Efficient Zero-Knowledge Arguments and Digital Signatures *via* Sharing Conversion *in the Head*

Jules Maire and Damien Vergnaud

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

Abstract. We present a novel technique within the MPC-in-the-Head framework, aiming to design efficient zero-knowledge protocols and digital signature schemes. The technique allows for the simultaneous use of additive and multiplicative sharings of secret information, enabling efficient proofs of linear and multiplicative relations. The applications of our technique are manifold. It is first applied to construct zero-knowledge arguments of knowledge for Double Discrete Logarithms (DDLP). The resulting protocol achieves improved communication complexity without compromising efficiency. We also propose a new zero-knowledge argument of knowledge for the Permuted Kernel Problem. Eventually, we suggest a short (candidate) post-quantum digital signature scheme constructed from a new one-way function based on simple polynomials known as fewnomials. This scheme offers simplicity and ease of implementation. Finally, we present two additional results inspired by this work but using alternative approaches. We propose a zero-knowledge argument of knowledge of an RSA plaintext for a small public exponent that significantly improves the state-of-the-art communication complexity. We also detail a more efficient forward-backward construction for the DDLP.

1 Introduction

Zero-knowledge protocols have emerged as a pivotal tool in ensuring robust computer security and enhancing cryptographic protocols. They offer a powerful solution by allowing one party to prove knowledge of certain information to another party, without revealing any additional details. With the rapid advancements in quantum computing technology, the need for post-quantum security in cryptography and computer security has become of paramount importance. Post-quantum cryptography aims to develop communication protocols that can withstand attacks from both classical and quantum computers.

Secure multi-party computation (MPC) enables a group of $n \geq 2$ parties, who do not trust each other, to collaboratively compute a joint function using their private inputs. In 2007, Ishai, Kushilevitz, Ostrovsky, and Sahai [IKOS07] demonstrated that semi-honest multiparty computation (i.e. where adversaries follow the protocol description but may try to learn arbitrary information) is sufficient for constructing zero-knowledge protocols. This theoretical paradigm, deemed *MPC-in-the-Head*, has received considerable practical attention recently

since it enables the construction of efficient and succinct protocols with good security properties. It has been used in particular to propose several innovative signature schemes with (alleged) post-quantum security. The goal of this article is to add another string to the *MPC-in-the-Head*'s bow by integrating secret sharing conversion, a technique that has already been used in general MPC [GPS12]. We show that this technique finds applications for (1) zero-knowledge arguments of knowledge of *Double Discrete Logarithms*, (2) zero-knowledge arguments of knowledge of *Permuted Kernel Problem* solutions, and (3) constructing a (candidate) post-quantum digital signature scheme from a new *somewhat minimalistic* one-way function in finite fields.

Related works and contributions of the paper. The MPC-in-the-Head (MPCitH) framework [IKOS07] has gained considerable popularity in recent times. This framework leverages secure MPC techniques, where the prover mentally shares its secret information and emulates a semi-honest MPC protocol involving N parties and independently commits each party's view. The verifier then challenges the prover to reveal the views of a randomly selected subset of $N-1$ parties. By design, no information about the original input is exposed, thereby achieving the zero-knowledge property. Besides, a malicious prover would need to deceive at least one party, which the verifier is likely to detect, ensuring the soundness property. In most practical applications, the secret is shared additively among the N parties, which makes proving linear relations easy but proofs of multiplicative relations more costly. Several techniques were introduced recently to improve the practical efficiency of the resulting schemes, for instance, the MPCitH with a *helper* as formalized in [Beu20], the MPCitH with *abort* introduced in [FMRV22] or the recent *hypercubing* optimization technique proposed in [MGH+23].

We present a new technique to expand this toolbox further by allowing a prover to use simultaneously in the MPC protocol additive sharings and multiplicative sharings of its secret information. The former are used for linear relations, while the latter are used to prove efficiently multiplicative relations. To ensure consistency, we propose a simple technique to transform a multiplicative share into an additive share of the same value. Converting shares from one type of secret sharing scheme into another is ubiquitous in MPC [GPS12] and the idea has already been used in the MPCitH realm [DGH+21] (but for different sharings). Our technique finds several applications in (post-quantum) zero-knowledge arguments and digital signature schemes.

Double Discrete Logarithm Problem (DDLP): A double discrete logarithm of an element $y \neq 1_{\mathbb{G}}$ in a cyclic group $\mathbb{G}$ of prime order q with respect to bases $g \in \mathbb{G}$ and $h \in \mathbb{F}_q^*$ (generators of $\mathbb{G}$ and $\mathbb{F}_q^*$ respectively) is an integer $x \in \{0, \ldots, q-1\}$ such that $y = g^{h^x}$. Initially introduced by Stadler [Sta96] for verifiable secret-sharing, this computational problem has found applications in various cryptographic protocols, including group signatures [CS97], blind signatures [ASM10], e-cash systems [CG07], credential systems [CGM16], and verifiable randomness generation [BTV20]. Stadler proposed a zero-knowledge protocol, which has a computational and communication complexity of $\Omega(\log q)$ (in terms of group

elements). However, in the recent work [BTV20], Blazy, Towa, and Vergnaud presented a new protocol that outputs arguments with only $O(\log \log q)$ group elements. It relies on the "*Bulletproofs*" technique proposed by Bünz, Bootle, Boneh, Poelstra, Wuille and Maxwell in 2018 [BBB+18]. This reduced communication complexity comes at a security price since the security analysis should rely on stronger idealized assumptions [GOP+22] or achieve only non-meaningful concrete security [DG23]. For a use-case considered in [BTV20], the length of Stadler arguments are 24.6 Kilobytes (KB) and those of Blazy *et al.* are 10.2KB long. As a first simple application of our conversion *in the head* technique, we present (for similar prover and verifier efficiency) arguments of size about 16.6KB (depending on the parameters). Even if this is longer than the previous approach, this still improves the communication complexity of Stadler's protocol by about 30%. By increasing the prover and verifier computational complexity, it is possible to decrease the communication complexity to 7.2KB (with better security guarantees than [BTV20]). It is worth mentioning that even by increasing the prover/verifier running times, the arguments of [Sta96,BTV20] cannot be shortened.

Permuted Kernel Problem (PKP): The PKP is a classical $\mathcal{NP}$-hard computational problem, where, given a matrix and a vector (of matching dimensions) defined over a finite field, one has to find a permutation of the vector coordinates that belongs to the matrix kernel. This problem was introduced in cryptography by Shamir [Sha90], who designed a zero-knowledge argument of knowledge of a solution of a PKP problem (and used it for a cryptographic post-quantum identification scheme). This protocol was improved subsequently in a long series of work [Ste94,BFK+19,Beu20,FJR23,Fen22a,BG22]. We apply our technique to this problem and obtain a zero-knowledge argument of knowledge protocol which does not involve permutations that are not easy to implement securely, in particular in the presence of side-channel attacks.

One-way functions from "Fewnomials": A cryptographic one-way function $f : S \to S$ is a function that is computationally easy to compute but computationally difficult to invert. If S is a finite field (e.g. $S = \mathbb{F}_p$ for some prime number p), then it is well-known that f can be represented as a polynomial in $\mathbb{F}_p[X]$ (with degree upper-bounded by $(p-1)$). Ad hoc examples of such functions are cryptographic hash functions or functions derived from block ciphers (using for instance the Davies-Meyer construction [Win84]). Still, the polynomial representations of such functions are usually of very high complexity (which makes them not convenient for the MPCitH paradigm). Several works were devoted to designing efficient symmetric cryptographic primitives suitable for efficient implementation using MPCitH (e.g. the Picnic [CDG+20,KZ22] and the Rainier [DKR+22] signature schemes). As a third application of our technique, we propose a reverse approach to design a cryptographic system with simplicity and minimal complexity. The motivation is to remove potential points of failure and to obtain schemes easier to implement correctly. To do so, we consider the simplest polynomials defined over a finite field $\mathbb{F}_p$ that are good one-way function candidates. The simplest polynomials are certainly the monomials $f_1 : \mathbb{F}_p \to \mathbb{F}_p$,

$x \mapsto f_1(x) = x^n \mod p$ but they are trivially not one-way. If n is coprime with $(p-1)$, this is a permutation on which one can apply the Davies-Meyer construction to obtain the binomials $f_2 : \mathbb{F}_p \to \mathbb{F}_p$, $x \mapsto f_2(x) = x^n + x \mod p$ which seem difficult to invert (the best-known algorithm for $n = \Omega(p)$ has arithmetic complexity $O(p^{1/2})$ [BCR13]). More generally, a *fewnomial* is a term used in algebraic geometry and computational algebra, to describe a polynomial with a few terms (i.e. with a relatively low number of monomials compared to its degree). If one considers a fewnomial of high degree with $t \geq 2$ monomials over $\mathbb{F}_p$, the best known algorithm has arithmetic complexity $O(p^{(t-1)/t})$ [BCR13]. These candidate one-way functions are not suitable for symmetric cryptography (since evaluating them is much more costly than popular hash functions and block ciphers) but they are particularly interesting for our new conversion technique. In particular, we propose (candidate) post-quantum signatures with lengths of about 10.5KB. The produced signatures are thus not the shortest ones, but our goal with this application is to propose a new simpler, and cleaner one-way function suitable for the MPCitH paradigm with competitive performances and to motivate future research in this area.

Other results: We present two additional results inspired by this work but using alternative approaches. We first describe a zero-knowledge argument of knowledge of an RSA plaintext for a small public exponent that significantly improves the state-of-the-art communication complexity [GQ90]. The scheme is very simple but seems to have been overlooked. Following a recent idea proposed by Joux [Jou23], we also propose a more efficient construction for the DDLP (without using our conversion in the head technique) achieving arguments about 6.6KB long. This improves the communication complexity of Stadler's protocol by about 75% (for the same security guarantees and overall efficiency).

2 Preliminaries

We denote $\mathbb{F}_q$ the finite field with q elements (for q some prime power). Let $N \geq 2$ be some integer. We make use of N-out-of-N *additive* and *multiplicative sharing* of field elements $x \in \mathbb{F}_q$ and $x \in \mathbb{F}_q^\times$ (respectively); they are vectors $[\![x]\!] = ([\![x]\!]_1, \ldots, [\![x]\!]_N) \in {\mathbb{F}_q}^N$ and $\langle x \rangle = (\langle x \rangle_1, \ldots, \langle x \rangle_N) \in {\mathbb{F}_q^\times}^N$ (respectively) such that

$$x = [\![x]\!]_1 + \cdots + [\![x]\!]_N \bmod q \text{ and } x = \langle x \rangle_1 \cdot \cdots \cdot \langle x \rangle_N \bmod q.$$

For a vector $v \in \mathbb{F}_q^n$, its sharing $[\![v]\!]$ (or $\langle v \rangle$) is seen coordinate-wisely. All logarithms are in base 2. We denote the security parameter by λ. The designation PPT stands for probabilistic polynomial-time in the security parameter. Random sampling from a finite set X according to the uniform distribution is denoted by $x \xleftarrow{\$} X$, whereas the symbol $\leftarrow$ is used for assignments from deterministic algorithms. We write $[m, n]$ to denote the set of integers $\{m, \ldots, n\}$ with $m < n$.

