsf{ext}_\circ(x_1, x_2) \iff T_{c_1,c_2}(\mathbf{v}) = \mathsf{ext}_\circ(x_1 \oplus c_1, x_2 \oplus c_2).$$

This seemingly simple observation turns out to be very useful. First, when being used with the lattice-based bit commitment scheme from [36] in the framework of Stern's protocol [63], it allows us to address a fundamental problem: proving arbitrary boolean relation among committed bits. Namely, for any binary boolean operation $\circ$, given commitment key $(\mathbf{a}, \mathbf{B}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{n \times m}$ and three commitments $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3 \in \mathbb{Z}_q^n$, we can prove knowledge of bits x_1, x_2, x_3 and randomness $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3 \in \{0,1\}^m$ such that:

$$(x_3 = x_1 \circ x_2) \ \wedge\ \big(\forall i = 1,2,3 :\ \mathbf{a} \cdot x_i + \mathbf{B} \cdot \mathbf{r}_i = \mathbf{c}_i \bmod q\big).$$

The above core building block in turn enables us to develop ZK argument for proving knowledge of committed inputs $x_1, \ldots, x_\ell$ satisfying a boolean circuit C. Suppose that circuit C has N gates labelled by binary operations $\circ_1, \ldots, \circ_N$. We proceed as follows. First, we commit to all the assignments to non-input wires. Then, to demonstrate that the evaluation at the j-th gate is correctly carried

out, we prove that the corresponding two input bits and output bit satisfy the boolean relation determined by $\circ_j$. Finally, when we arrive at the N-th gate, we additionally prove that the last commitment opens to the bit 1, which serves as the evidence that $C(x_1, \ldots, x_\ell) = 1$. In the whole process, we only have to assume the security of variants of the commitment scheme from [36], which is based on the worst-case hardness of the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ problem in general lattices.

It is also worth noting that our protocol can be easily modified in order to prove unary boolean operations among committed bits (e.g., $x_1 = \mathsf{NOT}(x_2)$ or $x_1 \equiv x_2$). As discussed above, this gives us the flexibility when choosing among equivalent circuits.

Furthermore, our protocol is readily extended to additionally prove that the committed input bits $x_1, \ldots, x_\ell$ satisfy other algebraic statements, which enables interesting applications. Our attribute-based signature for circuits is obtained exactly in this manner, where we smoothly employ the Stern-like techniques from [45,41] for the Boyen signature scheme [11,53] to demonstrate that $x_1, \ldots, x_\ell$ has been properly certified by the authority. We thus obtain an interactive argument of knowledge for the combined relation, which is then repeated $\omega(\log n)$ times to achieve negligible soundness error. Finally, we use the Fiat-Shamir transformation [20] to convert the resulting protocol into a signature in the random oracle model.

1.3 Related Work

Zero-knowledge proofs in lattice-based cryptography. Earlier works on interactive and non-interactive proof systems [24,55,34,57] for lattices mostly exploit the geometric structure of worst-case lattice problems, and do not find many applications in lattice-based cryptography. More recent methods of proving relations appearing in lattice-based cryptosystems belong to the following two main families.

The first family, introduced by Lyubashevsky [47,48], uses "rejection sampling" techniques, and recently lead to relatively efficient proofs of knowledge of small secret vectors [6,3,19,58], as well as proofs of linear and multiplicative relations among committed values [7,4] in the ideal lattice setting. However, due to the nature of "rejection sampling" mechanisms, even the honest prover may fail to convince the verifier with a tiny probability: i.e., protocols in this family do not have perfect completeness. Furthermore, when proving knowledge of vectors having norm bound β, the knowledge extractor of these protocols is only guaranteed to produce witnesses of norm bound $g \cdot \beta$, for some factor $g > 1$. This factor, called the "soundness slack" in [3,19], may have an undesirable consequence: if an extracted witness has to be used in the security proof to solve a challenge SIS instance, we have to rely on the $\mathsf{SIS}_{g \cdot \beta}$ assumption, which is stronger than the SIS_β assumption required by the protocol itself. Moreover, in some advanced cryptographic constructions such as the attribute-based signature scheme for circuits considered in this work, the coordinates of extracted vectors are expected to be in $\{0, 1\}$ and/or satisfy a specific pattern. Such issues seem hard to tackle using this family of protocols.

The second family, initiated by Ling *et al.* [44], rely on "decomposition-extension" techniques in lattice-based analogues [36] of Stern's protocol [63]. Stern-like systems are less efficient than those of the first family because each protocol execution admits a constant soundness error, requiring the protocols to be repeated $\omega(\log n)$ times, where n is the security parameter, in order to achieve a negligible soundness error. On the upside, Stern-like protocols do have perfect completeness and are capable of handling a wide range of lattice-based relations [40,45,43,41,42], especially when the witnesses are not only required to be small or binary, but should also have prescribed arrangements of coordinates. Moreover, unlike protocols of the first family, the extractor of Stern-like protocols are able to output witness vectors having exactly the same properties as those expected from valid witnesses. This feature is often crucial in the design of advanced protocols involving ZK proofs. Additionally, the "soundness slack" issue is completely avoided, so that the hardness assumptions are kept "in place".

Attribute-based signatures. Attribute-Based Signature (ABS), introduced in [51,52], is a generalized primitive of digital signature with fine-grained control over identifying information. In an ABS scheme, each user, provided from an authority a signing key associated to his attribute, can sign a message under a predicate that is satisfied by his attribute. The ability to define the predicate in the signing procedure makes ABS also related to ring signature [60] but ABS provides a much more fine-grained predicates: ring signatures only allow predicates which are disjunctions over the universe of identities.

Attribute-based signatures supporting a large class of predicates is an active direction. Okamoto and Takashima [56] proposed a scheme for non-monotone span programs from bilinear groups. Tang, Li, and Liang [64] suggested a scheme for any circuits from multilinear maps. Bellare and Fuchsbauer [5] introduced policy-based signatures and showed a generic construction of an attribute-based signature scheme from a policy-based signature scheme. One of their schemes can support the class of arbitrary circuits but it suffers from a large overhead due to a Karp-reduction to an NP-complete problem. In [61], Sakai, Attrapadung and Hanaoka presented an ABS scheme for circuits with the help of Groth-Ostrovsky-Sahai proofs in the bilinear setting. It is still an open question to construct a scheme supporting arbitrary circuits that also resists quantum attacks. As an application of our zero-knowledge argument for circuits, we propose a solution for this question in the lattice setting. Note that there have been proposed several lattice-based ABS schemes [65,66,2], but they only support a restricted class of predicates, namely either AND/OR or threshold predicates.

2 Preliminaries

Notations. We assume that all vectors are column vectors. If vector $\mathbf{x}$ has coordinates $x_1, \ldots, x_m$, then we write $\mathbf{x} = (x_1, \ldots, x_m)$. When concatenating column vectors $\mathbf{x} \in \mathbb{R}^k$ and $\mathbf{y} \in \mathbb{R}^m$, for simplicity, we often use the notation $(\mathbf{x} \| \mathbf{y}) \in \mathbb{R}^{k+m}$ (instead of $(\mathbf{x}^\top \| \mathbf{y}^\top)^\top$). The column concatenation of matrices

$\mathbf{A} \in \mathbb{R}^{n\times k}$ and $\mathbf{B} \in \mathbb{R}^{n\times m}$ is denoted by $[\mathbf{A} \,|\, \mathbf{B}] \in \mathbb{R}^{n\times(k+m)}$. If S is a finite set, then $x \xleftarrow{\$} S$ means that x is chosen uniformly at random over S. If n is a positive integer, then $[n]$ denotes the set $\{1, \ldots, n\}$ and $\mathsf{negl}(n)$ denotes a negligible function in n.

2.1 Some Lattice-Based Cryptographic Ingredients

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

Definition 1 ($\mathsf{SIS}^{\infty}_{n,m,q,\beta}$ [1,22]). *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 $q > \beta \cdot \widetilde{\mathcal{O}}(\sqrt{n})$, then 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., [22,54]). Specifically, when $\beta = 1$, $q = \widetilde{\mathcal{O}}(n^{1.5})$ and $m = \mathcal{O}(n \log q)$, the $\mathsf{SIS}^{\infty}_{n,m,q,1}$ problem is at least as hard as SIVP_γ with $\gamma = \widetilde{\mathcal{O}}(n)$.

In this work, we will employ two commonly used SIS-based cryptographic tools: the KTX commitment scheme [36], and the refined version of Boyen signature scheme [11] given in [53].

The KTX commitment scheme. The scheme works with security parameter n, prime modulus $q = \widetilde{\mathcal{O}}(n^{1.5})$, and dimension $m = 2n\lceil \log_2 q \rceil$. We will consider two flavours of the scheme.

In the variant that allows committing to one single bit, the commitment key is $(\mathbf{a}, \mathbf{B}) \xleftarrow{\$} \mathbb{Z}_q^n \times \mathbb{Z}_q^{n\times m}$. To commit to a bit x, one samples randomness $\mathbf{r} \xleftarrow{\$} \{0,1\}^m$, and outputs commitment $\mathbf{c} = \mathbf{a}\cdot x + \mathbf{B}\cdot\mathbf{r} \bmod q$. Then, to open the commitment, one simply reveals $x \in \{0,1\}$ and $\mathbf{r} \in \{0,1\}^m$.

If there exist two valid openings $(x_1, \mathbf{r}_1)$ and $(x_2, \mathbf{r}_2)$ for the same commitment $\mathbf{c}$, where $x_1 \neq x_2$, then one can compute a solution to the $\mathsf{SIS}^{\infty}_{n,m+1,q,1}$ problem associated with the uniformly random matrix $[\mathbf{a} \mid \mathbf{B}] \in \mathbb{Z}_q^{n\times(m+1)}$. Thus, the scheme is computationally binding, assuming the worst-case hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$. On the other hand, by the left-over hash lemma [59], the distribution of a valid commitment $\mathbf{c}$ is statistically close to uniform over $\mathbb{Z}_q^n$. This implies that the scheme is statistically hiding.

Kawachi et al. [36] extended the above bit 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.

In this work, we will employ the bit commitment variant to commit to secret bits (e.g., input bits to circuits), and use the string commitment scheme COM as a building block for Stern-like zero-knowledge protocols.

2.2 Zero-Knowledge Argument Systems and Stern-like Protocols

We will work with statistical zero-knowledge argument systems, namely, interactive protocols where the zero-knowledge property holds against *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 two 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 for any $\widehat{\mathcal{V}}(y)$, there exists a PPT simulator $\mathcal{S}(y)$ 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* [27], 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 $(y, w') \in \mathrm{R}$.

Stern-like protocols. The statistical zero-knowledge arguments of knowledge presented in this work are Stern-like [63] protocols. In particular, they are Σ-protocols in the generalized sense defined in [35,6] (where 3 valid transcripts are needed for extraction, instead of just 2). The basic protocol consists of 3 moves: commitment, challenge, response. If the KTX statistically hiding and computationally binding string commitment scheme [36] is employed in the first move, then one obtains a statistical zero-knowledge argument of knowledge (ZKAoK) with perfect completeness, constant soundness error $2/3$, and communication cost $\mathcal{O}(|w| \cdot \log q)$, where $|w|$ denotes the total bit-size of the secret vectors. In many applications, the protocol is repeated $\kappa = \omega(\log n)$ times, for security parameter n, to achieve negligible soundness error, and then made non-interactive via the Fiat-Shamir heuristic [20]. In the random oracle model, this results in a non-interactive zero-knowledge argument of knowledge ($\mathsf{NIZKAoK}$) with bit-size $\mathcal{O}(|w| \cdot \log q) \cdot \omega(\log n)$.

An abstraction of Stern's protocol. We recall an abstraction of Stern's protocol, proposed in [41]. Let K, D, q be positive integers, where $D \geq K$ and $q \geq 2$, and let VALID be a subset of $\mathbb{Z}^D$. Suppose that $\mathcal{S}$ is a finite set such that one can associate every $\phi \in \mathcal{S}$ with a permutation Γ_ϕ of D elements, satisfying the following conditions:

$$\begin{cases} \mathbf{w} \in \mathsf{VALID} \iff \Gamma_\phi(\mathbf{w}) \in \mathsf{VALID}, \\ \text{If } \mathbf{w} \in \mathsf{VALID} \text{ and } \phi \text{ is uniform in } \mathcal{S}, \text{ then } \Gamma_\phi(\mathbf{w}) \text{ is uniform in } \mathsf{VALID}. \end{cases} \tag{2}$$

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

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

Note that, Stern's original protocol corresponds to the special case when $\mathsf{VALID} = \{\mathbf{w} \in \{0,1\}^D : \mathsf{wt}(\mathbf{w}) = k\}$ (where $\mathsf{wt}(\cdot)$ denotes the Hamming weight

and $k < D$ is a given integer), $\mathcal{S} = \mathcal{S}_D$ - hereunder the set of all permutations of D elements, and $\Gamma_\phi(\mathbf{w}) = \phi(\mathbf{w})$.

The conditions in (2) play a crucial role in proving in ZK that $\mathbf{w} \in \mathsf{VALID}$: To do so, the prover samples $\phi \hookleftarrow U(\mathcal{S})$ and let the verifier check that $\Gamma_\phi(\mathbf{w}) \in \mathsf{VALID}$, while the latter cannot learn any additional information about $\mathbf{w}$ thanks to the randomness of ϕ. Furthermore, to prove in ZK that the linear equation holds, the prover samples a masking vector $\mathbf{r}_w \hookleftarrow U(\mathbb{Z}_q^D)$, and convinces the verifier instead that $\mathbf{M} \cdot (\mathbf{w} + \mathbf{r}_w) = \mathbf{M} \cdot \mathbf{r}_w + \mathbf{v} \bmod q$.

The interaction between prover $\mathcal{P}$ and verifier $\mathcal{V}$ is described in Figure 1. The protocol employs a statistically hiding and computationally binding string commitment scheme COM (e.g., the SIS-based scheme from [36]).

