Π: A Unified Framework for Verifiable Secret Sharing

Karim Baghery

COSIC, KU Leuven, Leuven, Belgium
firstname.lastname@kuleuven.be
May 23, 2024

Abstract. An (n, t)-Verifiable Secret Sharing (VSS) scheme allows a dealer to share a secret among n parties, s.t. all the parties can verify the validity of their shares and only a set of them, i.e., more than t, can access the secret. In this paper, we present $\mathbf{\Pi}$, as a unified framework for building VSS schemes in the honest majority setting. Notably, $\mathbf{\Pi}$ does not rely on homomorphic commitments; instead requires a random oracle and any commitment scheme that extra to its core attributes hiding and binding, it might be homomorphic and/or post-quantum (PQ) secure.

(i) When employing Discrete Logarithm (DL)-based commitments, $\mathbf{\Pi}$ enables the construction of two novel VSS schemes in the RO model, named $\mathbf{\Pi_P}$ and $\mathbf{\Pi_F}$. Compared to the well-known Pedersen and Feldman VSS schemes, both $\mathbf{\Pi_P}$ and $\mathbf{\Pi_F}$ require $O(1)$ (resp. $O(t)$) exponentiations in the verification (resp. reconstruction) process, as opposed to $O(t)$ (resp. $O(t^2)$), albeit at the expense of a constant factor slower sharing and increased communication.

(ii) By instantiating $\mathbf{\Pi}$ with a hash-based commitment, we obtain a novel PQ-secure VSS scheme, labeled $\mathbf{\Pi_{LA}}$ (pronounced [paɪˈla][1]). $\mathbf{\Pi_{LA}}$ outperforms the recent protocol by Atapoor, Baghery, Cozzo, and Pedersen from Asiacrypt'23 by a constant factor in *all* metrics. $\mathbf{\Pi_{LA}}$ can also be seen as an amplified version of the *simple* VSS scheme, proposed by Gennaro, Rabin, and Rabin at PODC'98.

(iii) Building upon $\mathbf{\Pi_F}$, we construct a Publicly VSS (PVSS) scheme, labeled $\mathbf{\Pi_S}$, that can be seen as a new variant of Schoenmakers' scheme from Crypto'99. To this end, we first define the Polynomial Discrete Logarithm (PDL) problem, as a generalization of DL and then build a variant of the Schnorr Proof of Knowledge (PoK) scheme based on the new hardness assumption. We think the PDL relation and the associated PoK scheme can be independently interesting for Shamir-based threshold protocols.

We believe $\mathbf{\Pi}$ is general enough to be employed in various contexts such as lattices, isogenies, and an extensive array of practical use cases.

Keywords: Verifiable Secret Sharing · Polynomial Discrete Logarithm

[1] In Turkish, 'Pay' (pronounced [paɪ]) is a noun for *Share* and 'Payla' means *Share it*.

1 Introduction

Secret sharing schemes have become foundational tools in threshold cryptography and secure multi-party computation. These schemes facilitate the secure distribution of sensitive information among multiple parties, allowing only qualified shareholders to reconstruct the original secret collaboratively.

Traditional secret sharing schemes, like Shamir's protocol [25], assume the presence of honest parties but lack provisions for security against malicious ones. To address this concern, Verifiable Secret Sharing (VSS) schemes [13, 15] have been developed, aiming to withstand various attacks, including incorrect share distribution by the dealer and malicious behavior by parties during the reconstruction phase. A Non-Interactive VSS (NI-VSS) scheme allows a dealer to non-interactively (in the happy path) distribute a secret among n parties, such that all the parties can verify the validity of their shares, and similar to a typical secret sharing scheme, only a specific number of them can access the secret. Numerous VSS schemes are built on regular secret-sharing protocols, adding verifiability features on top [1, 4, 13, 15, 18–20, 22, 24]. Many of known VSS schemes like Feldman [15] and Pedersen [22] use Shamir secret sharing and exploit the homomorphic property of the Discrete Logarithm (DL) and Pedersen commitment to achieve verifiability. To this end, the dealer sends the shares securely to parties and publishes the homomorphic commitments to the coefficients of the underlying secret polynomial. Then, they leverage the homomorphic property of the DL group to convince the shareholders that the secret sharing is performed correctly. Publicly Verifiable Secret Sharing (PVSS) schemes additionally allow an external verifier to verify the validity of the distributed shares (that are encrypted under the public key of the shareholders) in a single round [9, 17, 20, 24, 27].

