Removing the Strong RSA Assumption from Arguments over the Integers

Geoffroy Couteau, Thomas Peters, and David Pointcheval

ENS, CNRS, INRIA, PSL Research University, Paris, France

Abstract. Committing integers and proving relations between them is an essential ingredient in many cryptographic protocols. Among them, range proofs have shown to be fundamental. They consist of proving that a committed integer lies in a public interval. By the way, it can also be seen as a particular case of the more general Diophantine relations: for the committed vector of integers $\boldsymbol{x}$, there exists a vector of integers $\boldsymbol{w}$ such that $P(\boldsymbol{x}, \boldsymbol{w}) = 0$, where P is a polynomial.

In this paper, we revisit the security strength of the statistically hiding commitment scheme over the integers due to Damgård-Fujisaki, and the zero-knowledge proofs of knowledge of openings. Our first main contribution shows how to remove the Strong RSA assumption and replace it by the standard RSA assumption in the security proofs. This improvement naturally extends to generalized commitments and more complex proofs without modifying the original protocols.

Thereafter, we show that this commitment scheme over the integers is compatible with a commitment scheme modulo a prime p, which allows for more efficient proofs of relations between the committed values, still under the RSA assumption. Our second contribution is thus a more efficient and more secure interactive technique to prove Diophantine relations. We illustrate it with the most efficient range proofs. In addition, the security is proven under the sole RSA assumption.

Keywords. Public-key cryptography, Commitment schemes, Interactive arguments of knowledge, Zero-Knowledge proofs, RSA assumption.

1 Introduction

Commitment Schemes. The notion of commitment is one of the most fundamental and widely used in cryptography. A commitment scheme allows a committer $\mathscr{C}$ holding a secret value s to send a *commitment* c of s to a verifier $\mathscr{V}$, and later on to *open* this commitment to reveal the value s. Such a commitment should *hide* the committed value s to the verifier, but still guaranteeing one opening only (which is the *binding* property). A famous example of commitment scheme, that perfectly hides its input, is the Pedersen commitment scheme [Ped92], whose binding property relies on the discrete logarithm assumption: let $\mathbb{G}$ be a group of prime order p with two generators (g, h). To commit to $m \in \mathbb{Z}_p$, $\mathscr{C}$ picks at random $r \in \mathbb{Z}_p$ and sends $c = g^m h^r$.

Okamoto and Fujisaki [FO97] introduced the first *integer commitment scheme*, which was later generalized in [DF02]. Unlike classical commitment schemes, an integer commitment scheme allows $\mathscr{C}$ to commit to any $m \in \mathbb{Z}$. Intuitively, this is done by committing to m in a group $\mathbb{G}$ of unknown order.

Interactive Proofs of Knowledge. An interactive proof of knowledge is a two-party protocol in which a prover $\mathscr{P}$ wants to convince a verifier $\mathscr{V}$ of his knowledge of some values satisfying a public statement. It should be *knowledge-extractable*, which means that an extractor can get values satisfying the statement when interacting with a successful prover, and *zero-knowledge*, which means that no information about these values leaks to the verifier (except that they satisfy the statement). Such proofs of knowledge are useful in many cryptographic constructions. Commitment schemes are a core component of zero-knowledge proofs of knowledge. In particular, integer commitment schemes have been extensively used in various interactive protocols involving zero-knowledge proofs of knowledge.

Assumptions for Proofs on Integer Commitments. A commonly mentioned downside of proofs on integer commitments schemes is the assumption on which they rely. Indeed, the binding property of the Damgård-Fujisaki commitment scheme relies on the intractability of factoring composite integers. However, the knowledge-extractability of the proofs on these commitments is guaranteed under the $\mathsf{Strong\text{-}RSA}$ assumption [BP97, FO97]. The latter says that, given a composite integer n and a random element $u \in \mathbb{Z}_n^*$, it is hard to find a pair (v, e) such that $u = v^e \bmod n$. Unlike the RSA assumption [RSA78], where the exponent $e > 1$ is imposed, there are exponentially many solutions to a given instance of the $\mathsf{Strong\text{-}RSA}$ problem, the problem is thus easier to solve, hence the "stronger" assumption.

Range Proof. The most widespread reason to work over the integers is to prove that a committed value x lies in a public integer range $[\![a\,;b]\!]$. Indeed, working over the integers allows to show that $x - a$ and $b - x$ are positive by decomposing them as sum of four squares, following the well-known Lagrange's result. Lipmaa [Lip03] was the first to propose such a method by relying on a commitment over the integers. As a consequence, the knowledge extractability requires the $\mathsf{Strong\text{-}RSA}$ assumption.

1.1 Our Contribution

Our contributions in this paper are twofold. First, we revisit the Damgård-Fujisaki integer commitment scheme and show that the security of arguments of knowledge of openings can be based on the standard RSA assumption, instead of the $\mathsf{Strong\text{-}RSA}$ assumption. Our result extends to any protocols involving arguments or relations between committed integers. This implies that the security of numerous protocols, such as two-party computation [JS07, CPP15], e-cash [CHL05], e-voting [Gro05], secure generation of RSA keys [JG02, DM10, HMRT12], zero-knowledge primality tests [CM99a], password-protected secret sharing [JKK14], and range proofs [Lip03], among many others, can be proven under the RSA assumption instead of the $\mathsf{Strong\text{-}RSA}$ assumption. In addition, we believe that the ideas on which our proof relies could be used in several other constructions whose security was proven under the $\mathsf{Strong\text{-}RSA}$ assumption, and might allow to replace the $\mathsf{Strong\text{-}RSA}$ assumption by the standard RSA assumption in such constructions.

Second, we revisit a commitment scheme which was formally introduced in [Gen04]: $c = g^m R^\pi \bmod n$, for a message $m \in \mathbb{Z}_\pi$ and $R \in \mathbb{Z}_n^*$. It is perfectly hiding, and the binding property relies on the RSA assumption with exponent π in $\mathbb{Z}_n^*$. We prove, as for the Damgård-Fujisaki commitment scheme, that the security of an argument of knowledge of an opening can also be based on the classical RSA assumption. In addition, we identify an interesting property that is satisfied by this commitment, which corresponds informally to the possibility to see this commitment scheme either as an integer commitment scheme (i.e., $c = g^m h^r \bmod n$), or, after some secret has been revealed, as a commitment scheme over $\mathbb{Z}_\pi$ for some prime π (i.e., $c = g^m R^\pi \bmod n$). Note that in both situations, the security of the commitment scheme and the argument of knowledge relies on the RSA assumption only. We show how one can take advantage of this feature to improve the efficiency of zero-knowledge arguments over the integers. Our method allows to save communication and greatly reduces the work of the verifier, compared with a classical zero-knowledge argument for the same statement. We illustrate our method on range proofs [Lip03], a zero-knowledge argument of knowledge of an input to a commitment such that the input belongs to some public interval. Taken together, our contributions allow us to enhance both the security (by removing the $\mathsf{Strong\text{-}RSA}$ assumption) and the efficiency of numerous cryptographic protocols relying on integer commitment schemes.

1.2 Related Works

The Damgård-Fujisaki commitment scheme [FO97, DF02] is the only known *compact* statistically-hiding integer commitment scheme (bit-commitment schemes obviously allow to commit to arbitrary integers bit-by-bit). Arguments of knowledge over the integers were studied in [Lip03, KTY04, CCT07].

Range proofs were introduced in [BCDv88]. They are a core component in numerous cryptographic protocols, including e-cash [CHL05], e-voting [Gro05], private auctions [LAN01], group signatures [CM99b], and anonymous credentials [CL01], among many others. There are two classical methods for performing a range proof:

- Writing the number in binary notation [BCDv88, Gro11] or u-ary notation [CCs08], committing to its decomposition and performing a specific proof for each of these commitments For example, membership to $[\![0\,;2^\ell]\!]$ is proven in communication $O(\ell/(\log\ell - \log\log\ell))$ in the protocol of [CCs08], and in communication $O(\ell^{1/3})$ in the protocol of [Gro11].
- Using an integer commitment scheme [Bou00, Lip03, Gro05].

Note that protocols such as [CFT98] do also allow to prove that a committed integer x lies in a given interval $[\![0\,;a]\!]$, but not *exactly*: the proof shows only membership to $[\![0\,;(1+\delta)a]\!]$ for some accuracy parameter δ.

Eventually, several papers have proposed signatures based on the standard RSA assumption [HW09, HJK11, BHJ$^+$13] as alternatives to classical signature schemes based on the $\mathsf{Strong\text{-}RSA}$ assumption. Our work is in the same vein than these papers, replacing the $\mathsf{Strong\text{-}RSA}$ assumption by the RSA assumption in arguments over the integers. However, note that we do not actually propose a new argument system to get rid of the $\mathsf{Strong\text{-}RSA}$ assumption, but rather show that the security of the classical argument system is implied by the RSA assumption. As a consequence, the schemes using arguments over the integers do not need to be modified to benefit from our security analysis.

1.3 Organization.

Section 2 introduces the necessary background for what follows, and namely some useful facts on the RSA groups. Section 3 recalls the Damgård-Fujisaki commitment scheme, its properties, and the argument of knowledge of [DF02]. A new security proof of the latter, under the standard RSA assumption, is given in details Section 4. Section 5 illustrates some extensions of our result. First, we show how one can commit to vectors at once with generalized commitments. And then, we show how one can make range proofs under the standard RSA assumption. Section 6 revisits the commitment scheme of [Gen04] and shows how, by switching from the previous commitment to this one, we can get a new method for performing zero-knowledge arguments over the integers, that is much more efficient. Eventually, Section 7 illustrates our method on range proofs, with concrete efficiency comparisons.

For the sake of completeness, in the Appendix A we exhibit a flaw in the optimized version of Lipmaa's range proof [Lip03, Annex B]. We then propose a fix and prove it.

2 Backgrounds

Throughout this paper, κ denotes the security parameter. An algorithm is *efficient* when it runs in polynomial time in the (implicit) security parameter κ. A positive function f is *negligible* if for any polynomial p there exists a bound $B > 0$ such that, for any integer $k \geq B$, $f(k) \leq 1/|p(k)|$. An event depending on κ occurs with *overwhelming probability* when its probability is at least $1 - \varepsilon(\kappa)$ for a negligible function ε.

2.1 Notations

Given a finite set S, the notation $x \leftarrow_R S$ means a uniformly random affectation of an element of S to the variable x, then for any $s \in S$ we have $\Pr_S[x = s] = 1/|S|$ where $|S|$ denotes the cardinality of S. When an element s is represented by an integer, $|s|$ is the bit-length of the integer, and $||s||$ denotes its absolute value (or norm). Bold variables denote vectors. For a vector $\boldsymbol{x} = (x_1, \cdots, x_\ell)$, $g^{\boldsymbol{x}}$ denotes $(g^{x_1}, \cdots, g^{x_\ell})$.

The integer range $[\![a\,; b]\!]$ stands for $\{x \in \mathbb{Z} \mid a \leq x \leq b\}$, and $[\![a\,; b]\!]_c$ stands for $\{x \in \mathbb{Z} \mid a \leq x \leq b \ \wedge \ \gcd(x, c) = 1\}$. For any integers $a \leq b$, the statistical distance between two uniform distributions, over $U_a = [\![1\,; a]\!]$ and $U_b = [\![1\,; b]\!]$ respectively, is given by $\sum_{i=1}^{b} |\Pr_{U_a}[x = i] - \Pr_{U_b}[x = i]| = \sum_{i=1}^{a}(1/a - 1/b) + \sum_{i=a+1}^{b} 1/b = 2(b-a)/b$.

2.2 Commitment Scheme

We first recall the basic definition of a commitment scheme on the message space $\mathscr{M}$. This is an essential primitive in cryptography, that is used to lock a value in a box, so that the sender cannot change at the opening time (the *binding* property) but still the receiver has no information about the value before the opening (the *hiding* property). A *non-interactive* commitment scheme is defined by three algorithms $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$:

- $\mathsf{Setup}(1^\kappa)$, generates the public parameters pp, which also specifies the message space $\mathscr{M}$, the commitment space $\mathscr{C}$, the opening space $\mathscr{D}$, and the random source $\mathscr{R}$;
- $\mathsf{Commit}(\mathsf{pp}, m; r)$, given the message $m \in \mathscr{M}$ and some random coins $r \in \mathscr{R}$, outputs a commitment-opening pair (c, d). When there is no ambiguity, we will abuse the notation $(c, d) \leftarrow_R \mathsf{Commit}(m)$, for pp and $r \leftarrow_R \mathscr{R}$;
- $\mathsf{Verify}(\mathsf{pp}, c, d, m)$, outputs a bit whose value depends on the validity of the opening (m, d) with respect to the commitment c.

A commitment scheme *must* be

Correct. For any public parameters $\mathsf{pp} \leftarrow_R \mathsf{Setup}(1^\kappa)$, any message $m \in \mathscr{M}$, and any random coin $r \in \mathscr{R}$, if $(c, d) \leftarrow \mathsf{Commit}(\mathsf{pp}, m; r)$, then we necessarily have $\mathsf{Verify}(\mathsf{pp}, c, d, m) = 1$.

Hiding. No probabilistic polynomial-time adversary $\mathscr{A}$, that is first given $\mathsf{pp} \leftarrow_R \mathsf{Setup}(1^\kappa)$, can distinguish commitments on two messages (m_0, m_1) of its choice. The commitment scheme is said *statistically hiding* if the indistinguishability holds even for unbounded adversaries.

Binding. No probabilistic polynomial-time adversary $\mathscr{A}$ can open a commitment c on two different messages $m_0 \neq m_1$. The commitment scheme is said *statistically binding* if this is impossible even for unbounded adversaries.

A commitment scheme can also be *homomorphic*, if for a law $\oplus$ on the message space $\mathscr{M}$, from $(c_0, d_0) \leftarrow \mathsf{Commit}(\mathsf{pp}, m_0; r_0)$ and $(c_1, d_1) \leftarrow \mathsf{Commit}(\mathsf{pp}, m_1; r_1)$, one can generate c from c_0 and c_1 (and pp) as well as d from d_0 and d_1 (and pp) so that $\mathsf{Verify}(\mathsf{pp}, c, d, m_0 \oplus m_1) = 1$.

2.3 Interactive Proof Systems

We now recall the second tool we will use in this paper, the zero-knowledge proofs of knowledge, and their variants.

Zero-Knowledge Proofs and Arguments. Let R be an NP-relation over a set $\mathfrak{X}$ defining an NP-language $\mathscr{L} = \{x \in \mathfrak{X} \mid \exists w, \mathsf{R}(x,w) = 1\}$, where a w such that $\mathsf{R}(x,w) = 1$ is called a *witness* for the statement $x \in \mathscr{L}$.

A *zero-knowledge proof of knowledge* (ZKPoK) for a relation R and a word $x \in \mathfrak{X}$ is an interactive protocol $\langle \mathscr{P}(w), \mathscr{V}\rangle(x \in \mathscr{L})$ between a *prover* $\mathscr{P}$ holding a *witness* w for the statement $x \in \mathscr{L}$, and a verifier $\mathscr{V}$, both modeled as interactive probabilistic polynomial-time Turing machines. The purpose of a ZKPoK is for $\mathscr{P}$ to convince $\mathscr{V}$ of its knowledge of some witness w of the statement $x \in \mathscr{L}$, without revealing any information about this witness. More formally, let $\textsc{View}_{\mathscr{V}}[\langle \mathscr{P}(w), \mathscr{V}\rangle(x \in \mathscr{L})]$ be the view of $\mathscr{V}$ during the execution of the interactive protocol (i.e., all the messages it received when interacting with $\mathscr{P}$). A ZKPoK must be:

Correct. For every $x \in \mathscr{L}$, if $\mathscr{P}$ knows a witness w, and both $\mathscr{P}$ and $\mathscr{V}$ behave honestly, $\langle \mathscr{P}(w), \mathscr{V}\rangle(x \in \mathscr{L})$ is accepted with overwhelming probability by $\mathscr{V}$.