Two distributions $\{D_\lambda\}_\lambda$ and $\{\tilde{D}_\lambda\}_\lambda$ are called (t, ε)-indistinguishable if, for any algorithm $\mathcal{A}$ running in time at most $t(\lambda)$, we have

$$|\Pr[\mathcal{A}(1^\lambda, x) = 1 \mid x \xleftarrow{\$} D_\lambda] - \Pr[\mathcal{A}(1^\lambda, x) = 1 \mid x \xleftarrow{\$} \tilde{D}_\lambda]| \leq \varepsilon(\lambda).$$

A (ℓ, t, ε)-*pseudo-random generator* (PRG) is a deterministic algorithm G that, for all $\lambda \in \mathbb{N}$, on input a bit-string $x \in \{0,1\}^\lambda$ outputs $G(x) \in \{0,1\}^{\ell(\lambda)}$ with $\ell(\lambda) > \lambda$ such that the distributions $\{G(x) \mid x \xleftarrow{\$} \{0,1\}^\lambda\}_\lambda$ and $\{r \mid r \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}\}_\lambda$ are (t, ε)-indistinguishable. From such a generator, with $\ell(\lambda) = 2\lambda$, it is possible to construct a *tree PRG* [KKW18], which takes a root $x \in \{0,1\}^\lambda$ as input and generates $N = 2^t$ pseudo-random λ-bit strings in a structured fashion as follows: x is the label of the root of a depth-t complete binary tree in which the right/left child of each node is labeled with the λ most/least significant bits of the output of the PRG applied to the root label. This structure allows revealing $N - 1$ pseudo-random values of the leaves by revealing only $\log(N)$ labels of the tree (by revealing the labels on the siblings of the paths from the root to the one remaining leaf).

2.1 Commitment scheme

We define a commitment scheme as a pair of algorithms $(\mathsf{Com}, \mathrm{Verif})$ where:

- Com is a PPT taking as input a message m, that computes a commitment C of m and returns C and opening information ρ.
- Verif is a deterministic polynomial-time algorithm taking as input a message m, a commitment C and the opening information ρ, and returns a bit.

For all message m we have: $\forall (C, \rho) \xleftarrow{\$} \mathsf{Com}(m), \mathrm{Verif}(m, C, \rho) = 1$. A commitment scheme is said (t, ε)-computationally *hiding* if, for any two messages m_1, m_2, the distributions $\{c \mid c \xleftarrow{\$} \mathsf{Com}(m_1)\}$ and $\{c \mid c \xleftarrow{\$} \mathsf{Com}(m_2)\}$ are (t, ε)-indistinguishable. A commitment scheme is computationally *binding* if there exists a negligible function ν such that, for every PPT algorithm $\mathcal{A}$, the probability that the event

$$\left\{ \begin{array}{l} m_1 \neq m_2 \quad \wedge \\ \mathrm{Verif}(, m_1, C, \rho_1) = \mathrm{Verif}(m_2, C, \rho_2) = 1 \end{array} \;\middle|\; (m_1, m_2, \rho_1, \rho_2, C) \xleftarrow{\$} \mathcal{A}(1^\lambda) \right\}$$

occurs is upper-bounded by $\nu(\lambda)$. In the following, we consider a commitment scheme that outputs a 2λ bit-long commitment.

2.2 Zero-knowledge Arguments

A zero-knowledge protocol for a polynomial-time decidable binary relation $\mathcal{R}$ is defined by two interactive algorithms, a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. Both algorithms are given a common input x, and $\mathcal{P}$ is given an additional *witness* w such that $(x, w) \in \mathcal{R}$. The two algorithms then exchange messages until $\mathcal{V}$ outputs a

bit b ($b = 1$ to accept $\mathcal{P}$'s claim and $b = 0$ to reject). This sequence of messages and the answer b is referred to as a *transcript* and denoted $\langle \mathcal{P}(x, w), \tilde{\mathcal{V}}(x) \rangle$. In this paper, we consider *zero-knowledge argument of knowledge* which are protocols that allow a PPT prover to convince a PPT verifier that they *know* a witness w. There are three security notions underlying a zero-knowledge argument of knowledge.

Definition 1. *Let $t : \mathbb{N} \to \mathbb{N}$, $\varepsilon, \alpha, \zeta : \mathbb{N} \to [0, 1]$, and $\mathcal{R}$ be a polynomial-time decidable binary relation. A zero-knowledge argument $(\mathcal{P}, \mathcal{V})$ for $\mathcal{R}$ achieves:*

- α-completeness*, if for all $\lambda \in \mathbb{N}$ and all $(x, w) \in \mathcal{R}$, with $x \in \{0,1\}^\lambda$, $\Pr[\langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle = 1] \geq 1 - \alpha(\lambda)$ (*i.e. *$\mathcal{P}$ succeeds in convincing $\mathcal{V}$, except with probability α). When $\alpha = 1$, the protocol achieves perfect completeness.*
- ε-(special) soundness*, if for all PPT malicious prover $\tilde{\mathcal{P}}$ such that for all $\lambda \in \mathbb{N}$ and all $x \in \{0,1\}^\lambda$, $\tilde{\varepsilon}(\lambda) := \Pr[\langle \tilde{\mathcal{P}}(x), \mathcal{V}(x) \rangle = 1] > \varepsilon(\lambda)$, there exists a PPT algorithm $\mathcal{E}$ (called the* extractor*) which, given rewindable black-box access to $\tilde{\mathcal{P}}$ outputs a witness w such that $(x, w) \in \mathcal{R}$ in time $\mathsf{poly}(\lambda, (\tilde{\varepsilon} - \varepsilon)^{-1})$ with probability at least $1/2$.*
- (t, ζ)-zero-knowledge*, if for every PPT malicious verifier $\tilde{\mathcal{V}}$, there exists a PPT algorithm $\mathcal{S}$ (called the* simulator*) which, given the input statement $x \in \{0,1\}^\lambda$ and rewindable black-box access to $\tilde{\mathcal{V}}$, outputs a simulated transcript whose distribution is (t, ζ)-indistinguishable from $\langle \mathcal{P}(x, w), \tilde{\mathcal{V}}(x) \rangle$.*

If the zero-knowledge property holds only for the genuine verifier $\mathcal{V}$, then the protocol is deemed honest-verifier *zero-knowledge. In that case, $\mathcal{S}$ is given random challenges instead of a rewindable black-box access to $\tilde{\mathcal{V}}$.*

2.3 MPC in the head

The concept of MPCitH [IKOS07] provides a method for constructing zero-knowledge protocols using secure MPC protocols. Let f be some (one-way) function and y some public output of the function. Assume we have a secure MPC protocol Π_f with N parties such that, for any x in the domain of the function, the output of Π_f is 1 if $f(x) = y$ and 0 otherwise. Equivalently, the binary relation $\mathcal{R}$ underlying f can be used to define the output of Π_f: $\Pi_f(x) = 1 \iff (pp, x) \in \mathcal{R}$ where pp are public parameters. A prover $\mathcal{P}$ given a secret input x, generates a random N-out-of-N secret sharing of x, and mentally simulates all the parties' computation in Π_f. $\mathcal{P}$ sends commitments of each party's view in the protocol (including input share, secret random tape, and broadcast values). $\mathcal{V}$ selects $N - 1$ parties randomly and requests $\mathcal{P}$ to reveal their views. Upon receiving them, $\mathcal{V}$ verifies their consistency with an honest execution of Π_f and the commitments. Since the views of only $N - 1$ parties are disclosed, this does not disclose any information about the secret x.

MPCitH with Helper. In this work, we use the MPCitH with *Helper* paradigm introduced in [Beu20] by Beullens. This approach adds a trusted third party (called the helper) to the MPC protocol which runs a pre-processing phase. To

then remove the helper, one uses a cut-and-choose strategy. This approach is typically useful when some correlated randomness has to be generated in the MPC protocol. This randomness structure is desired for our sharing conversion: $\mathcal{P}$ has to produce a random couple of sharing $(\llbracket r \rrbracket, \langle s \rangle)$ with $r = s$. To prove the validity of this couple (i.e. $r = s$), we follow a cut-and-choose approach. $\mathcal{P}$ produces M couples of sharing $(\llbracket r^{[\ell]} \rrbracket, \langle s^{[\ell]} \rangle)_{\ell \in [1,M]}$ and commits to them. Then $\mathcal{V}$ asks to open all the couples except one and checks that each opened couple encodes an identical value. Hence, $\mathcal{V}$ can trust the unopened sharing with a soundness error of $1/M$. Note that the protocol requires converting more than one sharing (regardless of the emulated MPC protocol in the head). Indeed, to achieve a negligible soundness, we repeat the protocol. Hence, instead of opening $M - 1$ sharing, we remain unopened a subset of the M sharing.

Hypercube optimization. In [MGH+23], Melchor, Gama, Howe, Hülsing, Joseh, and Yue developed a geometrical approach for the MPC emulation phase. When dealing with additive and multiplicative secret sharing in finite fields, this optimization fits pretty well (due to the commutativity of the addition and multiplication laws). In the traditional approach of MPCitH, $\mathcal{P}$ simulates N parties during one emulation of one MPC protocol. By this hypercube approach, this number of parties can be reduced to $1 + \log_2 N$, with the same soundness error (see [MGH+23] for more details). In fact, instead of simulating one MPC with N parties, the prover emulates $\log_2 N$ MPC with 2 parties (2PC), thus a total of $2 \log_2 N$ parties, which can be optimized to $1 + (2 - 1) \log_2 N = 1 + \log_2 N$. Indeed, each MPC protocol should output the same result. Thus, once one protocol is computed, all but one parties (1 party here) is enough to determine the evaluation of the last party in the other 2PC. This optimization makes the MPC emulation less costly and allows us to take a larger number of parties (and get smaller sizes). For example, for the same soundness error, when the traditional approach needs to simulate 2^8 parties, we only need to emulate 9 parties. The computational gain is attenuated by the number of repetitions since the total number of parties to emulate is $\tau(1 + \log_2 N) \approx \lambda(1 + 1/\log_2 N)$, where τ is the number of repetitions of the protocol to get negligible soundness.

Sharing on the integers. We also make use of a technique developed in [FMRV22] by Feneuil, Maire, Rivain, and Vergnaud, to encode a binary secret $x \in \mathbb{F}_q$ over the integers in the MPCitH paradigm, i.e. $x = \sum_{i=1}^{N} \llbracket x \rrbracket_i + \Delta x$ with $\llbracket x \rrbracket_i \xleftarrow{\$} [0, A - 1]$ (with no modular reduction) with some $A \ll q$. To avoid information leakage, Feneuil *et al.* introduced the possibility for $\mathcal{P}$ to abort depending on the (last) challenge of $\mathcal{V}$. This induces a rejection rate in the protocol that can be decreased by increasing A (but this increases the communication complexity). Then they generalized this sharing to encode non-binary elements throughout the construction of a digital signature from Boneh-Halevi-Howgrave-Graham pseudo-random function. In this work, we use this last sharing to share on the integers some vector in $\mathbb{F}_q^n$. Contrary to the initial motivation, we take $A > q$, and the rejection rate of the sharing becomes $1 - \left(1 - \frac{q-1}{A}\right)^n$. This approach is only used when constructing a PKP argument of knowledge.