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

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

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

Verification: Receiving RSP, the verifier proceeds as follows:

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

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

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

Theorem 1 ([41]). *Assume that* COM *is a statistically hiding and computationally binding string commitment scheme. Then, the protocol in Figure 1 is a statistical* ZKAoK *with perfect completeness, soundness error* $2/3$*, and communication cost* $\mathcal{O}(D \log q)$*. In particular:*

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

The proof of the Theorem 1, appeared in [41], employs standard simulation and extraction techniques for Stern-like protocols [36,44]. For completeness, we recall the proof in Appendix A.

Looking ahead, all the relations we consider in this work (Sections 3.3, 3.4 and 4.2) will be reduced to instances of the above abstract protocol.

3 Our Zero-Knowledge Arguments for Boolean Circuits

This section presents the main contributions of the paper. We first recall several notations and techniques used in previous works [63,36,44,41,42] on Stern-like protocols. We then explain and prove our core technical observations, which allow us to generalize the technique for handling the bit-wise multiplication operation from [42] to the case of arbitrary binary boolean operation. Based on these new observations, we develop protocols for proving boolean relations among committed bits, and for satisfiability of boolean circuits.

3.1 Some Previous Notations and Techniques

For any positive integer $\mathfrak{m}$, let $\mathcal{S}_{\mathfrak{m}}$ be the set of all permutations of $\mathfrak{m}$ elements and let $\mathsf{B}^2_{\mathfrak{m}}$ denote the set of binary vectors in $\{0,1\}^{2\mathfrak{m}}$ with Hamming weight $\mathfrak{m}$. Note that, for any $\mathbf{x} \in \mathbb{Z}^{2\mathfrak{m}}$ and $\pi \in \mathcal{S}_{2m}$, we have:

$$\mathbf{x} \in \mathsf{B}^2_{\mathfrak{m}} \iff \pi(\mathbf{x}) \in \mathsf{B}^2_{\mathfrak{m}}. \tag{3}$$

Stern's original protocol and its first adaptation into the lattice setting by Kawachi et al. [36] rely on the equivalence (3) to prove knowledge of a secret vector $\mathbf{x} \in \mathsf{B}^2_{\mathfrak{m}}$, using a uniformly random permutation π. Ling et al. [44] later suggested an extension technique to handle any secret vector $\mathbf{x} \in \{0,1\}^m$. To this end, one first appends $\mathfrak{m}$ coordinates in $\{0,1\}$ with suitable numbers of 0's and 1's, so that the obtained vector $\mathbf{x}^*$ belongs to $\mathsf{B}^2_{\mathfrak{m}}$, and then employs the given permuting technique.

We next recall the Stern-like techniques for proving knowledge of a single secret bit, and for proving knowledge of 3 secret bits x_1, x_2, x_3 such that x_3 is the product $x_1 \cdot x_2$, from [43] and [42], respectively.

Let $\oplus$ denote the bit-wise XOR operation. For any bit $b \in \{0,1\}$, denote by $\overline{b}$ the bit $b = b \oplus 1$. Note that, for any $b, c \in \{0,1\}$, we have $\overline{b \oplus c} = b \oplus c \oplus 1 = \overline{b} \oplus c$. For any bit b, let $\mathsf{enc}(b) = (\overline{b}, b) \in \{0,1\}^2$.

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

$$\mathbf{v} = \mathsf{enc}(b) \iff F_c(\mathbf{v}) = \mathsf{enc}(b \oplus c). \tag{4}$$

The authors of [43] showed that the equivalence (4) is helpful for proving knowledge of a secret bit x that may appear in several correlated linear equations. To this end, one extends x to $\mathsf{enc}(x) \in \{0,1\}^2$, and permutes the latter using

F_c, where c is a uniformly random bit. Seeing the permuted vector $\mathsf{enc}(x \oplus c)$, the verifier should be convinced that the original vector $\mathsf{enc}(x)$ is well-formed - which in turn implies knowledge of some bit x, while c acts as a "one-time pad" that completely hides x.

Quadratic Relations. In the context of group encryption [37], Libert et al. [42] faced the problem of "quadratic relations", in which one has to prove that a secret vector is the matrix-vector product $\mathbf{X} \cdot \mathbf{s} \bmod q$, for some modulus q, where $\mathbf{X}$ and $\mathbf{s}$ are both secret and may satisfy other equations. To this end, they decomposed $\mathbf{X}$ and $\mathbf{s}$ to bits and reduced the problem to many sub-problems, where for each of which one has to prove knowledge of secret bits x_1, x_2, x_3 such that $x_3 = x_1 \cdot x_2$. While the knowledge of x_1, x_2 can be proved using the above technique with $\mathsf{enc}(\cdot)$ and $F_\cdot(\cdot)$, Libert et al. handled the bit-product relation $x_3 = x_1 \cdot x_2$ via the following extending-then-permuting technique.

For any two bits b_1, b_2, define the vector

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

that is an extension of the bit product $b_1 \cdot b_2$. Next, define a specific type of permutation, associated with two bits c_1, c_2, which is formalized as follows.

Definition 2 (Permutation $T_{\cdot,\cdot}(\cdot)$, first appeared in [42]). *For any two bits $c_1, c_2 \in \{0,1\}$, define T_{c_1,c_2} as the permutation that transforms integer vector $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1}) \in \mathbb{Z}^4$ into vector*

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

Then, the following equivalence holds. For any bits b_1, b_2, c_1, c_2 and any vector $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1}) \in \mathbb{Z}^4$,

$$\mathbf{v} = \mathsf{ext}(b_1, b_2) \iff T_{c_1,c_2}(\mathbf{v}) = \mathsf{ext}(b_1 \oplus c_1, b_2 \oplus c_2). \tag{5}$$

As a result, to prove $x_3 = x_1 \cdot x_2$, one can extend x_3 to vector $\mathsf{ext}(x_1, x_2)$, then permute the latter using T_{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., $\mathsf{ext}(x_1, x_2)$, 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 appearances of x_1 and x_2, respectively.

3.2 Our Observations

We aim to design a mechanism that can be used to prove knowledge of secret bits x_1, x_2, x_3 such that $x_3 = x_1 \circ x_2$, where $\circ$ could be an arbitrary binary boolean operation. Note that there are 16 such operations in total, including AND, OR, XOR, NAND and NOR. Our starting point is the technique from [42]

for proving bit product $x_3 = x_1 \cdot x_2$, which we recalled above. As the bit-wise multiplication operation $\cdot$ can be interpreted as the AND operation, this would give a method for proving $x_3 = x_1 \ \mathsf{AND}\ x_2$, but would not be sufficient for our purpose.

We then investigate the core ideas underlying the extending-then-permuting technique from [42], and observe that, the bit-wise operation in question does not necessarily have to be the AND operation. Indeed, somewhat interestingly, we find that the technique can be generalized to be applicable to arbitrary binary boolean operation $\circ$. Let us begin with the following definition.

Definition 3 (Extended vector $\mathsf{ext}_\circ(\cdot,\cdot)$). *For any binary boolean operation $\circ : \{0,1\} \times \{0,1\} \to \{0,1\}$, and for any $b_1, b_2 \in \{0,1\}$, define the vector $\mathsf{ext}_\circ(b_1, b_2) \in \{0,1\}^4$ as follows:*

$$\mathsf{ext}_\circ(b_1, b_2) = \big(\overline{b}_1 \circ \overline{b}_2,\ \overline{b}_1 \circ b_2,\ b_1 \circ \overline{b}_2,\ b_1 \circ b_2\big).$$

Our definition of $\mathsf{ext}_\circ(\cdot,\cdot)$ subsumes the definition of $\mathsf{ext}(\cdot,\cdot)$ from [42] as a special case when $\circ$ is AND. Next, we prove in Lemma 1 that the equivalence (5) still holds with respect to $\mathsf{ext}_\circ(\cdot,\cdot)$ and permutation $T_{\cdot,\cdot}(\cdot)$.

Lemma 1. *For any binary boolean operation $\circ$, any $b_1, b_2, c_1, c_2 \in \{0,1\}$ and any $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1}) \in \mathbb{Z}^4$, it holds that:*

$$\mathbf{v} = \mathsf{ext}_\circ(b_1, b_2) \iff T_{c_1,c_2}(\mathbf{v}) = \mathsf{ext}_\circ(b_1 \oplus c_1, b_2 \oplus c_2). \tag{6}$$

Proof. Let $\mathbf{w} = (w_{0,0}, w_{0,1}, w_{1,0}, w_{1,1}) \in \mathbb{Z}^4$ be the vector $T_{c_1,c_2}(\mathbf{v})$. By definition 2, we have:

$$\begin{aligned} & w_{0,0} = v_{c_1,c_2},\ w_{0,1} = v_{c_1,\overline{c}_2},\ w_{1,0} = v_{\overline{c}_1,c_2},\ w_{1,1} = v_{\overline{c}_1,\overline{c}_2} \\ \iff & w_{i,j} = v_{c_1 \oplus i,\, c_2 \oplus j},\ \forall (i,j) \in \{0,1\} \times \{0,1\} \end{aligned}$$

Meanwhile, let $\mathbf{t} = (t_{0,0}, t_{0,1}, t_{1,0}, t_{1,1}) \in \{0,1\}^4$ be the vector $\mathsf{ext}_\circ(b_1 \oplus c_1, b_2 \oplus c_2)$. By Definition 3, we have:

$$\begin{aligned} & \begin{cases} t_{0,0} = (\overline{b_1 \oplus c_1}) \circ (\overline{b_2 \oplus c_2}) = (\overline{b}_1 \oplus c_1 \oplus 0) \circ (\overline{b}_2 \oplus c_2 \oplus 0) \\ t_{0,1} = (\overline{b_1 \oplus c_1}) \circ (b_2 \oplus c_2) = (\overline{b}_1 \oplus c_1 \oplus 0) \circ (\overline{b}_2 \oplus c_2 \oplus 1) \\ t_{1,0} = (b_1 \oplus c_1) \circ (\overline{b_2 \oplus c_2}) = (\overline{b}_1 \oplus c_1 \oplus 1) \circ (\overline{b}_2 \oplus c_2 \oplus 0) \\ t_{1,1} = (b_1 \oplus c_1) \circ (b_2 \oplus c_2) = (\overline{b}_1 \oplus c_1 \oplus 1) \circ (\overline{b}_2 \oplus c_2 \oplus 1) \end{cases} \\ \iff & t_{i,j} = (\overline{b}_1 \oplus c_1 \oplus i) \circ (\overline{b}_2 \oplus c_2 \oplus j),\ \forall (i,j) \in \{0,1\} \times \{0,1\}. \end{aligned}$$

Hence, the following equivalences hold for any $b_1, b_2, c_1, c_2 \in \{0,1\}$ and any $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1}) \in \mathbb{Z}^4$:

$$\begin{aligned} & T_{c_1,c_2}(\mathbf{v}) = \mathsf{ext}_\circ(b_1 \oplus c_1, b_2 \oplus c_2) \iff \mathbf{w} = \mathbf{t} \\ \iff & v_{c_1 \oplus i,\, c_2 \oplus j} = (\overline{b}_1 \oplus c_1 \oplus i) \circ (\overline{b}_2 \oplus c_2 \oplus j),\ \forall (i,j) \in \{0,1\} \times \{0,1\} \\ \iff & v_{i,j} = (\overline{b}_1 \oplus i) \circ (\overline{b}_2 \oplus j),\ \forall (i,j) \in \{0,1\} \times \{0,1\} \\ \iff & v_{0,0} = \overline{b}_1 \circ \overline{b}_2;\ \ v_{0,1} = \overline{b}_1 \circ b_2;\ \ v_{1,0} = b_1 \circ \overline{b}_2;\ \ v_{1,1} = b_1 \circ b_2 \\ \iff & \mathbf{v} = \mathsf{ext}_\circ(b_1, b_2). \end{aligned}$$

This concludes the lemma. □

In the next subsections, we will employ the above techniques to develop lattice-based zero-knowledge protocols for boolean relations among committed bits and for circuit satisfiability.

3.3 Proving Boolean Relations Among Committed Bits

Let us consider the following fundamental problem: proving knowledge of committed bits x_1, x_2, x_3 satisfying the relation $x_1 \circ x_2 = x_3$, where $\circ$ is an arbitrary binary boolean operation. We will work with the lattice-based bit commitment scheme proposed by Kawachi et al. [36], which is recalled in Section 2.1.

Let the commitment key be $(\mathbf{a}, \mathbf{B}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{n \times m}$, where n is the security parameter, $q = \widetilde{\mathcal{O}}(n^{1.5})$ and $m = 2n\lceil \log q \rceil$. We will construct a zero-knowledge argument of knowledge for the relation $\mathrm{R}_{\mathrm{bit}}$ defined below.

$$\mathrm{R}_{\mathrm{bit}} = \Big((\mathbf{a}, \mathbf{B}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{n \times m} \times (\mathbb{Z}_q^n)^3 ; (\{x_i, \mathbf{r}_i \in \{0,1\} \times \{0,1\}^m\}_{i=1}^3) : \\ (x_3 = x_1 \circ x_2) \quad \wedge \quad \big(\forall i = 1,2,3 : \ \mathbf{a} \cdot x_i + \mathbf{B} \cdot \mathbf{r}_i = \mathbf{c}_i \bmod q\big) \Big).$$

Our strategy is to reduce the relation $\mathrm{R}_{\mathrm{bit}}$ to an instance of the abstract relation $\mathrm{R}_{\mathrm{abstract}}$ considered in Section 2.2. In the process, we will perform several transformations with public matrices and secret vectors, and rely on the Stern-like techniques discussed in Sections 3.1 and 3.2.