Knowledge Extractable. For any prover $\mathscr{P}$' which succeeds in convincing $\mathscr{V}$ of $x \in \mathscr{L}$ with non-negligible probability, there exists a simulator $\mathscr{S}im_{\mathsf{KE}}$, running in expected polynomial time, which extracts a witness w for $x \in \mathscr{L}$ from $\mathscr{P}$'.

Zero-Knowledge: For any verifier $\mathscr{V}$', there exists a simulator $\mathscr{S}im_{\mathsf{ZK}}$ such that for every $x \in \mathscr{L}$, $\mathscr{S}im_{\mathsf{ZK}}(x)$ and $\textsc{View}_{\mathscr{V}'}[\langle \mathscr{P}(w), \mathscr{V}'\rangle(x \in \mathscr{L})]$, where w is a witness for $x \in \mathscr{L}$, are indistinguishable.

If the knowledge-extractability holds only for a computationally-bounded $\mathscr{P}'$, the protocol is a zero-knowledge *argument* of knowledge (ZKAoK). If the verifier is restricted to being honest in the zero-knowledge property, the proof is an *honest-verifier* zero-knowledge proof.

Zero-Knowledge Arguments from Diophantine Relations. A *Diophantine set* $S \subseteq \mathbb{Z}^k$ is a set of vectors over $\mathbb{Z}^k$ defined by a *representing polynomial* $P_S(X, W)$ with $X = (X_1, \cdots, X_k)$ and $W = (Y_1, \cdots, Y_\ell)$, i.e. a set of the form $S = \{\boldsymbol{x} \in \mathbb{Z}^k \mid \exists \boldsymbol{w} \in \mathbb{Z}^\ell, P_S(\boldsymbol{x}, \boldsymbol{w}) = 0\}$ for some polynomial P_S. It was shown in [DPR61] that any recursively enumerable set is Diophantine. An interesting class for cryptographic applications is the class $\mathbf{D}$ of Diophantine sets S such that each $\boldsymbol{x} \in S$ has at least one witness $\boldsymbol{w}$ satisfying $||\boldsymbol{w}||_1 \leq (||\boldsymbol{x}||_1)^{O(1)}$. It is widely conjectured that $\mathbf{D} = \mathsf{NP}$, as $\mathbf{D}$ contains several NP-complete problems, and it was shown in [Pol03] that if $\mathsf{co\text{-}NLOGTIME} \subseteq \mathbf{D}$, then $\mathbf{D} = \mathsf{NP}$. The class $\mathbf{D}$ was introduced in [AM76] and its cryptographic relevance was pointed out in [Lip03]. For example, the set $\mathbb{Z}_+$ of positive integers is in $\mathbf{D}$, as by a well-known result of Lagrange, it can be defined as $\mathbb{Z}_+ = \{x \in \mathbb{Z} \mid \exists (w_1, w_2, w_3, w_4) \in \mathbb{Z}^4, x - (w_1^2 + w_2^2 + w_3^2 + w_4^2) = 0\}$. In addition, each w_i is of bounded size $||w_i|| \leq ||x||$.

Lipmaa [Lip03] has shown that zero-knowledge arguments of membership to a set $S \in \mathbf{D}$, with representing polynomial P over k-vector inputs and ℓ-vector witnesses, can be constructed using an integer commitment scheme, such as [DF02]. The size of the argument (the communication between $\mathscr{P}$ and $\mathscr{V}$) depends on k, ℓ, and $\deg(P)$, the degree of P. As noted in [Lip03], intervals, unions of intervals, exponential relations (i.e., set of tuples (x, y, z) such that $z = x^y$) and gcd relation (i.e., set of tuples (x, y, z) such that $z = \gcd(x, y)$) are all in $\mathbf{D}$, with parameters (k, ℓ and $\deg(P)$) small enough for cryptographic applications.

2.4 RSA Group Structure

In this paper we focus on $\mathbb{Z}_n^*$ for a strong RSA modulus $n = pq$ where p, q are distinct safe primes. That means that $p = 2p' + 1$ and $q = 2q' + 1$ for two other primes so that p, p', q, q' are all distinct, and $\varphi(n) = 4p'q'$. We write $a = b \bmod n$ to specify that $a = b$ in $\mathbb{Z}_n$ and we write $a \leftarrow [b \bmod n]$ to affect the smallest positive integer to a so that $a = b \bmod n$.

By $\mathsf{GenMod}(1^\kappa)$, we denote a probabilistic efficient algorithm that, given the security parameter κ, generates a strong RSA modulus n and secret parameters (p, q) of at least κ bits each with the specification that $n = pq$. In the following, we write $(n, (p, q)) \leftarrow_R \mathsf{GenMod}(1^\kappa)$. We will sometimes abuse the notation $n \leftarrow_R \mathsf{GenMod}(1^\kappa)$ to say that the modulus n has been generated according to this distribution.

Computational Assumptions. In such RSA groups, there are some famous computational assumptions, such that the intractability of the factorization, but also the so-called RSA and Strong-RSA assumptions. In our construction, we will have some restriction on the exponent e, for which the RSA assumption holds.

Integer Factorization Assumption. It states that finding a non-trivial factor of $n \leftarrow_R \mathsf{GenMod}(1^\kappa)$ is hard for any probabilistic polynomial-time algorithm.

RSA Assumption [RSA78]. It states that, for $n \leftarrow_R \mathsf{GenMod}(1^\kappa)$ and any exponent e, prime to $\varphi(n)$, this is hard to find the e-th root modulo n, for a random $y \leftarrow_R \mathbb{Z}_n^*$, for any probabilistic polynomial-time algorithm.

Strong-RSA Assumption [BP97, FO97]. It lets the choice of e to the algorithm: It states that, for $n \leftarrow_R \mathsf{GenMod}(1^\kappa)$, this is hard to find the e-th root modulo n, for a random $y \leftarrow_R \mathbb{Z}_n^*$, for any probabilistic polynomial-time algorithm, for an exponent $e > 1$ of its choice.

It is well-known that breaking the integer factorization assumptions allows to break the two others. From the definition, it is clear that the Strong-RSA assumption gives more degree of freedom to the adversary, so it is way stronger. This is the reason why one always tries to avoid it.

In this paper, we will rely on the RSA assumption, in the sense that the exponent is not chosen by the adversary, but it will be assumed to be randomly chosen in some set, hence our so-called Random-RSA assumption, for a set S, in which e is randomly drawn (in addition to be prime to $\varphi(n)$), in the same vein as in [HW09]. But, as they say, this is just a way to specify the choice of the exponent in the standard RSA assumption.

Properties of Strong RSA Groups. One can note that in such groups, p and q are Blum primes: $p = q = 3 \bmod 4$. If we denote QR_n the subgroup of the squares, $\mathsf{QR}_n = \{a \in \mathbb{Z}_n^* \mid \exists b \in \mathbb{Z}_n^*, a = b^2 \bmod n\}$, this is a cyclic subgroup of $\mathbb{Z}_n^*$ of order $\varphi(n)/4 = p'q'$.

Proposition 1. *The following facts hold:*

1. $-1 \notin \mathsf{QR}_n$;
2. *any square* $a \in \mathsf{QR}_n$ *has four square roots, with exactly one* QR_n;
3. *for a random element* $h \in \mathsf{QR}_n$, *finding a square root of* h *is equivalent to factor the modulus* n;
4. *for random elements* $g, h \in \mathsf{QR}_n$, *finding non-zero integers* a, b *such that* $g^a = h^b \bmod n$ *is equivalent to factoring the modulus* n;
5. *for an RSA instance* (n, e, y), *finding* $x \in \mathbb{Z}_n^*$ *and* e' *prime to* e *such that* $x^e = y^{e'} \bmod n$ *is equivalent to finding an* e*-th root of* y *modulus* n.

Proof. Let us briefly explain why these facts hold, using the Jacobi symbol function $J_n(x) = J_p(x) \times J_q(x)$ in $\mathbb{Z}_n^*$, as the extension of the Legendre symbol on $\mathbb{Z}_p^*$ for prime p, $J_p(x) = (x)^{(p-1)/2}$, which determines whether x is a square or not in $\mathbb{Z}_p^*$. Since p and q are Blum primes, $J_p(-1) = J_q(-1) = -1$, and so $J_n(-1) = 1$, but -1 is not in QR_n, hence the fact 1. The four square roots of 1, in $\mathbb{Z}_n^*$ are 1 and -1, both with Jacobi symbol $+1$, but respectively $(+1, +1)$ and $(-1, -1)$ for the Legendre symbols in $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$, and α, and $-\alpha$, both with Jacobi symbol -1, but respectively $(+1, -1)$ and $(-1, +1)$ for the Legendre symbols in $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$. As a consequence, given a square $h \in \mathsf{QR}_n$, and a square root u, the four square roots are $u, -u$, and $\alpha u, -\alpha u$, one of which being in QR_n, since the four kinds of Legendre symbols. This leads to the fact 2.

For fact 3, if one chooses a random $u \in \mathbb{Z}_n^*$ and sets $h = u^2 \bmod n$, $J_n(u)$ is completely hidden. Another square root v has probability one-half to have $J_n(v) = -J_n(u)$. This means that $u^2 = v^2 \bmod n$, but $u \neq \pm v \bmod n$. Then, $\gcd(u - v, n)$ gives a non-trivial factor of n.

For fact 4, if one chooses a random $u \in \mathbb{Z}_n^*$ and a large random scalar α and sets $h = u^2 \bmod n$ and $g = h^\alpha \bmod n$, h is likely a generator of QR_n, and then $g^a = h^b \bmod n$ means that $m = b - a\alpha$ is a multiple of $p'q'$, the order of the subgroup of the squares. Let us note $m = 2^v \cdot t$, for an odd t, then $p'q'$ divides t: let us choose a random element $u \in \mathbb{Z}_n^*$, with probability close to one-half, $J_n(u) = -1$, and so $J_n(u^t) = -1$ while u^t is a square root of 1. As in the proof of the previous fact 3, we can obtain a non-trivial factor of n.

Eventually, for fact 5, using Bézout relation $ue + ve' = 1$, then $x^{ve} = y^{ve'} = y^{1-ue} \bmod n$. So $y = (x^v y^u)^e \bmod n$. □

3 Commitment of Integers Revisited

In [FO97], Okamoto and Fujisaki proposed a statistically-hiding commitment scheme allowing commitment to arbitrary-size integers. Their commitment was later generalized in [DF02]. It relies on the fact that when the factorization is unknown, it is infeasible to know the order of the sub-group QR_n of the squares in $\mathbb{Z}_n^*$, where n is a strong RSA modulus. Hence, the only way for a computationally-bounded committer to open a commitment is to do it over the integers.

In addition, [FO97] gave an argument of knowledge of an opening of a commitment and proved that the knowledge extractability of the argument is implied by the Strong-RSA assumption. A flaw in the original proof was later identified and corrected in [DF02]. We will revisit the argument of knowledge of an opening due to Damgård-Fujisaki [DF02] and provide an extended proof for its knowledge extractability, in order to remove the requirement of the Strong-RSA assumption. Our proof requires the standard RSA assumption only, with an exponent randomly chosen in a polynomially-bounded set.

3.1 Commitments over the Integers

Description. Let us recall the commitment of one integer m:

- $\mathsf{Setup}(1^\kappa)$ runs $(n, (p, q)) \leftarrow_R \mathsf{GenMod}(1^\kappa)$, and picks two random generators g, h of QR_n. It returns $\mathsf{pp} = (n, g, h)$;
- $\mathsf{Commit}(\mathsf{pp}, m; r)$, for $\mathsf{pp} = (n, g, h)$, a message $m \in \mathbb{Z}$, and some random coins $r \leftarrow_R [\![0\,; n]\!]$, computes $c = g^m h^r \bmod n$, and returns (c, d) with $d = r$;
- $\mathsf{Verify}(\mathsf{pp}, c, d, m)$ parses pp as $\mathsf{pp} = (n, g, h)$ and outputs 1 if $c = \pm g^m h^d \bmod n$ and 0 otherwise.

One should note that an honest user will always open such that $c = g^m h^d \bmod n$. But the binding property cannot exclude the change of sign. In this description, we provide a trusted setup algorithm. But as we see below, the guarantees for the prover (the hiding property of the commitment) just rely on the existence of α such that $g = h^\alpha \bmod n$. For the verifier to be convinced, one can just let him generate the parameters (n, g, h), and prove the existence of such an α to the prover.

Security Analysis. The above commitment scheme is obviously *correct*. The *hiding* property relies on the existence of α such that $g = h^\alpha \bmod n$ (they are both generators of the same subgroup QR_n), and so

$$\begin{aligned} c = \mathsf{Commit}(\mathsf{pp}, m; r) &= g^m h^r = h^{r+\alpha m} = h^{(r+\alpha(m-m'))+\alpha m'} \\ &= g^{m'} h^{r+\alpha(m-m')} = \mathsf{Commit}(\mathsf{pp}, m'; r'), \end{aligned}$$

for any $m' \in \mathbb{Z}$, with $r' \leftarrow [r + \alpha(m - m') \bmod p'q']$, that is smaller than n. The binding property relies on the Integer Factorization assumption: indeed, from two different openings m_0, d_0, m_1, d_1 for a commitment c, with $d_1 > d_0$, the validity checks show that $g^{m_0}h^{d_0} = \pm g^{m_1}h^{d_1} \bmod n$, and so $g^{m_0-m_1} = \pm h^{d_1-d_0} \bmod n$. Since g and h are squares, and -1 is not a square, necessarily $g^{m_0-m_1} = h^{d_1-d_0} \bmod n$. The Fact 4 from Proposition 1 leads to a non-trivial factor of n.

3.2 Zero-Knowledge Argument of Opening

Let us now study the argument of knowledge of a valid opening for such a commitment. The common inputs are the public parameters pp and the commitment $c = g^x h^r \bmod n$, together with the bit-length k_x of the message x, that is then assumed to be in $[\![-2^{k_x} ; 2^{k_x}]\!]$, while $r \in [\![0 ; n]\!]$ and x are the private inputs, i.e. the witness of the prover. We stress that k_x is chosen by the prover, since this reveals some information about the integer x, while r is always in the same set, whatever the committed element x is.

Description of the Protocol. The protocol works as follows:

Initialize: $\mathscr{P}$ and $\mathscr{V}$ decide to run the protocol on input $(\mathsf{pp}, \kappa, c, k_x)$;
Commit: $\mathscr{P}$ computes $d = g^y h^s \bmod n$, for randomly chosen $y \leftarrow_R [\![0 ; 2^{k_x+2\kappa}]\!]$ and $s \leftarrow_R [\![0 ; 2^{|n|+2\kappa}]\!]$, and sends d to the $\mathscr{V}$;
Challenge: $\mathscr{V}$ outputs $e \leftarrow_R [\![0 ; 2^{\kappa}]\!]$;
Response: $\mathscr{P}$ computes and outputs the integers $z = ex + y$ and $t = er + s$;
Verify: $\mathscr{V}$ accepts the proof and outputs 1 if $c^e d = g^z h^t \bmod n$. Otherwise, $\mathscr{V}$ rejects the proof and outputs 0.