In [18, Section 2], Gennaro, Rabin, and Rabin (GRR) proposed a *simple* VSS scheme for $n \geq 2t + 1$ that does not need homomorphic commitments. However, their construction achieves a weaker security in terms of reconstruction. Namely, from the n distributed shares, any different $t + 1$ honest shareholders might reconstruct a different secret. The reason is that in their construction [18, Fig. 1], the dealer does not prove that all the shares are generated using a unique degree-t polynomial $f(X)$. To deal with this concern, they propose an amplified version of their simple construction, that uses homomorphic commitments (i.e., Pedersen commitment) and achieves the stronger notion of extractability, which guarantees that any different $t + 1$ honest shareholders reconstruct a unique secret $f(0)$. The VSS schemes in plain model, which are based on homomorphic commitments, e.g., [4, 15, 18, 22], have at the best $O(t\lambda)$ communication complexity and require $O(n)$ or $O(t)$ exponentiations, in the sharing and verification sides, respectively, where λ denotes the security parameter. In [2, Section 3.1], Backes, Kate, and Patra proposed the first VSS scheme for $n \geq 2t + 1$, that do not require homomorphic commitments. However, their construction uses bivariate polynomials [4] to achieve verifiability, that requires $O(n^2)$ commitments and $O(n^2\lambda)$ bits of broadcast, $O(n^2\lambda)$ bits of private communication in the sharing phase and also imposes $O(n^2\lambda)$ broadcasts in the reconstruction phase. In an elegant recent work, Atapoor, Baghery, Cozzo, and Pedersen (ABCP) [1] introduced the first Post-Quantum (PQ) secure VSS scheme for $n \geq 2t+1$ which uses a quantum Random Oracle (RO) and a (collapsing) hash-based commitment scheme and boasts computational and communication costs of $O(n)$ and $O(n\lambda)$, respectively. Notably, their scheme relies solely on *lightweight* operations, such as hashing and polynomial evaluations, making it significantly more efficient than previous schemes in this setting. In a different setting, Shoup and Smart [26] also recently unveiled a novel lightweight asynchronous VSS scheme that similarly employs a random oracle (or a random beacon) and *lightweight* cryptographic operations, specifically hashing and polynomial evaluations. Shoup and Smart's scheme is tailored for the asynchronous communication model and necessitates at least 2/3 of the participants to be honest. Both the mention works [1, 26] have used hash-based commitments to build PQ secure and lightweight VSS schemes. Our study is in the synchronous setting, assumes that the majority of parties are honest, and aims to harness the strengths of both lightweight and heavyweight cryptography. More precisely, we aim to construct new VSS schemes that either improve existing constructions in terms of efficiency or by sacrificing PQ security (and lightweightness), can achieve unique features such as Information-Theoretical (IT) unpredictability (such as Pedersen VSS [22]) or public verifiability (as in Schoenmakers PVSS [24]). These features cannot be achieved in VSS schemes that use hash-based commitments.

The starting point for ABCP [1] is the construction of a PQ-secure Non-Interactive Threshold Zero-Knowledge (NI-TZK) proof scheme for the following n-distributed relations $R_1, \ldots, R_n$:

$$R_i = \{(f_i, f(X)) | f(i) = f_i\}, \quad i = 1, \cdots, n. \tag{1}$$

Here $f(X)$ represents a witness polynomial in X of degree (at most) t with coefficients defined over the ring $\mathbb{Z}_N$, and f_i are the shares received by n parties.

NI-TZK proofs are formally defined and studied by Boneh et al. [7]. In a NI-TZK proof scheme, a prover aims to convince n verifiers, holding a piece of the statement, e.g., f_i, that the main statement, e.g., $f_1 \parallel \cdots \parallel f_n$ (hidden from an individual verifier) belongs to a specific language. Similar to the typical cases, such proof systems must be complete, meaning that if the main statement is in language, an honest prover will be able to convince honest verifiers. They should satisfy soundness, meaning that if the main statement is not in the language, then all verifiers will reject the verification except for a negligible probability. However, in some cases a subset of verifiers, e.g., up to t of them, may be malicious and collude with an adversarial prover. Finally, they need to satisfy a variant of ZK, so-called *Threshold ZK* (TZK)[2], as introduced by Boneh et al. [7]. TZK implies that any subset of the verifiers up to t, should learn no additional information about the main statement, beyond their own shares of statement and the fact that the main statement belongs to the language.