Reduction from $\mathrm{R}_{\mathrm{bit}}$ *to* $\mathrm{R}_{\mathrm{abstract}}$. We first observe that the three equations $\{\mathbf{a} \cdot x_i + \mathbf{B} \cdot \mathbf{r}_i = \mathbf{c}_i \bmod q\}_{i=1,2,3}$ can be unified as:

$$\underbrace{\begin{bmatrix} \mathbf{a} & & \\ & \mathbf{a} & \\ & & \mathbf{a} \end{bmatrix}}_{\widetilde{\mathbf{A}} \in \mathbb{Z}_q^{3n \times 3}} \cdot \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}}_{\tilde{\mathbf{x}} \in \{0,1\}^3} + \underbrace{\begin{bmatrix} \mathbf{B} & & \\ & \mathbf{B} & \\ & & \mathbf{B} \end{bmatrix}}_{\widetilde{\mathbf{B}} \in \mathbb{Z}_q^{3n \times 3m}} \cdot \underbrace{\begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{pmatrix}}_{\tilde{\mathbf{r}} \in \{0,1\}^{3m}} = \underbrace{\begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{pmatrix}}_{\mathbf{v} \in \mathbb{Z}_q^{3n}} \bmod q. \tag{7}$$

Our task is to prove knowledge of $x_1, x_2 \in \{0,1\}$ and $\tilde{\mathbf{r}} \in \{0,1\}^{3m}$ such that (7) holds for $\tilde{\mathbf{x}} = (x_1,\ x_2,\ x_1 \circ x_2)$. To this end, we use the techniques of Sections 3.1 and 3.2 to extend $\tilde{\mathbf{x}}$ to $\mathbf{x}^* = \big(\ \mathsf{enc}(x_1) \parallel \mathsf{enc}(x_2) \parallel \mathsf{ext}_\circ(x_1, x_2)\ \big) \in \{0,1\}^8$, i.e., $\mathbf{x}^* = \big(\ \overline{x}_1,\ x_1,\ \overline{x}_2,\ x_2,\ \overline{x}_1 \circ \overline{x}_2,\ \overline{x}_1 \circ x_2,\ x_1 \circ \overline{x}_2,\ x_1 \circ x_2\ \big)$.

We also append $3m$ coordinates in $\{0,1\}$ to vector $\tilde{\mathbf{r}}$ to obtain an extended vector $\mathbf{r}^* \in \{0,1\}^{6m}$ of Hamming weight exactly $3m$, i.e., $\mathbf{r}^* \in \mathsf{B}_{3m}^2$.

Given the above extensions of secret vectors $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{r}}$, we can insert suitable zero-columns to matrices $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$ from equation (7) to obtain public matrices $\mathbf{A}^* \in \mathbb{Z}_q^{3n \times 8}$ and $\mathbf{B}^* \in \mathbb{Z}_q^{3n \times 6m}$, respectively, so that the equation is preserved. Namely, we have:

$$\mathbf{A}^* \cdot \mathbf{x}^* + \mathbf{B}^* \cdot \mathbf{r}^* = \mathbf{v} \bmod q.$$

Now, we form the matrix $\mathbf{M} = \big[\mathbf{A}^* \mid \mathbf{B}^*\big] \in \mathbb{Z}_q^{K \times D}$, where $K = 3n$ and $D = 6m + 8$. Then, we have equation $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$, where

$$\mathbf{w} = (\ \mathbf{x}^* \parallel \mathbf{r}^*\ ) = (\ \mathsf{enc}(x_1) \parallel \mathsf{enc}(x_2) \parallel \mathsf{ext}_\circ(x_1, x_2) \parallel \mathbf{r}^*\ ) \in \{0,1\}^D.$$

Having performed the above transformations, we now specify the VALID that contains the obtained vector $\mathbf{w}$, the set $\mathcal{S}$ and permutations $\{\Gamma_\phi : \phi \in \mathcal{S}\}$, such that the conditions in (2) hold. To this end, we let VALID be the set of all vectors in $\{0,1\}^D$ that have the form $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{r}^*)$, where $\mathbf{r}^* \in \mathsf{B}_{3m}^2$ and

$$\exists\, x_1, x_2 \in \{0,1\} \text{ such that } \mathbf{y}_1 = \mathsf{enc}(x_1),\ \mathbf{y}_2 = \mathsf{enc}(x_2) \text{ and } \mathbf{y}_3 = \mathsf{ext}_\circ(x_1, x_2).$$

It is clear that our vector $\mathbf{w}$ belongs to this set VALID. Next, we define $\mathcal{S} = \{0,1\}^2 \times \mathcal{S}_{6m}$. (Recall that $\mathcal{S}_{6m}$ is the set of all permutations of $6m$ elements.) For every $\phi = \big((b_1, b_2), \pi\big) \in \mathcal{S}$, we let Γ_ϕ be the permutation that, when acting on vector $\mathbf{w} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{r}^*) \in \mathbb{Z}^D$, where $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{Z}^2$, $\mathbf{y}_3 \in \mathbb{Z}^4$ and $\mathbf{r}^* \in \mathbb{Z}^{6m}$, it transforms $\mathbf{w}$ into vector:

$$\Gamma_\phi(\mathbf{w}) = \big(\ F_{b_1}(\mathbf{y}_1) \parallel F_{b_2}(\mathbf{y}_2) \parallel T_{b_1,b_2}(\mathbf{y}_3) \parallel \pi(\mathbf{r}^*)\ \big).$$

Based on the equivalences observed in (4), (6) and (3), it can be seen that VALID, $\mathcal{S}$ and Γ_ϕ satisfy the conditions specified in (2). Hence, we have reduced the relation $\mathrm{R}_{\mathrm{bit}}$ to an instance of the abstract relation from Section 2.2.

The interactive protocol. Given the above preparations, our interactive protocol works as follows.

- The public input consists of matrix $\mathbf{M}$ and vector $\mathbf{v}$, which are built from $(\mathbf{a}, \mathbf{B}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, as discussed above.
- The prover's witness is vector $\mathbf{w} \in \mathsf{VALID}$, which is obtained from the original witnesses $\{x_i, \mathbf{r}_i\}_{i=1}^3$, as described above.

Both parties then run the protocol of Figure 1. The protocol uses the KTX string commitment scheme COM, which is statistically hiding and computationally binding if the $\mathsf{SIVP}_{\tilde{\mathcal{O}}(n)}$ problem is hard. We therefore obtain the following result, as a corollary of Theorem 1.

Lemma 2. *Let us assume that the* $\mathsf{SIVP}_{\tilde{\mathcal{O}}(n)}$ *problem is hard. Then the protocol described above is a statistical* ZKAoK *of the relation* $\mathrm{R}_{\mathrm{bit}}$*, with perfect completeness, soundness error* $2/3$ *and communication cost* $\mathcal{O}(m \log q)$*.*

Proof. For simulation, we simply run the simulator of Theorem 1. As for extraction, we invoke the knowledge extractor of Theorem 1 to obtain a vector $\mathbf{w}' \in \mathsf{VALID}$ such that $\mathbf{M} \cdot \mathbf{w}' = \mathbf{v} \bmod q$. Then, by "backtracking" the transformations being done, we can extract from $\mathbf{w}'$ bits x_1', x_2', x_3' and vectors $\mathbf{r}_1', \mathbf{r}_2', \mathbf{r}_3' \in \{0,1\}^m$ such that:

$$\big(\forall i = 1,2,3 :\ \mathbf{a} \cdot x_i' + \mathbf{B} \cdot \mathbf{r}_i' = \mathbf{c}_i \bmod q\big) \quad \wedge \quad \big(x_1' \circ x_2' = x_3'\big).$$

The perfect completeness, soundness error and communication cost of the protocol directly follow from those of the abstract protocol in Section 2.2. $\square$

We remark that, the above protocol can easily be modified to handle the simpler scenarios where one has to prove unary relations among committed bits x_1, x_2, such as $x_1 = x_2$ or $x_1 = \mathsf{NOT}(x_2)$. Indeed, in both cases, one can translate the two equations $\{\mathbf{a} \cdot x_i + \mathbf{B} \cdot \mathbf{r}_i = \mathbf{c}_i \bmod q\}_{i=1,2}$ into one of the form $\widetilde{\mathbf{A}} \cdot \mathsf{enc}(x_2) + \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \cdot \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix}$, where $\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \mathbf{a} \\ \mathbf{0} & \mathbf{a} \end{bmatrix}$ if $x_1 = x_2$, and $\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{0} & \mathbf{a} \end{bmatrix}$ if $x_1 = \mathsf{NOT}(x_2)$. This allows to reduce the statement into a special case of the abstract one.

Once we can prove binary and unary operations among committed bits, we can combine such sub-protocols to develop argument systems for proving satisfiability of boolean circuits for committed inputs. Our protocols are flexible in the sense that we are able to work with various functionally complete sets of operations, such as $\{\mathsf{AND}, \mathsf{OR}, \mathsf{NOT}\}$, $\{\mathsf{NAND}\}$, $\{\mathsf{NOR}\}$, $\{\mathsf{IMPLY}, \mathsf{NOT}\}$ or $\{\mathsf{IMPLY}, \mathsf{XOR}\}$. The sub-protocols smoothly interact with each other, because they all operate in Stern's framework.

3.4 Proving Satisfiability of Boolean Circuits

We now present a lattice-based zero-knowledge argument system for proving knowledge of a committed input satisfying a given boolean circuit. As discussed above, we can flexibly handle arbitrary circuits of fan-ins at most 2. However, in the following, for simplicity of description, we consider circuits represented entirely by binary gates.

Let C be a boolean circuit that has N gates labelled by operations $\circ_1, \ldots, \circ_N$. The topology of C is specified by two publicly known functions g and h mapping $\{1, \ldots, N\}$ to $\{1, \ldots, \ell+N-1\}$. Given an ℓ-bit input $(x_1, \ldots, x_\ell)$, the assignments to non-input wires in C are denoted as $x_{\ell+1}, \ldots, x_{\ell+N}$, and are computed as:

$$\forall\, j = 1, \ldots, N : \; x_{\ell+j} = x_{g(j)} \circ_j x_{h(j)}.$$

Note that, $C(x_1, \ldots, x_\ell) = 1$ if and only if $x_{\ell+N} = x_{g(N)} \circ_N x_{h(N)} = 1$.

We again will work with the bit commitment scheme from [36], which is computationally binding assuming the worst-case hardness of the $\mathsf{SIVP}_{\tilde{\mathcal{O}}(n)}$ problem. The general idea of the protocol is as follows. In the initialization phase, the prover computes commitments $\mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N}$ to all the bits $x_1, \ldots, x_{\ell+N}$. Then, in order to convince the verifier that $C(x_1, \ldots, x_\ell) = 1$, the prover demonstrates to the latter that:

- He knows all the bits $x_1, \ldots, x_{\ell+N}$ and randomness $\mathbf{r}_1, \ldots, \mathbf{r}_{\ell+N}$ used to compute $\mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N}$, which justifies the validity of the commitments.
- The circuit evaluation on input $x_1, \ldots, x_\ell$ is correctly carried out: For all $j \in [N]$, at the j-th gate in circuit C, the committed inputs $x_{g(j)}, x_{h(j)}$ and committed output $x_{\ell+j}$ satisfy $x_{\ell+j} = x_{g(j)} \circ_j x_{h(j)}$;
- $\mathbf{c}_{\ell+N}$ is a valid commitment to the bit 1, with randomness $\mathbf{r}_{\ell+N}$.

More formally, our protocol an argument of knowledge for the relation $\mathrm{R}_{\mathrm{circuit}}$ defined below.

Definition 4. *Define* $\mathrm{R}_{\mathrm{circuit}} = \big\{(\mathbf{a}, \mathbf{B}, \mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N}), x_1, \ldots, x_{\ell+N}, \mathbf{r}_1, \ldots, \mathbf{r}_{\ell+N}\big\}$, *as a relation where*

$$\begin{cases} \mathbf{a} \in \mathbb{Z}_q^n;\ \mathbf{B} \in \mathbb{Z}_q^{n\times m};\ \mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N} \in \mathbb{Z}_q^n \\ x_1, \ldots, x_{\ell+N} \in \{0,1\};\ \mathbf{r}_1, \ldots, \mathbf{r}_{\ell+N} \in \{0,1\}^m \end{cases}$$

satisfy

1. *For all* $i = 1, \ldots, \ell$: $\mathbf{a} \cdot x_i + \mathbf{B} \cdot \mathbf{r}_i = \mathbf{c}_i \bmod q$.
2. *For all* $j = 1, \ldots, N$:

$$\begin{cases} \mathbf{a} \cdot x_{g(j)} + \mathbf{B} \cdot \mathbf{r}_{g(j)} = \mathbf{c}_{g(j)} \bmod q, \\ \mathbf{a} \cdot x_{h(j)} + \mathbf{B} \cdot \mathbf{r}_{h(j)} = \mathbf{c}_{h(j)} \bmod q, \\ \mathbf{a} \cdot \big(x_{g(j)} \circ_j x_{h(j)}\big) + \mathbf{B} \cdot \mathbf{r}_{\ell+j} = \mathbf{c}_{\ell+j} \bmod q. \end{cases}$$

3. $\mathbf{B} \cdot \mathbf{r}_{\ell+N} = \mathbf{c}_{\ell+N} - \mathbf{a} \cdot 1 \bmod q$.