In the rest of this section, we prove this protocol is indeed a zero-knowledge argument of knowledge of an opening. Which means it is correct, zero-knowledge, and knowledge-extractable.

Correctness. First, the correctness is quite obvious: if $c = g^x h^r \bmod n$, with $z = ex + y$ and $t = er + s$, we have $g^z h^t = (g^x h^r)^e \cdot g^y h^s = c^e d \bmod n$.

Zero-Knowledge. For the zero-knowledge property, in the honest-verifier setting, the simulator $\mathscr{S}\!im$ (that is $\mathscr{S}\!im_{\mathsf{ZK}}$ in this case) can simply do as follows:

1. $\mathscr{S}\!im$ chooses a random challenge $e \leftarrow_R [\![0 ; 2^{\kappa}]\!]$;
2. $\mathscr{S}\!im$ chooses random responses $z \leftarrow_R [\![0 ; 2^{k_x+2\kappa}]\!]$ and $t \leftarrow_R [\![0 ; 2^{|n|+2\kappa}]\!]$;
3. $\mathscr{S}\!im$ sets $d = g^z h^t c^{-e} \bmod n$.

The simulated transcript is the tuple $(d, e, (z, t))$.

Actually, the real distribution of the transcript is for tuples that follow:

$$\mathscr{D}_0 : \left\{ \begin{array}{l} y \leftarrow_R [\![0 ; 2^{k_x+2\kappa}]\!], s \leftarrow_R [\![0 ; 2^{|n|+2\kappa}]\!], \\ e \leftarrow_R [\![0 ; 2^{\kappa}]\!], z = xe + y, t = re + y, d = g^y h^s \bmod n \end{array} \right\}.$$

This is exactly the same as

$$\mathscr{D}_1 : \left\{ \begin{array}{l} z \leftarrow_R [\![xe ; 2^{k_x+2\kappa} + xe]\!], t \leftarrow_R [\![re ; 2^{|n|+2\kappa} + re]\!], \\ e \leftarrow_R [\![0 ; 2^{\kappa}]\!], d = g^{z-xe} h^{t-re} \bmod n \end{array} \right\}$$

which can be rewritten as

$$\mathscr{D}_2 : \left\{ \begin{array}{l} z \leftarrow_R [\![xe ; 2^{k_x+2\kappa} + xe]\!], t \leftarrow_R [\![re ; 2^{|n|+2\kappa} + re]\!], \\ e \leftarrow_R [\![0 ; 2^{\kappa}]\!], d = g^z h^t c^{-r} \bmod n \end{array} \right\},$$

while the distribution generated by the simulator $\mathscr{S}im$ is

$$\mathscr{D}_3 : \left\{ \begin{array}{l} z \leftarrow_R [\![0 \,; 2^{k_x+2\kappa}]\!], t \leftarrow_R [\![0 \,; 2^{|n|+2\kappa}]\!], \\ e \leftarrow_R [\![0 \,; 2^{\kappa}]\!], d = g^z h^t c^{-r} \bmod n \end{array} \right\}.$$

As just said, this is clear that $\mathscr{D}_0 = \mathscr{D}_1 = \mathscr{D}_2$, while the distance between $\mathscr{D}_2$ and $\mathscr{D}_3$ is the sum of the distances between the distributions of z and t, respectively in $\mathscr{Z}_2 = [\![xe \,; 2^{k_x+2\kappa} + xe]\!]$ and $\mathscr{Z}_3 = [\![0 \,; 2^{k_x+2\kappa}]\!]$, and $\mathscr{T}_2 = [\![re \,; 2^{|n|+2\kappa} + re]\!]$ and $\mathscr{T}_3 = [\![0 \,; 2^{|n|+2\kappa}]\!]$:

$$\begin{aligned} \varDelta_z &= \sum_{Z=0}^{2^{k_x+2\kappa}+xe} |\Pr[z \leftarrow_R \mathscr{Z}_2 : z = Z] - \Pr[z \leftarrow_R \mathscr{Z}_3 : z = Z]| \\ &= \sum_{Z=0}^{xe-1} 2^{-k_x-2\kappa} + \sum_{Z=2^{k_x+2\kappa}+1}^{2^{k_x+2\kappa}+xe} 2^{-k_x-2\kappa} = 2 \cdot xe \cdot 2^{-k_x-2\kappa} \leq 2 \cdot 2^{k_x+\kappa} \cdot 2^{-k_x-2\kappa} \end{aligned}$$

that is bounded by $2 \cdot 2^{-\kappa}$. Similarly, $\varDelta_t \leq 2 \cdot 2^{-\kappa}$. Hence the statistical zero-knowledge property, since the real distribution $\mathscr{D}_0$ and the simulated distribution $\mathscr{D}_3$ have a negligible distance bounded by $2^{-\kappa+2}$.

Knowledge-Extractability. The last property is the most intricate, and this is the one that required the Strong-RSA assumption in the original proof of Damgård and Fujisaki [DF02]. We first give an intuition of how we can get rid of it, and then present the proof in details in the next section, since this is the main contribution of the paper.

Damgård and Fujisaki consider a classical simulator that rewinds the prover to get two related transcripts $(d, e_0, (z_0, t_0))$ and $(d, e_1, (z_1, t_1))$. In the “good” event the simulator can immediately extract a valid opening of the commitment. Then in the “bad” event two cases can appear: case 1 corresponds to an event which can be reduced to a Strong-RSA solver and we will discuss that case below. Case 2 is shown to happen with probability at most 1/2 using an information theoretic argument, but independently from the view of the adversary. Hence, when the “good” event does not happen, there is a probability at least 1/2 for the case 2 not to happen either. Damgård and Fujisaki are thus left with the case 1. In this case, they explain how one can extract a pair $(\tilde{h}, w)$ such that $\tilde{h}^w = h \bmod n$, where $h \leftarrow y$ comes from the Strong-RSA-challenge[1].

Our starting point is the observation that this case 1 can be refined if one considers further rewindings to rely on the standard RSA assumption instead of the Strong-RSA assumption. Roughly speaking, we divide this case 1 into two subcases, depending on whether the exponent w such that $h = \tilde{h}^w \bmod n$ remains unchanged when rewinding further, or changes. If the adversary does not consistently use the same w, then we show how to extract non-trivial values (a, b) such that $g^a = h^b \bmod n$, which leads to a non-trivial factor of n (see Fact 4 from Proposition 1). If, however, the adversary keeps using the same w, which corresponds to $(a, b) = (0, 0)$, then we observe that this w must divide all the challenge-differences for which the adversary answered with a valid proof[2]. But as we know that the adversary succeeds in the proof with some non-negligible probability ε, we have an upper-bound B on the size of w: if w is too big, the probability that it divides a bunch of random independent values would become lower than ε. This upper-bound implies that if the simulator is given some RSA challenge e picked as a uniformly random integer in $[\![2 \,; B]\!]_{\varphi(n)}$, then

[1] Actually, they extract $(\mu, \tilde{h}, w)$ such that $h = \mu\tilde{h}^w \bmod n$ for some μ in a very small subgroup, but this is shown to be sufficient to break the Strong-RSA assumption.

[2] More precisely, w must divide all the e'_i where $e'_i = e_0 - e_i$, e_0 being the very first challenge for which the adversary produced a valid transcript, and e_i being any other challenge for which the adversary returned a valid proof.

either w is even or $w = e$ with non-negligible probability (since we will show that $B \leq 4/\varepsilon$). When w is even, we can compute a square root of h and recover the factorization of n, from which we can solve any RSA challenge. When $w = e$, we have solved our RSA challenge (n, e, y) if $h \leftarrow y^2 \bmod n$.

As a consequence, unless one can break the RSA assumption, our extractor is likely in the “good” event, and gets a valid opening. More precisely, we prove the following theorem.

Theorem 2. *Given a prover $\mathscr{P}$' able to convince a verifier $\mathscr{V}$ of its knowledge of an opening of c for random system parameters $\mathsf{pp} = (n, g, h)$ with probability greater than ε within time t, one either breaks the RSA assumption with expected time upper-bounded by $64t/\varepsilon^3$, or outputs a valid opening with expected time upper-bounded by $16t/\varepsilon^2$.*

4 Proof of Theorem 2

Since this proof is the main technical contribution of the paper, with many practical applications, we provide it in details. We start with some preliminaries, and then discuss various cases.

4.1 Preliminaries

The proof will make use of the splitting lemma [PS96, PS00], that we recall below:

Lemma 3. *Let $A \subset X \times Y$ such that $\Pr[(x, y) \in A] \geq \varepsilon$. For any $\varepsilon' < \varepsilon$, if one defines $B = \big\{(x, y) \in X \times Y \mid \Pr_{y' \in Y}[(x, y') \in A] \geq \varepsilon - \varepsilon'\big\}$, then it holds that:*

$$(i)\, \Pr[B] \geq \varepsilon' \qquad (ii)\ \forall (x, y) \in B, \Pr_{y' \in Y}[(x, y') \in A] \geq \varepsilon - \varepsilon' \qquad (iii)\, \Pr[B \mid A] \geq \varepsilon'/\varepsilon.$$

4.2 Detailed Proof

Let us suppose the extractor $\mathscr{S}im$ (that is $\mathscr{S}im_{\mathsf{KE}}$ in this case) is given a $4/\varepsilon$-RSA challenge (n, e, u). It sets $h \leftarrow u^2 \bmod n$ and $g \leftarrow h^\alpha \bmod n$ for a random exponent $\alpha \leftarrow_R \mathbb{Z}_{n^2}$. It sets $\mathsf{pp} = (n, g, h)$. Note that as u is random, (g, h) are indeed distributed as in the real protocol. We consider an adversary $\mathscr{A}$ that provides a convincing proof of knowledge of an opening of c with probability ε, with the parameters $(\mathsf{pp} = (n, g, h), \kappa, c, k_x)$.

Note that the distribution probability of a protocol execution is $D = (R, e)$, where R is the adversary's random tape that determines d, and the random challenge from e the verifier. Since this is a “good” adversary”, we assume that on a random pair (R, e), its probability to output a valid transcript (d, e, z, t) is greater than ε. We apply the splitting lemma with $\varepsilon' = \varepsilon/2$ for the distribution $D = \{R\} \times \{e\}$: after one execution, with probability greater than ε, we obtain a successful transcrit (d, e_0, z_0, t_0). In such a case, with probability greater than $1/2$, R is a good random tape, which means that another execution with the same R but a random challenge e_i will lead to another successful transcript (d, e_i, z_i, t_i) with probability $\varepsilon' = \varepsilon/2$. Note that since R is kept unchanged, d is the same. Globally, with probability greater than $\varepsilon^2/4$, after 2 executions of the protocol, one gets two related successful transcripts: (d, e_0, z_0, t_0) and (d, e_1, z_1, t_1).

Without loss of generality, we may assume $e_0 \geq e_1$. Writing $e'_1 \leftarrow e_0 - e_1$, $z'_1 \leftarrow z_0 - z_1$, and $t'_1 \leftarrow t_0 - t_1$, the two valid tuples lead to the relation $c^{e'_1} = g^{z'_1} h^{t'_1} \bmod n$.

Case 1: e'_1 divides both z'_1 and t'_1. $\mathscr{S}im$ simply outputs the pair of integers $(x_1, r_1) \leftarrow (z'_1/e'_1, t'_1/e'_1)$. If e'_1 is odd, and thus prime to $\varphi(n)$, we have $c = g^{x_1} h^{r_1} \bmod n$. However, if $e'_1 = 2^v \rho$ for an odd ρ and $v \geq 1$, $(c^{-1} g^{x_1} h^{r_1})^{2^v} = 1 \bmod n$: from the Fact 2 from Proposition 1, $(c^{-1} g^{x_1} h^{r_1})^2 = 1 \bmod n$:

- either $c^{-1}g^{x_1}h^{r_1} = \pm 1 \bmod n$, and so $c = \pm g^{x_1}h^{r_1} \bmod n$ (valid opening);
- or we have a non-trivial square root of 1, which leads to the factorization of n (see Proposition 1).

We denote p_1 the probability of case 1 happening when one got two successful transcripts. Recall that the two successful transcripts are obtained with probability greater than $\varepsilon^2/4$. Then, either $p_1 \geq \varepsilon^2/8$, and in this case $\mathscr{S}im$ extracts an opening of c with probability $\varepsilon^2/8$ (under the Integer Factorization assumption), or $\varepsilon^2/4 - p_1 \geq \varepsilon^2/8$. Let us now assume case 1 does not happen often enough.

Case 2: e'_1 divides $\alpha z'_1 + t'_1$ (but does not divide both z'_1 and t'_1). Let us argue that this happens with probability at most 1/2 given that case 1 does not occur, and that this probability is completely independent of the actions of $\mathscr{A}$. Note that this is exactly the case 2 from [DF02]. The intuition behind the proof is that the only information that $\mathscr{A}$ can get about α is from $g = h^\alpha \bmod n$. However, this leaks only $\alpha \bmod p'q'$, while α was taken at random in $\mathbb{Z}_{n^2}$: all the information on its most significant bits is *perfectly* hidden. We recall below the proof given by Damgård and Fujisaki, for completeness.

Let Q be a prime factor of e'_1 and j be an integer such that Q^j divides e'_1 but Q^{j+1} does not divide e'_1, and at least one of z'_1 or t'_1 is non-zero modulo Q^j. As we are not in case 1, e'_1 does not divide both z'_1 and t'_1, so such a prime Q does necessarily exist. If Q^j divides z'_1, as it divides e'_1, it must also divide $\alpha z'_1 + t'_1$ and therefore t'_1, which again cannot happen as we are not in case 1. Therefore, $z'_1 \neq 0 \bmod Q^j$. Then, it holds that $\alpha = [\alpha \bmod p'q'] + \lambda p'q'$ for some λ. Let us write $\mu = [\alpha \bmod p'q']$. The tuple (n, g, h) uniquely determines μ, whereas λ is perfectly unknown to the prover. As Q^j divides $\alpha z'_1 + t'_1$, it holds that

$$\alpha z'_1 + t'_1 = \lambda z'_1 p'q' + \mu z'_1 + t'_1 = 0 \bmod Q^j.$$

Note that $p'q' \neq 0 \bmod Q$. And from the view of the adversary, λ is chosen uniformly at random among at least n values, and must satisfy the above equation for case 2 to occur. But since this equation has at most $\gcd(z'_1 p'q', Q^j)$ solutions, which is a power of Q (and at most Q^{j-1}), and since n is larger than Q^j by a factor (way) bigger than 2^κ, the distribution of $\lambda \bmod Q^j$ is statistically close to uniform in $\mathbb{Z}_{Q^j}$, and the probability that λ satisfies the above equation is bounded by $1/Q - 2^{-\kappa} \leq 1/2$, independently of the actions of $\mathscr{A}$. Hence, under the condition that we are not in case 1, with probability at least 1/2, the case 2 does *not* occur either: with probability greater than $\varepsilon^2/16$, the following case 3 happens.

Case 3: e'_1 does not divide $\alpha z'_1 + t'_1$. We will now prove that when case 3 occurs, $\mathscr{S}im$ can solve an RSA instance, which is the difference with the original proof. Let $\beta_1 = \gcd(e'_1, \alpha z'_1 + t'_1) < e'_1$. Let $\mathit{\Gamma}_1 \leftarrow e'_1/\beta_1$ and $F_1 \leftarrow (\alpha z'_1 + t'_1)/\beta_1$: $F_1/\mathit{\Gamma}_1$ is the irreducible fraction form of $(\alpha z'_1 + t'_1)/e'_1$ and $\mathit{\Gamma}_1 > 1$.