ABCP [1] coined the term "Shamir relation" to describe the n-distributed relation in Eq. (1). Then, they used the proposed PQ-secure NI-TZK for the

[2] We adopt the term "Threshold ZK" from [1] to refer this variant of zero-knowledge, and it is called "Strong ZK" in [7].

Shamir relation and built an extremely efficient computationally secure VSS scheme in the majority-honest setting, which uses hash functions and polynomial evaluations. Drawing upon the NI-TZK proofs, they also introduced a new approach for secret reconstruction in VSS schemes. In certain scenarios, this approach can lead to the development of more efficient threshold protocols, such as Distributed Key Generation (DKG) protocols and threshold signatures.

1.1 Our Contributions

$\mathbf{\Pi}$: A Unified Framework for VSS Schemes. We present a unified framework $\mathbf{\Pi}$ designed for constructing VSS schemes in the honest majority setting. The framework is based on Shamir secret sharing and draws inspiration from the VSS scheme recently introduced by Atapoor, Baghery, Cozzo, and Pedersen [1] and the *simple* construction presented by Gennaro, Rabin, and Rabin [18]. In its general form, $\mathbf{\Pi}$ uses a (classic or quantum) random oracle, and does not necessarily need a homomorphic commitment scheme (as in Feldman [15] and Pedersen [22] schemes). Nevertheless, the option remains to instantiate it with homomorphic commitments, to construct new VSS schemes that are more efficient than current schemes, or achieve unique properties such as Public Verifiability (PV) [24] and IT unpredictability [22]. At its core, the framework boasts a general and efficient construction, enabling the construction of VSS schemes with diverse features.

Sharing. In the main construction of $\mathbf{\Pi}$, given a secret f_0, a hiding and binding commitment scheme $\mathcal{C}$, and a (classic or quantum) random oracle $\mathcal{H}$, the dealer proceeds as follows: 1) Does Shamir secret sharing: samples a random degree-t polynomial $f(X)$ with free term f_0 and sets $f_i = f(i)$ for $i = 1, \ldots, n$. 2) Samples another random degree-t polynomial $r(X)$ and sets $r_i = r(i)$ for $i = 1, \ldots, n$. 3) Sets $c_i = \mathcal{C}(f_i, r_i)$ (or $c_i = \mathcal{C}((f_i, r_i), \gamma_i)$, where $\gamma_i = \gamma(i)$ are evaluations of a new random degree-t polynomial $\gamma(X)$ in point i) for $i = 1, \cdots, n$ and $z(X) = r(X) + d \cdot f(X)$; where $d = \mathcal{H}(c_1, \ldots, c_n)$. Finally, securely transmit f_i (and the randomizer γ_i employed in the commitment $\mathcal{C}$, if applicable) to party P_i, and publish $\pi_{Share} := (z(X), c_1, \ldots, c_n)$. In general, sharing can be as efficient as performing two (or three) Shamir secret sharing in addition to n commitments.

Verification. In the general form, given $(f_i, z(X), \{c_i\}_{i=1}^n)$, to verify the received share f_i (and γ_i if applicable), party P_i checks if $z(X)$ is a degree-t polynomial, computes $d = \mathcal{H}(c_1, \ldots, c_n)$ and checks if $c_i = \mathcal{C}(f_i, z(i) - d \cdot f_i)$ (or $c_i = \mathcal{C}((f_i, z(i) - d \cdot f_i), \gamma_i)$). If the checks do not pass, P_i broadcasts a complaint against the dealer. If player P_j broadcasted a complaint, then the dealer broadcasts the share f_j (and γ_j if applicable), such that $c_j = \mathcal{C}(f_j, z(j) - d \cdot f_j)$ (or $c_j = \mathcal{C}((f_j, z(j) - d \cdot f_j), \gamma_j)$). If the dealer does not follow the protocol, he is disqualified, otherwise the protocol continues as usual.