Reduction from $\mathrm{R}_{\mathrm{circuit}}$ *to* $\mathrm{R}_{\mathrm{abstract}}$. As in Section 3.3, our strategy is to reduce $\mathrm{R}_{\mathrm{circuit}}$ into an instance of the abstract relation from Section 2.2. Let us first combine all the secret objects appearing in the equations considered by $\mathrm{R}_{\mathrm{circuit}}$ into vector $\tilde{\mathbf{x}} \in \{0,1\}^{(\ell+3N)(m+1)}$ of the form

$$\tilde{\mathbf{x}} = \big(\, x_1, \ldots,\ x_\ell,\ x_{g(1)},\ x_{h(1)},\ x_{g(1)} \circ_1 x_{h(1)}, \ldots,\ x_{g(N)},\ x_{h(N)},\ x_{g(N)} \circ_N x_{h(N)} \| \\ \mathbf{r}_1 \,\|\, \ldots \,\|\, \mathbf{r}_\ell \,\|\, \mathbf{r}_{g(1)} \,\|\, \mathbf{r}_{h(1)} \,\|\, \mathbf{r}_{\ell+1} \,\|\, \ldots \,\|\, \mathbf{r}_{g(N)} \,\|\, \mathbf{r}_{h(N)} \,\|\, \mathbf{r}_{\ell+N} \,\big).$$

Next, we let $\mathbf{v} \in \mathbb{Z}_q^K$, where $K = (\ell + 3N + 1)n$, be the concatenation of public vectors on the right-hand side of the considered equations, i.e.,

$$\mathbf{v} = \big(\, \mathbf{c}_1 \,\|\, \ldots \,\|\, \mathbf{c}_\ell \,\|\, \mathbf{c}_{g(1)} \| \,\mathbf{c}_{h(1)} \,\|\, \mathbf{c}_{\ell+1} \,\|\, \ldots \,\|\, \mathbf{c}_{g(N)} \,\|\, \mathbf{c}_{h(N)} \,\|\, \mathbf{c}_{\ell+N} \,\|\, (\mathbf{c}_{\ell+N} - \mathbf{a}) \,\big).$$

Then, note that, one can build public matrix $\widetilde{\mathbf{M}}$ such that all the considered equations are unified to one of the form: $\widetilde{\mathbf{M}} \cdot \tilde{\mathbf{x}} = \mathbf{v} \bmod q$. Now, we extend vector $\tilde{\mathbf{x}}$ to vector $\mathbf{w} \in \{0,1\}^D$, where $D = 2\ell + 8N + 2m(\ell + 3N)$, such that $\mathbf{w}$ has the form:

$$\big(\mathbf{y}_1 \,\|\, \ldots \,\|\, \mathbf{y}_\ell \,\|\, \mathbf{z}_{1,1} \,\|\, \mathbf{z}_{1,2} \,\|\, \mathbf{t}_1 \,\|\, \ldots \,\|\, \mathbf{z}_{N,1} \,\|\, \mathbf{z}_{N,2} \,\|\, \mathbf{t}_N \,\|\, \mathbf{r}^* \,\big), \tag{8}$$

where:

- For each $i \in [\ell]$: $\mathbf{y}_i = \mathsf{enc}(x_i) \in \{0,1\}^2$.
- For each $j \in [N]$: $\mathbf{z}_{j,1} = \mathsf{enc}(x_{g(j)}) \in \{0,1\}^2$, $\mathbf{z}_{j,2} = \mathsf{enc}(x_{h(j)}) \in \{0,1\}^2$ and

$$\mathbf{t}_j = \mathsf{ext}_{\circ_j}(x_{g(j)}, x_{h(j)}) \in \{0,1\}^4.$$

- $\mathbf{r}^* \in \mathsf{B}^2_{m(\ell+3N)}$, obtained by appending $m(\ell+3N)$ suitable $\{0,1\}$-coordinates to vector $(\mathbf{r}_1 \,\|\, \ldots \,\|\, \mathbf{r}_\ell \,\|\, \mathbf{r}_{g(1)} \,\|\, \mathbf{r}_{h(1)} \,\|\, \mathbf{r}_{\ell+1} \,\|\, \ldots \,\|\, \mathbf{r}_{g(N)} \,\|\, \mathbf{r}_{h(N)} \,\|\, \mathbf{r}_{\ell+N})$.

At the same time, we insert suitable zero-columns to matrix $\widetilde{\mathbf{M}}$ to obtain public matrix $\mathbf{M} \in \mathbb{Z}_q^{K \times D}$, so that we get the desired equation $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$.

Now, let us define VALID as the set of all vectors in $\{0,1\}^D$, having the form (8), such that the following hold:

1. There exist bits $x_1, \ldots, x_\ell$, $e_{1,1}, e_{1,2}, \ldots, e_{N,1}, e_{N,2}$ satisfying
   - For all $i \in [\ell]$: $\mathbf{y}_i = \mathsf{enc}(x_i)$.
   - For all $j \in [N]$: $\mathbf{z}_{j,1} = \mathsf{enc}(e_{j,1})$, $\mathbf{z}_{j,2} = \mathsf{enc}(e_{j,2})$ and $\mathbf{t}_j = \mathsf{ext}_{\circ_j}(e_{j,1}, e_{j,2})$.
2. $\mathbf{r}^* \in \mathsf{B}^2_{m(\ell+3N)}$.

By construction, our vector $\mathbf{w}$ belongs to this tailored set VALID. Next, we define the set $\mathcal{S}$ and permutations $\{\Gamma_\phi : \phi \in \mathcal{S}\}$ so that the conditions in (2) are satisfied. To this end, we let $\mathcal{S} := \{0,1\}^{\ell+2N} \times \mathcal{S}_{2m(\ell+3N)}$, and associate each element

$$\phi = \big((b_1, \ldots, b_\ell, p_{1,1}, p_{1,2}, \ldots, p_{N,1}, p_{N,2}), \pi \big) \in \mathcal{S}$$

with the permutation Γ_ϕ that, when acting on vector $\mathbf{w} \in \mathbb{Z}^D$, whose blocks are denoted as in (8), Γ_ϕ transforms the blocks of $\mathbf{w}$ as follows:

$$\begin{aligned}\Gamma_\phi(\mathbf{w}) = \ & \big(F_{b_1}(\mathbf{y}_1) \,\|\, \ldots \,\|\, F_{b_\ell}(\mathbf{y}_\ell) \,\|\, F_{p_{1,1}}(\mathbf{z}_{1,1}) \,\|F_{b_{1,2}}(\mathbf{z}_{1,2}) \,\|T_{p_{1,1},p_{1,2}}(\mathbf{t}_1) \,\|\, \ldots \\ & \|\, F_{p_{N,1}}(\mathbf{z}_{N,1}) \,\|\, F_{p_{N,2}}(\mathbf{z}_{N,2}) \,\|\, T_{p_{N,1},p_{N,2}}(\mathbf{t}_N) \,\|\, \pi(\mathbf{r}^*) \big),\end{aligned}$$

Based on the equivalences observed in (4), (6) and (3), it can be checked that VALID, $\mathcal{S}$ and Γ_ϕ satisfy the conditions specified in (2). Hence, we have reduced the relation $\mathrm{R}_{\mathrm{circuit}}$ to an instance of the abstract relation from Section 2.2.

The interactive protocol. Given the above preparations, our interactive protocol works as follows.

- The public input consists of matrix $\mathbf{M}$ and vector $\mathbf{v}$, which are built from $(\mathbf{a}, \mathbf{B}, \mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N})$, as discussed above.
- The prover's witness is vector $\mathbf{w} \in \mathsf{VALID}$, which is obtained from the original witnesses $\{x_i, \mathbf{r}_i\}_{i=1}^{\ell+N}$, as described above.

Both parties then run the protocol of Figure 1. The protocol uses the KTX string commitment scheme COM, which is statistically hiding and computationally binding if the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ problem is hard. We therefore obtain the following result.

Theorem 2. *Assume that* $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ *is hard. Then the protocol described above is a statistical* ZKAoK *for the relation* $\mathrm{R}_{\mathrm{circuit}}$ *with perfect completeness, soundness error* $2/3$ *and communication cost* $\mathcal{O}(m(\ell+N)\log q)$.

Moreover, the witness $(x'_1, \ldots, x'_\ell, x'_{\ell+1}, \ldots, x'_{\ell+N}, \mathbf{r}'_1, \ldots, \mathbf{r}'_{\ell+N})$ *produced by the knowledge extractor satisfies* $C(x'_1, \ldots, x'_\ell) = 1$.

Proof. The perfect completeness, soundness error 2/3 and communication cost $\mathcal{O}(m(\ell + N)\log q)$ of the protocol directly follow from those of the abstract protocol from Section 2.2. (Note that, we have $D = 2\ell + 8N + 2m(\ell + 3N)$ here.)

For simulation, we simply run the simulator of Theorem 1. For extraction, we run the knowledge extractor of Theorem 1 to obtain a vector $\mathbf{w}' \in \mathsf{VALID}$ such that $\mathbf{M} \cdot \mathbf{w}' = \mathbf{v} \bmod q$. By "backtracking" the transformations being done, we can extract from vector $\mathbf{w}'$ the following bits and vectors

- ℓ bits $x'_1, \ldots, x'_\ell$, and $2N$ bits denoted by $e_{1,1}, e_{1,2}, \ldots, e_{N,1}, e_{N,2}$;
- Vectors $\mathbf{r}'_1, \ldots, \mathbf{r}'_{\ell+N} \in \{0,1\}^m$, as well as $2N$ vectors in $\{0,1\}^m$ denoted by $\mathbf{s}_{1,1}, \mathbf{s}_{1,2}, \ldots, \mathbf{s}_{N,1}, \mathbf{s}_{N,2}$,

such that the below conditions hold

1. For each $i \in [\ell]$: $\mathbf{a} \cdot x'_i + \mathbf{B} \cdot \mathbf{r}'_i = \mathbf{c}_i \bmod q$.
2. For each $j \in [N]$,
$$\begin{cases} \mathbf{a} \cdot e_{j,1} + \mathbf{B} \cdot \mathbf{s}_{j,1} = \mathbf{c}_{g(j)} \bmod q, \\ \mathbf{a} \cdot e_{j,2} + \mathbf{B} \cdot \mathbf{s}_{j,2} = \mathbf{c}_{h(j)} \bmod q, \\ \mathbf{a} \cdot \left(e_{j,1} \circ_j e_{j,2}\right) + \mathbf{B} \cdot \mathbf{r}'_{\ell+j} = \mathbf{c}_{\ell+j} \bmod q. \end{cases}$$
3. $\mathbf{a} \cdot 1 + \mathbf{B} \cdot \mathbf{r}'_{\ell+N} = \mathbf{c}_{\ell+N} \bmod q$.

Given $x'_1, \ldots, x'_\ell$ and the public functions g, h, we can compute $x'_{\ell+1}, \ldots, x'_{\ell+N}$ as $x'_{\ell+j} = x'_{g(j)} \circ_j x'_{h(j)}$ for each $j \in [N]$. Before proceeding further, we remark that, assuming the hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$, the bit commitment scheme being used is computationally binding.

Now, for j runs from 1 to N, observe that $\mathbf{c}_{g(j)}$ opens to $x'_{g(j)}$ (with randomness $\mathbf{r}'_{g(j)}$) on one hand, and to $e_{j,1}$ (with randomness $\mathbf{s}_{j,1}$) on the other hand. Based on the computational binding property of the bit commitment scheme, we can deduce that $e_{j,1} = x'_{g(j)}$. Similarly, we can deduce that $e_{j,2} = x'_{h(j)}$. This in turn implies that $\mathbf{c}_{\ell+j}$ opens to $e_{j,1} \circ_j e_{j,2} = x'_{g(j)} \circ_j x'_{h(j)} = x'_{\ell+j}$.

After N steps, we finally obtain that $\mathbf{c}_{\ell+N}$ opens to $x'_{\ell+N}$ on one hand, and to the bit 1 on the other hand. This implies that $x'_{\ell+N} = 1$. In other words, we have $C(x'_1, \ldots, x'_\ell) = 1$. □

4 Attribute-Based Signatures for Circuits from Lattices

This section presents an application of our zero-knowledge protocols to attribute-based signatures (ABS). The background on ABS, including its definition and its associated security notions, is recalled in Appendix B. We describe our lattice-based ABS scheme for circuits in Section 4.1. The zero-knowledge argument underlying the scheme is then presented in Section 4.2. Finally, we provide detailed analysis in Section 4.3 and Appendix C.

4.1 Description of Our Scheme

Our scheme follows a common approach for designing ABS, put forward in [51,52], which employs an ordinary signature scheme and a non-interactive proof of knowledge, in the following manner. The user, who requests a secret key for an attribute $\mathbf{x}$, is issued a signature $\sigma = \mathsf{sk}_{\mathbf{x}}$ on the "message" $\mathbf{x}$ by the authority. Then, when generating an ABS associated with predicate/circuit C, the user proves in zero-knowledge that: (i) he possesses a valid message-signature pair $(\mathbf{x}, \sigma)$ with respect to the authority's verification key; (ii) $\mathbf{x}$ satisfies the predicate/circuit C. If the ordinary signature scheme is unforgeable and the proof of knowledge is sound, then the resulting ABS scheme is unforgeable. Furthermore, the scheme is private if the proof system is at least witness-indistinguishable. When instantiating this generic construction based on concrete computational assumptions, the main technical difficulty consists of designing proof systems that are capable of handling the possibly non-algebraic statements determined by C. Another notable difficulty is to extend such proof system to simultaneously and smoothly address the algebraic statements associated with the signature layer.

Our zero-knowledge argument from Section 3.4 is therefore a good starting point towards designing lattice-based ABS scheme for circuits. Our next step is to choose a lattice-based signature scheme that admits a proof of knowledge of a valid message-signature pair, which is compatible with our protocol. To this end, we employ Boyen's signature [11] and its supporting protocol proposed by Ling et al.[45]. Then, using Ling et al.'s techniques, we manage to extend our protocol from Section 3.4 to additionally prove knowledge of a Boyen signature on the secret bits $x_1, \ldots, x_\ell$ committed in $\mathbf{c}_1, \ldots, \mathbf{c}_\ell$. The resulting protocol is then converted into a non-interactive ZKAoK in the random oracle model with negligible soundness error, which effectively gives us an ABS scheme via the Fiat-Shamir heuristic [20].

Boyen's lattice-based signature scheme. The scheme works with security parameter n, message space $\{0,1\}^\ell$, prime modulus $q = \mathcal{O}(\ell \cdot n^2)$, dimension $m \geq 2n \log q$, and a norm bound $\beta = \widetilde{\mathcal{O}}(\sqrt{\ell n})$. The public key is a tuple $(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u})$, and the signing key is a trapdoor $\mathbf{T}_{\mathbf{A}}$, generated together with matrix $\mathbf{A}$ by a PPT algorithm $\mathsf{GenTrap}(n, m, q)$ (see [22,53]).