Let $\mathscr{S}im$ rewind the protocol once more and get a third valid transcript (d, e_2, z_2, t_2) (recall that this happens with probability $\varepsilon/2$), it can compute $\mathit{\Gamma}_2$ in the same way it computed $\mathit{\Gamma}_1$. We consider the following two situations :

- **Subcase 3.a.** $\mathit{\Gamma}_2 = \mathit{\Gamma}_1$, with probability $p_a \geq \varepsilon/4$;
- **Subcase 3.b.** $\mathit{\Gamma}_2 \neq \mathit{\Gamma}_1$, with probability $p_b \geq \varepsilon/4$.

Subcase 3.a. If $p_a \geq \varepsilon/4$, this means that with probability greater than $\varepsilon/4$, $\mathit{\Gamma}_2 = \mathit{\Gamma}_1$ is a constant value $\mathit{\Gamma}$. This $\mathit{\Gamma}$ divides e'_2 for any new successful transcript that would lead to subcase 3.a. Since

$e_2 \leftarrow_R [\![0\,;2^\kappa]\!]$, independently of $\varGamma$, the probability that $\varGamma$ divides $e'_2 = e_0 - e_2$ for a uniformly random e_2 is at most $1/\varGamma$:

$$\varepsilon/4 \leq p_a = \Pr_{e_2}[\text{Succes in subcase 3.2}] \leq \Pr_{e_2}[\varGamma \text{ divides } (e_0 - e_2)] \leq 1/\varGamma.$$

Hence, it must hold that $\varGamma \leq 4/\varepsilon$. In addition, since $\beta_1 < e'_1$, we can assume $\varGamma \in [\![2\,;4/\varepsilon]\!]$. In order to simplify the notations, after one rewind, we get (e', z', t') so that $c^{e'} = g^{z'} h^{t'} \bmod n$ and $\beta = \gcd(e', \alpha z' + t')$ with $1 < \varGamma = e'/\beta \leq 4/\varepsilon$, with global probability greater $\varepsilon^2/32$.

We note $e' = \beta\varGamma$ and $\alpha z' + t' = \beta k$ for relatively prime integers $\varGamma, k$. Since $h = u^2 \bmod n$ and $c^{e'} = h^{\alpha z' + t'} \bmod n$, we have $c^{e'} = u^{2(\alpha z' + t')} \bmod n$, which reduces to $c^{\varGamma} = c^{e'/\beta} = \pm u^{2(\alpha z' + t')/\beta} = \pm u^{2k} \bmod n$, where $\varGamma$ and k are relatively prime, and $\varGamma > 1$:

- if $\varGamma = 2^a$ with $a \geq 1$, we thus have an odd k such that $c^{2^a} = u^{2k} \bmod n$: $c^{2^{a-1}}$ and u^k are two square roots of the same value. Since no information leaks about the actual square roots $\{u, -u\}$ known for h, nor for $h^k \bmod n$, so $c^{2^{a-1}} \neq \pm u^k \bmod n$ with probability $1/2$, which leads to the factorization of n (see Proposition 1);
- if $\varGamma = 2^a v$ with an odd $v > 1$, we thus have $C^v = u^{2k} \bmod n$, for $C = \pm c^{2^a}$ and $\gcd(v, 2k) = 1$, since $v|\varGamma$ and v is odd. Using the Fact 5 from Proposition 1, one gets the v-th root of u modulo n, for $v \in [\![3\,;4/\varepsilon]\!]$.

As a consequence, if $p_a \geq 4/\varepsilon$, after just one rewind, and thus with probability greater than $\varepsilon^2/32$, $\mathscr{S}im$ extracts a v-th root of u modulo n, for $2 \leq v \leq 4/\varepsilon$. Since our simulation that uses the RSA challenge (n, u, e) does not leak *any* information about e, $v = e$ with probability greater than $\varepsilon/4$, if the exponent e is randomly chosen in $[\![2\,;4/\varepsilon]\!]$.

Subcase 3.b. We now assume that $p_b \geq \varepsilon/4$: after two rewindings, we have two relations $c^{e'_1} = g^{z'_1} h^{t'_1} \bmod n$ and $c^{e'_2} = g^{z'_2} h^{t'_2} \bmod n$, and so $c^{e'_1 e'_2} = g^{e'_2 z'_1} h^{e'_2 t'_1} = g^{e'_1 z'_2} h^{e'_1 t'_2} \bmod n$. This leads, for $\varDelta_z = e'_2 z'_1 - e'_1 z'_2$ and $\varDelta_t = e'_2 t'_1 - e'_1 t'_2$, to

$$g^{\varDelta_z} = g^{e'_2 z'_1 - e'_1 z'_2} = h^{e'_1 t'_2 - e'_2 t'_1} = h^{-\varDelta_t} \bmod n.$$

If $\varDelta_z = \varDelta_t = 0$, then it holds that $z'_2/e'_2 = z'_1/e'_1$ and $t'_2/e'_2 = t'_1/e'_1$:

$$\frac{F_2}{\varGamma_2} = \frac{\alpha z'_2 + t'_2}{e'_2} = \alpha \cdot \frac{z'_2}{e'_2} + \frac{t'_2}{e'_2} = \alpha \cdot \frac{z'_1}{e'_1} + \frac{t'_1}{e'_1} = \frac{\alpha z'_1 + t'_1}{e'_1} = \frac{F_1}{\varGamma_1}.$$

Since they are both the irreducible notation of the same fraction, we necessarily have $\varGamma_1 = \varGamma_2$ and $F_1 = F_2$, whereas we excluded this case. Hence, when subcase 3.b happens, the pair $(\varDelta_z, \varDelta_t)$ is non-trivial and and leads to the factorization of n, from the Fact 4 from Proposition 1.

As a consequence, under the RSA assumption (which implies the Integer Factorization assumption), our extractor falls in case 1 with probability greater than $\varepsilon^2/8$ and outputs an opening of c.

5 Classical Extensions and Applications

We revisit the natural implications of the commitment scheme of Section 3 and its argument of knowledge. More precisely, we generalize the results of previous sections while we commit to vectors of integers. Then, we also show the security of Lipmaa's range proofs [Lip03] under the RSA assumption to illustrate how the result of Section 4 extends to more general arguments over the integers.

5.1 Generalized Commitment of Integers

The following commitment scheme allows committing to a vector of integers $(m_1, \ldots, m_\ell)$ with a single element of the form $c = g_1^{m_1} \cdots g_\ell^{m_\ell} h^r \bmod n$:

- $\mathsf{Setup}(1^\kappa, \ell)$ runs $(n, (p, q)) \leftarrow_R \mathsf{GenMod}(1^\kappa)$, and picks $\ell + 1$ random generators $(g_1, \ldots, g_\ell, h)$ of QR_n. It returns $\mathsf{pp} = (n, g_1, \ldots, g_\ell, h)$;
- $\mathsf{Commit}(\mathsf{pp}, \boldsymbol{m}; r)$, for $\mathsf{pp} = (n, g_1, \ldots, g_\ell, h)$, a vector $\boldsymbol{m} = (m_1, \ldots, m_\ell) \in \mathbb{Z}^\ell$, and some random coins $r \leftarrow_R [\![0\,; n]\!]$, computes $c = g_1^{m_1} \cdots g_\ell^{m_\ell} h^r \bmod n$, and returns (c, d) with $d = r$;
- $\mathsf{Verify}(\mathsf{pp}, c, d, \boldsymbol{m})$ parses pp as $\mathsf{pp} = (n, g_1, \ldots, g_\ell, h)$ and outputs 1 if $c = g_1^{m_1} \cdots g_\ell^{m_\ell} h^d \bmod n$ and 0 otherwise.

Again, the above commitment scheme is obviously *correct*. The *hiding* property relies on the existence of α_i such that $g_i = h^{\alpha_i} \bmod n$ for $i = 1, \ldots, \ell$, and so

$$\begin{aligned} c = \mathsf{Commit}(\mathsf{pp}, \boldsymbol{m}; r) &= g_1^{m_1} \cdots g_\ell^{m_\ell} h^r = h^{r + \sum \alpha_i m_i} \\ &= h^{(r + \sum \alpha_i (m_i - m_i')) + \sum \alpha_i m_i'} = g_1^{m_1'} \cdots g_\ell^{m_\ell'} h^{r + \sum \alpha_i (m_i - m_i')} \\ &= \mathsf{Commit}(\mathsf{pp}, \boldsymbol{m'}; r'), \end{aligned}$$

for any $\boldsymbol{m'} = (m_1', \ldots, m_\ell') \in \mathbb{Z}$, with $r' \leftarrow [r + \sum \alpha_i (m_i - m_i') \bmod p'q']$, that is smaller than n.

The binding property relies on the Integer Factorization assumption: indeed, from two different openings $(\boldsymbol{m}, d)$ and $(\boldsymbol{m'}, d')$ for a commitment c, with $d' > d$, the validity checks show that $g_1^{m_1} \cdots g_\ell^{m_\ell} h^d = g_1^{m_1'} \cdots g_\ell^{m_\ell'} h^{d'} \bmod n$, and so, if one has chosen β_i such that $g_i = g^{\beta_i} \bmod n$, for a random square g, then one knows $g^{\sum \beta_i (m_i - m_i')} = h^{d' - d} \bmod n$. The Fact 4 from Proposition 1 leads to the conclusion.

To avoid a trusted setup, one can note that the guarantees for the prover (the hiding property) just rely on the existence of α_i such that $g_i = h^{\alpha_i} \bmod n$ for $i = 1, \ldots, \ell$. The well-formedness of the RSA modulus is for the verifier guarantees. It is important for him that the prover cannot break the RSA assumption. So the setup can be run by the verifier, with an additional proof of existence of α_i such that $g_i = h^{\alpha_i} \bmod n$ for $i = 1, \ldots, \ell$ to the prover.

5.2 Zero-Knowledge Argument of Opening

An argument of knowledge of an opening of a commitment $c = g_1^{x_1} \cdots g_\ell^{x_\ell} h^r \bmod n$ in the general case can be easily adapted from the normal case leading to a transcript of the form $(d, e, (z_1, \ldots, z_\ell, t))$ with $d = g_1^{y_1} \cdots g_\ell^{y_\ell} h^s$, and $c^e d = g_1^{z_1} \cdots g_\ell^{z_\ell} h^t \bmod n$.

As above, the knowledge-extractor rewinds the execution for the same d, but two different challenges $e_0 \neq e_1$. Doing the quotient of the two relations, d cancels out: $c^{e'} = g_1^{z_1'} \cdots g_\ell^{z_\ell'} h^{t'} \bmod n$.

Let us assume that one would have set $g_i = g^{a_i} h^{b_i} \bmod n$, we would have

$$c^{e'} = g^{\sum a_i z_i'} h^{\sum b_i z_i' + t'} \bmod n.$$

Under the RSA assumption, we are in the above case 1: e' divides both $\sum a_i z_1'$ and $\sum b_i z_i' + t'$ with non-negligible probability. Since the coefficients a_i's and b_i's are random, this means that e' divides all the z_i''s and t'. Hence, one can set $\mu_i = z_i'/e'$, for $i = 1, \ldots, \ell$ and $\tau = t'/e'$, and $c = g_1^{\mu_1} \cdots g_\ell^{\mu_\ell} h^\tau \bmod n$ is a valid opening of c.

5.3 Equally Efficient Range Proofs from RSA

We show that Lipmaa's range proof [Lip03] also benefits from our technique as the $\mathsf{Strong}\text{-}\mathsf{RSA}$ assumption can also be avoided in the security analysis.

Range Proof from Integer Commitment Scheme. Let $c = g^x h^r \bmod n$ be a commitment of a value x and $[\![a\,;b]\!]$ be a public interval. As the commitment is homomorphic, one can efficiently compute a commitment c_a of $x - a$ and a commitment c_b of $b - x$ from c. To prove that $x \in [\![a\,;b]\!]$, this is enough to show that c_a and c_b commit to positive values. Let us focus on the proof that $c_a = g^{x-a} h^r \bmod n$ commits to a positive value, since the same method applies for c_b. To do so, the prover computes (x_1, x_2, x_3, x_4) such that $x - a = \sum_{i=1}^{4} x_i^2$. By a famous result from Lagrange, such a decomposition exists if and only if $x - a \geq 0$. Moreover, this decomposition can be efficiently computed by the Rabin-Shallit algorithm [RS86], for which Lipmaa [Lip03] also suggested some optimizations. The prover commits to (x_1, x_2, x_3, x_4) in (c_1, c_2, c_3, c_4), where $c_i = g^{x_i} h^{r_i} \bmod n$ for each $i = 1$ to 4. Now, the prover proves his knowledge of openings $x - a$, x_1, x_2, x_3, x_4 (along with random coins r, r_1, r_2, r_3, r_4) of c_a, c_1, c_2, c_3, c_4 satisfying $\sum_{i=1}^{4} x_i^2 = x - a$ over the integers.

The reason allowing to solely relies on the RSA assumption in the range proof comes from the fact that the first part of the argument reduces to an argument of knowledge of openings x_1, x_2, x_3, x_4 of c_1, c_2, c_3, c_4 while the remaining part simply ensures the relation $\sum_{i=1}^{4} x_i^2 = x - a$ to hold. Indeed, once the witnesses are extracted, this is implied by the representation $c_a = \prod_{i=1}^{4} c_i^{x_i} h^{r - \sum x_i r_i} \bmod n$ which can be seen as generalized commitment scheme with basis (c_1, c_2, c_3, c_4, h) from which the opening cannot change. Therefore, the argument can be seen as five parallel arguments of knowledge, the fifth one being an argument of knowledge for a generalized commitment, where the opening for the last argument is the vector of the openings for the other arguments. A formal proof of an optimized version of this protocol under the intractability of the RSA assumption is presented in Appendix A.

Extension. Since most of the arguments of knowledge of a solution to a system of equations over the integers [CCT07] can be split into parallel arguments of knowledge of affectations to the variables and a proof of membership (in the language composed of all the solutions of the system), which is expressed as representations corresponding to generalized commitments, our analysis extends to all “discrete-logarithm relation set” (see [KTY04]): the description of the protocol is unchanged but the security only relies on the standard RSA assumption.

6 Commitment with Knowledge-Delayed Order

Arguments of knowledge of openings for the Dåmgard-Fujisaki commitment scheme can rely on the RSA assumption rather than the $\mathsf{Strong\text{-}RSA}$ assumption. In this section, we show that this scheme enjoys in addition a very particular feature, which corresponds informally to the possibility to convert a commitment of an *integer* into a commitment of an *element of* $\mathbb{Z}_\pi$, for some prime π. We propose to perform zero-knowledge arguments over the integers which exploit this feature. Our method improves upon the classical method for zero-knowledge arguments over the integers on several aspects. We then illustrate our technique on the famous example of range proofs.

6.1 RSA-based Commitments with Known Order

We revisit a commitment scheme which, as far as we know, was proposed by Gennaro [Gen04] as a commitment for a message $m \in \mathbb{Z}_\pi$. The order of the commitment is a known prime $\pi > 2^\kappa$.