3 Sharing Conversion and Design Principle

RSA in the Head. In the MPCitH paradigm, when the secret is shared additively, multiplicative relations are costly to prove, and vice versa. Whence converting secret sharing *in the Head* naturally comes to mind. However, there exists a natural application where the conversion is not necessary, which seems to have been overlooked in the literature. Indeed, assume that we want to prove the knowledge of an RSA plaintext for a public exponent e, i.e. $x^e = y \bmod n$ where n is some RSA modulus. Then we could imagine sharing x multiplicatively as $x = \prod_{j=1}^{N} \langle x \rangle_j \bmod n$ and the corresponding MPC protocol consists simply in locally computing $\langle x \rangle^e$. Using straightforward techniques from MPCitH, this simple observation improves the communication complexity of the seminal protocol from Guillou and Quisquater [GQ90] for the public exponent $e = 3$ from around 20.4KB to 6.6KB for a 2048-bit modulus n and has similar efficiency. The communication complexity could be made even smaller by increasing N (but at the cost of an increased computational complexity). Interestingly, even if the hypercube technique [MGH+23] could be applied here, this would result in worse computation complexity. See annex B for the protocol and its security proofs. To apply the hypercube optimization, we have to build the new shares of the $1 + \log_2 N$ new parties, and each of these shares is the product of $N/2$ original shares. Assume d is the number of operations during the fast exponentiation. Then, for each party, there are $d + \frac{N}{2} - 1$ operations, hence a total of $(1+\log_2 N)(d+\frac{N}{2}-1)$ operations, to compare with dN operations for our method (since our protocol requires N fast exponentiation to perform). Therefore, our method is asymptotically better, and e.g., when $e = 3$, $d = 2$ and as long as $N > 4$, the hypercube optimization increases the computational cost.

Sharing conversion. Let $\langle x \rangle$ be a multiplicative sharing of some field element $x \in \mathbb{F}_q$. The aim is to securely compute an additive sharing $[\![x]\!]$ of x. For the sharing conversion considered in the following, we need a uniformly random precomputed couple of sharing $([\![r]\!], \langle s \rangle)$ such that $r = s \in \mathbb{F}_q^\times$. As explained in section 2, we work in the MPCitH with helper paradigm and follow a cut-and-choose approach. The MPC protocol is the following:

Input: The parties have $\langle x \rangle$. Output: The parties get $[\![x]\!]$.
Preprocessing phase: A trusted dealer generates random sharing $r = \sum_{i=1}^{N} [\![r]\!]_i$, $s = \prod_{i=1}^{N} \langle s \rangle_i$, such that $r = s$. They give $([\![r]\!]_i, \langle s \rangle_i)$ to party P_i for $i \in [1, N]$. **Online phase:** 1. The parties compute $\langle \alpha \rangle = \langle x \rangle / \langle s \rangle$ and broadcast it. 2. The parties locally compute $\alpha[\![r]\!] := [\![x]\!]$.

Protocol 1: Sharing conversion protocol Π_{conv}

In practice, during the preprocessing phase, we start by generating

$$\{[\![r]\!]_i\}_{1\leq i\leq N} \xleftarrow{\$} \mathbb{F}_q \text{ and } \{\langle s\rangle_i\}_{1\leq i\leq N} \xleftarrow{\$} \mathbb{F}_q^\times.$$

Then define $r = \sum_{i=1}^N [\![r]\!]_i$, and compute Δs such that $r = \Delta s \prod_{i=1}^N \langle s\rangle_i := s$. If $r = s = 0$, i.e. $\Delta s = 0$, we start again. In terms of communication complexity (during the interaction between the prover and the verifier in the zero-knowledge argument), this offline step introduces one auxiliary value Δs to send. Since there is also the value α to communicate when running Π_{conv} (because it is a broadcasted value), the sharing conversion protocol needs 2 field elements to communicate for each single conversion (we can not reuse the couple of sharing for another conversion).

Correctness and security. Π_{conv} actually outputs an additive sharing of x if $r = s$. It provides security in the semi-honest model (which is sufficient for the MPCitH paradigm) and α does not reveal any information on the parties' share thanks to the randomness of s.

General protocol. We develop a 5-round protocol with helper, presented in a general manner, and that can be adapted to each of the problems considered in this work. Let f be a one-way function, along with y a public output of the function. Let Π_f be a secure MPC protocol with N parties such that, for any x (assume $x \in \mathbb{F}_q$) in the domain of the function, the output of Π_f is 1 if $f(x) = y$ and 0 otherwise. Π_f takes as input a secret sharing of x which is either $[\![x]\!]$, $\langle x\rangle$, or a sharing on the integers from [FMRV22]. It also takes as input a couple (or many couples) of secret sharing $([\![r]\!], \langle s\rangle)$ with $r = s \in \mathbb{F}_q^\times$ that is generated during a pre-processing phase. For the PKP application, Π_f takes as additional input, some prime number q' greater than q.

Soundness error. Let ε be the soundness of the protocol. We perform τ parallel repetitions of the protocol to get a soundness error $\varepsilon^\tau < (1/2)^\lambda$. As explained in the previous paragraph, each of these repetitions uses a cut-and-chose phase to prove the helper. Instead of performing $\tau \approx \lambda/\log_2(N)$ parallel cut-and-chose phases each resulting in trusting one couple of sharing $([\![r^{[\ell]}]\!], \langle s^{[\ell]}\rangle)$ among M, we follow the more efficient approach from [KKW18] and perform a global cut-and-choose phase resulting in τ trusted sharing among a larger M. The idea is that $\mathcal{V}$ asks to reveal $M - \tau$ out of M master seeds. The remaining τ executions of the pre-processing phase are used to emulate τ independent instances of the MPC protocol. When opening all but one seed, a wrong couple of sharing will not be detected with probability

$$\frac{1}{N} + \left(1 - \frac{1}{N}\right)\beta, \tag{1}$$

where β is the false positive probability of the MPC protocol. This β will be zero when considering the DDLP and the fewnomial pre-image problem. If a cheating

prover produces $M - k \leq \tau$ wrong couples of sharing, they will not be detected during the first phase (when revealing $M - \tau$ master seeds) with probability

$$\binom{k}{M-\tau}\binom{M}{M-\tau}^{-1}.$$

This leads to the soundness error

$$\varepsilon = \max_{M-\tau \leq k \leq M} \left\{ \frac{\binom{k}{M-\tau}\left(\frac{1}{N} + (1-\frac{1}{N})\beta\right)^{k-M+\tau}}{\binom{M}{M-\tau}} \right\}$$

(see [KKW18] for additional details).

We describe the identification scheme 2 used in the remainder of the paper. The protocol makes use of a pseudo-random generator PRG, a tree-based pseudo-random generator TreePRG, four collision-resistant hash functions $\mathcal{H}_i$ for $i \in [1, 4]$ and a commitment scheme (Com, Verif). The red part of the protocol has to be adapted depending on the problem considered. We choose to use $[\![\cdot]\!]$ in the protocol for the sharing of x and $f(x)$, but it can be substituted by $\langle\cdot\rangle$.

Parameters selection. Recall that we are dealing with a preprocessing phase, that is proved with a cut-and-choose strategy. The total number of parties to set up is MN, which impacts the prover/verifier computational complexity, hence we choose sets of parameters to keep a reasonable signing time. We start by fixing a number of parties N to be either 2^5 or 2^8. Then we look for the best trade-off between τ, M while keeping a soundness error below $2^{-\lambda}$. Decreasing τ leads to better sizes but to higher M and so slower signatures. The MPC emulation does not impact a lot the signing speed, since the hypercube optimization is consistent with our scheme (see section 2). To have a relatively small computational cost, we limit the number of parties for the MPC protocol to 2^5 for the first set of parameters. Thanks to the hypercube optimization and a gain in the emulation of the MPC protocol, we choose a second set of parameters with $N = 2^8$ parties and get an acceptable running time. Note that our MPC protocols are easy to implement and parallelize since most of the computation is done locally by the parties, except for the broadcasting during the sharing conversion protocol 1. To estimate the cut-and-choose's impact on the computation cost of the tree expansion, randomness generation, share preparation, and commitments computation, we use the benchmark proposed in [Fen22a], assuming a 4-core processor.

4 Proving Knowledge of a Double Discrete Logarithm

We present the Double Discrete Logarithm Problem (DDLP) which has found numerous applications in cryptography [CS97,ASM10,CG07,CGM16,BTV20].

Prover $\mathcal{P}$ $x \in \mathbb{F}_q$		Verifier $\mathcal{V}$ $y = f(x)$
$\mathsf{mseed}^{[0]} \xleftarrow{\$} \{0,1\}^\lambda$ $(\mathsf{mseed}^{[e]})_{e\in[1,M]} \leftarrow \text{TreePRG}(\mathsf{mseed}^{[0]})$ For each $e \in [1, M]$: $(\mathsf{seed}_i^{[e]}, \rho_i^{[e]})_{i\in[1,N]} \leftarrow \text{TreePRG}(\mathsf{mseed}^{[e]})$ For each $i \in [1, N]$: $([\![x^{[e]}]\!]_i, [\![r^{[e]}]\!]_i, \langle s^{[e]}\rangle_i) \leftarrow \text{PRG}(\mathsf{seed}_i^{[e]})$ $\mathsf{com}_i^{[e]} = \mathsf{Com}(\mathsf{seed}_i^{[e]}; \rho_i^{[e]})$ $\Delta x^{[e]} = x - \sum_i [\![x]\!]_i^{[e]}$ $r^{[e]} = \sum_i [\![r]\!]_i^{[e]}$ $\Delta s^{[e]} = r^{[e]} / \prod_i \langle s\rangle_i^{[e]}$ $s^{[e]} = \Delta s^{[e]} \prod_i \langle s\rangle_i^{[e]}$ $h_e = \mathcal{H}_1(\Delta s^{[e]}, \mathsf{com}_1^{[e]}, \ldots, \mathsf{com}_N^{[e]})$ $h = \mathcal{H}_2(h_1, \ldots, h_M)$		▷ $[\![x^{[e]}]\!]_i, [\![r^{[e]}]\!]_i \in \mathbb{F}_q$, $\langle s^{[e]}\rangle_i \in \mathbb{F}_q^\times$
	$\xrightarrow{h}$	
		$J \xleftarrow{\$} \{J \subset [1, M]\ ;\ \lvert J\rvert = \tau\}$
	$\xleftarrow{J}$	
For each $e \in J$: The parties computes $\Pi_f([\![x^{[e]}]\!], \Delta x^{[e]})$ $h'_e = \mathcal{H}_3(\Delta x^{[e]}, [\![y^{[e]}]\!], \alpha^{[e]})$ $h' = \mathcal{H}_4((h'_e)_{e\in J})$		▷ $\alpha^{[e]}$ is the broadcasted value in Π_{conv} called in Π_f
	$\xrightarrow{h',\ (\mathsf{mseed}^{[e]})_{e\in[1,M]\setminus J}}$	
		$L = \{\ell_e\}_{e\in J} \xleftarrow{\$} [1, N]^\tau$
	$\xleftarrow{L}$	
	$\left(\begin{matrix}(\mathsf{seed}_i^{[e]}, \rho_i^{[e]})_{i\neq\ell_e} \\ \Delta x^{[e]}, \Delta s^{[e]}, \alpha^{[e]},\ \mathsf{com}_{\ell_e}^{[e]}\end{matrix}\right)_{e\in J}$	
	$\xrightarrow{\quad}$	
		For each $e \notin J$: Compute h_e using $\mathsf{mseed}^{[e]}$ For each $e \in J$: For all $i \neq \ell_e$ $\mathsf{com}_i^{[e]} = \mathsf{Com}(\mathsf{seed}_i^{[e]}; \rho_i^{[e]})$ Rerun the party i as the prover to get $[\![y^{[e]}]\!]_i$ $[\![y^{[e]}]\!]_{\ell_e} = y - \sum_{i\neq\ell_e} [\![y^{[e]}]\!]_i$ $h_e = \mathcal{H}_1(\Delta s^{[e]}$ $\mathsf{com}_1^{[e]}, \ldots, \mathsf{com}_N^{[e]})$ $h'_e = \mathcal{H}_3(\Delta x^{[e]}, [\![y^{[e]}]\!], \alpha^{[e]})$ Check $h = \mathcal{H}_2(h_1, \ldots, h_M)$ Check $h' = \mathcal{H}_4((h'_e)_{e\in J})$ Return 1

Protocol 2: Identification scheme of a pre-image for a function f.