The signature on a message $(x_1, \ldots, x_\ell)$ is an integer vector $\mathbf{s} \in [-\beta, \beta]^{2m}$ satisfying $\left[\mathbf{A} \mid \mathbf{A}_0 + \sum_{i=1}^{\ell} x_i \cdot \mathbf{A}_i\right] \cdot \mathbf{s} = \mathbf{u} \bmod q$. It is generated using $\mathbf{T}_{\mathbf{A}}$ via the basis extension algorithm from [15] and the pre-image sampling algorithm from [22]. It follows from the improved security reduction in [53] that the scheme is unforgeable under adaptive chosen-message attack if the $\mathsf{SIS}^{\infty}_{n,m,q,\beta'}$ problem is hard for some $\beta' = \ell \cdot \widetilde{\mathcal{O}}(n)$. Therefore, for the given parameters, the security of the scheme can be based on the worst-case hardness of $\mathsf{SIVP}_{\ell \cdot \widetilde{\mathcal{O}}(n^2)}$.

Let us now describe our ABS scheme. The scheme uses security parameter n, prime modulus $q = \mathcal{O}(\ell \cdot n^2)$, dimension $m = 2n\lceil \log q \rceil$. The infinity norm bound for signatures from Boyen's scheme is set as integer $\beta = \widetilde{\mathcal{O}}(\sqrt{\ell n})$. Let $\kappa =$

$\omega(\log n)$ as the number of protocol repetitions, and let $\mathcal{H}_{\mathsf{FS}} : \{0,1\}^* \to \{1,2,3\}^\kappa$ be a collision-resistant hash function to be modelled as a random oracle.

$\mathsf{Setup}(n, \ell)$. This algorithm performs the following steps:

1. Generate verification key $(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u})$ and signing key $\mathbf{T}_\mathbf{A}$ for Boyen's signature scheme.
2. Generate commitment key $(\mathbf{a}, \mathbf{B}) \xleftarrow{\$} \mathbb{Z}_q^n \times \mathbb{Z}_q^{n \times m}$.
3. Output $\mathsf{pp} = \big(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}, \mathbf{a}, \mathbf{B}\big)$ and $\mathsf{msk} = \mathbf{T}_\mathbf{A}$.

$\mathsf{Keygen}(\mathsf{pp}, \mathsf{msk}, \mathbf{x}) \to \mathsf{sk}_\mathbf{x}$. On input pp, the master secret key $\mathsf{msk} = \mathbf{T}_\mathbf{A}$ and an attribute $\mathbf{x} = (x_1, \ldots, x_\ell) \in \{0,1\}^\ell$, outputs $\mathsf{sk}_\mathbf{x} = \big(\mathbf{x}, \mathbf{s}\big)$, where $\mathbf{s} \in [-\beta, \beta]^{2m}$ is a Boyen signature on the message $(x_1, \ldots, x_\ell)$. Note that the following equation holds:

$$\big[\mathbf{A} \mid \mathbf{A}_0 + \sum_{i=1}^{\ell} x_i \cdot \mathbf{A}_i\big] \cdot \mathbf{s} = \mathbf{u} \bmod q. \tag{9}$$

$\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}_\mathbf{x}, M, C) \to \Sigma$. Let circuit C consist of N gates labelled by operations $\circ_1, \ldots, \circ_N$, and let $g, h : \{1, \ldots, N\} \to \{1, \ldots, \ell + N - 1\}$ be the functions specifying the topology of C. The signer parses $\mathsf{sk}_\mathbf{x}$ as $\big((x_1, \ldots, x_\ell), \mathbf{s}\big)$ and proceeds as follows:

1. For each $j \in [N]$, compute the assignment to each non-input wire in C as: $x_{\ell+j} = x_{g(j)} \circ_j x_{h(j)}$.
2. For each $i \in [\ell + N]$, compute a commitment to the bit x_i as

$$\mathbf{c}_i = \mathbf{a} \cdot x_i + \mathbf{B} \cdot \mathbf{r}_i \bmod q,$$

   where $\mathbf{r}_i \xleftarrow{\$} \{0,1\}^m$.
3. Generate a non-interactive zero-knowledge argument of knowledge $\mathit{\Pi}$ to prove knowledge of $(\,\mathbf{s},\ x_1, \ldots,\ x_\ell,\ x_{\ell+1}, \ldots,\ x_{\ell+N},\ \mathbf{r}_1, \ldots,\ \mathbf{r}_{\ell+N}\,)$, such that the following hold:
   (a) $\big((x_1, \ldots, x_\ell), \mathbf{s}\big)$ is a valid message-signature pair, i.e., equation (9) holds with $\mathbf{s} \in [-\beta, \beta]^{2m}$.
   (b) For each $i \in [\ell + N]$: $\mathbf{c}_i$ is valid a commitment to x_i with randomness $\mathbf{r}_i \in \{0,1\}^m$.
   (c) $\mathbf{x} = (x_1, \ldots, x_\ell)$ satisfies circuit C.

   This is done by running the ZKAoK for the relation $\mathrm{R}_{\mathrm{abs}}$ from Section 4.2, with public input $(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}, \mathbf{a}, \mathbf{B}, \mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N})$ and prover's witness $(\,\mathbf{s},\ x_1, \ldots,\ x_\ell,\ x_{\ell+1}, \ldots,\ x_{\ell+N},\ \mathbf{r}_1, \ldots,\ \mathbf{r}_{\ell+N}\,)$. The protocol is repeated $\kappa = \omega(\log n)$ times to make the soundness error negligibly small, and then made non-interactive via the Fiat-Shamir heuristic as a triple $\mathit{\Pi} = (\{\mathrm{CMT}_i\}_{i=1}^\kappa, \mathrm{CH}, \{\mathrm{RSP}_i\}_{i=1}^\kappa)$, where

$$\mathrm{CH} = (Ch_1, \ldots, Ch_\kappa) = \mathcal{H}_{\mathsf{FS}}\big(\mathsf{pp}, C, M, \{\mathbf{c}_i\}_{i=1}^{\ell+N}, \{\mathrm{CMT}_i\}_{i=1}^\kappa\big) \in \{1,2,3\}^\kappa.$$

4. Output signature

$$\Sigma = \big(\mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N}, \Pi\big). \tag{10}$$

$\mathsf{Verify}(\mathsf{pp}, M, C, \Sigma) \to 1/0$. Given the public parameter pp, message M, circuit C and signature Σ, this algorithm proceeds as follows:

1. Parse Σ as in (10), and parse Π as

$$\Pi = (\{\mathrm{CMT}_i\}_{i=1}^{\kappa}, (Ch_1, \ldots, Ch_\kappa), \{\mathrm{RSP}_i\}_{i=1}^{\kappa}).$$

   Return 0 if $(Ch_1, \ldots, Ch_\kappa) \neq \mathcal{H}_{\mathsf{FS}}\big(\mathsf{pp}, C, M, \{\mathbf{c}_i\}_{i=1}^{\ell+N}, \{\mathrm{CMT}_i\}_{i=1}^{\kappa}\big)$.
2. For each $i = 1$ to κ, run the verification phase of the protocol from Section 4.2 with public input $(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}, \mathbf{a}, \mathbf{B}, \mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N})$ to check the validity of RSP_i with respect to CMT_i and Ch_i. If any of the conditions does not hold, then return 0.
3. Return 1.

4.2 The Underlying Zero-Knowledge Argument System

We now present the zero-knowledge argument system underlying the ABS scheme from Section 4.1. The protocol is an argument of knowledge for the relation $\mathrm{R}_{\mathrm{abs}}$ defined below.

Definition 5. *Define*

$$\mathrm{R}_{\mathrm{abs}} = \Big\{(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}, \mathbf{a}, \mathbf{B}, \mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N}),\ \mathbf{s},\ x_1, \ldots,\ x_{\ell+N},\ \mathbf{r}_1, \ldots, \mathbf{r}_{\ell+N}\Big\}$$

as a relation, where

$$\begin{cases} \mathbf{A},\ \mathbf{A}_0, \ldots,\ \mathbf{A}_\ell \in \mathbb{Z}_q^{n \times m};\ \mathbf{u} \in \mathbb{Z}_q^n \\ \mathbf{a} \in \mathbb{Z}_q^n;\ \mathbf{B} \in \mathbb{Z}_q^{n \times m};\ \mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N} \in \mathbb{Z}_q^n \\ \mathbf{s} \in [-\beta, \beta]^{2m};\ x_1, \ldots, x_{\ell+N} \in \{0,1\};\ \mathbf{r}_1, \ldots,\ \mathbf{r}_{\ell+N} \in \{0,1\}^m \end{cases}$$

satisfy

1. $\big[\mathbf{A} \mid \mathbf{A}_0 + \sum_{i=1}^{\ell} x_i \cdot \mathbf{A}_i\big] \cdot \mathbf{s} = \mathbf{u} \bmod q$.
2. $\big((\mathbf{a}, \mathbf{B}, \mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N}), x_1, \ldots, x_{\ell+N}, \mathbf{r}_1, \ldots, \mathbf{r}_{\ell+N}\big) \in \mathrm{R}_{\mathrm{circuit}}$, *where* $\mathrm{R}_{\mathrm{circuit}}$ *is the relation specified in Definition 4.*

At a high level, the protocol combines the Stern-like protocol for proving knowledge of a message-signature pair for Boyen signature, presented in [45,41], with the our protocol for circuit satisfiability from Section 3.4. The commitments $\mathbf{c}_1, \ldots, \mathbf{c}_\ell$ serve as the bridge between these two protocols. Since the former protocol involves witness $\mathbf{s}$ with constraint $\|\mathbf{s}\|_\infty \leq \beta$, we have to employ specific Stern-like techniques for handling such constraint.

Decompositions. Let $\delta_\beta = \lfloor \log_2 \beta \rfloor + 1$. It was shown in [44] and later formalized in [41] that, for any positive integers m, β, there exists "decomposition matrix" $\mathbf{H}_{m,\beta} \in \mathbb{Z}^{m \times m\delta_\beta}$, where $\delta_\beta = \lfloor \log_2 \beta \rfloor + 1$, such that the following holds:

1. For any vector $\mathbf{z} \in [-\beta, \beta]^m$, there exists an efficiently computable vector $\tilde{\mathbf{z}} \in \{-1, 0, 1\}^{m\delta_\beta}$ such that $\mathbf{H}_{m,\beta} \cdot \tilde{\mathbf{z}} = \mathbf{z}$.
2. Conversely, if $\tilde{\mathbf{z}} \in \{-1, 0, 1\}^{m\delta_\beta}$ and $\mathbf{z} = \mathbf{H}_{m,\beta} \cdot \tilde{\mathbf{z}}$, then $\mathbf{z} \in [-\beta, \beta]^m$.[3]

Let us now write the signature vector $\mathbf{s} \in [-\beta, \beta]^{2m}$ as $\mathbf{s} = (\mathbf{s}_1 \| \mathbf{s}_2)$, where $\mathbf{s}_1, \mathbf{s}_2 \in [-\beta, \beta]^m$. Using the decomposition techniques from [44,41], we can compute $\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2 \in \{-1, 0, 1\}^{m\delta_\beta}$ such that $\{\mathbf{s}_i = \mathbf{H}_{m,\beta} \cdot \tilde{\mathbf{s}}_i\}_{i=1,2}$.

The equation $\left[\mathbf{A} \mid \mathbf{A}_0 + \sum_{i=1}^{\ell} x_i \cdot \mathbf{A}_i\right] \cdot \mathbf{s} = \mathbf{u} \bmod q$ then can be rewritten as:

$$\mathbf{A} \cdot \mathbf{s}_1 + \mathbf{A}_0 \cdot \mathbf{s}_2 + \sum_{i=1}^{\ell} \mathbf{A}_i \cdot (x_i \cdot \mathbf{s}_2) = \mathbf{u} \bmod q$$

$$\iff (\mathbf{A} \cdot \mathbf{H}_{m,\beta}) \cdot \tilde{\mathbf{s}}_1 + (\mathbf{A}_0 \cdot \mathbf{H}_{m,\beta}) \cdot \tilde{\mathbf{s}}_2 + \sum_{i=1}^{\ell} (\mathbf{A}_i \cdot \mathbf{H}_{m,\beta}) \cdot (x_i \cdot \tilde{\mathbf{s}}_2) = \mathbf{u} \bmod q$$

$$\iff \left[\mathbf{A} \cdot \mathbf{H}_{m,\beta} \mid \mathbf{A}_0 \cdot \mathbf{H}_{m,\beta} \mid \mathbf{A}_1 \cdot \mathbf{H}_{m,\beta} \mid \cdots \mid \mathbf{A}_\ell \cdot \mathbf{H}_{m,\beta}\right] \cdot \widehat{\mathbf{s}} = \mathbf{u} \bmod q, \quad (11)$$

where $\widehat{\mathbf{s}} = \left(\tilde{\mathbf{s}}_1 \parallel \tilde{\mathbf{s}}_2 \parallel x_1 \cdot \tilde{\mathbf{s}}_2 \parallel \ldots \parallel x_\ell \cdot \tilde{\mathbf{s}}_2\right) \in \{-1, 0, 1\}^{(\ell+2)m\delta_\beta}$.