Description of the Generalized Commitment Scheme. Let us describe the commitment of vectors of integers $(m_1, \ldots, m_\ell)$:

- $\mathsf{Setup}(1^\kappa)$ runs $(n, (p, q)) \leftarrow_R \mathsf{GenMod}(1^\kappa)$, and picks ℓ random generators $g_1, \ldots, g_\ell$ of QR_n. Then, it picks a random prime $\pi \in [\![2^{\kappa+1}\,; 2^{\kappa+2}]\!]$, and returns $\mathsf{pp} = (n, g_1, \ldots, g_\ell, \pi)$;

- $\mathsf{Commit}(\mathsf{pp}, \boldsymbol{m}; r)$, for $\mathsf{pp} = (n, g_1, \dots, g_\ell, \pi)$, a vector $\boldsymbol{m} = (m_1, \dots, m_\ell) \in \mathbb{Z}_\pi^\ell$, and some random coins $r \leftarrow_R \mathbb{Z}_n$, computes $c = g_1^{m_1} \cdots g_\ell^{m_\ell} r^\pi \bmod n$, and returns (c, d) with $d = r$;
- $\mathsf{Verify}(\mathsf{pp}, c, d, \boldsymbol{m})$ parses pp as $\mathsf{pp} = (n, g_1, \dots, g_\ell, \pi)$ and outputs 1 if $c = g_1^{m_1} \cdots g_\ell^{m_\ell} r^\pi \bmod n$, and 0 otherwise.

The above commitment scheme is obviously *correct*. The *hiding* property relies on the bijectivity of the π-th power modulo n (as π is prime): for any message $\boldsymbol{m'} = (m'_1, \dots, m'_\ell) \in \mathbb{Z}_\pi^\ell$, we have $c = g_1^{m'_1} \cdots g_\ell^{m'_\ell} \times g_1^{m_1 - m'_i} \cdots g_\ell^{m_\ell - m'_i} \times r^\pi \bmod n$. By noting s the π-th root of $g_1^{m_1 - m'_i} \cdots g_\ell^{m_\ell - m'_i}$, $c = \mathsf{Commit}(\mathsf{pp}, \boldsymbol{m'}; rs)$. The binding property uses an extension of the Fact 5 from Proposition 1: if one has chosen β_i such that $g_i = u^{2\beta_i}$, for a challenge RSA $u \in \mathbb{Z}_n^*$, two distinct openings $(\boldsymbol{m}, r) \neq (\boldsymbol{m'}, s)$ satisfy $g_1^{m_1} \cdots g_\ell^{m_\ell} r^\pi = g_1^{m'_1} \cdots g_\ell^{m'_\ell} s^\pi \bmod n$, and so $(s/r)^\pi = u^{2a} \bmod n$, where $a = \sum \beta_i (m_i - m'_i) = a_1 \pi + a_0$, with $0 \leq a_0 < \pi$. Let us note α and β the integers such that $\alpha\pi + \beta 2 a_0 = \gcd(\pi, 2a_0) = 1$, and output $u_0 := u^{\alpha - 2a_1\beta} \cdot (s/r)^\beta \bmod n$, then

$$u_0^\pi = u^{\alpha\pi - 2a_1\beta\pi} \cdot (s/r)^{\beta\pi} = u^{1 - 2(a_0 + a_1\pi)\beta} \cdot u^{2a\pi} = u \bmod n.$$

This breaks the RSA assumption with exponent π.

Homomorphic-Opening. In addition, this commitment scheme is homomorphic in $\mathbb{Z}_\pi$: given $c = g_1^{m_1} \cdots g_\ell^{m_\ell} r^\pi \bmod n$ and $d = g_1^{m'_1} \cdots g_\ell^{m'_\ell} s^\pi \bmod n$ with known openings, we can efficiently open the commitment $c \cdot d \bmod n$ to $\bar{\boldsymbol{m}} = (\bar{m}_1, \dots, \bar{m}_\ell)$, with $\bar{m}_i = m_i + m'_i \bmod \pi$ for $1 \leq i \leq \ell$, and a random coin $rs \prod g_i^{(m_i + m'_i) \div \pi} \bmod n$, where $a \div b$ is the quotient of the Euclidean division. We emphasize this property to be essential not to work with integers in the arguments of knowledge of an opening: the prover can "reduce" its openings since π is known.

Argument of Opening. Given $\mathsf{pp} = (n, g_1, \dots, g_\ell, \pi)$ and $c = g_1^{x_1} \cdots g_\ell^{x_\ell} r^\pi \bmod n$, with witness $(x_1, \dots, x_\ell, r)$, we can describe a standard argument of knowledge of an opening:

Initialize: $\mathscr{P}$ and $\mathscr{V}$ decide to run the protocol on input (pp, κ, c);
Commit: $\mathscr{P}$ computes $d = g_1^{y_1} \cdots g_\ell^{y'_\ell} s^\pi$, for $y_i \leftarrow_R \mathbb{Z}_\pi$, and $s \leftarrow_R \mathbb{Z}_n^*$, and sends d to $\mathscr{V}$;
Challenge: $\mathscr{V}$ outputs $e \leftarrow_R [\![0\,;2^\kappa]\!]$;
Response: $\mathscr{P}$ computes k_i, z_i, t such that $ex_i + y_i = k_i\pi + z_i$, with $0 \leq z_i < \pi$, and $t = g_1^{k_1} \cdots g_\ell^{k_\ell} \cdot r^e s \bmod n$. $\mathscr{P}$ outputs $(z = (z_i)_i, t)$;
Verify: $\mathscr{V}$ accepts the proof and outputs 1 if, for each i, $0 \leq z_i < \pi$, and $c^e d = g_1^{z_1} \cdots g_\ell^{z_\ell} t^\pi \bmod n$. Otherwise, $\mathscr{V}$ rejects the proof and outputs 0.

Completeness and *zero-knowledge* are straightforwards. Then, let us focus on the *knowledge-extractability*: From two related valid transcripts, for the same d, we get as usual $c^{e-e'} = g_1^{z_1 - z'_1} \cdots g_\ell^{z_\ell - z'_\ell} \cdot (t/t')^\pi \bmod n$. Since the prime $\pi > 2^\kappa \geq ||e - e'||$, the simulator can compute $\alpha(e - e') + \beta\pi = 1$ and we have

$$c^{1 - \beta\pi} = c^{\alpha(e - e')} = g_1^{\alpha(z_1 - z'_1)} \cdots g_\ell^{\alpha(z_\ell - z'_\ell)} \cdot (t/t')^{\alpha\pi} \bmod n.$$

Then, for $\alpha(z_i - z'_i) = l_i\pi + x'_i$ with $0 \leq x'_i < \pi$, and $t' = c^\beta \cdot g_1^{l_1} \cdots g_\ell^{l_\ell} \cdot (t/t')^\alpha \bmod n$, we have a valid opening $(x'_1, \dots, x'_\ell, t')$ of c.

Size-Independent. The size of our committed values is independent to the order of the commitment itself: even if we take a quite large π, the commitment remains in $\mathbb{Z}_n$. As far as we know, no Pedersen-like commitments of known prime order enjoyed this feature: the cardinality of the cyclic groups always bounded the order of the commitment.

6.2 Commitment with Knowledge-Delayed Order

Now, we show how we can hide the above commitment scheme with known prime order π into a commitment scheme of Section 3 with hidden order.

Description of the Commitment Scheme. As explained earlier, the setup could have been run by the verifier, with an additional proof of existence of α, such that $g = h^\alpha \bmod n$, to guarantee the hiding property. In this protocol, the verifier runs the setup:

- $\mathsf{Setup}(1^\kappa)$ runs $(n, (p, q)) \leftarrow_R \mathsf{GenMod}(1^\kappa)$, and picks $h_0 \leftarrow_R \mathsf{QR}_n$ and a random prime $\pi \in [\![2^{\kappa+1} ; 2^{\kappa+2}]\!]$. Then, it picks $\rho \leftarrow_R [\![0 ; n^2]\!]_\pi$ and sets $g \leftarrow h_0^\rho \bmod n$ and $h \leftarrow h_0^\pi \bmod n$. Finally, it returns $\mathsf{pp} = (n, g, h)$ and keeps $\mathsf{sk} = (\pi, h_0)$. Actually, we have $h^\rho = g^\pi \bmod n$. So, if one sets $\alpha = \rho \cdot \pi^{-1} \bmod \varphi(n)$, one has $g = h^\alpha \bmod n$, and proves it;
- $\mathsf{Commit}(\mathsf{pp}, m; r)$ parses pp as above and commits to $m \in \mathbb{Z}$ by picking $r \leftarrow_R \mathbb{Z}_n$ and computing $c = g^m h^r \bmod n$. It returns (c, r);
- $\mathsf{Verify}(\mathsf{pp}, c, m, r)$ parses $\mathsf{pp} = (n, g, h)$ and outputs 1 if $c = \pm g^m h^r \bmod n$ and 0 otherwise;
- $\mathsf{Reveal}(\mathsf{pp}, \mathsf{sk})$ returns $\mathsf{sk} = (\pi, h_0)$;
- $\mathsf{Adapt}(\mathsf{pp}, \mathsf{sk}, c, m, r)$ first parses $\mathsf{sk} = (\pi, h_0)$ and checks whether $h = h_0^\pi \bmod n$. Then, it adapts the opening by computing $m = k\pi + \bar{m}$ for $0 \leq \bar{m} < \pi$ and $t = g^k h_0^r \bmod n$. It outputs $(\bar{m}, t)$;
- $\mathsf{Verify}'(\mathsf{pp}, \pi, c, \bar{m}, t)$ outputs 1 if $c = g^{\bar{m}} t^\pi \bmod n$, and 0 otherwise.

This construction easily extends to commitments of vectors. Note that from $g^{\bar{m}} t^\pi = c = g^{\bar{m}'} t'^\pi \bmod n$, with $\bar{m} \neq \bar{m}' \bmod \pi$, setting $h_0 = y^2$ from an RSA challenge (n, y) of exponent $\pi > 2^\kappa$, we obtain $y^{2\rho(\bar{m} - \bar{m}')} = (t'/t)^\pi \bmod n$, with $2\rho(\bar{m} - \bar{m}') \neq 0 \bmod \pi$, which leads to the π-th root of y modulo n (using Fact 5 from Proposition 1).

Switching between Commitments. Our goal is to use the more efficient commitment scheme given in Section 6.1, that we denote com_π, and also the associated proofs of relations in $\mathbb{Z}_\pi$: in the case of a single integer $m \in \mathbb{Z}_\pi$, $\mathsf{com}_\pi(m; r) = g^m r^\pi \bmod n$, for $r \leftarrow_R \mathbb{Z}_n^*$. But let $\mathscr{V}$ run the setup from Section 6.2, which outputs $\mathsf{pp} = (n, g, h)$ (while keeping $\mathsf{sk} = (\pi, h_0)$), as in Section 3: this reveals no information about π. Now, $\mathscr{P}$ can use (n, g, h) for the Damgård-Fujisaki integer commitment scheme that we denote com: for an integer $m \in \mathbb{Z}$ and $r \leftarrow_R \mathbb{Z}_n$, $c = \mathsf{com}(m; r) = g^m h^r \bmod n$. After some time, $\mathscr{V}$ reveals (π, h_0), which allows $\mathscr{P}$ to open c as a *commitment over* $\mathbb{Z}_\pi$ of $\mathfrak{r}_\pi(m) = m \bmod \pi$:

$$\mathsf{com}(m; r) = \mathsf{com}_\pi(\mathfrak{r}_\pi(m); g^{\mathfrak{q}_\pi(m)} h_0^r), \tag{1}$$

where $\mathfrak{q}_\pi(m)$ and $\mathfrak{r}_\pi(m)$ indeed denote the quotient and remainder of the euclidean division of m by π. This then allows to use efficient proofs on com_π, but still with good properties on the integers, since the prover did not know π at the commit time.

6.3 Improving Zero-Knowledge Arguments over the Integers

In this section, we thus introduce our new technique to build zero-knowledge arguments for statements over the integers, while using com_π. We restrict our attention to statements that can be expressed as membership to a set $S \in \mathbf{D}$. Our technique allows us to provide more efficient membership arguments, with a lower communication and a smaller verifier work (applying the technique *delegates* some of the work of the verifier to the prover). The core component of our technique is the commitment com that we can later switch to com_π, in which the order of the message space is revealed *after* the prover has committed to values. We call such commitment a *commitment with knowledge-delayed order.*

Membership Argument for D. Let us consider a set $S \in \mathbf{D}$ with representing polynomial P_S with k-vector input and ℓ-vector witness. We assume that $\mathscr{P}$ and $\mathscr{V}$ have agreed on a bound t such that each $\boldsymbol{x} \in S$ has a witness $\boldsymbol{w}$ of size $||\boldsymbol{w}||_1 \leq (||\boldsymbol{x}||_1)^t$ ($S \in \mathbf{D}$, so there is always such a t. As shown in [Lip03], $t < 2$ is sufficient for most cryptographic applications).

Let $\boldsymbol{x}$ be a secret vector held by $\mathscr{P}$, and $\boldsymbol{w}$ be a *witness* for $\boldsymbol{x} \in S$ (i.e., a vector satisfying $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0$). It is known that zero-knowledge arguments can be constructed for polynomial relations over committed inputs (see e.g. [BS02]). Intuitively, this is done by committing to intermediate values, and proving additive and multiplicative relationships between those values and the inputs. To prove a multiplicative relationship $z = xy$ between values (x, y, z) committed in (c_x, c_y, c_z), $\mathscr{P}$ proves his knowledge of inputs (x, y, z) and random coins (r_x, r_y, r_z) such that $c_x = g^x r_x^\pi \bmod n$, $c_y = g^y r_y^\pi \bmod n$, and $c_z = c_x^y r_z^\pi$. Let us now consider the following situation, where commitments are applied component-wise:

1. $\mathscr{P}$ picks random coins $(\boldsymbol{r_x}, \boldsymbol{r_w})$ and commits to $(\boldsymbol{x}, \boldsymbol{w})$ with $(\boldsymbol{r_x}, \boldsymbol{r_w})$ as $(\boldsymbol{c_x}, \boldsymbol{c_w}) \leftarrow (\mathsf{com}_\pi(\boldsymbol{x}; \boldsymbol{r_x}), \mathsf{com}_\pi(\boldsymbol{w}; \boldsymbol{r_w}))$;
2. $\mathscr{P}$ performs a zero-knowledge argument with $\mathscr{V}$ to prove his knowledge of four vectors $(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{r_x}, \boldsymbol{r_w})$ such that $(\boldsymbol{c_x}, \boldsymbol{c_w}) = (\mathsf{com}_\pi(\boldsymbol{x}; \boldsymbol{r_x}), \mathsf{com}_\pi(\boldsymbol{w}; \boldsymbol{r_w}))$ and $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0 \bmod \pi$.

As com_π is a commitment scheme over $\mathbb{Z}_\pi$, this protocol is an argument of knowledge of $(\boldsymbol{x}, \boldsymbol{w})$ such that $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0 \bmod \pi$. But it does, by no mean, prove the knowledge of *integers* belonging to the Diophantine set S. However, our main observation is that com_π can also be seen as an *integer commitment scheme* (the commitment scheme we denoted com).

Argument of knowledge of the inputs and witnesses.