Double Discrete Logarithm Problem (DDLP).

Let $\mathbb{G}$ be a cyclic group of prime order q with some generator $g \in \mathbb{G}$, and let $h \in \mathbb{F}_q^*$ of prime order p with $p|(q-1)$. Given $(y, g, h) \in \mathbb{G} \setminus \{1_\mathbb{G}\} \times \mathbb{G} \times \mathbb{F}_q^*$, the *DDLP* asks to find some $x \in \mathbb{F}_p^\times$ such that $y = g^{h^x}$.

We first propose a direct application of our sharing conversion technique. However, even if it improves the state-of-the-art in terms of communication complexity (compared to schemes with the same assumptions), we present this scheme primarily for pedagogical purposes. Indeed, we then build an efficient zero-knowledge argument of knowledge based on a forward-backward technique.

First construction. Consider the function $f : \mathbb{F}_p^\times \to \mathbb{G}, x \mapsto f(x) = g^{h^x}$ realizing the "double discrete exponentiation". We present an MPC protocol Π_{DDLP} to securely compute the corresponding binary relation (via the computation of a multiplicative sharing of $f(x) \in \mathbb{G}$).

Input: $y \neq 1_\mathbb{G}$ in a cyclic group $\mathbb{G}$ of prime order q, $h \in \mathbb{F}_q^*$ of prime order p with $p\|(q-1)$, and an additive sharing of $x \in \mathbb{F}_p^\times$.
Output: 1 if $y = g^{h^x}$, 0 otherwise.
1. Parties locally compute a multiplicative sharing $\langle h^x \rangle$ via $h^x = \prod_{j=1}^N h^{[\![x]\!]_j} \bmod q$.
2. Parties convert it into an additive sharing $[\![h^x]\!]$ over $\mathbb{F}_q$ using Π_{conv} 1.
3. Parties locally compute $\langle g^{h^x} \rangle$ via $g^{h^x} = \prod_{j=1}^N g^{[\![h^x]\!]_j}$.
4. Parties broadcast $\langle g^{h^x} \rangle$ and output 1 if $g^{h^x} = y$ and 0 otherwise.

Protocol 3: MPC protocol Π_{DDLP}

The correctness of Π_{DDLP} comes from the fact that $h^x = h^{\sum_{j=1}^N [\![x]\!]_j \bmod p} = \prod_{j=1}^N h^{[\![x]\!]_j}$ since h has order p. The same reasoning holds for step 3 because g has order q. Plugging Π_{DDLP} into the red part of protocol 2, with $\alpha^{[e]} = h^{[e]}/s^{[e]}$, we readily get a zero-knowledge argument of knowledge of a solution to the given DDLP instance. Note that we should also slightly adapt protocol 2 since $x \in \mathbb{F}_p$ (with $p \leq q$), and $y^{[e]} := g^{h^{x\,[e]}}$ is shared multiplicatively, but this is straightforward.
To estimate the communication complexity, we remark that for each iteration of the protocol, three values have to be communicated: the auxiliary value $\Delta x \in \mathbb{F}_p$ to fix the secret, and $(\Delta s, \alpha) \in \mathbb{F}_q^2$ from the sharing conversion protocol 1 (there is a sole conversion). This leads to a total communication cost of at most:

$$4\lambda + \lambda\tau \log_2 \frac{M}{\tau} + \tau \left[2 \log_2(q) + \log_2 p + \lambda \log_2 N + 2\lambda\right] \text{ bits},$$

where M the number of parallel phases in the cut-and-choose, and τ the number of unrevealed phases (see section 3).

Theorem 1 (Security Proofs).
Considered an instance $(y, g, h) \in \mathbb{G} \setminus \{1_\mathbb{G}\} \times \mathbb{G} \times \mathbb{F}_q^*$ of the DDLP. Then, the identification scheme 2 combined with MPC protocol 3 is an honest-verifier zero-knowledge argument of knowledge of $x \in \mathbb{F}_p^\times$ such that $g^{h^x} = y$, with perfect

completeness, and special soundness ε equals to

$$\max_{M-\tau \leq k \leq M} \left\{ \frac{\binom{k}{M-\tau}}{\binom{M}{M-\tau} N^{k-M+\tau}} \right\} .$$

Proofs can be found in appendix C.

In [BTV20], the authors considered the case of a group $\mathbb{G}$ of prime order $q = (4p + 18)p + 1$ where p is the Sophie Germain prime $p = 2^{1535} + 554415$ that divides $q-1$. Their arguments involve $2\lceil \log_2(2(\lceil \log_2(\ell) \rceil + 1)) \rceil + 8$ elements in $\mathbb{G}$ and 5 elements in $\mathbb{F}_q$. Taking $\mathbb{G}$ as the subgroup of order q in $\mathbb{F}_\ell^*$ for $\ell = 1572q + 1$, one obtains an argument of size 10.2KB for [BTV20] and of size 24.6KB for [Sta96] (for a soundness error of 2^{-128}). Table 1 presents the communication complexity of our arguments. Note that they are always shorter than those from [Sta96] and provide better security guarantees than [BTV20] (as mentioned in the introduction). Contrary to [BTV20], we could compress our argument size and construct parameter sets with argument size below 10KB (but at the cost of an increase in computational complexity for the prover and the verifier). We propose another set of parameters, more adapted to our scheme, by considering p, q as $\sim$2048-bit prime.

Second construction. Actually, we could greatly improve the performance of our zero-knowledge argument of knowledge by considering another approach. This is based on an idea of Joux [Jou23], a forward-backward technique. Again, we start by sharing x additively. Then the prover $\mathcal{P}$ commits to the values

$$y_i := \left(\left(\left(g^{h^{[\![x]\!]_1}} \right)^{h^{[\![x]\!]_2}} \right)^{\cdots} \right)^{h^{[\![x]\!]_i}} \quad \text{for } i \in [1, N].$$

The correctness of this approach relies on the fact that $y_N = y$. The verifier $\mathcal{V}$ sends a challenge $i^* \xleftarrow{\$} [1, N]$. The prover $\mathcal{P}$ answers by sending the seeds $\{\mathsf{seed}_i\}_{i \neq i^*}$ (i.e. opens all the shares of x except the i^*th) to $\mathcal{V}$. This last can recompute all the committed values by a forward-backward technique: they iteratively compute y_i as

- $y_{i-1}^{h^{[\![x]\!]_i}}$ if $1 \leq i \leq i^* - 1$;
- $y_{i+1}^{-h^{[\![x]\!]_{i+1}}}$ if $i^* \leq i \leq N - 1$;

with $y_0 = g$ and $y_N = y$. In particular, $\mathcal{V}$ can check the consistency of the subsequences $\{y_1, \ldots, y_{i^*-1}\}$ and $\{y_{i^*}, \ldots, y_N = y\}$ thanks to the revealed seeds, but $\mathcal{V}$ can not check the whole sequence $\{y_1, \ldots, y_N\}$ since $[\![x]\!]_{i^*}$ is missing to link y_{i^*-1} and y_{i^*}. Hence, except if a malicious prover is cheating on $[\![x]\!]_{i^*}$, they are detected. This yields the following MPC protocol 4, where a party $\mathcal{P}_i$ uses the output of $\mathcal{P}_{i-1}$ to compute their output (for $1 < i \leq N$).

The overall zero-knowledge argument of knowledge protocol with its security proofs can be found in appendix A.

Input: $y \neq 1_{\mathbb{G}}$ in a cyclic group $\mathbb{G}$ of prime order q, $h \in \mathbb{F}_q^*$ of prime order p with $p\|(q-1)$, and an additive sharing of $x \in \mathbb{F}_p^\times$. **Output:** 1 if $y = g^{h^x}$, 0 otherwise.
1. Party $\mathcal{P}_1$ computes $y_1 = g^{h^{[\![x]\!]_1}}$ and broadcast it. 2. For each $i \in [2, N]$: Party $\mathcal{P}_i$ computes $y_i = y_{i-1}^{h^{[\![x]\!]_i}}$ and broadcast it. 3. If $y_N = y$ parties output 1, otherwise 0.

Protocol 4: MPC protocol with the forward-backward technique

Performances. Since no more cut-and-choose has to be produced, the argument size is shortened compared to the sharing conversion approach (and only one element field has to be communicated). This leads to the performances in table 1, where the communication complexity of the argument proof is reduced by 77% compared to [Sta96], and beats the bulletproof approach of [BTV20].

protocol	Parameters				Argument size (KB)
	$\log_2 q$	τ	N	M	
protocol 3	2048	16	2^8	4096	16.6
protocol 3	2048	17	2^8	1744	17.2
protocol 4	3072	16	2^8	4096	5.6
protocol 4	3072	17	2^8	1744	5.9

Table 1: Achieved performances of our zero-knowledge protocol for proving the knowledge of a solution of a DDLP instance.

5 Proving Knowledge of a PKP Solution

We denote $\mathcal{S}_n$ the symmetric group of degree n. For a permutation $\pi \in \mathcal{S}_n$ and a vector $v \in \mathbb{F}_q^n$, $\pi(v)$ is the action of the permutation on the coordinates of v.

Permuted Kernel Problem (PKP/IPKP).

Let (q, m, n) be positive integers, $H \in \mathbb{F}_q^{m \times n}$ a random matrix, and a vector $v \in \mathbb{F}_q^n$. The PKP is to find a permutation $\pi \in \mathcal{S}_n$, such that $H\pi(v) = 0$. The inhomogeneous version of the problem (IPKP) is, given a target vector $y \in \mathbb{F}_q^m$, to find a permutation $\pi \in \mathcal{S}_n$, such that $H\pi(v) = y$.

We consider the PKP variant, but this work can be straightly extended for the IPKP (without loss of performances since y is public). We want to prove the knowledge of a solution to a PKP instance, i.e., some $x \in \mathbb{F}_q^n$ and $\pi \in \mathcal{S}_n$ such that $Hx = 0$ and $\pi(v) = x$. For this purpose, we adapt the protocol 2 as follows:

– the input $x \in \mathbb{F}_q^n$ is a vector, so we should consider one conversion by coordinate;
– the sharing of x is over the integers, so $[\![x^{[e]}]\!]_j \in [0, A-1]^n$ for some $A > q$. Thus, we should add a rejection rule as explained in section 2;
– $\mathcal{V}$ sends an additional challenge $g \xleftarrow{\$} \mathbb{F}_{q'}^{\times}$ (as an evaluation point) at the same time as the challenge J, where q' is a prime greater than q whose choice is explained afterward.

Proving the knowledge of a permutation. Consider the polynomial $f_{x,v}(X) = \sum_{i=1}^n X^{x_i} - \sum_{i=1}^n X^{v_i}$ of degree at most $q-1$ (x_i, v_i denotes the components of the vectors x, v), and some uniformly random element $g \in \mathbb{F}_{q'}$. If $x = \pi(v)$ for some $\pi \in \mathcal{S}_n$, then $f_{x,v}$ is identically zero. If there is no permutation $\pi \in \mathcal{S}_n$ such that $\pi(v) = x$, then via the Schwartz-Zippel lemma [Sch80,Zip79], the probability that $f_{x,v}(g) = 0 \bmod q'$ is bounded by $(q-1)/q'$. Indeed, the probability that a random polynomial in $\mathbb{F}_{q'}[X]$ of degree at most $q-1$ be vanished by a random element in $\mathbb{F}_{q'}$ is at most $(q-1)/q'$.