Extensions. Next, we apply some extending-then-permuting techniques on the vector $\widehat{\mathbf{s}}$ obtained above. Let $\mathsf{B}^3_{m\delta_\beta}$ be the set of vectors in $\{-1, 0, 1\}^{3m\delta_\beta}$ that have exactly $m\delta_\beta$ coordinates equal to j, for every $j \in \{-1, 0, 1\}$. We append $2m\delta_\beta$ coordinates in $\{-1, 0, 1\}$ (with suitable numbers of -1's, 0's and 1's) to vectors $\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2 \in \{-1, 0, 1\}^{m\delta_\beta}$ in order to obtain vectors $\mathbf{s}_1^*, \mathbf{s}_2^* \in \mathsf{B}^3_{m\delta_\beta}$. Then, we build the vector $\mathbf{s}^* \in \{-1, 0, 1\}^{(2\ell+2)3m\delta_\beta}$ of the form:

$$\mathbf{s}^* = \left(\mathbf{s}_1^* \parallel \mathbf{s}_2^* \parallel \overline{x}_1 \cdot \mathbf{s}_2^* \parallel x_1 \cdot \mathbf{s}_2^* \parallel \ \ldots \parallel \overline{x}_\ell \cdot \mathbf{s}_2^* \parallel x_\ell \cdot \mathbf{s}_2^*\right) \quad (12)$$

As vector $\mathbf{s}^*$ is an extension of $\widehat{\mathbf{s}}$, we can insert suitable zero-columns to the public matrix on the left-hand side of (11) to obtain matrix $\mathbf{A}^* \in \mathbb{Z}_q^{n \times (2\ell+2)3m\delta_\beta}$ such that $\mathbf{A}^* \cdot \mathbf{s}^* = \mathbf{u} \bmod q$.

Permutations. Now, for any ℓ bits $x_1, \ldots, x_\ell$, we define $\mathsf{Mix}(x_1, \ldots, x_\ell)$ as the set of all vectors in $\{-1, 0, 1\}^{(2\ell+2)3m\delta_\beta}$, that have the form (12), for certain vectors $\mathbf{s}_1^*, \mathbf{s}_2^* \in \mathsf{B}^3_{m\delta_\beta}$. Obviously, $\mathbf{s}^*$ belongs to this set. Next, for any permutations $\tau_1, \tau_2 \in \mathcal{S}_{3m\delta_\beta}$, any bits $d_1, \ldots, d_\ell$, we define the permutation $P_{\tau_1,\tau_2,d_1,\ldots,d_\ell}$, that when acting on vector $\mathbf{z} = \left(\mathbf{z}_{-1} \parallel \mathbf{z}_0 \parallel \mathbf{z}_1^0 \parallel \mathbf{z}_1^1 \parallel \ldots \parallel \mathbf{z}_\ell^0 \parallel \mathbf{z}_\ell^1\right) \in \mathbb{Z}^{(2\ell+2)3m\delta_\beta}$ consisting of $2\ell + 2$ blocks of size $3m\delta_\beta$, it transforms the blocks as follows:

$$P_{\tau_1,\tau_2,d_1,\ldots,d_\ell}(\mathbf{z}) = \left(\tau_1(\mathbf{z}_{-1}) \parallel \tau_2(\mathbf{z}_0) \parallel \tau_2(\mathbf{z}_1^{d_1}) \parallel \tau_2(\mathbf{z}_1^{\overline{d}_1}) \parallel \ldots \parallel \tau_2(\mathbf{z}_\ell^{d_\ell}) \parallel \tau_2(\mathbf{z}_\ell^{\overline{d}_\ell})\right)$$

[3] This fact is crucial in the knowledge extraction process, as it ensures that the extracted vector has the same infinity norm bound as the real witness.

It can be checked that the following equivalence holds. For any $\mathbf{z} \in \mathbb{Z}^{(2\ell+2)3m\delta_\beta}$, any bits $d_1, \ldots, d_\ell$, any permutations $\tau_1, \tau_2 \in \mathcal{S}_{3m\delta_\beta}$,

$$\mathbf{z} \in \mathsf{Mix}(x_1, \ldots, x_\ell) \iff P_{\tau_1,\tau_2,d_1,\ldots,d_\ell}(\mathbf{z}) \in \mathsf{Mix}(x_1 \oplus d_1, \ldots, x_\ell \oplus d_\ell). \quad (13)$$

Let us summarize the steps we have done so far: We have transformed the equation $\left[\mathbf{A} \mid \mathbf{A}_0 + \sum_{i=1}^{\ell} x_i \cdot \mathbf{A}_i\right] \cdot \mathbf{s} = \mathbf{u} \bmod q$ with $\mathbf{s} \in [-\beta, \beta]^{2m}$ into the equivalent equation $\mathbf{A}^* \cdot \mathbf{s}^* = \mathbf{u} \bmod q$ with $\mathbf{s}^* \in \mathsf{Mix}(x_1, \ldots, x_\ell)$, and the set $\mathsf{Mix}(x_1, \ldots, x_\ell)$ is supported by equivalence (13). Next, we will combine this "signature layer" with the "circuit layer".

Recall that in the protocol for relation $\mathrm{R}_{\mathrm{circuit}}$ from Section 3.4, we transformed all the involved equations into one equation of the form $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$, where $\mathbf{w}$ belongs to the set VALID defined therein. Therefore, all the equations considered by the relation $\mathrm{R}_{\mathrm{abs}}$ in Definition 5 can be unified into equation $\mathbf{M}_{\mathrm{abs}} \cdot \mathbf{w}_{\mathrm{abs}} = \mathbf{v}_{\mathrm{abs}} \bmod q$, where

$$\mathbf{M}_{\mathrm{abs}} = \begin{bmatrix} \mathbf{A}^* & \\ & \mathbf{M} \end{bmatrix} \in \mathbb{Z}_q^{K_{\mathrm{abs}} \times D_{\mathrm{abs}}}, \quad \mathbf{v}_{\mathrm{abs}} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \in \mathbb{Z}_q^{K_{\mathrm{abs}}}, \quad \mathbf{w}_{\mathrm{abs}} = \begin{pmatrix} \mathbf{s}^* \\ \mathbf{w} \end{pmatrix},$$

for $K_{\mathrm{abs}} = (\ell + 3N + 2)n$ and $D_{\mathrm{abs}} = (2\ell+2)3m\delta_\beta + 2\ell + 8N + 2m(\ell + 3N)$.

Now, we define $\mathsf{VALID}_{\mathrm{abs}}$ as the set of all vectors in $\{-1, 0, 1\}^{D_{\mathrm{abs}}}$ having the form $(\mathbf{s}^* \| \mathbf{w})$, such that the following conditions hold.

1. $\mathbf{w} = \big(\mathbf{y}_1 \| \ldots \| \mathbf{y}_\ell \| \mathbf{z}_{1,1} \| \mathbf{z}_{1,2} \| \mathbf{t}_1 \| \ldots \| \mathbf{z}_{N,1} \| \mathbf{z}_{N,2} \| \mathbf{t}_N \| \mathbf{r}^* \big)$, as in (8).
2. There exist bits $x_1, \ldots, x_\ell$, $e_{1,1}, e_{1,2}, \ldots, e_{N,1}, e_{N,2}$ such that
   - $\mathbf{s}^* \in \mathsf{Mix}(x_1, \ldots, x_\ell)$.
   - For all $i \in [\ell]$: $\mathbf{y}_i = \mathsf{enc}(x_i)$.
   - For all $j \in [N]$: $\mathbf{z}_{j,1} = \mathsf{enc}(e_{j,1})$, $\mathbf{z}_{j,2} = \mathsf{enc}(e_{j,2})$ and $\mathbf{t}_j = \mathsf{ext}_{\circ_j}(e_{j,1}, e_{j,2})$.
3. $\mathbf{r}^* \in \mathsf{B}^2_{m(\ell+3N)}$.

By construction, we have $\mathbf{w}_{\mathrm{abs}} \in \mathsf{VALID}_{\mathrm{abs}}$. Next, we define the set $\mathcal{S}_{\mathrm{abs}}$ and permutations $\{\Gamma_{\phi_{\mathrm{abs}}} : \phi_{\mathrm{abs}} \in \mathcal{S}_{\mathrm{abs}}\}$ so that the conditions in (2) are satisfied. To this end, we let $\mathcal{S}_{\mathrm{abs}} = (\mathcal{S}_{3m\delta_\beta})^2 \times \{0,1\}^{\ell+2N} \times \mathcal{S}_{2m(\ell+3N)}$, and associate each element

$$\phi_{\mathrm{abs}} = \big((\tau_1, \tau_2),\ (b_1, \ldots,\ b_\ell,\ p_{1,1},\ p_{1,2}, \ldots,\ p_{N,1},\ p_{N,2}),\ \pi \big) \in \mathcal{S}_{\mathrm{abs}}$$

with the permutation $\Gamma_{\phi_{\mathrm{abs}}}$ that transforms vector $\mathbf{t} = (\mathbf{s}^* \| \mathbf{w}) \in \mathbb{Z}^{D_{\mathrm{abs}}}$, to vector

$$\Gamma_{\phi_{\mathrm{abs}}}(\mathbf{t}) = \big(P_{\tau_1,\tau_2,d_1,\ldots,d_\ell}(\mathbf{s}^*) \parallel \Gamma_\phi(\mathbf{w}) \big),$$

where $\phi = \big((b_1, \ldots,\ b_\ell,\ p_{1,1},\ p_{1,2}, \ldots,\ p_{N,1},\ p_{N,2}),\ \pi \big)$ and Γ_ϕ is as defined in Section 3.4. It can be checked that $\mathsf{VALID}_{\mathrm{abs}}$, $\mathcal{S}_{\mathrm{abs}}$ and $\Gamma_{\phi_{\mathrm{abs}}}$ satisfy the conditions specified in (2), based on the equivalence (13) and the discussions from Section 3.4. In other words, we have reduced the relation $\mathrm{R}_{\mathrm{abs}}$ to an instance of the abstract relation from Section 2.2.

The interactive protocol. Given all the above preparations, our interactive protocol works as follows.

- The public input consists of matrix $\mathbf{M}_{\text{abs}}$ and vector $\mathbf{v}_{\text{abs}}$, which are built from $(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}, \mathbf{a}, \mathbf{B}, \mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N})$, as discussed above.
- The prover's witness is vector $\mathbf{w} \in \mathsf{VALID}_{\text{abs}}$, which is obtained from the original witnesses $\mathbf{s}, x_1, \ldots, x_{\ell+N}, \mathbf{r}_1, \ldots, \mathbf{r}_{\ell+N}$, as described above.

Both parties then run the protocol of Figure 1. The protocol uses the KTX string commitment scheme COM, which is statistically hiding and computationally binding if the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ problem is hard. We therefore obtain the following result.

Lemma 3. *Assume that* $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ *is hard. Then the protocol described above is a statistical* ZKAoK *for the relation* R_{abs} *with perfect completeness, soundness error* $2/3$ *and communication cost* $\mathcal{O}(D_{\text{abs}} \log q) = \widetilde{\mathcal{O}}(n(\ell + N))$.

Proof. The perfect completeness, soundness error $2/3$ and communication cost of the protocol directly follow from those of the abstract protocol from Section 2.2. Note that we work with dimension $D_{\text{abs}} = (2\ell+2)3m\delta_\beta + 2\ell + 8N + 2m(\ell+3N)$, and thus, we have $\mathcal{O}(D_{\text{abs}} \log q) = \widetilde{\mathcal{O}}(n(\ell + N))$ for the parameter setting of Section 4.1.

For simulation, we simply run the simulator of Theorem 1. For knowledge extraction, we run the knowledge extractor of Theorem 1 to obtain a vector $\mathbf{w}'_{\text{abs}} = ((\mathbf{s}^*)' \| \mathbf{w}') \in \mathsf{VALID}_{\text{abs}}$ such that $\mathbf{M}_{\text{abs}} \cdot \mathbf{w}'_{\text{abs}} = \mathbf{v}_{\text{abs}} \bmod q$.

We then have $\mathbf{w}' \in \mathsf{VALID}$, where the set VALID is as defined in Section 3.4. If we proceed as in the proof of Theorem 2, then we can extract from $\mathbf{w}'$ witness $(x'_1, \ldots, x'_\ell, x'_{\ell+1}, \ldots, x'_{\ell+N}, \mathbf{r}'_1, \ldots, \mathbf{r}'_{\ell+N})$ satisfying the relation $\mathrm{R}_{\text{circuit}}$. In particular, we have $C(x'_1, \ldots, x'_\ell) = 1$.

Meanwhile, we also have $(\mathbf{s}^*)' \in \mathsf{Mix}(x'_1, \ldots, x'_\ell)$. By "backtracking" the transformations we have done previously for the "signature layer", we can obtain from $(\mathbf{s}^*)'$ vector $\mathbf{s}' \in [-\beta, \beta]^{2m}$ such that $\left[\mathbf{A} \mid \mathbf{A}_0 + \sum_{i=1}^{\ell} x'_i \cdot \mathbf{A}_i\right] \cdot \mathbf{s}' = \mathbf{u} \bmod q$.

Putting pieces together, $(\mathbf{s}', x'_1, \ldots, x'_\ell, x'_{\ell+1}, \ldots, x'_{\ell+N}, \mathbf{r}'_1, \ldots, \mathbf{r}'_{\ell+N})$ is a satisfying witness for the relation R_{abs}. This concludes the proof. □

4.3 Analysis of the Attribute-Based Signature Scheme

We summarize the properties of the proposed ABS scheme in the following theorem, whose proof can be found in Appendix C.

Theorem 3. *The* ABS *scheme described in Section 4.1 is correct, has public parameter size* $\widetilde{\mathcal{O}}(\ell \cdot n^2)$ *and produces signatures of bit-size* $\widetilde{\mathcal{O}}(n \cdot (\ell + N))$.