1. $\mathscr{V}$ runs the setup from the Section 6.2, which generates $\mathsf{pp} = (n, g, h)$ and $\mathsf{sk} = (\pi, h_0)$: this defines $\mathsf{com} : (x; r) \mapsto g^x h^r \bmod n$. It additionally proves the existence of α such that $g = h^\alpha \bmod n$;
2. $\mathscr{P}$ picks random coins $(\boldsymbol{r_x}, \boldsymbol{r_w})$ and commits to $(\boldsymbol{x}, \boldsymbol{w})$ with $(\boldsymbol{r_x}, \boldsymbol{r_w})$ as $(\boldsymbol{c_x}, \boldsymbol{c_w}) \leftarrow (\mathsf{com}(\boldsymbol{x}; \boldsymbol{r_x}), \mathsf{com}(\boldsymbol{w}; \boldsymbol{r_w}))$;
3. $\mathscr{P}$ performs a $\mathsf{ZKAoK}\{(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{r_x}, \boldsymbol{r_w}) \mid \boldsymbol{c_x} = g^{\boldsymbol{x}} h^{\boldsymbol{r_x}} \wedge \boldsymbol{c_w} = g^{\boldsymbol{w}} h^{\boldsymbol{r_w}}\}$, we thereafter refer to ZK_1, with $\mathscr{V}$. If the argument fails, $\mathscr{V}$ aborts the protocol.

Argument of knowledge of $(\boldsymbol{x'}, \boldsymbol{w'})$ *such that* $P_S(\boldsymbol{x'}, \boldsymbol{w'}) = 0 \bmod \pi$.

1. $\mathscr{V}$ reveals (π, h_0) to $\mathscr{P}$ who checks whether $h = h_0^\pi \bmod n$ or not, to switch to com_π. Let $(\boldsymbol{x}', \boldsymbol{w}') = (\mathfrak{r}_\pi(\boldsymbol{x}), \mathfrak{r}_\pi(\boldsymbol{w})) = (\boldsymbol{x}, \boldsymbol{w}) \bmod \pi$.
2. $\mathscr{P}$ performs a $\mathsf{ZKAoK}\{(\boldsymbol{x}', \boldsymbol{w}', \boldsymbol{R_x}, \boldsymbol{R_w})\}$, we thereafter refer to ZK_2, such that $(\boldsymbol{c_x}, \boldsymbol{c_w}) = (\mathsf{com}_\pi(\boldsymbol{x}; \boldsymbol{R_x}), \mathsf{com}_\pi(\boldsymbol{w}; \boldsymbol{R_w}))$ and $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0 \bmod \pi$. Note that $(\boldsymbol{c_x}, \boldsymbol{c_w})$ are now seen as commitments over $\mathbb{Z}_\pi$, using the fact that $\mathsf{com}(\boldsymbol{x}; \boldsymbol{r_x}) = \mathsf{com}_\pi(\mathfrak{r}_\pi(\boldsymbol{x}); \boldsymbol{R_x})$ and $\mathsf{com}(\boldsymbol{w}; \boldsymbol{r_w}) = \mathsf{com}_\pi(\mathfrak{r}_\pi(\boldsymbol{w}); \boldsymbol{R_w})$, with appropriate $(\boldsymbol{R_x}, \boldsymbol{R_w})$. If the argument succeeds, $\mathscr{V}$ returns accept.

Theorem 4. *Under the* RSA *assumption, the above protocol is a statistical zero-knowledge argument of knowledge of openings of* $(\boldsymbol{c_x}, \boldsymbol{c_w})$ *to vectors of integers* $(\boldsymbol{x}, \boldsymbol{w})$ *such that* $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0$*: which proves that* $\boldsymbol{x} \in S$*.*

Proof. The intuition behind Theorem 4 is that ZK_1 proves that $\mathscr{P}$ knows $(\boldsymbol{x}, \boldsymbol{w})$ in $(\boldsymbol{c_x}, \boldsymbol{c_w})$, and ZK_2 proves that $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0 \bmod \pi$ for a κ-bit prime π which was revealed *after* $(\boldsymbol{x}, \boldsymbol{w})$ *were committed.* Hence, $\mathscr{P}$ knew vectors of integer $(\boldsymbol{x}, \boldsymbol{w})$ such that $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0 \bmod \pi$ for a random κ-bit prime π. This has a negligible probability to happen unless $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0$ holds over the integers, since P_S is a polynomial. The full proof consists of the three properties: correctness, zero-knowledge, and knowledge-extractability.

Correctness. It easily follows from the correctness of ZK_1 and ZK_2: if $\mathscr{P}$ knows $(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{r_x}, \boldsymbol{r_w})$ such that $(\boldsymbol{c_x}, \boldsymbol{c_w}) = (\mathsf{com}(\boldsymbol{x}; \boldsymbol{r_x}), \mathsf{com}(\boldsymbol{w}; \boldsymbol{r_w}))$ and $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0$, then the argument of knowledge of $(\boldsymbol{x}, \boldsymbol{r_x})$ such that $\boldsymbol{c_x} = \mathsf{com}(\boldsymbol{x}; \boldsymbol{r_x})$ will succeed, and it holds that $(\boldsymbol{c_x}, \boldsymbol{c_w}) = (\mathsf{com}_\pi(\boldsymbol{x} \bmod \pi; v^{\mathfrak{q}_\pi(\boldsymbol{x})} \tilde{h}^{\boldsymbol{r_x}}), \mathsf{com}_\pi(\boldsymbol{w} \bmod \pi; v^{\mathfrak{q}_\pi(\boldsymbol{x})} \tilde{h}^{\boldsymbol{r_x}}))$. Moreover, as P_S is a polynomial, the modular reduction applies, and leads to $P_S(\boldsymbol{x} \bmod \pi, \boldsymbol{w} \bmod \pi) = P_S(\boldsymbol{x}, \boldsymbol{w}) = 0 \bmod \pi$.

Zero-Knowledge. It also follows from the zero-knowledge of ZK_1 and ZK_2, and the hiding property of the commitments. Let $\mathscr{S}im_{\mathsf{ZK}}$ be the following simulator: one first generates dummy commitments $(\boldsymbol{c_x}, \boldsymbol{c_w})$, which does not make any difference under the hiding perperty, and runs the simulator of ZK_1. Once (π, h_0) is revealed, $\mathscr{S}im_{\mathsf{ZK}}$ runs the simulator of ZK_2.

Since the commitment is statistically hiding, ZK_1 is our statistically zero-knowledge argument of knowledge of opening from Section 3 and ZK_2 is an argument of relations on commitments with known order π (since $h = h_0^\pi \bmod n$) that is possible in statistical zero-knowledge, the full protocol is statistically zero-knowledge.

Knowledge Extractability. Consider a $\mathscr{P}$' which succeeds in providing a convincing argument with probability ε, which means that the two protocols ZK_1 and ZK_2 succeed with probability greater than ε.

We first use the extractor of ZK_1 to extract the inputs-witnesses and random coins $(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{r_x}, \boldsymbol{r_w})$ such that $\boldsymbol{c_x} = g^{\boldsymbol{x}} h^{\boldsymbol{r_x}}$ and $\boldsymbol{c_w} = g^{\boldsymbol{w}} h^{\boldsymbol{r_w}}$. This extraction is successful under the RSA assumption.

Then, (π, h_0) is revealed and we use the extractor of ZK_2 to extract the inputs-witnesses and random coins $(\boldsymbol{x}', \boldsymbol{w}', \boldsymbol{R_x}, \boldsymbol{R_w})$ such that both relations $(\boldsymbol{c_x}, \boldsymbol{c_w}) = (\mathsf{com}_\pi(\boldsymbol{x}'; \boldsymbol{R_x}), \mathsf{com}_\pi(\boldsymbol{w}'; \boldsymbol{R_w}))$ and $P_S(\boldsymbol{x}', \boldsymbol{w}') = 0 \bmod \pi$ are satisfied. Again, this extraction is successful under the RSA assumption.

Now, let us consider two situations:

- If $\boldsymbol{x}' = \boldsymbol{x} \bmod \pi$ and $\boldsymbol{w}' = \boldsymbol{w} \bmod \pi$, then the value committed over the integers, *before π was revealed*, satisfy $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0 \bmod \pi$, for a random $\pi \in [\![2^{\kappa+1}\,; 2^{\kappa+2}]\!]$. We stress that the view of (n, g, h) does not reveal any information on the prime π.
  Since there are approximately $2^{\kappa+1}/\kappa$ primes in this set, and this extraction works with probability greater than ε^2, $P_S(\boldsymbol{x}, \boldsymbol{w}) = 0 \bmod Q$, for $Q \geq 2^{2^\kappa/\varepsilon^2}$, which is much larger than the values that can be taken in the integers, since the inputs and the witnesses have a size polynomial in κ, and the polynomial P_S has a bounded degree.
- If $\boldsymbol{x}' \neq \boldsymbol{x} \bmod \pi$ or $\boldsymbol{w}' \neq \boldsymbol{w} \bmod \pi$, wlog, we can assume that $\boldsymbol{x}' \neq \boldsymbol{x} \bmod \pi$: one knows
  - $(\boldsymbol{x}, \boldsymbol{r_x})$ such that (1) $\boldsymbol{c_x} = \pm g^{\boldsymbol{x}} h^{\boldsymbol{r_x}} = g^{\mathfrak{r}_\pi(\boldsymbol{x})} (\pm g^{\mathfrak{q}_\pi(\boldsymbol{x})} h_0^{\boldsymbol{r_x}})^\pi \bmod n$;
  - and $(\boldsymbol{x}', \boldsymbol{R_x})$ such that (2) $\boldsymbol{c_x} = g^{\boldsymbol{x}'} \boldsymbol{R_x}^\pi \bmod n$.

  Hence, $g^{\mathfrak{r}_\pi(\boldsymbol{x})} (\pm g^{\mathfrak{q}_\pi(\boldsymbol{x})} h_0^{\boldsymbol{r_x}})^\pi = g^{\boldsymbol{x}'} \boldsymbol{R_x}^\pi \bmod n$, and so $g^{\mathfrak{r}_\pi(\boldsymbol{x}) - \boldsymbol{x}'} = S^\pi \bmod n$, for $S = \boldsymbol{R_x} / (\pm g^{\mathfrak{q}_\pi(\boldsymbol{x})} h_0^{\boldsymbol{r_x}}) \bmod n$. If one would have set $h_0 = y^2$ from an RSA challenge (n, y, π) of exponent $\pi > 2^\kappa$, and thus $g = y^{2\rho}$, using Fact 5 from Proposition 1, one gets the π-th root of y modulo n.

This concludes the proof of the knowledge-extractability of the protocol, under the RSA assumption over $\mathbb{Z}_n$. □

On the Efficiency of the Method. The advantages of this method compared to the classical method are twofold. First, most of the work in the protocol comes from the computation of exponentiations; with our technique, most of the work is transfered from $\mathscr{V}$ to $\mathscr{P}$. This comes from the fact that verifying an equation such as $\boldsymbol{c} = \mathsf{com}(x; r)$ involves exponentiations by integers of size $O(\log n + \kappa)$ while verifying the equation $\boldsymbol{c} = \mathsf{com}_\pi(x \bmod \pi; R)$ involves only two exponentiations by κ-bit values, so the work of $\mathscr{V}$ is reduced. However, $\mathscr{P}$ will have to compute exponentiations by integers of size $O(\log n + \kappa)$ to construct the random coin R associated to the commitment mod π

(using the identity 1 in Section 6.2). $\mathscr{V}$ will still need to perform exponentiations by integers during ZK_1, but his work during this step can be made essentially independent of the number N of inputs and witnesses (up to a small $\log N$ additive term) and completely independent of the degree of the representing polynomial.

Second, our method separates the argument of *knowledge* of inputs to a Diophantine equation from the argument that they do indeed satisfy the equation. The arguments of knowledge of an opening of a commitment can be very efficiently batched: if $\mathscr{P}$ commits to $(x_1, \cdots, x_N)$ with random coins $(r_1, \cdots, r_N)$ as $(c_1, \cdots, c_N)$, the verifier can simply send a random seed $\lambda \leftarrow_R \{0,1\}^\kappa$ from which both players compute $(\lambda_1, \cdots, \lambda_N)$ using a pseudo-random generator[3]. Then, $\mathscr{P}$ performs a *single* argument of knowledge of an opening $(\sum_i \lambda_i x_i; \sum_i \lambda_i r_i)$ of the commitment $\prod_i c_i^{\lambda_i}$ (see [BGR98a, BGR98b] for more details). Therefore, when performing multiple membership arguments, $\mathscr{P}$ and $\mathscr{V}$ will have to perform a single argument for ZK_1 (of size essentially independent of the number of committed values).

In general, the higher the degree of the representing polynomial, the better our method will perform (in terms of communication). Still, we show in the following section that even for the case of range proofs, which can be seen as membership proofs to a Diophantine set whose representing polynomial is of degree 2, our method provides efficiency improvements.

Further Improvements. $\mathscr{V}$ can set h to $h_0^{\prod_i \pi_i}$ for several primes π_i instead of h^π. For some integer i, let $p_i \leftarrow \prod_{j \neq i} \pi_j$. Doing so allows $\mathscr{V}$ to reveal $(h_0^{p_i}, \pi_i)$ instead of (h_0, π) in our method. Hence, in addition to allowing arbitrary parrallel arguments with a single prime π, a single setting is sufficient to perform a polynomial number of sequential arguments (fixed in advance) with different primes π_i. In addition, we explained that commitments with knowledge-delayed order allow splitting the arguments of knowledge of the witnesses, denoted ZK_1, and the argument that they indeed belong to a Diophantine set, denoted ZK_2. The arguments ZK_1 can be batched as described above but, for efficiency reason, we should not generate $(\lambda_1, \lambda_2 \ldots, \lambda_N)$ as $(\lambda, \lambda^2, \ldots, \lambda^N)$. Indeed, $|\lambda^j|$ growth linearly with j over the integers. However, for the argument ZK_2, the order of the commitment has been revealed. Hence, we can now do use batch technique with such $\lambda_j = \lambda^j$ since the prover is able to reduce the exponents modulo π at this stage. That means that our technique consisting of efficiently revealing the order of the commitment between ZK_1 and ZK_2 allows to use any tricks that were only available for discrete-log based proofs of statement over (pairing-free) known-order groups. For instance, we can get a sub-linear size argument to show that a committed matrix is the Hadamard products *over the integers* of two other committed matrices. Indeed, we can commit the rows of the matrices using a generalized commitment and make a batch proof for ZK_1, which remain sub-linear in the number of entrees, and then we can import the results of [Gro09, BG12] to ZK_2, preserving its sub-linearity.

Full-Fledge Zero-Knowledge. With an honest verifier, there is no need to prove the existence of α such that $g = h^\alpha$. In the malicious setting, this proof guarantees the hiding property of the commitments to the prover, who additionally checks $h = h_0^\pi \bmod n$ when they are revealed. Then we can use classical techniques to make the HVZK protocol to ZK, such as an equivocable commitment of the challenge by the verifier, before the commitments from the prover.

[3] The classical trick that consists of using $\lambda_i = \lambda^i$ is not efficient here since we are in the integers, and so no reduction can be applied.

7 Application to Range Proofs

7.1 Lipmaa's Compact Argument for Positivity

As explained before, Lipmaa [Lip03] proposed an efficient argument for positivity, using generalized Damgåard-Fujisaki commitments, and the proof that an integer is positive if and only if it can be written as the sum of four squares. However, it appears that the explicit construction given in [Lip03, annex B] is flawed — although the high-level description is correct: any prover can provide a convincing argument for positivity, regardless of the sign of the committed integer, and so without holding valid witnesses.

This might raise some concerns as the protocol of Lipmaa is the "textbook" range proof based on hidden order groups. Hence the protocol is suggested in several papers, and was implemented in e.g. [AMAR05]. In Appendix A, we recall the argument of [Lip03], identify its flaw, and provide a correct optimized version together with a full proof of security.