Reconstruction. Each shareholder broadcasts their shares f_i (and γ_i if applicable). Subsequently, the disclosed shares are verified using the verification process of the target VSS scheme. Following verification, $t + 1$ valid shares are utilized for the reconstruction of the secret polynomial $f(X)$ (and $\gamma(X)$ if applicable), resulting in the main secret $s = f(0)$.

Security. We show that, given a secure commitment scheme, in the majority-honest scenario where $t+1$ of the parties are honest, with $n \geq 2t+1$, the general construction satisfies *verifiability* (implied by soundness against prover and t malicious verifiers), and unpredictability (implied by threshold zero-knowledge) in the random oracle model. These properties ensure that a malicious dealer cannot convince honest parties except with a negligible probability. Consequently, any set of $t+1$ honest parties will be able to collectively reconstruct a *unique* secret $f_0 = f(0)$ in the reconstruction phase.

It's crucial to mention that when we use $\mathbf{\Pi}$ to build VSS schemes aiming to satisfy computational unpredictability, the dealer commits to *random* values $\{f_i, r_i\}_{i=1}^{n}$, where possess sufficient entropy. Consequently, there is no need for an additional randomizer in the commitment process, even if $z(i) = r_i + d \cdot f_i$ and d are public. However, when we use $\mathbf{\Pi}$ to build an VSS scheme that satisfies IT unpredictability, like Pedersen scheme, i.e., in case the secret f_0 does not have enough entropy, the dealer must use an additional randomizer in the commitment (e.g., an extension of Pedersen commitment with three random generators from [8]). Then, the use of a separate randomizer, i.e., γ_i, becomes essential. As can be seen, the strength of the new framework lies in its simplicity and generality and it is flexible enough to be tailored for constructing various VSS schemes with distinct properties. In addition to Shamir's secret sharing, it only requires a secure commitment scheme and a classic/quantum RO. Commitment schemes and classic/quantum RO can be efficiently constructed respectively using various fundamental primitives (such one-way functions) and hash functions, rendering $\mathbf{\Pi}$ a unified framework for building VSS schemes that can also achieve IT unpredictability, public verifiability and/or PQ security. Leveraging $\mathbf{\Pi}$, we present a range of novel and efficient VSS schemes that, in general, can outperform current alternatives, particularly in the verification and reconstruction phases.

New VSS Schemes from DL-Based (Homomorphic) Commitments. By instantiating $\mathbf{\Pi}$ with standard Pedersen commitment scheme [22], we introduce an efficient alternative for the well-known Feldman scheme [15], labeled as $\mathbf{\Pi_F}$. When dealing with an extension of Pedersen commitment (i.e., using three random generators, instead of two) $\mathbf{\Pi}$ enables the construction of a novel IT-secure VSS scheme referred to as $\mathbf{\Pi_P}$. Similar to the Pedersen scheme, in $\mathbf{\Pi_P}$, fewer than t parties learn nothing about the main secret, ensuring IT unpredictability.

In terms of efficiency, both $\mathbf{\Pi_F}$ and $\mathbf{\Pi_P}$ require $O(1)$ (resp. $O(t)$) exponentiations in the verification (resp. reconstruction) process, as opposed to $O(t)$ (resp. $O(t^2)$) in the Feldman [15] and Pedersen [22] schemes, where t represents the threshold parameter. This improvement comes at the cost of a constant factor of overhead in the sharing phase and communication and using a random oracle.

More Efficient VSS Scheme from Hash Functions. Through the instantiation of $\mathbf{\Pi}$ using a quantum RO and a non-homomorphic commitment scheme, such as those based on hash functions, we obtain a novel PQ-secure VSS scheme, named $\mathbf{\Pi_{LA}}$. It can be viewed as an alternative to the recent scheme by ABCP [1], which similarly employs a quantum RO and a hash-based commitment scheme. Compared to the ABCP VSS scheme [1], $\mathbf{\Pi_{LA}}$ outperforms by constant factor in terms of all efficiency metrics. From a different view, $\mathbf{\Pi_{LA}}$ can be seen as an amplified version of the *weak* VSS scheme proposed by GRR in [18, Section 2], as their *simple* construction also uses non-homomorphic commitments to commit (f_i, r_i) for $i = 1, \ldots, n$. However, compared to their *weak* VSS scheme, $\mathbf{\Pi_{LA}}$ satisfies the stronger notion of constructability, i.e., any different set of $t+1$ honest parties will reconstruct a *unique* secret.