In the random oracle model, the scheme is statistically private, and is adaptively unforgeable assuming the hardness of the $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ *problem and the security of Boyen's signature.*

References

1. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th Annual ACM Symposium on Theory of Computing*, pages 99–108. ACM Press, May 1996.
2. Rachid El Bansarkhani and Ali El Kaafarani. Post-quantum attribute-based signatures from lattice assumptions. *IACR Cryptology ePrint Archive*, 2016:823, 2016.
3. Carsten Baum, Ivan Damgård, Kasper Green Larsen, and Michael Nielsen. How to prove knowledge of small secrets. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 478–498. Springer, Heidelberg, August 2016.
4. Carsten Baum, Ivan Damgård, Sabine Oechsner, and Chris Peikert. Efficient commitments and zero-knowledge protocols from ring-SIS with applications to lattice-based threshold cryptosystems. Cryptology ePrint Archive, Report 2016/997, 2016. `http://eprint.iacr.org/2016/997`.
5. Mihir Bellare and Georg Fuchsbauer. Policy-based signatures. In Hugo Krawczyk, editor, *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 520–537. Springer, Heidelberg, March 2014.
6. Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 551–572. Springer, Heidelberg, December 2014.
7. Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl, editors, *ESORICS 2015: 20th European Symposium on Research in Computer Security, Part I*, volume 9326 of *Lecture Notes in Computer Science*, pages 305–325. Springer, Heidelberg, September 2015.
8. Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In Shafi Goldwasser, editor, *ITCS 2012: 3rd Innovations in Theoretical Computer Science*, pages 326–349. Association for Computing Machinery, January 2012.
9. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 327–357. Springer, Heidelberg, May 2016.
10. Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 16: 23rd Conference on Computer and Communications Security*, pages 1006–1018. ACM Press, October 2016.
11. Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 499–517. Springer, Heidelberg, May 2010.

12. Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 501–519. Springer, Heidelberg, March 2014.
13. Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.
14. Ernest F. Brickell, David Pointcheval, Serge Vaudenay, and Moti Yung. Design validations for discrete logarithm based signature schemes. In Hideki Imai and Yuliang Zheng, editors, *PKC 2000: 3rd International Workshop on Theory and Practice in Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 276–292. Springer, Heidelberg, January 2000.
15. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, Heidelberg, May 2010.
16. David Chaum. Showing credentials without identification: Signatures transferred between unconditionally unlinkable pseudonyms. In Franz Pichler, editor, *Advances in Cryptology – EUROCRYPT'85*, volume 219 of *Lecture Notes in Computer Science*, pages 241–244. Springer, Heidelberg, April 1986.
17. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20. Springer, Heidelberg, December 2011.
18. Ronald Cramer and Ivan Damgård. Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free? In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 424–441. Springer, Heidelberg, August 1998.
19. Ronald Cramer, Ivan Damgård, Chaoping Xing, and Chen Yuan. Amortized complexity of zero-knowledge proofs revisited: Achieving linear soundness slack. In *EUROCRYPT 2017*, volume 10210 of *Lecture Notes in Computer Science*, pages 479–500. Springer, 2017.
20. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, Heidelberg, August 1987.
21. Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam D. Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology*, 28(4):820–843, October 2015.
22. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206. ACM Press, May 2008.
23. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 99–108. ACM Press, June 2011.
24. Oded Goldreich and Shafi Goldwasser. On the limits of non-approximability of lattice problems. In *30th Annual ACM Symposium on Theory of Computing*, pages 1–9. ACM Press, May 1998.

25. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 171–185. Springer, Heidelberg, August 1987.
26. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th Annual ACM Symposium on Theory of Computing*, pages 291–304. ACM Press, May 1985.
27. Jens Groth. Evaluating security of voting schemes in the universal composability framework. In Markus Jakobsson, Moti Yung, and Jianying Zhou, editors, *ACNS 04: 2nd International Conference on Applied Cryptography and Network Security*, volume 3089 of *Lecture Notes in Computer Science*, pages 46–60. Springer, Heidelberg, June 2004.
28. Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 192–208. Springer, Heidelberg, August 2009.
29. Jens Groth. Short non-interactive zero-knowledge proofs. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 341–358. Springer, Heidelberg, December 2010.
30. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 321–340. Springer, Heidelberg, December 2010.
31. Jens Groth. Efficient zero-knowledge arguments from two-tiered homomorphic commitments. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 431–448. Springer, Heidelberg, December 2011.
32. Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3):11:1–11:35, 2012.
33. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, Heidelberg, April 2008.
34. Venkatesan Guruswami, Daniele Micciancio, and Oded Regev. The complexity of the covering radius problem. *Computational Complexity*, 14(2):90–121, 2005.
35. Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 663–680. Springer, Heidelberg, December 2012.
36. Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In Josef Pieprzyk, editor, *Advances in Cryptology – ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389. Springer, Heidelberg, December 2008.
37. Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Group encryption. In Kaoru Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 181–199. Springer, Heidelberg, December 2007.

38. Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th Annual ACM Symposium on Theory of Computing*, pages 723–732. ACM Press, May 1992.
39. Thijs Laarhoven. Sieving for shortest vectors in lattices using angular locality-sensitive hashing. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 3–22. Springer, Heidelberg, August 2015.
40. Adeline Langlois, San Ling, Khoa Nguyen, and Huaxiong Wang. Lattice-based group signature scheme with verifier-local revocation. In Hugo Krawczyk, editor, *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 345–361. Springer, Heidelberg, March 2014.
41. Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 373–403. Springer, Heidelberg, December 2016.
42. Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 101–131. Springer, Heidelberg, December 2016.
43. Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 1–31. Springer, Heidelberg, May 2016.
44. San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013: 16th International Conference on Theory and Practice of Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 107–124. Springer, Heidelberg, February / March 2013.
45. San Ling, Khoa Nguyen, and Huaxiong Wang. Group signatures from lattices: Simpler, tighter, shorter, ring-based. In Jonathan Katz, editor, *PKC 2015: 18th International Conference on Theory and Practice of Public Key Cryptography*, volume 9020 of *Lecture Notes in Computer Science*, pages 427–449. Springer, Heidelberg, March / April 2015.
46. Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 169–189. Springer, Heidelberg, March 2012.
47. Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In Ronald Cramer, editor, *PKC 2008: 11th International Workshop on Theory and Practice in Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 162–179. Springer, Heidelberg, March 2008.
48. Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, Heidelberg, April 2012.

49. Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP 2006*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2006.
50. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, Heidelberg, May 2010.
51. Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. *IACR Cryptology ePrint Archive*, 2008:328, 2008.
52. Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 376–392. Springer, Heidelberg, February 2011.
53. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, Heidelberg, April 2012.
54. Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with small parameters. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 21–39. Springer, Heidelberg, August 2013.
55. Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 282–298. Springer, Heidelberg, August 2003.
56. Tatsuaki Okamoto and Katsuyuki Takashima. Efficient attribute-based signatures for non-monotone predicates in the standard model. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011: 14th International Conference on Theory and Practice of Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 35–52. Springer, Heidelberg, March 2011.
57. Chris Peikert and Vinod Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 536–553. Springer, Heidelberg, August 2008.
58. Rafaël Del Pino and Vadim Lyubashevsky. Amortization with fewer equations for proving knowledge of small secrets. *IACR Cryptology ePrint Archive*, 2017:280, 2017.
59. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93. ACM Press, May 2005.
60. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, Heidelberg, December 2001.
61. Yusuke Sakai, Nuttapong Attrapadung, and Goichiro Hanaoka. Attribute-based signatures for circuits from bilinear map. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 9614 of *Lecture Notes in Computer Science*, pages 283–300. Springer, Heidelberg, March 2016.

62. Jae Hong Seo. Round-efficient sub-linear zero-knowledge arguments for linear algebra. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011: 14th International Conference on Theory and Practice of Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 387–402. Springer, Heidelberg, March 2011.
63. Jacques Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.
64. Fei Tang, Hongda Li, and Bei Liang. Attribute-based signatures for circuits from multilinear maps. In Sherman S. M. Chow, Jan Camenisch, Lucas Chi Kwong Hui, and Siu-Ming Yiu, editors, *ISC 2014: 17th International Conference on Information Security*, volume 8783 of *Lecture Notes in Computer Science*, pages 54–71. Springer, Heidelberg, October 2014.
65. Qingbin Wang and Shaozhen Chen. Attribute-based signature for threshold predicates from lattices. *Security and Communication Networks*, 8(5):811–821, 2015.
66. Qingbin Wang, Shaozhen Chen, and Aijun Ge. A new lattice-based threshold attribute-based signature scheme. In *Information Security Practice and Experience, ISPEC 2015*, volume 9065 of *Lecture Notes in Computer Science*, pages 406–420. Springer, 2015.
67. Xiang Xie, Rui Xue, and Minqian Wang. Zero knowledge proofs from ring-LWE. In Michel Abdalla, Cristina Nita-Rotaru, and Ricardo Dahab, editors, *CANS 13: 12th International Conference on Cryptology and Network Security*, volume 8257 of *Lecture Notes in Computer Science*, pages 57–73. Springer, Heidelberg, November 2013.
68. Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 160–164. IEEE Computer Society Press, November 1982.

A Proof of Theorem 1

We provide the proof of Theorem 1, as appeared in [41]. We first restate the theorem.

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

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

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

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

Zero-Knowledge Property. We construct a PPT simulator SIM interacting with a (possibly dishonest) verifier $\widehat{\mathcal{V}}$, such that, given only the public input, SIM

outputs with probability negligibly close to 2/3 a simulated transcript that is statistically close to the one produced by the honest prover in the real interaction.

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

Case $\overline{Ch} = 1$: Using basic linear algebra over $\mathbb{Z}_q$, SIM computes a vector $\mathbf{w}' \in \mathbb{Z}_q^D$ such that $\mathbf{M} \cdot \mathbf{w}' = \mathbf{v} \bmod q$. Next, it samples $\mathbf{r}_w \leftarrow U(\mathbb{Z}_q^D)$, $\phi \leftarrow U(\mathcal{S})$, and randomness ρ_1, ρ_2, ρ_3 for COM. Then, it sends the commitment $\mathrm{CMT} = \big(C'_1, C'_2, C'_3\big)$ to $\widehat{\mathcal{V}}$, where

$$C'_1 = \mathsf{COM}(\phi, \mathbf{M} \cdot \mathbf{r}_w; \rho_1),$$
$$C'_2 = \mathsf{COM}(\Gamma_\phi(\mathbf{r}_w); \rho_2), \quad C'_3 = \mathsf{COM}(\Gamma_\phi(\mathbf{w}' + \mathbf{r}_w); \rho_3).$$

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

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

Case $\overline{Ch} = 2$: SIM samples $\mathbf{w}' \leftarrow U(\mathsf{VALID})$, $\mathbf{r}_w \leftarrow U(\mathbb{Z}_q^D)$, $\phi \leftarrow U(\mathcal{S})$, and randomness ρ_1, ρ_2, ρ_3 for COM. Then it sends the commitment $\mathrm{CMT} = \big(C'_1, C'_2, C'_3\big)$ to $\widehat{\mathcal{V}}$, where

$$C'_1 = \mathsf{COM}(\phi, \mathbf{M} \cdot \mathbf{r}_w; \rho_1),$$
$$C'_2 = \mathsf{COM}(\Gamma_\phi(\mathbf{r}_w); \rho_2), \quad C'_3 = \mathsf{COM}(\Gamma_\phi(\mathbf{w}' + \mathbf{r}_w); \rho_3).$$

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

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

Case $\overline{Ch} = 3$: SIM samples $\mathbf{w}' \leftarrow U(\mathsf{VALID})$, $\mathbf{r}_w \leftarrow U(\mathbb{Z}_q^D)$, $\phi \leftarrow U(\mathcal{S})$, and randomness ρ_1, ρ_2, ρ_3 for COM. Then it sends the commitment $\mathrm{CMT} = \big(C'_1, C'_2, C'_3\big)$ to $\widehat{\mathcal{V}}$, where $C'_2 = \mathsf{COM}(\Gamma_\phi(\mathbf{r}_w); \rho_2)$, $C'_3 = \mathsf{COM}(\Gamma_\phi(\mathbf{w}' + \mathbf{r}_w); \rho_3)$ as in the previous two cases, while

$$C'_1 = \mathsf{COM}(\phi, \mathbf{M} \cdot (\mathbf{w}' + \mathbf{r}_w) - \mathbf{v}; \rho_1).$$

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

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

We observe that, in every case we have considered above, since COM is statistically hiding, the distribution of the commitment CMT and the distribution of

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

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

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

Since COM is computationally binding, we can deduce that

$$\begin{cases} \mathbf{t}_w \in \mathsf{VALID};\ \phi_2 = \phi_3;\ \mathbf{t}_r = \Gamma_{\phi_3}(\mathbf{w}_3);\ \mathbf{t}_w + \mathbf{t}_r = \Gamma_{\phi_2}(\mathbf{w}_2) \bmod q; \\ \mathbf{M} \cdot \mathbf{w}_2 - \mathbf{v} = \mathbf{M} \cdot \mathbf{w}_3 \bmod q. \end{cases} \tag{14}$$

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

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

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

B Attribute-Based Signatures

Here, we recall the definitions and security notions for ABS, as presented in previous works [51,52,61]. An ABS scheme consists of a tuple of four polynomial-time algorithms $(\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$, defined as follows:

$\mathsf{Setup}(n, \ell) \to (\mathsf{pp}, \mathsf{msk})$. The setup algorithm takes as input the security parameter n and the length ℓ of attributes, and outputs the public parameter pp and the master secret key msk.

$\mathsf{Keygen}(\mathsf{pp}, \mathsf{msk}, \mathbf{x}) \to \mathsf{sk}_{\mathbf{x}}$. The signing key generation algorithm takes as input the public parameter pp, the master secret key msk and attribute $\mathbf{x}$, and outputs the signing key $\mathsf{sk}_{\mathbf{x}}$ for $\mathbf{x}$.

$\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}_{\mathbf{x}}, M, C) \to \Sigma$. The signing algorithm takes as input the public parameter pp, the signing key $\mathsf{sk}_{\mathbf{x}}$, message M and circuit C, and outputs signature Σ.

$\mathsf{Verify}(\mathsf{pp}, M, C, \Sigma) \to 1/0$. The verification algorithm takes as input the public parameter pp, message M, circuit C and signature Σ, and outputs 1 or 0 indicating the validity or invalidity of the signature, respectively.

We next describe the following requirements for attribute-based signatures: correctness, statistical privacy and unforgeability.

Definition 6 (Correctness). *An* ABS *scheme* $(\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ *is called correct if for all* $n, \ell \in \mathbb{N}$; $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(n, \ell)$; $\mathbf{x} \in \{0,1\}^{\ell}$; $\mathsf{sk}_{\mathbf{x}} \leftarrow \mathsf{Keygen}(\mathsf{pp}, \mathsf{msk}, \mathbf{x})$; $M \in \{0,1\}^*$ *and* C *such that* $C(\mathbf{x}) = 1$, *it holds that* $\mathsf{Verify}\big(\mathsf{pp}, M, C, \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}_{\mathbf{x}}, M, C)\big) = 1$.