In the following, we describe a range proof in the same vein as the positivity argument of Lipmaa: an integer x belongs to an interval $[\![a\,;b]\!]$ if and only if $(x-a)(b-x) \geq 0$. In addition, we take into account the following improvement suggested by Groth [Gro05]: x is positive if and only if $4x+1$ can be written as the sum of three squares, and such a decomposition can be computed in polynomial time by the prover. We view this range proof as an optimized version of the textbook range proof with integer commitments, to which we will compare our new method with knowledge-delayed order commitments.

7.2 Three-Square Range Proof

To prove that $x \in [\![a\,;b]\!]$, for x committed with an integer commitment scheme, we prove that $4(x-a)(b-x)+1$ can be written as the sum of three squares. Let (n, g, h) be the public parameters of the Damgård-Fujisaki commitment scheme, generated by the verifier. The three-square range proof ($\mathsf{3SRP}$) is described in full details on Figure 1. Basically, both $\mathscr{P}$ and $\mathscr{V}$ know that c_a contains $4(x-a)$ and c_0 contains $(b-x)$. The latter, with c_1, c_2, c_3 containing respectively x_1, x_2, x_3, is proven in a classical way, and the last part of the proof shows that $c_a^{x_0} g$, which implicitly contains $4(x-a)(b-x)+1$ also contains $x_1^2 + x_2^2 + x_3^2$.

We then illustrate the technique introduced in Section 6.3 on this $\mathsf{3SRP}$ protocol. The full converted protocol, denoted $\mathsf{3SRP\text{-}KDO}$, is described on Figure 2. We also combine in parallel the two arguments: in the integers with a random combination using $(\lambda_i)_i$ and in $\mathbb{Z}_\pi$.

7.3 Results

Let $B = \log(b-a)$. Note that for all $i \in \{0, 1, 2, 3\}$, $x_i^2 \leq (b-a)^2$ hence $\log x_i \leq B$. An exponentiation by a t-bit value takes $1.5t$ multiplications using a square-and-multiply algorithm; we do not take into account possible optimizations from multi-exponentiation algorithms. Table 1 sums up the communication complexity and the computational complexity of both the $\mathsf{3SRP}$ and the $\mathsf{3SRP\text{-}KDO}$ arguments for the execution of N parallel range proofs on the same interval $[\![a\,;b]\!]$, as classical batch techniques [BGR98a, BGR98b] allow to batch arguments of knowledge.

Note that we omit constant terms. The communication is given in bits, while the work is given as a number of multiplications of elements of QR_n. When comparing the work of the prover, we also omit the cost of the decomposition in sum of squares, as it is the same in both protocols. Similarly, we omit the cost of the initial proof of $g = h^\alpha \bmod n$ by the verifier to the prover.

For $\mathsf{pp} = (n, g, h)$ generated by $\mathscr{V}$, $\mathscr{P}$ has sent c, for which he knows (x, r) such that $c = g^x h^r \bmod n$ and $x \in [\![a\,;b]\!]$. Let $H : \mathbb{Z}_n^5 \mapsto \{0,1\}^{2\kappa}$ be a collision-resistant hash function. $\mathscr{V}$ compute $c_a = (cg^{-a})^4 \bmod n$ and $c_0 = c^{-1} g^b \bmod n$; $\mathscr{P}$ computes c_a.

1. $\mathscr{P}$ computes $(x_i)_{1 \le i \le 3}$ such that $4(b-x)(x-a)+1 = \sum_{i=1}^3 x_i^2$. $\mathscr{P}$ commits to $(x_i)_{1 \le i \le 3}$ with random coins $(r_i)_{1 \le i \le 3} \leftarrow_R [\![0\,;n]\!]^3$ as $(c_i = g^{x_i} h^{r_i} \bmod n)_{1 \le i \le 3}$. Let $x_0 \leftarrow (b-x)$ and $r_0 \leftarrow r$.
2. $\mathscr{P}$ picks $(m_0, \cdots, m_3) \leftarrow_R [\![0\,;2^{B+2\kappa}]\!]^4$, $(s_0, \cdots, s_3) \leftarrow_R [\![0\,;2^{2\kappa} n]\!]^4$, $\sigma \leftarrow_R [\![0\,;2^{B+2\kappa} n]\!]$, and sends $\Delta = H((g^{m_i} h^{s_i} \bmod n)_{0 \le i \le 3}, h^\sigma c_a^{m_0} \prod_{i=1}^3 c_i^{-m_i} \bmod n)$.
3. $\mathscr{V}$ picks a challenge $e \leftarrow_R [\![0\,;2^\kappa]\!]$ and sends it to $\mathscr{V}$.
4. $\mathscr{P}$ computes and sends $z_i = ex_i + m_i$ and $t_i = er_i + s_i$ for $i \in \{0,1,2,3\}$, and $\tau = \sigma + e(x_0 r_0 - \sum_{i=1}^3 x_i r_i)$.
5. $\mathscr{V}$ accepts the argument if

$$\Delta = H\left((g^{z_i} h^{t_i} c_i^{-e} \bmod n)_{0 \le i \le 3}, h^\tau g^e c_a^{z_0} (\prod_{i=1}^3 c_i^{-z_i}) \bmod n \right).$$

Fig. 1: Three-Square Range Proof (3SRP)

For $\mathsf{pp} = (n, g, h)$ and $\mathsf{sk} = (\pi, h_0)$ generated by $\mathscr{V}$, $\mathscr{P}$ has sent c, for which he knows (x, r) such that $c = g^x h^r \bmod n$ and $x \in [\![a\,;b]\!]$. Let $H : \mathbb{Z}_n^6 \mapsto \{0,1\}^{2\kappa}$ be a collision-resistant hash function. $\mathscr{V}$ compute $c_a = (cg^{-a})^4 \bmod n$ and $c_0 = c^{-1} g^b \bmod n$; $\mathscr{P}$ computes c_a.

1. $\mathscr{P}$ computes $(x_i)_{1 \le i \le 3}$ such that $4(b-x)(x-a)+1 = \sum_{i=1}^3 x_i^2$. $\mathscr{P}$ commits to $(x_i)_{1 \le i \le 3}$ with random coins $(r_i)_{1 \le i \le 3} \leftarrow_R [\![0\,;n]\!]^3$ as $(c_i = g^{x_i} h^{r_i} \bmod n)_{1 \le i \le 3}$. Let $x_0 \leftarrow (b-x)$ and $r_0 \leftarrow r$.
2. $\mathscr{P}$ picks $m \leftarrow_R [\![0\,;2^{B+3\kappa}]\!]$, $(m_0, \cdots, m_3) \leftarrow_R [\![0\,;2^\kappa]\!]^4$, $s \leftarrow_R [\![0\,;2^{3\kappa} n]\!]$, $(s_0, \cdots, s_3) \leftarrow_R [\![0\,;n]\!]^4$, $\sigma \leftarrow_R [\![0\,;2^{B+2\kappa} n]\!]$, and sends $\Delta = H(g^m h^s \bmod n, (g^{m_i} h^{s_i} \bmod n)_{0 \le i \le 3}, h^\sigma c_a^{m_0} \prod_{i=1}^3 c_i^{-m_i} \bmod n)$.
3. $\mathscr{V}$ picks a challenge $e' \leftarrow_R [\![0\,;2^\kappa]\!]$ and sends (e', π, h_0) to $\mathscr{P}$.
4. $\mathscr{P}$ extends the challenge e' into $(e, (\lambda_i)_{0 \le i \le 3}) \in [\![0\,;2^\kappa]\!]^5$, computes and sends $z = e \sum \lambda_i x_i + m$ and $t = e \sum \lambda_i r_i + s$, as well as $z_i = \mathfrak{r}_\pi(ex_i + m_i)$ and $T_i = h_0^{er_i + s_i} g^{\mathfrak{q}_\pi(ex_i + m_i)} \bmod n$ for $i \in \{0,1,2,3\}$, and $T = h_0^{\sigma + e(x_0 r_0 - \sum_{i=1}^3 x_i r_i)} c_a^{\mathfrak{q}_\pi(ex_0 + m_0)} \prod_{i=1}^3 c_i^{-\mathfrak{q}_\pi(ex_i + m_i)} \bmod n$.
5. $\mathscr{V}$ accepts the argument if

$$\Delta = H\left(g^z h^t (\prod_{i=0}^3 c_i^{\lambda_i})^{-e} \bmod n, (g^{z_i} T_i^\pi c_i^{-e} \bmod n)_{i=0}^3, T^\pi g^e c_a^{z_0} (\prod_{i=1}^3 c_i^{-z_i}) \bmod n \right)$$

Fig. 2: Three-Square Range Proof with Knowledge-Delayed Order (3SRP-KDO)

	3SRP	3SRP-KDO
Communication	$N(8\log n + 18\kappa + 5B) + 3\kappa$	$N(8\log n + 4\kappa) + 10\kappa + 2\log n + B + \log N$
Prover's work	$1.5N(8\log n + 12B + 26\kappa + \log a)$	$1.5(N(13\log n + 13B + 18\kappa + \log a) + \log n + B + 6\kappa + \log N)$
Verifier's work	$1.5(N(5\log n + 9B + 30\kappa + \log a + \log b) + \kappa)$	$1.5(N(12\kappa + \log a + \log b) + \log n + B + 10\kappa + \log N)$

Table 1: Complexities of 3SRP and 3SRP-KDO

Efficiency Analysis. We now provide a detailed comparison between the 3SRP and the 3SRP-KDO protocols. We set the order of the modulus n to 2048 bits and the security parameter κ to 128. As the communication of the protocols does also depend on the bound 2^B on the size of the interval, we consider various bounds in our estimation. For the sake of simplicity, we assume $B = \log b$.

Small Intervals and Large Intervals. As pointed out in [CCs08], several practical applications of range proofs, such as e-voting [Gro05] and e-cash [CHL05], involve quite small intervals (say, of size at most 2^{30}, and so $B \leq 30$). However, in numerous cryptographic schemes, range proofs on very large intervals are involved. Examples include anonymous credentials [CL01], mutual private set intersection protocols [KLC12], secure generation of RSA keys [JG02, DM10, HMRT12], zero-knowledge primality tests [CM99a], and some protocols for performing non-arithmetic operations on Paillier ciphertexts [GMS10, CPP15]. In such protocols, B typically range from 1024 to 8000. We note that such intervals are exactly the ones for which range proofs based on groups of hidden order are likely to be used, since for for small intervals, protocols based on some u-ary decomposition of the input [CCs08, Gro11] will in general have better performances (essentially because they avoid the need of the Rabin-Shallit algorithm, which is computationally involved).

Comparisons. Table 2 gives a summary of our results. As already noted, the overhead of the work of the prover in 3SRP-KDO is measured by comparing the works *without considering the cost of the Rabin-Shallit algorithm*; the latter one, however, is by far the dominant cost when B is large (as it runs in expected $O(B^2 \log B \cdot M(\log B))$ time, where $M(\log B)$ is the time taken to perform a multiplication of $(\log B)$-bit integers). Therefore, for a large B, the overhead of the work of the prover in 3SRP-KDO is very small, whereas there is a huge gain for the verifier. As expected, the 3SRP-KDO protocol provides interesting performances in settings where:

- The verifier is computationally weak (e.g. in secure Cloud computing), and/or
- Multiples range proofs are likely to be used in parallel, and/or
- The intervals are large.

	communication overhead	prover's work overhead	verifier's work overhead
$B = 30, N = 1$	+16%	+60.2%	−66%
$B = 1024, N = 1$	−3.7%	+44%	−71.7%
$B = 2048, N = 1$	−17%	+36.4%	−74.1%
$B = 30, N = 10$	−7.6%	+47.5%	−86.8%
$B = 1024, N = 10$	−26.5%	+33.2%	−87.7%
$B = 2048, N = 10$	−39.1%	+26.5%	−88%

This is for various interval sizes (2^B) and numbers N of parallel executions
Percentages indicate $100 \times (\mathsf{cost}(\mathsf{3SRP\text{-}KDO}) - \mathsf{cost}(\mathsf{3SRP}))/\mathsf{cost}(\mathsf{3SRP})$, where prover's cost does not consider the 3-square decomposition.

Table 2: Comparison between the 3SRP and the 3SRP-KDO

Acknowledgments

This work was supported in part by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 339563 – CryptoCloud).

References

AM76. L. Adleman and K. Manders. Diophantine complexity. In *Proceedings of the 17th Annual Symposium on Foundations of Computer Science*, SFCS '76, pages 81–88, Washington, DC, USA, 1976. IEEE Computer Society.

AMAR05. A. André, R. Markus, and S. Ahmad-Reza. Non-interactive watermark detection for a correlation-based watermarking scheme. In *Communications and Multimedia Security: 9th IFIP TC-6 TC-11 International Conference, CMS 2005*, pages 129–139, 2005.

BCDv88. E. F. Brickell, D. Chaum, I. Damgård, and J. van de Graaf. Gradual and verifiable release of a secret. In *CRYPTO'87*, *LNCS* 293, pages 156–166. Springer, Heidelberg, August 1988.

BG12. S. Bayer and J. Groth. Efficient zero-knowledge argument for correctness of a shuffle. In *EUROCRYPT 2012*, *LNCS* 7237, pages 263–280. Springer, Heidelberg, April 2012.

BGR98a. M. Bellare, J. A. Garay, and T. Rabin. Batch verification with applications to cryptography and checking. In *LATIN 1998*, *LNCS* 1380, pages 170–191. Springer, Heidelberg, April 1998.

BGR98b. M. Bellare, J. A. Garay, and T. Rabin. Fast batch verification for modular exponentiation and digital signatures. In *EUROCRYPT'98*, *LNCS* 1403, pages 236–250. Springer, Heidelberg, May / June 1998.

BHJ$^+$13. F. Böhl, D. Hofheinz, T. Jager, J. Koch, J. H. Seo, and C. Striecks. Practical signatures from standard assumptions. In *EUROCRYPT 2013*, *LNCS* 7881, pages 461–485. Springer, Heidelberg, May 2013.

Bou00. F. Boudot. Efficient proofs that a committed number lies in an interval. In *EUROCRYPT 2000*, *LNCS* 1807, pages 431–444. Springer, Heidelberg, May 2000.

BP97. N. Bari and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT'97*, *LNCS* 1233, pages 480–494. Springer, Heidelberg, May 1997.

BS02. E. Bresson and J. Stern. Proofs of knowledge for non-monotone discrete-log formulae and applications. In *ISC 2002*, *LNCS* 2433, pages 272–288. Springer, Heidelberg, September / October 2002.

CCs08. J. Camenisch, R. Chaabouni, and a. shelat. Efficient protocols for set membership and range proofs. In *ASIACRYPT 2008*, *LNCS* 5350, pages 234–252. Springer, Heidelberg, December 2008.

CCT07. S. Canard, I. Coisel, and J. Traoré. Complex zero-knowledge proofs of knowledge are easy to use. In *ProvSec 2007*, *LNCS* 4784, pages 122–137. Springer, Heidelberg, November 2007.

CFT98. A. H. Chan, Y. Frankel, and Y. Tsiounis. Easy come - easy go divisible cash. In *EUROCRYPT'98*, *LNCS* 1403, pages 561–575. Springer, Heidelberg, May / June 1998.

CHL05. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *EUROCRYPT 2005*, *LNCS* 3494, pages 302–321. Springer, Heidelberg, May 2005.