Initially, the sharing over the integers was introduced to share small values. In this work, when computing $f_{x,v}(g)$ over $\mathbb{F}_{q'}$ in a distributed way, the challenge g may not satisfy $g^q = 1 \bmod q'$ and then the modular sharing would lead to a wrong computation. This is the motivation for using a sharing over the integers for x.

Slack management. Recall that the verifier knows that $[\![x_i]\!]_j \in [0, A-1]$ (this is verified for open parties) and they check that $-A + q \leq x_i - [\![x_i]\!]_{j^*} \leq 0$. This implies that they are convinced by the fact that $-A + q \leq x_i \leq A - 1$. In particular, the degree of the polynomial is bounded by $A-1$, whence the slack. Indeed, the degree should be bounded by $q-1$, but a malicious prover may choose some x whose coordinates are upper bounded by $A-1$. This is not a problem as long as the modulus q' is large enough compared to A (for the Schwartz-Zippel lemma). This leads to the next MPC protocol.

MPC protocol. We describe the MPC protocol Π_{PKP} to plug in the red part of protocol 2. As input, x is shared among the parties via a secret sharing over the integers, i.e., $[\![x]\!]_j \xleftarrow{\$} [0, A-1]^n$ for $j \in [1, N]$. The rejection rate of the sharing is $1 - \left(1 - \frac{q-1}{A}\right)^n$ (see section 2). Parties also get some $g \in \mathbb{F}_{q'}^{\times}$ with q' a prime number greater than $\beta(A-1)$, where $1/\beta$ is the false positive probability of the MPC protocol. We present the following MPC protocol Π_{PKP} to securely compute the corresponding binary relation via the computation of a sharing of $\{Hx, f_{x,v}(g)\}$.

Notice that the correctness of $g^{x_i} = g^{\sum_{j=1}^N [\![x_i]\!]} \bmod q'$ follows from the sharing over the integers (and would not be necessarily true if x_i was shared as $\sum_{j=1}^N [\![x_i]\!]_j \bmod q$).

Size. For each coordinate, there is one conversion (so two values over $\mathbb{F}_{q'}$) and one auxiliary value for the secret (over $[0, A-1]$). Hence, the obtained argument

Input: $x \in \mathbb{F}_q^n$ shared over the integers as $x = \sum_{j=1}^{N} [\![x]\!]_j$, $H \in \mathbb{F}_q^{m \times n}$, $g \xleftarrow{\$} \mathbb{F}_{q'}^{\times}$ with q' the next prime after $\beta(A-1)$. **Output:** 1 if $Hx = 0 \bmod q$ and $\pi(x) = v$ for some $\pi \in \mathcal{S}_n$, 0 otherwise.
1. From the sharing over the integers of each x_i, parties locally compute $\langle g^{x_i} \rangle$, a multiplicative sharing of $g^{x_i} = \prod_{j=1}^{N} g^{[\![x_i]\!]_j} \bmod q'$, for each $i \in [1, n]$. 2. Parties convert it into an additive sharing $[\![g^{x_i}]\!]$ using Π_{conv} 1, for each $i \in [1, n]$. 3. Parties locally compute their share of $[\![f_{x,v}(g)]\!] = \sum_{i=1}^{n} [\![g^{x_i}]\!] - \sum_{i=1}^{n} g^{v_i} \bmod q'$. 4. Parties locally compute their share of $[\![Hx]\!] = H[\![x]\!] \bmod q$. 5. Parties broadcast $[\![Hx]\!]$ and $[\![f_{x,v}(g)]\!]$. If $Hx = 0$ and $f_{x,v}(g) = 0$, parties output 1, otherwise 0.

Protocol 5: MPC protocol Π_{PKP}

size is

$$4\lambda + \lambda\tau \log_2 \frac{M}{\tau} + \tau \left[n(2 \log_2 q' + \log_2(A-1)) + \lambda \log_2 N + 2\lambda \right] \text{ bits},$$

where M is the number of parallel phases in the cut-and-choose, and τ the number of unrevealed phases.

Theorem 2 (Security Proofs).
Considered an instance $(H, v) \in \mathbb{F}_q^{m \times n} \times \mathbb{F}_q^n$ of the PKP. Then, the identification scheme protocol 2 combined with MPC protocol 5 is an honest-verifier zero-knowledge argument of knowledge of $(x, \sigma) \in \mathbb{F}_q^n \times \mathcal{S}_n$ such that $Hx = 0$ and $\sigma(x) = v$, with $\left(1 - \frac{q-1}{A}\right)^{\tau\, n}$-completeness and special soundness ε equals to

$$\max_{M-\tau \le k \le M} \left\{ \frac{\binom{k}{M-\tau} \left(\frac{1}{N} + (1 - \frac{1}{N}) \frac{1}{q'} \right)^{k-M+\tau}}{\binom{M}{M-\tau}} \right\}.$$

The proof of Theorem 2 can be found in annex E.

Performances. The security of the PKP/IPKP has been well-studied for many years (see section 1). We consider the parameter sets proposed in [BFK+19] to achieve 128 bits of security, i.e. $n = 61$, $m = 28$, $q = 997$. The choice of the remaining parameters τ and M are chosen as a trade-off between argument size and signing speed. We fix $\beta = 2^8$, i.e., q' is the next prime after $2^8(A-1)$. Hence, the rate of false-positive when checking the existence of a permutation is smaller than $1/2^8$. Based on the benchmark realized in [Fen22b], the first set of parameters achieves around 20ms for the signing/verification speed, and the second set around 200ms.

6 Proving Knowledge of a Fewnomial Pre-image

We propose a new (candidate) post-quantum one-way function and a digital signature scheme constructed as an argument of knowledge of a pre-image of

Parameters				Argument size (KB)	Rej. rate
τ	N	A	M		
19	2^8	$2^{14}q$	1289	13.3	0.068
27	2^5	$2^{14}q$	541	16.9	0.096

Table 2: Obtained performances for proving the knowledge of a witness of a PKP instance.

the public key using the MPCitH paradigm. Our goal is to design a simple and somewhat minimalistic scheme.

We consider a prime number p and the simplest one-way polynomials defined over the finite field $\mathbb{F}_p$. Those polynomials are called *fewnomials* and are simply polynomials with a relatively low number of monomials compared to their degree. If one considers a fewnomial with $t \geq 2$ monomials of large degrees over $\mathbb{F}_p$, the best known classical algorithm has arithmetic complexity $O(p^{(t-1)/t})$ [BCR13]. Combining this algorithm with Grover's algorithm [Gro96], leads to the best-known quantum algorithm with complexity $O(p^{(t-1)/2t})$.

Fewnomial Inversion Problem (FIP).

Let q be a Sophie Germain prime number where $p = 2q+1$ is also a prime number. Let $t \geq 2$ be an integer and $f : \mathbb{F}_p \to \mathbb{F}_p$ be a fewnomial with t monomials defined as $f(X) = \sum_{i \in S} X^i$ where S is a set of t integers in $[\lceil q/2 \rceil, q-1]$. The Fewnomial Inversion Problem is given $y = f(x) \in \mathbb{F}_p$ to find $x' \in \mathbb{F}_p$ such that $y = f(x')$.

We construct a digital signature scheme based on the hardness of the FIP. Note that we consider the case of unitary monomials but adding non-zero (public) coefficients does not change the following analysis and performances. The choice of t and p will be discussed later on. It is worth mentioning that the monomial $X^n \bmod p$ would be easy to invert except if we replace the prime p by a modulus with unknown factorization, and this would be essentially an RSA instance with a larger modulus (and we can use the construction outlined in section 3)

MPC protocol. The prover/signer shares x multiplicative. We present the MPC protocol Π_{FIP} to plug in protocol 2, in which parties securely compute the corresponding binary relation via the computation of a sharing of $f(x)$.

Proof size. In the MPC protocol 6, parties apply the conversion protocol for each $i \in S^\times$ and each conversion requests to communicate 2 field elements as explained in section 3. The rest of the communication is standard. Hence, the communication cost of the protocol is

$$4\lambda + \lambda\tau \log_2 \frac{M}{\tau} + \tau \left[(1+2s)\log_2 p + \lambda \log_2 N + 2\lambda\right] \text{ bits},$$

Input: $x \in \mathbb{F}_p^\times$ shared multiplicative, i.e. $x = \prod_{j=1}^{N} \langle x \rangle_j \bmod p$. A fewnomial $f : X \to \sum_{i \in S} X^i$, with a finite subset $S \subset \mathbb{N}^t$, and some public value $y \in \mathbb{F}_p$. **Output:** 1 if $f(x) = y$, and 0 otherwise.
1. Parties locally compute $\langle x^i \rangle$ via $x^i = \prod_{j=1}^{N} \langle x \rangle_j^i \mod p$ for $i \in S^\times$. 2. For each $i \in S^\times$, parties convert $\langle x^i \rangle$ into an additive sharing $[\![x^i]\!]$ using Π_{conv} 1. 3. Parties locally compute $[\![f(x)]\!] = \sum_{i \in S} [\![x^i]\!]$. 4. Parties broadcast $[\![f(x)]\!]$, and output 1 if $f(x) = y$ and 0 otherwise.

Protocol 6: MPC protocol Π_{FIP}

where s is the size of $S^\times$, M the number of parallel phases in the cut-and-choose, and τ the number of unrevealed phases (see Section 3).

Theorem 3 (Security Proofs).
Considered an instance $(f, y) \in \mathbb{F}_p[X] \times \mathbb{F}_p$ of the FIP. Then, the identification scheme protocol 2 combined with MPC protocol 6 is an honest-verifier zero-knowledge argument of knowledge of $x \in \mathbb{F}_p$ such that $f(x) = y$, with perfect completeness, and special soundness ε equals to

$$\max_{M-\tau \le k \le M} \left\{ \frac{\binom{k}{M-\tau}}{\binom{M}{M-\tau} N^{k-M+\tau}} \right\} .$$

Proof can be found in annex D.

Fiat-Shamir heuristic. We apply the Fiat-Shamir transform [FS87] to get a non-interactive protocol, and so a signature scheme. Since our protocols have 5 rounds, we have to take into consideration the attack of Kales and Zaverucha [KZ20] for the security of the signature. The forgery cost of the signature scheme is then given by

$$\min_{M-\tau \le k \le M} \left\{ \frac{\binom{M}{M-\tau}}{\binom{k}{M-\tau}} + N^{k-M+\tau} \right\} .$$

Signature scheme. To build a signature, we choose $x \in \mathbb{F}_p^\times$ as the private key and $y = f(x) \bmod p$ as the public key. To achieve a forgery cost of $1/\varepsilon$, we could increase τ, but this would not lead to an efficient scheme. Instead, we transform our 5-round protocol into a 3-round before applying the Fiat-Shamir transform, hence [KZ20] attack does not apply anymore. The 5-to-3-round convert's idea is to emulate M MPC protocols before the first round of communication, i.e., before getting the challenge for the cut-and-choose. After values are committed, $\mathcal{V}$ sends both challenges during the same round. It leads to an overhead in terms of signing speed, i.e., there are $M(1 + \log_2(N))$ parties to emulate instead of $\tau(1 + \log_2 N)$, but the hypercube optimization attenuates it. Moreover, the communication cost is slightly greater for the 3-round version, the size of the signature scheme is then

$$4\lambda + 3\lambda\tau \log_2 \frac{M}{\tau} + \tau \left[(1 + 2s) \log_2 p + \lambda \log_2 N + 2\lambda \right] \text{ bits},$$

with s the size of $S^\times$, M the number of parallel phases in the cut-and-choose, and τ the number of unrevealed phases. The resulting 3-round protocol is also an honest-verifier zero-knowledge proof with the same soundness. It can be checked that the round reduction described here does not impact the proofs of Theorem 3 in annex D.