Generalizing DL Relation and Schnorr's Protocol Over Polynomials. The well-known Schnorr ID protocol [23] allows one to prove knowledge of witness for the relation $R_{DL} = \{(g, F), f \mid F = g^f\}$ where g is the group generator, $f \in \mathbb{Z}_q$ is the witness value, which can also be interpreted as a degree-0 polynomial with a single coefficient defined over $\mathbb{Z}_q$. In Sec. 5, we generalize R_{DL} relation over polynomials and introduce the Polynomial Discrete Logarithm (PDL) relation denoted as R_{PDL}, which is defined as follows,

$$R_{PDL} = \{(g, x_i, F_i), f(X) \mid F_i = g^{f(x_i)}\} \text{ for } i = 1, 2, \ldots, n.$$

Here, $f(X) \in \mathbb{Z}_q[X]_t$ is a (at most) degree $t \leq n-1$ witness polynomial with coefficients from $\mathbb{Z}_q$, and $\{x_i\}_{i=1}^{n}$ are n *distinct* elements from $\mathbb{Z}_q$. Then, we present a Non-Interactive Zero-Knowledge (NIZK) Proof-of-Knowledge (PoK) scheme π_{PDL} based on Schnorr's protocol, that allows a prover to prove knowledge of a witness for R_{PDL} relation. We believe π_{PDL} can be a useful proof scheme for constructing threshold protocols based on Shamir secret sharing, specifically for $n \geq 2t+1$ and $x_1 = 1, \ldots, x_n = n$.

To the best of our knowledge, this marks the first explicit definition of the PDL problem, even though it has been implicitly employed in certain prior VSS schemes and protocols [9,10,15,24]. In [10], Cascudo and David similarly defined a slightly different variant of R_{PDL} and built a sigma protocol for this variant. However, there are some issues in their proposed sigma protocol, which will be discussed in detail in Sec. 5. They also presented a probabilistic verification protocol for the R_{PDL} relation, which achieves soundness, in contrast to our proposed NIZK proof π_{PDL}. A detailed comparison of our construction with theirs is provided later in this paper.

A Novel PVSS Scheme Based on DL. Using the new NIZK PoK scheme π_{PDL}, and building upon a variant of VSS scheme $\mathbf{\Pi_F}$, we introduce a novel Publicly Verifiable Secret Sharing (PVSS) scheme, designated as $\mathbf{\Pi_S}$. $\mathbf{\Pi_S}$ serves as a more efficient alternative to Schoenmakers' PVSS scheme from Crypto'99 [24]. Compared to Schoenmakers' scheme [24], $\mathbf{\Pi_S}$ streamlines the verification complexity from $O(nt)$ to $O(n)$, accelerates the sharing phase by more than two times, and reduces the communication cost slightly. In essence, $\mathbf{\Pi_S}$ improves all efficiency metrics in Schoenmakers' scheme with no additional expense. In [9,10], Cascudo and David have proposed different variants of Schoenmakers' PVSS scheme, all reducing the verification complexity to $O(n)$. In comparison to their schemes from [9], $\mathbf{\Pi_S}$ generally demonstrates superior efficiency. Notably, its verification process is at least $3-4\times$ faster than the verification of their schemes. In their later work [10], Cascudo and David extended and optimized their RO-based scheme from [9] to support packed secret sharing. Notably, we found that $\mathbf{\Pi_S}$ shares similarities with the unpacked case of their scheme [10]. Using optimization employed in $\mathbf{\Pi_S}$, the unpacked version of their scheme can achieve the same performance to $\mathbf{\Pi_S}$. It is important to note that $\mathbf{\Pi_S}$ is developed in a generic manner, with its security reduced to the PDL problem, providing a clearer and simpler security proof. The scheme by Cascudo and David [10] relies on a sigma protocol tailored for a variant of the R_{PDL} relation. However, in their security proof of sigma protocol [10, Proposition 1], there is a lack of a clear reduction to a hardness assumption, and their security proof for special soundness lacks an extraction algorithm. We elaborate more on this matter later in Sec. 5.