CL01. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, *LNCS* 2045, pages 93–118. Springer, Heidelberg, May 2001.

CM99a. J. Camenisch and M. Michels. Proving in zero-knowledge that a number is the product of two safe primes. In *EUROCRYPT'99*, *LNCS* 1592, pages 107–122. Springer, Heidelberg, May 1999.

CM99b. J. Camenisch and M. Michels. Separability and efficiency for generic group signature schemes. In *CRYPTO'99*, *LNCS* 1666, pages 413–430. Springer, Heidelberg, August 1999.

CPP15. G. Couteau, T. Peters, and D. Pointcheval. Encryption switching protocols. Cryptology ePrint Archive, Report 2015/990, 2015. `http://eprint.iacr.org/`.

DF02. I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *ASIACRYPT 2002*, *LNCS* 2501, pages 125–142. Springer, Heidelberg, December 2002.

DM10. I. Damgård and G. L. Mikkelsen. Efficient, robust and constant-round distributed RSA key generation. In *TCC 2010*, *LNCS* 5978, pages 183–200. Springer, Heidelberg, February 2010.

DPR61. M. Davis, H. Putnam, and J. Robinson. The decision problem for exponential diophantine equations. *Annals of Mathematics*, pages 425–436, 1961.

FO97. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO'97*, *LNCS* 1294, pages 16–30. Springer, Heidelberg, August 1997.

Gen04. R. Gennaro. Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks. In *CRYPTO 2004*, *LNCS* 3152, pages 220–236. Springer, Heidelberg, August 2004.

GMS10. J. Guajardo, B. Mennink, and B. Schoenmakers. Modulo reduction for Paillier encryptions and application to secure statistical analysis. In *FC 2010*, *LNCS* 6052, pages 375–382. Springer, Heidelberg, January 2010.

Gro05. J. Groth. Non-interactive zero-knowledge arguments for voting. In *ACNS 05*, *LNCS* 3531, pages 467–482. Springer, Heidelberg, June 2005.

Gro09. J. Groth. Linear algebra with sub-linear zero-knowledge arguments. In *CRYPTO 2009*, *LNCS* 5677, pages 192–208. Springer, Heidelberg, August 2009.

Gro11. J. Groth. Efficient zero-knowledge arguments from two-tiered homomorphic commitments. In *ASIACRYPT 2011*, *LNCS* 7073, pages 431–448. Springer, Heidelberg, December 2011.

HJK11. D. Hofheinz, T. Jager, and E. Kiltz. Short signatures from weaker assumptions. In *ASIACRYPT 2011*, *LNCS* 7073, pages 647–666. Springer, Heidelberg, December 2011.

HMRT12. C. Hazay, G. L. Mikkelsen, T. Rabin, and T. Toft. Efficient RSA key generation and threshold Paillier in the two-party setting. In *CT-RSA 2012*, *LNCS* 7178, pages 313–331. Springer, Heidelberg, February / March 2012.

HW09. S. Hohenberger and B. Waters. Short and stateless signatures from the RSA assumption. In *CRYPTO 2009*, *LNCS* 5677, pages 654–670. Springer, Heidelberg, August 2009.

JG02. A. Juels and J. Guajardo. RSA key generation with verifiable randomness. In *PKC 2002*, *LNCS* 2274, pages 357–374. Springer, Heidelberg, February 2002.

JKK14. S. Jarecki, A. Kiayias, and H. Krawczyk. Round-optimal password-protected secret sharing and T-PAKE in the password-only model. In *ASIACRYPT 2014, Part II*, *LNCS* 8874, pages 233–253. Springer, Heidelberg, December 2014.

JS07. S. Jarecki and V. Shmatikov. Efficient two-party secure computation on committed inputs. In *EUROCRYPT 2007*, *LNCS* 4515, pages 97–114. Springer, Heidelberg, May 2007.

KLC12. M. Kim, H. T. Lee, and J. H. Cheon. Mutual private set intersection with linear complexity. In *WISA 11*, *LNCS* 7115, pages 219–231. Springer, Heidelberg, August 2012.

KTY04. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *EUROCRYPT 2004*, *LNCS* 3027, pages 571–589. Springer, Heidelberg, May 2004.

LAN01. H. Lipmaa, N. Asokan, and V. Niemi. Secure vickrey auctions without threshold trust. Cryptology ePrint Archive, Report 2001/095, 2001. `http://eprint.iacr.org/2001/095`.

Lip03. H. Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *ASIACRYPT 2003*, *LNCS* 2894, pages 398–415. Springer, Heidelberg, November / December 2003.

Ped92. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO'91*, *LNCS* 576, pages 129–140. Springer, Heidelberg, August 1992.

Pol03. C. Pollett. On the bounded version of hilbert's tenth problem. *Arch. Math. Log.*, 42(5):469–488, 2003.

PS96. D. Pointcheval and J. Stern. Security proofs for signature schemes. In *EUROCRYPT'96*, *LNCS* 1070, pages 387–398. Springer, Heidelberg, May 1996.

PS00. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

RS86. M. O. Rabin and J. O. Shallit. Randomized algorithms in number theory. *Communications on Pure and Applied Mathematics*, 39(S1):S239–S256, 1986.

RSA78. R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signature and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, 1978.

A A Correction on Lipmaa's Argument for Positivity

A.1 Initial Protocol

Lipmaa [Lip03, Annex B] proposed an efficient zero-knowledge argument of positivity. Since the exact protocol was not fully detailled, we describe on Figure 3 our understanding from reading its proof of correctness. Unfortunately, it is not sound, and the flaw comes from the fact that the original protocol is described as using a generalized Damgård-Fujisaki commitment scheme. However, the same basis is used to commit to masks m_1, m_2, m_3, m_4, which implies that the prover will only be (computationally) binded to $\sum_i x_i$ in the argument.

Actually, it does not seem possible to rely on generalized commitments to get a more efficient protocol. Concretely, let us consider a prover $\mathscr{P}^*$ holding (x, r) such that $c = g^x h^r$ and $x = -1$. $\mathscr{P}^*$ commits $x_1 = 0, x_2 = 1, x_3 = 0$ and $x_4 = 0$, and computes d_1, d_2 honestly. After receiving a challenge, however, $\mathscr{P}^*$ sets $\bar{x}_1 = 2, \bar{x}_2 = -1, \bar{x}_1 = 0, \bar{x}_1 = 0$, and sends $\bar{z}_i = e\bar{x}_i + m_i$ for $i = 1$ to 4

instead of the correct z_i, and $\bar{t}_2 = e(r - \sum_i \bar{x}_i r_i) + s_2$ instead of the correct t_2. The values $\bar{x}_i$ were chosen so that $\sum_i \bar{x}_i = \sum_i x_i$, hence $\sum_i \bar{z}_i = e(\sum_i \bar{x}_i) + \sum_i m_i = e(\sum_i x_i) + \sum_i m_i = \sum_i z_i$, and so the check that $(c_1 c_2 c_3 c_4)^e \cdot d_1 = g^{\bar{z}_1 + \bar{z}_2 + \bar{z}_3 + \bar{z}_4} h^{t_1}$ succeeds. The second verification is equivalent to checking that $\sum_i x_i \cdot \bar{x}_i = x$, which is the case here $(-1 = 0 \times 2 + 1 \times (-1) + 0 \times 0 + 0 \times 0)$: $\mathscr{V}$ accepts the argument even though the value x known by $\mathscr{P}^*$ is strictly negative.

A natural way to fix this flaw without increasing the communication would be to require the verifier to send a seed λ between step 1 and step 2, from which pseudo-random values $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are stretched, to send $d_1 = \sum_i \lambda_i m_i$, $t_1 = e \sum_i \lambda_i r_i + s_1$ and to adapt the verification equation accordingly. However, an attack quite similar to the one we've just described succeeds with good probability in this case (it is sufficient that the gcd of λ_i and λ_j is small, for some $i \neq j$, for the attack to succeed). The interesting point is that we cannot batch the arguments of knowledge and the proof of membership at the same time.

A.2 Corrected Protocol

In this section, we propose a variant of Lipmaa's protocol [Lip03] proving that a committed x is a sum of four squares. There are two correct ways to construct an optimized argument of positivity. A first possibility is to rely on a collision-resistant hash function to strongly reduce the length of the flow sent by $\mathscr{P}$ in step 2 (note that we only require the hash function to be collision-resistant, hence the protocol is in the standard model). An alternative would be to let $\mathscr{P}$ send all individual values $(d_i)_i$ and d in step 2 instead of a single hash, and to stretch pseudo-random values from e in step 4 to batch all the t_i into a single value. We describe the former solution, on Figure 4, as it is slightly more efficient than the latter in terms of communication and enjoys a better security reduction.

A.3 Proof of Security

Correctness immediately follows from a careful inspection of the protocol.

Zero-Knowledge Property. We now argue that the protocol is honest-verifier zero-knowledge: given c and a challenge e, the simulator $\mathscr{S}im_{\mathsf{ZK}}$ sends random group elements c_1, c_2, c_3, c_4, and picks random $(z_i, t_i) \leftarrow_R [\![0\,;2^{B/2+2\kappa}]\!] \times [\![0\,;2^{2\kappa}n]\!]$ for $i = 1$ to 4, and a random $t \leftarrow_R [\![0\,;2^{B/2+2\kappa}n]\!]$. In step 2, $\mathscr{S}im_{\mathsf{ZK}}$ sends $\varDelta = H\left((g^{z_i} h^{t_i} c_i^{-e} \bmod n)_{i=1}^4, \prod_{i=1}^4 c_i^{z_i} h^t c^{-e} \bmod n\right)$. The commitments $(c_i)_i$ are perfectly indistinguishable from valid commitments, and $((z_i)_i, (t_i)_i, t)$ are statistically indistinguishable from honestly computed integers, with a similar analysis as in Section 3.

Knowledge Extractability. Let us now prove the knowledge extractability of the protocol under the RSA assumption. A prover $\mathscr{P}'$ which succeeds in providing a convincing proof with probability ε is rewinded, to provide two valid proofs for the same initial commitments $c_1, c_2, c_3, c_4, \varDelta$. Under the collision-resistance of the hash function: $g^{z_i} h^{t_i} c_i^{-e} = g^{z'_i} h^{t'_i} c_i^{-e'} \bmod n$, for $i = 1$ to 4, and $\prod c_i^{z_i} h^t c^{-e} = \prod c_i^{z'_i} h^{t'} c^{-e'} \bmod n$.

Hence, we have, for $i = 1$ to 4, $c_i^{e'-e} = g^{z_i - z'_i} h^{t_i - t'_i} \bmod n$, and $c^{e'-e} = \prod c_i^{z_i - z'_i} h^{t-t'} \bmod n$. Using a similar argument as in the proof of Theorem 2, unless one can break the RSA assumption, $e' - e$ likely divides all the other differences and so, with $\rho_i = (z_i - z'_i)/(e' - e)$ and $w_i = (t_i - t'_i)/(e' - e)$ for $i = 1$ to 4, and $w = (t - t')/(e' - e)$, we have $c_i = g^{\rho_i} h^{w_i}$, and $c = \prod_{i=1}^4 c_i^{\rho_i} h^w$.

Altogether, this implies that $c = \prod_{i=1}^4 g^{\rho_i^2} h^{w_i \rho_i} h^w = g^{\sum \rho_i^2} h^{w + \sum w_i \rho_i} \bmod n$. The commitment c thus contains $x = \sum \rho_i^2$, that is necessarily positive.

$\mathscr{P}$ knows (x, r) such that $c = g^x h^r \bmod n$ and $x \geq 0$. $\mathscr{V}$ knows c.

1. $\mathscr{P}$ computes $(x_i)_{i \leq 4}$ such that $x = \sum_{i=1}^4 x_i^2$. $\mathscr{P}$ commits the x_i's with fresh random coins $r_i \leftarrow_R [\![0\,;n]\!]$ as $c_i = g^{x_i} h^{r_i} \bmod n$. $\mathscr{P}$ sends c_1, c_2, c_3, c_4 to $\mathscr{V}$.
2. $\mathscr{P}$ picks $(m_i)_{i=1}^4 \leftarrow_R [\![0\,; 2^{B/2+2\kappa}]\!]^4$, $s_1 \leftarrow_R [\![0\,; 2^{2\kappa+|n|}]\!]$, and $s_2 \leftarrow_R [\![0\,; 2^{B/2+|n|+2\kappa}]\!]$. Then, $\mathscr{P}$ sends $d_1 = g^{m_1+m_2+m_3+m_4} \cdot h^{s_1} \bmod n$ and $d_2 = c_1^{m_1} c_2^{m_2} c_3^{m_3} c_4^{m_4} \cdot h^{s_2} \bmod n$.
3. $\mathscr{V}$ picks a challenge $e \leftarrow_R [\![0\,; 2^\kappa]\!]$ and sends it to $\mathscr{P}$.
4. $\mathscr{P}$ computes and sends $z_i = ex_i + y_i$, for $i = 1$ to 4, $t_1 = e \sum_i r_i + s_1$ and $t_2 = e(r - \sum_i x_i r_i) + s_2$.
5. $\mathscr{V}$ accepts the argument if both

$$(c_1 c_2 c_3 c_4)^e \cdot d_1 = g^{z_1+z_2+z_3+z_4} h^{t_1} \quad \text{and} \quad c^e \cdot d_2 = c_1^{z_1} c_2^{z_2} c_3^{z_3} c_4^{z_4} \cdot h^{t_2}.$$

Fig. 3: Lipmaa's Compact Argument for Positivity

$\mathscr{P}$ knows (x, r) such that $c = g^x h^r$ and $x \geq 0$. $\mathscr{V}$ knows c. Let $H : \mathbb{Z}_n^5 \mapsto \{0,1\}^{2\kappa}$ be a collision-resistant hash function.

1. $\mathscr{P}$ computes $(x_i)_{i \leq 4}$ such that $x = \sum_{i=1}^4 x_i^2$. $\mathscr{P}$ commits the x_i's with fresh random coins $r_i \leftarrow_R [\![0\,;n]\!]$ as $c_i = g^{x_i} h^{r_i} \bmod n$. $\mathscr{P}$ sends c_1, c_2, c_3, c_4 to $\mathscr{V}$.
2. $\mathscr{P}$ picks $(m_i)_{i=1}^4 \leftarrow_R [\![0\,; 2^{B/2+2\kappa}]\!]^4$, $(s_i)_{i=1}^4 \leftarrow_R [\![0\,; 2^{2\kappa} n]\!]^4$, $s \leftarrow_R [\![0\,; 2^{B/2+2\kappa} n]\!]$, computes $(d_i = g^{m_i} h^{s_i} \bmod n)_{i=1}^4$, $d = \prod_{i=1}^4 c_i^{m_i} h^s$, and sends the commitment $\Delta = H(d_1, d_2, d_3, d_4, d)$ to $\mathscr{V}$.
3. $\mathscr{V}$ picks a challenge $e \leftarrow_R [\![0\,; 2^\kappa]\!]$ and sends it to $\mathscr{P}$.
4. $\mathscr{P}$ computes and sends $z_i = ex_i + m_i$ and $t_i = er_i + s_i$ for $i = 1$ to 4, and $t = e(r - \sum x_i r_i) + s$.
5. $\mathscr{V}$ accepts the argument if $\Delta = H\left((g^{z_i} h^{t_i} c_i^{-e} \bmod n)_{i=1}^4, \prod_{i=1}^4 c_i^{z_i} h^t c^{-e} \bmod n\right)$.

Fig. 4: Variant of Lipmaa's Compact Argument for Positivity