Performances. As mentioned above, considering a fewnomial with $t \geq 2$ monomials over $\mathbb{F}_p$, the best (classical) known algorithm has arithmetic complexity $O(p^{(t-1)/t})$ [BCR13]. Hence, there is a trade-off between the size of the modulus p and the number of monomials t to consider achieving (classical) 128 bits of security. The optimal one to minimize the proof size is a trinomial over a prime of 170 bits. Based on the benchmark realized in [Fen22b], the estimated signing/verification speed is around 15ms for the first set of parameters and 200ms for the second set.

Parameters			Signature size (KB)
τ	N	M	
28	2^5	389	12.2
18	2^8	1251	10.6

Table 3: Achieved signature size based on FIP.

Acknowledgements. The authors are supported in part by the French ANR SANGRIA project (ANR-21-CE39-0006).

References

ASM10. M. H. Au, W. Susilo, and Y. Mu. Proof-of-knowledge of representation of committed value and its applications. In R. Steinfeld and P. Hawkes, eds, *ACISP 10*, vol. 6168 of *LNCS*, p. 352–369, Sydney, NSW, Australia, 2010. Springer, Heidelberg, Germany.

BBB+18. B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, p. 315–334, San Francisco, CA, USA, 2018. IEEE Computer Society Press.

BCR13. J. Bi, Q. Cheng, and J. M. Rojas. Sub-linear root detection, and new hardness results, for sparse polynomials over finite fields. In M. Kauers, ed., *International Symposium on Symbolic and Algebraic Computation, ISSAC'13, Boston, MA, USA, June 26-29, 2013*, p. 61–68. ACM, 2013.

Beu20. W. Beullens. Sigma protocols for MQ, PKP and SIS, and Fishy signature schemes. In A. Canteaut and Y. Ishai, eds, *EUROCRYPT 2020, Part III*, vol. 12107 of *LNCS*, p. 183–211, Zagreb, Croatia, 2020. Springer, Heidelberg, Germany.

BFK+19. W. Beullens, J.-C. Faugère, E. Koussa, G. Macario-Rat, J. Patarin, and L. Perret. PKP-based signature scheme. In F. Hao, S. Ruj, and S. Sen Gupta, eds, *INDOCRYPT 2019*, vol. 11898 of *LNCS*, p. 3–22, Hyderabad, India, 2019. Springer, Heidelberg, Germany.

BG22. L. Bidoux and P. Gaborit. Compact post-quantum signatures from proofs of knowledge leveraging structure for the pkp, sd and rsd problems. *CoRR*, abs/2204.02915, 2022.

BTV20. O. Blazy, P. Towa, and D. Vergnaud. Public-key generation with verifiable randomness. In S. Moriai and H. Wang, eds, *ASIACRYPT 2020, Part I*, vol. 12491 of *LNCS*, p. 97–127, Daejeon, South Korea, 2020. Springer, Heidelberg, Germany.

CDG+20. M. Chase, D. Derler, S. Goldfeder, J. Katz, V. Kolesnikov, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, X. Wang, and G. Zaverucha. The Picnic Signature Scheme – Design Document. Version 2.2 – 14 April 2020, 2020.

CG07. S. Canard and A. Gouget. Divisible e-cash systems can be truly anonymous. In M. Naor, ed., *EUROCRYPT 2007*, vol. 4515 of *LNCS*, p. 482–497, Barcelona, Spain, 2007. Springer, Heidelberg, Germany.

CGM16. M. Chase, C. Ganesh, and P. Mohassel. Efficient zero-knowledge proof of algebraic and non-algebraic statements with applications to privacy preserving credentials. In M. Robshaw and J. Katz, eds, *CRYPTO 2016, Part III*, vol. 9816 of *LNCS*, p. 499–530, Santa Barbara, CA, USA, 2016. Springer, Heidelberg, Germany.

CS97. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In B. S. Kaliski Jr., ed., *CRYPTO'97*, vol. 1294 of *LNCS*, p. 410–424, Santa Barbara, CA, USA, 1997. Springer, Heidelberg, Germany.

DG23. Q. Dao and P. Grubbs. Spartan and bulletproofs are simulation-extractable (for free!). In C. Hazay and M. Stam, eds, *EUROCRYPT 2023, Part II*, vol. 14005 of *Lecture Notes in Computer Science*, p. 531–562. Springer, 2023.

DGH+21. I. Dinur, S. Goldfeder, T. Halevi, Y. Ishai, M. Kelkar, V. Sharma, and G. Zaverucha. MPC-friendly symmetric cryptography from alternating moduli: Candidates, protocols, and applications. In T. Malkin and C. Peikert, eds, *CRYPTO 2021, Part IV*, vol. 12828 of *LNCS*, p. 517–547, Virtual Event, 2021. Springer, Heidelberg, Germany.

DKR+22. C. Dobraunig, D. Kales, C. Rechberger, M. Schofnegger, and G. Zaverucha. Shorter signatures based on tailor-made minimalist symmetric-key crypto. In H. Yin, A. Stavrou, C. Cremers, and E. Shi, eds, *ACM CCS 2022*, p. 843–857, Los Angeles, CA, USA, 2022. ACM Press.

Fen22a. T. Feneuil. Building MPCitH-based signatures from MQ, MinRank, rank SD and PKP. Cryptology ePrint Archive, Report 2022/1512, 2022.

Fen22b. T. Feneuil. Building mpcith-based signatures from mq, minrank, rank SD and PKP. *IACR Cryptol. ePrint Arch.*, pp. 1512, 2022.

FJR23. T. Feneuil, A. Joux, and M. Rivain. Shared permutation for syndrome decoding: new zero-knowledge protocol and code-based signature. *Des. Codes Cryptogr.*, 91(2):563–608, 2023.

FMRV22. T. Feneuil, J. Maire, M. Rivain, and D. Vergnaud. Zero-knowledge protocols for the subset sum problem from MPC-in-the-head with rejection. In *ASIACRYPT 2022, Part II*, LNCS, p. 371–402. Springer, Heidelberg, Germany, 2022.

FS87. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, ed., *CRYPTO'86*, vol. 263 of *LNCS*, p. 186–194, Santa Barbara, CA, USA, 1987. Springer, Heidelberg, Germany.

GOP+22. C. Ganesh, C. Orlandi, M. Pancholi, A. Takahashi, and D. Tschudi. Fiat-shamir bulletproofs are non-malleable (in the algebraic group model). In O. Dunkelman and S. Dziembowski, eds, *EUROCRYPT 2022, Part II*, vol. 13276 of *LNCS*, p. 397–426, Trondheim, Norway, 2022. Springer, Heidelberg, Germany.

GPS12. H. Ghodosi, J. Pieprzyk, and R. Steinfeld. Multi-party computation with conversion of secret sharing. *Des. Codes Cryptogr.*, 62(3):259–272, 2012.

GQ90. L. C. Guillou and J.-J. Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In S. Goldwasser, ed., *CRYPTO'88*, vol. 403 of *LNCS*, p. 216–231, Santa Barbara, CA, USA, 1990. Springer, Heidelberg, Germany.

Gro96. L. K. Grover. A fast quantum mechanical algorithm for database search. In *28th ACM STOC*, p. 212–219, Philadephia, PA, USA, 1996. ACM Press.

IKOS07. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Zero-knowledge from secure multiparty computation. In D. S. Johnson and U. Feige, eds, *39th ACM STOC*, p. 21–30, San Diego, CA, USA, 2007. ACM Press.

Jou23. A. Joux. MPC in the head for isomorphisms and group actions. *IACR Cryptol. ePrint Arch.*, pp. 664, 2023.

KKW18. J. Katz, V. Kolesnikov, and X. Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In D. Lie, M. Mannan, M. Backes, and X. Wang, eds, *ACM CCS 2018*, p. 525–537, Toronto, ON, Canada, 2018. ACM Press.

KZ20. D. Kales and G. Zaverucha. An attack on some signature schemes constructed from five-pass identification schemes. In S. Krenn, H. Shulman, and S. Vaudenay, eds, *CANS 20*, vol. 12579 of *LNCS*, p. 3–22, Vienna, Austria, 2020. Springer, Heidelberg, Germany.

KZ22. D. Kales and G. Zaverucha. Efficient lifting for shorter zero-knowledge proofs and post-quantum signatures. Cryptology ePrint Archive, Paper 2022/588, 2022.

MGH+23. C. A. Melchor, N. Gama, J. Howe, A. Hülsing, D. Joseph, and D. Yue. The return of the sdith. In C. Hazay and M. Stam, eds, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, vol. 14008 of *Lecture Notes in Computer Science*, p. 564–596. Springer, 2023.

Sch80. J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.

Sha90. A. Shamir. An efficient identification scheme based on permuted kernels (extended abstract) (rump session). In G. Brassard, ed., *CRYPTO'89*, vol. 435 of *LNCS*, p. 606–609, Santa Barbara, CA, USA, 1990. Springer, Heidelberg, Germany.

Sta96. M. Stadler. Publicly verifiable secret sharing. In U. M. Maurer, ed., *EUROCRYPT'96*, vol. 1070 of *LNCS*, p. 190–199, Saragossa, Spain, 1996. Springer, Heidelberg, Germany.

Ste94. J. Stern. Designing identification schemes with keys of short size. In Y. Desmedt, ed., *CRYPTO'94*, vol. 839 of *LNCS*, p. 164–173, Santa Barbara, CA, USA, 1994. Springer, Heidelberg, Germany.

Win84. R. S. Winternitz. A secure one-way hash function built from DES. In *Proceedings of the 1984 IEEE Symposium on Security and Privacy, Oakland, California, USA, April 29 - May 2, 1984*, p. 88–90. IEEE Computer Society, 1984.

Zip79. R. Zippel. Probabilistic algorithms for sparse polynomials. In E. W. Ng, ed., *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, vol. 72 of *Lecture Notes in Computer Science*, p. 216–226. Springer, 1979.

A Zero-Knowledge Argument of Knowledge for the DDLP with MPC Protocol 4

We first described the zero-knowledge argument of knowledge protocol for the DDLP where protocol 4 is plugged in the red part of the next protocol 7.

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$x \in \mathbb{F}_q$		$y = g^{h^x}$
$\mathsf{mseed} \xleftarrow{\$} \{0,1\}^\lambda$		
$(\mathsf{seed}_i, \rho_i)_{i\in[1,N]} \leftarrow \mathrm{TreePRG}(\mathsf{mseed})$		
For each $i \in [1, N]$:		
$\quad [\![x]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$		
$\quad \mathsf{com}_i = \mathsf{Com}(\mathsf{seed}_i; \rho_i)$		
$\Delta x = x - \sum_i [\![x]\!]_i$		
The parties computes $\Pi_f([\![x]\!], \Delta x)$		
$h = \mathcal{H}_1(\mathsf{com}_1, \ldots, \mathsf{com}_N, \Delta x, \{y_i\}_{i\in[1,N]})$		
	$\xrightarrow{\quad h \quad}$	
		$ch = i^* \xleftarrow{\$} [1, N]$
	$\xleftarrow{\quad i^* \quad}$	
	$\xrightarrow{(\mathsf{seed}_i, \rho_i)_{i\neq i^*},\ \Delta x,\ \mathsf{com}_{i^*}}$	
		For all $i \neq i^*$
		$\quad \mathsf{com}_i = \mathsf{Com}(\mathsf{seed}_i; \rho_i)$
		Run the forward-backward technique
		$\quad$ to get $y_1, \ldots, y_N$
		Check h
		Return 1

Protocol 7: Identification scheme of a solution of a DDLP instance

Theorem 4 (Security Proofs).
Considered an instance $(y, g, h) \in \mathbb{G} \setminus \{1_\mathbb{G}\} \times \mathbb{G} \times \mathbb{F}_q^*$ of the DDLP. Then, the identification scheme 7 combined with MPC protocol 4 is a zero-knowledge argument of knowledge of $x \in \mathbb{F}_p^\times$ such that $g^{h^x} = y$, with perfect completeness, and special soundness ε equals to $\frac{1}{N}$.