The standard security notions for an ABS scheme are privacy and unforgeability. Statistical privacy requires the signature to not leak any information on the signer's identity and attribute beyond the fact that the attribute satisfies the predicate, even if the adversary is computationally unbounded and is given all the signing keys. Unforgeability demands that, it is infeasible for any collusion of signers to produce a new signature with respect to a predicate not satisfied by any attribute in the collusion, even if signatures on maliciously chosen messages are given.

Definition 7 (Privacy). *An* ABS *scheme* $(\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ *is statistically private if the success probability of any adversary in the following experiment is at most* $1/2 + \mathsf{negl}(n)$.

1. *The experiment generates* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(n, \ell)$, *and sends* $(\mathsf{pp}, \mathsf{msk})$ *to the adversary, who returns* $(M^\star, C^\star, \mathbf{x}_0^\star, \mathbf{x}_1^\star)$ *such that* $C^\star(\mathbf{x}_0^\star) = C^\star(\mathbf{x}_1^\star) = 1$.
2. *The experiment picks* $b \xleftarrow{\$} \{0,1\}$, *obtains* $\mathsf{sk}_b \leftarrow \mathsf{Keygen}(\mathsf{pp}, \mathsf{msk}, \mathbf{x}_b^\star)$, *and returns* $\Sigma \leftarrow \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}_b, M^\star, C^\star)$.
3. *The adversary outputs* b', *and succeeds if* $b' = b$.

Definition 8 (Unforgeability). *An* ABS *scheme* $(\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ *is adaptively unforgeable if the success probability of any PPT adversary in the following experiment is at most* $\mathsf{negl}(n)$.

1. *The experiment generates* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(n, \ell)$, *and sends* pp *to the adversary.*
2. *The adversary is allowed to access the key reveal oracle and the signing oracle. For the former, given a query* $\mathbf{x}$, *returns* $\mathsf{sk}_{\mathbf{x}} \leftarrow \mathsf{Keygen}(\mathsf{pp}, \mathsf{msk}, \mathbf{x})$. *For the latter, given a query* (M, C), *returns* $\Sigma \leftarrow \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}_{\mathbf{x}}, M, C)$ *with arbitrary* $\mathsf{sk} \leftarrow \mathsf{Keygen}(\mathsf{pp}, \mathsf{msk}, \mathbf{x})$ *such that* $C(\mathbf{x}) = 1$.
3. *The adversary outputs a forgery* $(M^\star, C^\star, \Sigma^\star)$.
4. *The adversary succeeds if the following three conditions hold:*
   (a) $\mathsf{Verify}\big(\mathsf{pp}, M^\star, C^\star, \Sigma^\star\big) = 1$.
   (b) *The adversary did not query* $\mathbf{x}$ *such that* $C^\star(\mathbf{x}) = 1$.
   (c) *The adversary did not query* $(M^\star, C^\star)$ *to the signing oracle.*

C Proof of Theorem 3

Efficiency and correctness. The correctness of the proposed ABS scheme directly follows from the perfect completeness of the underlying argument system (see Lemma 3).

Since $\mathsf{pp} = \big(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}, \mathbf{a}, \mathbf{B}\big)$, it has size $(\ell+3)nm\log q + 2n\log q = \widetilde{\mathcal{O}}(\ell \cdot n^2)$ bits. The ABS signature Σ is dominated by the $\mathsf{NIZKAoK}$ Π. The size of the latter is roughly $\kappa = \omega(\log n)$ times the communication cost of the protocol of Section 4.2. Thus, Σ has size $\widetilde{\mathcal{O}}(n \cdot (\ell + N))$ bits.

Privacy. Next, we prove that the scheme satisfies the statistical privacy requirement.

Lemma 4. *In the random oracle model, the* ABS *scheme described in Section 4.1 is statistically private in the sense of Definition 7.*

Proof. The proof relies on the statistical zero-knowledge property of the argument system from Section 4.2 and the fact that the distribution of commitments $(\mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N})$ is statistically uniform over $(\mathbb{Z}_q^n)^{\ell+N}$. Specifically, we consider the following two games.

Game $G_0^{(b)}$. This is the experiment of Definition 7. The adversary is given $\mathsf{pp} = \big(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}, \mathbf{a}, \mathbf{B}\big)$ and $\mathsf{msk} = \mathbf{T}_{\mathbf{A}}$. It then returns $(M^\star, C^\star, \mathbf{x}_0^\star, \mathbf{x}_1^\star)$ such that $C^\star(\mathbf{x}_0^\star) = C^\star(\mathbf{x}_1^\star) = 1$. The experiment picks $b \xleftarrow{\$} \{0,1\}$, obtains $\mathsf{sk}_b \leftarrow \mathsf{Keygen}(\mathsf{pp}, \mathsf{msk}, \mathbf{x}_b^\star)$, and returns

$$\Sigma = \big(\mathbf{c}_1, \ldots, \mathbf{c}_{\ell+N}, \Pi\big) \leftarrow \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}_b, M^\star, C^\star).$$

The adversary outputs b' and succeeds if $b' = b$.

Game G_1. In this game, instead of faithfully generating Σ, we simulate it as follows. First, we sample $\mathbf{c}_1^\star, \ldots, \mathbf{c}_{\ell+N}^\star \xleftarrow{\$} \mathbb{Z}_q^n$. Then, for each $i \in [\kappa]$, we run the simulator of the underlying zero-knowledge argument system with input $\big(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}, \mathbf{a}, \mathbf{B}, \mathbf{c}_1^\star, \ldots, \mathbf{c}_{\ell+N}^\star\big)$, and program the random oracle $\mathcal{H}_{\mathsf{FS}}$ accordingly. Let $\Pi^\star$ be the simulated non-interactive argument, and let $\Sigma^\star = (\mathbf{c}_1^\star, \ldots, \mathbf{c}_{\ell+N}^\star, \Pi^\star)$. Observe that, $\Sigma^\star$ is accepted by the verification algorithm, and its distribution is statistically close to that of Σ in **Game** $G_0^{(b)}$. Thus, the two games are statistically close.

As **Game** G_1 does not depend on the bit b, the adversary succeeds in this game with probability $1/2$. It then follows that its success probability in **Game** $G_0^{(b)}$ is at most $1/2 + \mathsf{negl}(n)$. This concludes the proof. □

Unforgeability. Finally, we prove that the proposed ABS scheme satisfies the unforgeability requirement.

Lemma 5. *Assume that:*

1. *The* $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ *is hard (which implies the computational binding of the bit commitment scheme and the string commitment scheme from [36]).*

2. Boyen's signature scheme [11,53] is unforgeable.

Then, in the random oracle model, the ABS *scheme described in Section 4.1 is adaptively unforgeable in the sense of Definition 8.*

Proof. Let $\mathcal{A}$ be a PPT adversary with non-negligible success probability ϵ in the unforgeability experiment of Definition 8. Assuming the computational binding of the bit commitment scheme and the string commitment scheme being used, we construct a PPT forger $\mathcal{F}$ for Boyen's signature scheme, whose success probability is non-negligible.

The forger $\mathcal{F}$ is given the verification key $(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u})$ for Boyen's signature scheme. It generates commitment key $(\mathbf{a}, \mathbf{B}) \xleftarrow{\$} \mathbb{Z}_q^n \times \mathbb{Z}_q^{n \times m}$. Next, $\mathcal{F}$ starts interacting with the adversary $\mathcal{A}$ by sending $\mathsf{pp} = \big(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u}, \mathbf{a}, \mathbf{B}\big)$, the distribution of which is exactly as in the real scheme. Then, $\mathcal{F}$ handles the queries from $\mathcal{A}$ as follows:

- Queries to the random oracle $\mathcal{H}_{\mathsf{FS}}(.)$ are answered by outputting uniformly random elements of the range $\{1, 2, 3\}^\kappa$. Of course, the same answer is returned in case the same query occurs more than once. Suppose that $\mathcal{A}$ makes $Q_{\mathcal{H}}$ random oracle queries.
- Queries to the key reveal oracle are handled as follows. For a query with attribute $\mathbf{x} \in \{0,1\}^\ell$, algorithm $\mathcal{F}$ first queries its own signing oracle for Boyen's signature on message $\mathbf{x}$, then it forwards the answer to $\mathcal{A}$.
- Queries for ABS are answered with simulated signatures. For any query of the form (M, C), algorithm $\mathcal{F}$ returns with a simulated Σ', which is computed as in **Game** G_1 from Lemma 4 and which is indistinguishable from a legitimate ABS.

Eventually, $\mathcal{A}$ outputs a forgery $(M^\star, C^\star, \Sigma^\star)$ satisfying the requirements of Definition 8. If we parse $\Sigma^\star$ as $\big(\mathbf{c}_1^\star, \ldots, \mathbf{c}_{\ell+N}^\star, \Pi^\star\big)$ and parse the argument $\Pi^\star$ as $(\{\mathrm{CMT}_i^\star\}_{i=1}^\kappa, \mathrm{CH}^\star, \{\mathrm{RSP}_i^\star\}_{i=1}^\kappa)$, then with overwhelming probability, $\mathcal{A}$ must have queried the random oracle on input $\big(\mathsf{pp}, C^\star, M^\star, \{\mathbf{c}_i^\star\}_{i=1}^{\ell+N}, \{\mathrm{CMT}_i^\star\}_{i=1}^\kappa\big)$. Otherwise, the probability that $\mathrm{CH}^\star = \mathcal{H}_{\mathsf{FS}}\big(\mathsf{pp}, C^\star, M^\star, \{\mathbf{c}_i^\star\}_{i=1}^{\ell+N}, \{\mathrm{CMT}_i^\star\}_{i=1}^\kappa\big)$ would be smaller than $3^{-\kappa}$, making $\mathcal{A}$'s success probability negligible. Thus, with probability at least $\epsilon' := \epsilon - 3^{-\kappa}$, the tuple $\big(\mathsf{pp}, C^\star, M^\star, \{\mathbf{c}_i^\star\}_{i=1}^{\ell+N}, \{\mathrm{CMT}_i^\star\}_{i=1}^\kappa\big)$ has been the input of a random oracle query and we call $t^\star \in \{1, \ldots, Q_{\mathcal{H}}\}$ the index of this specific query.

Then, algorithm $\mathcal{F}$ runs up to $32 \cdot Q_{\mathcal{H}}/(\epsilon - 3^{-\kappa})$ extra executions of the adversary $\mathcal{A}$ with the *same* random tape and input as in the first execution. In each new run, all queries receive exactly the same answers as in the first run until the $t^\star$-th random oracle query where a forking occurs. Namely, the first $t^\star - 1$ $\mathcal{H}_{\mathsf{FS}}$-queries – which must coincide with those of the first run given that $\mathcal{A}$ is provided with the same random tape – obtain the same responses $\mathrm{CH}_1, \ldots, \mathrm{CH}_{t^\star - 1}$ as in the first run. This implies that the $t^\star$-th query necessarily involves the same input $\big(\mathsf{pp}, C^\star, M^\star, \{\mathbf{c}_i^\star\}_{i=1}^{\ell+N}, \{\mathrm{CMT}_i^\star\}_{i=1}^\kappa\big)$ as in the initial run. The forking occurs at the moment of the $t^\star$-th query from which $\mathcal{A}$'s $\mathcal{H}_{\mathsf{FS}}$-queries

receive fresh random responses $\mathrm{CH}'_{t^\star}, \ldots, \mathrm{CH}'_{Q_H}$ at each new run. The Forking Lemma of Brickell *et al.* [14] tells us that, with probability at least $1/2$, $\mathcal{F}$ can obtain a 3-fork involving the same tuple $\big(\mathsf{pp}, C^\star, M^\star, \{\mathbf{c}_i^\star\}_{i=1}^{\ell+N}, \{\mathrm{CMT}_i^\star\}_{i=1}^{\kappa}\big)$ with pairwise distinct responses $\mathrm{CH}_{t^\star}^{(1)}, \mathrm{CH}_{t^\star}^{(2)}, \mathrm{CH}_{t^\star}^{(3)} \in \{1,2,3\}^\kappa$. With probability $1-(7/9)^\kappa$, the results of [14] imply that there exists $j \in \{1, \ldots, \kappa\}$ for which the j-th components of $\mathrm{CH}_{t^\star}^{(1)}, \mathrm{CH}_{t^\star}^{(2)}, \mathrm{CH}_{t^\star}^{(3)}$ are $(Ch_{t^\star,j}^{(1)}, Ch_{t^\star,j}^{(2)}, Ch_{t^\star,j}^{(3)}) = (1,2,3)$.

Now, $\mathcal{F}$ parses the 3 forgeries corresponding to the fork branches to obtain 3 valid responses with respect to 3 different challenges for the same commitment. Then, since $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ is hard and the commitment schemes being used are computationally binding, $\mathcal{F}$ can invoke the knowledge extractor of the underlying argument system (Lemma 3) to obtain $\mathbf{x}' = (x'_1, \ldots, x'_\ell) \in \{0,1\}^\ell$ and $\mathbf{s}' \in [-\beta, \beta]^{2m}$ such that $C^\star(\mathbf{x}') = 1$ and

$$\big[\mathbf{A} \mid \mathbf{A}_0 + \sum_{i=1}^{\ell} x'_i \cdot \mathbf{A}_i\big] \cdot \mathbf{s}' = \mathbf{u} \bmod q.$$

It then follows from the requirements of Definition 8 that $\big(\mathbf{s}', (x'_1, \ldots, x'_\ell)\big)$ is a valid forgery for the Boyen signature with respect to the verification key $(\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{u})$.

Furthermore, the above analysis shows that, if $\mathcal{A}$ has non-negligible success probability and runs in polynomial time, then so does $\mathcal{F}$. This concludes the proof. $\square$