Proof. Completeness. For any sampling of the random coins of $\mathcal{P}$ and $\mathcal{V}$, if the computation described in protocol 7 combined with protocol 4 is honestly performed, all the checks of $\mathcal{V}$ pass. The completeness is hence perfect.

Special soundness. For the sake of simplicity, we assume that the commitment scheme is perfectly binding, otherwise if it was computationally binding, we would have to deal with extra cases where the extractor would produce a commitment collision. Assume that we can extract two successful transcripts T_1 and T_2 (meaning that they pass all the tests of the $\mathcal{V}$) with the same commitment h, and with different challenges $ch_{T_1} = i^*_{T_1} \neq i^*_{T_2} = ch_{T_2}$. Either the revealed shares are not consistent between the two transcripts, and then we find a hash collision, or the openings are unique and hence the underlying witness is uniquely defined. Let $[\![x]\!]^{(1)}$ be the sharing of x from the first transcript, and $[\![x]\!]^{(2)}$ the sharing from the second transcript. Then, define $[\![x]\!]_i = [\![x]\!]^{(1)}_i$ if $i \neq i^*_{T_1}$, and $[\![x]\!]_{i^*_{T_1}} = [\![x]\!]^{(2)}_{i^*_{T_1}}$. This witness $\sum_{i=1}^{N} [\![x]\!]_i$ enables us to build a solution for the DDLP instance.

Now, we construct the extractor that outputs these transcripts, and then we analyze it. This has been already done in [FJR23,FMRV22], we provide a summary. We assume that this extractor only gets transcripts with consistent shares since otherwise, the extractor would find a hash collision. We denote by R_h the randomness of $\tilde{\mathcal{P}}$ which is used to generate the initial commitment h, and r_h is a possible realization of R_h. Then we describe the following extractor $\mathcal{E}$:

```
Repeat +∞ times:
  Run P̃ with honest V to get transcript T_1
  If T_1 is not a successful transcript, go to the next iteration
  Do η times:
    Run P̃ with honest V and same r_h as T_1 to get transcript T_2
    If T_2 is a successful transcript, i*_{T_1} ≠ i*_{T_2} and (T_1, T_2) reveals a good witness,
      Return (T_1, T_2)
```

In the following, we analyze the complexity of $\mathcal{E}$ by computing the average number of calls to $\tilde{\mathcal{P}}$. We denote $\mathsf{succ}_{\tilde{\mathcal{P}}}$ the event that $\tilde{\mathcal{P}}$ succeeds in convincing $\mathcal{V}$, and by hypothesis $\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}] = \tilde{\varepsilon}$. Since $\tilde{\varepsilon} > \varepsilon$, there exists some $\alpha \in (0,1)$ such that $(1-\alpha)\tilde{\varepsilon} > \varepsilon$. Let r_h be a possible realization of R_h. We will say that r_h is *good* if

$$\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}} \mid R_h = r_h] \geq (1-\alpha)\,\tilde{\varepsilon}\ . \tag{2}$$

By the Splitting lemma we have

$$\Pr[R_h \text{ good} \mid \mathsf{succ}_{\tilde{\mathcal{P}}}] \geq \alpha\ . \tag{3}$$

Let's assume we sample a successful transcript T_1 with some realization r_h of R_h. Assume r_h is good. There exists a successful transcript T_2 with $i^*_{T_2} \neq i^*_{T_1}$ because $\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}} \mid R_h = r_h] \geq (1-\alpha)\,\tilde{\varepsilon} > \varepsilon = \frac{1}{N}$. Hence, there exists a unique and well-defined witness $[\![x]\!]$ corresponding to these transcripts. Let us assume that the witness $[\![x]\!]$ in T_1 is a bad witness (i.e. $g^{h^x} \not\equiv y \bmod q$ where $x := \sum_i [\![x]\!]_i$). To have a successful transcript, the best strategy for the prover is to cheat for the simulation of one party, and the probability of being successful

is at most $1/N$. Thus, $\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}} \mid R_h = r_h] \leq \frac{1}{N} = \varepsilon$, meaning that r_h is *not* good. By contraposition, we get that if r_h is good, then $[\![x]\!]$ is a good witness.

Now, given a good R_h, we estimate the probability that one iteration of the inner loop finds a successful transcript T_2 (denoted as $\mathsf{succ}^{T_2}_{\tilde{\mathcal{P}}}$) such that $i^*_{T_1} \neq i^*_{T_2}$:

$$\begin{aligned}
&\Pr[\mathsf{succ}^{T_2}_{\tilde{\mathcal{P}}} \cap (i^*_{T_1} \neq i^*_{T_2}) \mid R_h \text{ good}] \\
&\qquad = \Pr[\mathsf{succ}^{T_2}_{\tilde{\mathcal{P}}} \mid R_h \text{ good}] - \Pr[\mathsf{succ}^{T_2}_{\tilde{\mathcal{P}}} \cap (i^*_{T_1} = i^*_{T_2}) \mid R_h \text{ good}] \\
&\qquad \overset{2}{\geq} (1-\alpha)\tilde{\varepsilon} - \Pr[i^*_{T_1} = i^*_{T_2} \mid R_h \text{ good}] \\
&\qquad = (1-\alpha)\tilde{\varepsilon} - \Pr[i^*_{T_1} = i^*_{T_2}] \\
&\qquad = (1-\alpha)\tilde{\varepsilon} - 1/N \\
&\qquad \geq (1-\alpha)\tilde{\varepsilon} - \varepsilon \ .
\end{aligned}$$

Let define $p_0 := (1-\alpha)\,\tilde{\varepsilon} - \varepsilon$. By running $\tilde{\mathcal{P}}$ η times with the same r_h as for the successful transcript T_1, we obtain a second successful non-colliding transcript T_2 with probability at least $1/2$ when

$$\eta \approx \frac{\ln(2)}{\ln\left(\frac{1}{1-p_0}\right)} \leq \frac{\ln(2)}{p_0} \ . \tag{4}$$

Let C denotes the total number of calls to $\tilde{\mathcal{P}}$ from the extractor. While entering a new iteration, the extractor makes one call to $\tilde{\mathcal{P}}$ to obtain T_1. Then, if T_1 is not successful, which occurs with probability $(1 - \Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}])$, the extractor continues to the next iteration and makes an average of $\mathbb{E}[C]$ calls to $\tilde{\mathcal{P}}$. Otherwise, if T_1 is successful, which occurs with probability $\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}]$, either r_h is good which occurs with probability α (equation 3), and the extractor makes at most η calls to $\tilde{\mathcal{P}}$ in the inner loop of $\mathcal{E}$ and output a pair (T_1, T_2) with probability $1/2$. Or the extractor makes η calls to $\tilde{\mathcal{P}}$ in the inner loop of $\mathcal{E}$ without stopping, with probability at most $(1 - \frac{\alpha}{2})$. The average number of calls to $\tilde{\mathcal{P}}$ hence satisfies:

$$\begin{aligned}
\mathbb{E}[C] &\leq 1 + (1 - \Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}])\,\mathbb{E}[C] + \Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}]\left(\eta + \left(1 - \frac{\alpha}{2}\right)\mathbb{E}[C]\right) \\
&\leq 1 + (1-\tilde{\varepsilon})\,\mathbb{E}[C] + \tilde{\varepsilon}\left(\eta + \left(1 - \frac{\alpha}{2}\right)\mathbb{E}[C]\right) \\
&\leq 1 + \tilde{\varepsilon}\,\eta + \mathbb{E}[C]\left(1 - \frac{\tilde{\varepsilon}\,\alpha}{2}\right) \\
&\leq \frac{2}{\alpha\,\tilde{\varepsilon}}\,(1 + \tilde{\varepsilon}\eta) \\
&\overset{4}{\leq} \frac{2}{\alpha\,\tilde{\varepsilon}}\left(1 + \tilde{\varepsilon}\,\frac{\ln(2)}{(1-\alpha)\,\tilde{\varepsilon} - \varepsilon}\right)
\end{aligned}$$

By picking $\alpha = \frac{1}{2}\left(1 - \frac{\varepsilon}{\tilde{\varepsilon}}\right)$, we get

$$\mathbb{E}[C] \leq \frac{4}{\tilde{\varepsilon} - \varepsilon}\left(1 + \tilde{\varepsilon}\,\frac{2\,\ln(2)}{\tilde{\varepsilon} - \varepsilon}\right) \ .$$

Zero-knowledge. We build a PPT simulator $\mathcal{S}$ (*i.e.* an algorithm that outputs transcripts that are indistinguishable from real transcripts without knowing a valid witness) that has oracle access to some PPT $\tilde{\mathcal{V}}$, and works as follows:

1. Sample a challenge $i^* \xleftarrow{\$} [1, N]$.
2. Sample $\mathsf{mseed} \xleftarrow{\$} \{0,1\}^\lambda$.
3. Compute parties' seeds $(\mathsf{seed}_1, \rho_1), \ldots, (\mathsf{seed}_N, \rho_N)$ with TreePRG(mseed).
4. For each party $i \in [1, N] \backslash \{i^*\}$:
   - $[\![x]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$
   - $\mathsf{com}_i = \mathsf{Com}(\mathsf{seed}_i; \rho_i)$
5. Simulate the forward-backward execution with $\{[\![x]\!]_i\}_{i \neq i^*}$, and get $\{y_i\}_{i \in [1,N]}$.
6. Sample a random commitment com_{i^*}
7. $\Delta x \xleftarrow{\$} \mathbb{F}_p^\times$
8. Call $\tilde{\mathcal{V}}$ with the hash digest h of $\Delta x, \{y_i\}_{i \in [1,N]}$, and of the commitments of the seed and associated randomness of each party. Get a challenge $\tilde{i}^*$. If $\tilde{i}^* \neq i^*$, then $\mathcal{S}$ restarts the simulation from scratch.
9. Output the transcript

$$\left(h, (\mathsf{seed}_i, \rho_i)_{i \neq i^*}, \mathsf{com}_{i^*}, \{y_i\}_{i \in [1,N]}, \Delta x\right) .$$

The output transcript is identically distributed to the genuine transcript except for the commitment of the party i^*. Distinguishing them means breaking the commitment hiding property or the PRG security. The above simulator $\mathcal{S}$ is a PPT algorithm since the challenge set $[1, N]$ (from which i^* is sampled) has a size that is polynomial in the security level.

B Zero-Knowledge Argument of Knowledge of an RSA Plaintext

First, we formally describe the MPC protocol 8 that is realized by the parties emulated by the prover, and that has been mentioned in the introduction of section 3.

Input: RSA ciphertext, modulus and public exponent (y, n, e). A multiplicative sharing $x = \prod_{j=1}^{N} \langle x \rangle_j \bmod n$ of some element $x \in \mathbb{Z}/n\mathbb{Z}$. **Output:** 1 if $x^e = y \bmod n$, 0 otherwise.
1. Each party i locally computes their multiplicative share $\langle x \rangle_i^e$ and broadcast it. 2. If $x^e = y \bmod n$ parties output 1, otherwise 0.

Protocol 8: MPC protocol Π_{RSA}

Now, we slightly adapt the protocol 7 to get a zero-knowledge argument of knowledge protocol of an RSA plaintext. We consider the protocol 7 where

protocol 8 is plugged in the red part and with two minor modifications. Instead of committed to $\{y_i\}_{i\in[1,N]}$, $\mathcal{P}$ commits to $\langle y\rangle$. Therefore, instead of applying the forward-backward technique, $\mathcal{V}$ reruns the MPC protocol 8 for all the parties $i \neq i^*$ and sets $y_{i^*} = y/\prod_{i\neq i^*}\langle y\rangle_i$.

Proofs of security are identical to those in Theorem 4. The completeness and the soundness proofs are the same (both arguments of knowledge rely on the same protocol 7). Step 5 in the simulator $\mathcal{S}$ has to be substituted by "Simulate the computation of all the parties $i \neq i^*$ to get $\{\langle x\rangle_i^e\}_{i\neq i^*}$, and fix $\langle x\rangle_{i^*}^e := y/\prod_{i\neq i^*}\langle x\rangle_i^e$".

C Proof of Theorem 1

Proof (Theorem 1). Completeness. For any sampling of the random coins of $\mathcal{P}$ and $\mathcal{V}$, if the computation described in the protocol 2 combined with protocol 3 is honestly performed, all the checks of $\mathcal{V}$ pass. The completeness is hence perfect.

Soundness. To prove the special soundness, one builds an efficient knowledge extractor that returns 3 specific transcripts, from which we can extract a solution of the DDLP instance. This extractor has rewindable black-box access to $\tilde{\mathcal{P}}$. Assume that we can get three transcripts $T_i = (h, J^{(i)}, \textsc{Rsp}_1^{(i)}, \{\ell_j^{(i)}\}_{j\in J^{(i)}}, \textsc{Rsp}_2^{(i)})$ for $i \in \{1,2,3\}$ from $\tilde{\mathcal{P}}$ with the same first commitment. We additionally require that there exists $j_0 \in (J^{(1)}\cap J^{(2)})\setminus J^{(3)}$ such that $\ell_{j_0}^{(1)} \neq \ell_{j_0}^{(2)}$. Moreover, T_1 and T_2 are supposed to be successful transcripts (i.e. which pass all the tests of $\mathcal{V}$). Finally, we assume that $\mathsf{seed}^{[j_0]}$ from $\textsc{Rsp}_1^{(3)}$ is consistent with the $(x^{[j_0]}, r^{[j_0]}, s^{[j_0]})$ from T_1 and T_2.

We show how to extract a solution of the DDLP instance (y, g, h) from the three transcripts. First, we can assume that all the revealed shares are mutually consistent between the three transcripts. Otherwise, we find a hash collision (since they have the same first commitment). Thus, we know all the shares for the iteration j_0 from T_1 and T_2. For the sake of clarity, we only consider the variables of the j_0-th iteration, and this notation is omitted in the following. Consider $x' := \sum_{j=1}^N [\![x]\!]_j \bmod p$ as a natural candidate solution for x. Then, following the MPC protocol 3 we compute:

- $h^{x'} = h^{\sum_{j=1}^N [\![x]\!]_j} = \prod_{j=1}^N h^{[\![x]\!]_j} = \prod_{j=1}^N \langle h^x\rangle_j \bmod q$
- the broadcasting of $\langle\alpha\rangle = \frac{\langle h^x\rangle}{\langle s\rangle}$ i.e. $\alpha = \frac{h^x}{s} \bmod q$
- an additive sharing of h^x via $\alpha[\![r]\!] = \frac{h^x}{s}[\![r]\!] = [\![h^x]\!]$, since from the checked equations at the end of T_3 we get that $r = s$.
- $y = \prod_{j=1}^N \langle y\rangle_j \bmod q$ with $\langle y\rangle_j = g^{[\![h^x]\!]_j} \bmod q$.

Hence, $g^{h^{x'}} = g^{\prod_{j=1}^N \langle h^x\rangle_j} = g^{\sum_{j=1}^N [\![h^x]\!]_j} = \prod_{j=1}^N g^{[\![h^x]\!]_j} = \prod_{j=1}^N \langle y\rangle_j = y \bmod q$. Therefore, x' is a solution of the considered DDLP.

Our protocol 2 has the same structure as the protocol 5 in [FJR23], with the same first challenge for revealing $M - \tau$ out of M cut-and-choose phases, and

the second challenge for opening $N-1$ views. Hence, we can use the extractor described in appendix E of [FJR23] for producing the above three transcripts T_1, T_2, T_3 by calling in average at most

$$\frac{4}{\tilde{\varepsilon}-\varepsilon}\left(1+\tilde{\varepsilon}\frac{8M}{\tilde{\varepsilon}-\varepsilon}\right)$$

times $\tilde{\mathcal{P}}$ (the analysis of the average number of calls to $\tilde{\mathcal{P}}$ is also identical).

Honest-verifier zero-knowledge. We build a PPT simulator $\mathcal{S}$ (*i.e.* an algorithm that outputs transcripts that are indistinguishable from real transcripts without knowing a valid witness) given random challenges J and L (because we assume an honest verifier), and works as follows:

1. Sample $J \xleftarrow{\$} \{J \subset [1,M]; |J| = \tau\}$ and $L = \{\ell_e\}_{e\in J} \xleftarrow{\$} [1,N]^\tau$
2. Sample $\mathsf{mseed}^{[0]} \xleftarrow{\$} \{0,1\}^\lambda$
3. $(\mathsf{mseed}^{[e]})_{e\in[1,M]} \leftarrow \mathrm{TreePRG}(\mathsf{mseed}^{[0]})$
4. For $e \in [1,M]\backslash J$, follow honestly the protocol and deduce h_e.
5. For $e \in J$:
   - Compute $(\mathsf{seed}_1^{[e]}, \rho_1^{[e]}), \ldots, (\mathsf{seed}_N^{[e]}, \rho_N^{[e]})$ with $\mathrm{TreePRG}(\mathsf{mseed}^{[e]})$.
   - For each party $j \in [1,N]\backslash\{\ell_e\}$:
     - $([\![x^{[e]}]\!]_j, [\![r^{[e]}]\!]_j, \langle s^{[e]}\rangle_j) \leftarrow \mathrm{PRG}(\mathsf{seed}_j^{[e]})$
     - $\mathsf{com}_j^{[e]} = \mathsf{Com}(\mathsf{seed}_j^{[e]}; \rho_j^{[e]})$
   - Sample $\Delta x^{[e]} \xleftarrow{\$} \mathbb{F}_p, [\![x^{[e]}]\!]_{\ell_e} \xleftarrow{\$} \mathbb{F}_p, [\![r^{[e]}]\!]_{\ell_e} \xleftarrow{\$} \mathbb{F}_q, \langle s^{[e]}\rangle_{\ell_e} \xleftarrow{\$} \mathbb{F}_q^\times$
   - $\Delta s^{[e]} = \sum_{j=1}^N [\![r^{[e]}]\!]_j / \prod_j \langle s\rangle_j^{[e]} \bmod q$
   - $\alpha^{[e]} = h^{\sum_{j=1}^N [\![x^{[e]}]\!]_j + \Delta x^{[e]}} / (\Delta s^{[e]} \prod_{j=1}^N \langle s^{[e]}\rangle_j) \bmod q$
   - $\langle g^{h^{x^{[e]}}}\rangle_j = g^{\alpha^{[e]}[\![r^{[e]}]\!]_j} \bmod q$
   - Adapt the output of the party ℓ_e: $\langle g^{h^{x^{[e]}}}\rangle_{\ell_e} = y / \prod_{j\neq\ell_e} \langle g^{h^{x^{[e]}}}\rangle_j \bmod q$
   - Sample a random commitment $\mathsf{com}_{\ell_e}^{[e]}$
   - Compute
     - $h_e = \mathcal{H}_1(\Delta s^{[e]}, \mathsf{com}_1^{[e]}, \ldots, \mathsf{com}_n^{[e]})$
     - $h'_e = \mathcal{H}_3(\Delta x^{[e]}, \langle g^{h^{x^{[e]}}}\rangle, \alpha^{[e]})$
6. Compute
   - $h = \mathcal{H}_2(h_1, \ldots, h_M)$
   - $h' = \mathcal{H}_4((h'_e)_{e\in J})$
7. Outputs the transcript

$$\left(h, h', (\mathsf{mseed}^{[e]})_{e\in[1,M]\backslash J}, ((\mathsf{seed}_i^{[e]}, \rho_i^{[e]})_{i\neq\ell_e}, \mathsf{com}_{\ell_e}^{[e]}, \Delta x^{[e]}, \Delta s^{[e]}, \alpha^{[e]})_{e\in J}\right) .$$

The distribution of the output transcript is identical to a real one, except for the commitment of the party ℓ_e in each execution $e \in J$. Distinguishing them means breaking the commitment hiding property or the PRG security.

D Proof of Theorem 3

Proofs of completeness, special soundness, and honest-verifier zero-knowledge are identical to those in annex C (they both use the same protocol 2), with a slight adaptation. In the soundness, we adapt to the FIP the proof that the natural solution x' extracted from the first two transcripts satisfies $f(x') = y \bmod p$. In the simulator $\mathcal{S}$, step 5 has to be adapted to the FIP.

E Proof of Theorem 2

Proof. Completeness. For any sampling of the random coins of $\mathcal{P}$ and $\mathcal{V}$, if the computation described in the protocol 2 (with the slight adaptation described in section 5 for the first challenge and the rejection rule) combined with MPC protocol 5 is honestly performed and if there is *no abort*, all the checks of $\mathcal{V}$ pass. Since the probability of abortion is $1 - \left(1 - \frac{q-1}{A}\right)^{\tau n}$ (see section 2), the completeness probability is hence equals to $\left(1 - \frac{q-1}{A}\right)^{\tau n}$.

Special soundness. Compared to the DDLP identification scheme, the ongoing scheme has a larger first challenge $J \cup \{g \xleftarrow{\$} \mathbb{F}_{q'}\}$ where J is the cut-and-choose challenge, but the second challenge is the same. Consider the same extractor $\mathcal{E}$ as in annex C which outputs some specific three transcripts T_1, T_2, T_3. Define $J^{(i)} \cup \{g^{(i)} \xleftarrow{\$} \mathbb{F}_{q'}\}$ the first challenge in transcript T_i for $i \in [1, 3]$. Thus, $J^{(1)}, J^{(2)}$ and $J^{(3)}$ satisfy the conditions enumerated in proof of Theorem 1, but $g^{(1)}, g^{(2)}$ and $g^{(3)}$ are random. As detailed in proof of Theorem 1, T_1 and T_2 aims to recover some natural candidate solution x'. T_3 provides a valid couple of sharing $(\llbracket r \rrbracket, \langle r \rangle)$. All together, we can reconstruct the polynomial $f_{x',v}(X)$. Since T_1 is successful, $f_{x',v}(g^{(1)}) = 0 \bmod q'$.

Honest-verifier zero-knowledge. We detail the change to carry out in the simulator $\mathcal{S}$ from annex C. First, we shall sample $g \xleftarrow{\$} \mathbb{F}_{q'}$ during step 1. We adapt step 5 accordingly to MPC protocol 5. Finally, we add "Abort with probability $1 - \left(1 - \frac{q-1}{A}\right)^{n\tau}$" before step 7. The final analysis still holds.