Efficiency Comparisons of New Schemes. Table 1 provides a summary of performance metrics for the proposed VSS and PVSS schemes, including $\mathbf{\Pi_F}$, $\mathbf{\Pi_P}$, and $\mathbf{\Pi_{LA}}$, as well as the PVSS scheme $\mathbf{\Pi_S}$. These metrics are compared with relevant schemes from the literature [1,9,10,15,22,24]. In [21], authors proposed a VSS scheme in the Common Reference String (CRS) and RO models, which

Table 1. A comparison of new VSS and PVSS schemes with those of Feldman [15], Pedersen [22], Schoenmakers [24], Cascudo-David [9,10], and ABCP [1]. Commu.: Communication, DL: Discrete Logarithm, PDL: Polynomial Discrete Logarithm, DDH: Decisional Diffie-Hellman, DBS: Decisional Bilinear Square, IT-U: Information Theoretically Unpredictable, Classic: Classical security, PQ: Post-quantum security, RO: Random Oracle, Plain: Plain Model, BC: Broadcast, n: Number of parties, t: threshold parameter ($t \approx n/2$), $P_{\mathbb{G}}$: Pairing Operation, $E_{\mathbb{G}}$: Exponentiation in group $\mathbb{G}$, $M_{\mathbb{G}}$: Multiplication in group $\mathbb{G}$, $\mathcal{PE}$: degree-t Polynomial Evaluation, $\mathcal{H}$: Hashing, $\lvert\mathbb{G}\rvert$: $\mathbb{G}$ element size, $\lvert\mathbb{Z}_q\rvert$: $\mathbb{Z}_q$ element size, $\lvert\mathbb{Z}_N\rvert$: $\mathbb{Z}_N$ element size, $\lvert\mathcal{H}\rvert$: Output size of $\mathcal{H}$.

(P)VSS & Security	Share	Dealer's Commu.	Verification	Reconstruction
Feldman [15] (DL, Plain, Classic)	$0.5n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	Private: $1n\lvert\mathbb{Z}_q\rvert$ BC: $0.5n\lvert\mathbb{G}\rvert$	$t\ E_{\mathbb{G}}\ +$ $t\ M_{\mathbb{G}}$	$t^2\ E_{\mathbb{G}}\ +$ $t^2\ M_{\mathbb{G}}$
Sec. 4.1, $\mathbf{\Pi_F}$ (DL, RO, Classic)	$2n\ E_{\mathbb{G}}$ $2n\ \mathcal{PE}$	Private: $1n\lvert\mathbb{Z}_q\rvert$ BC: $n\lvert\mathbb{G}\rvert + 0.5n\lvert\mathbb{Z}_q\rvert$	$2\ E_{\mathbb{G}}\ +$ $1\ \mathcal{PE} + 1\ \mathcal{H}$	$2t\ E_{\mathbb{G}}\ +$ t $\mathcal{PE}$ + t $\mathcal{H}$
Pedersen [22] (DL, Plain, IT-U)	$1n\ E_{\mathbb{G}}$ $2n\ \mathcal{PE}$	Private: $2n\lvert\mathbb{Z}_q\rvert$ BC: $0.5n\lvert\mathbb{G}\rvert$	$t\ E_{\mathbb{G}}$ $+\ t\ M_{\mathbb{G}}$	$t^2\ E_{\mathbb{G}}$ $+\ t^2\ M_{\mathbb{G}}$
Sec. 4.2, $\mathbf{\Pi_P}$ (DL, RO, IT-U)	$3n\ E_{\mathbb{G}}$ $3n\ \mathcal{PE}$	Private: $2n\lvert\mathbb{Z}_q\rvert$ BC: $n\lvert\mathbb{G}\rvert + 0.5n\lvert\mathbb{Z}_q\rvert$	$3\ E_{\mathbb{G}}\ +$ $1\ \mathcal{PE} + 1\ \mathcal{H}$	$3t\ E_{\mathbb{G}}\ +$ $t\ \mathcal{PE} + t\ \mathcal{H}$
ABCP [1] (Hash, RO, PQ)	$2n\ \mathcal{H}$ $2n\ \mathcal{PE}$	Private: $1n\lvert\mathbb{Z}_N\rvert$ BC: $2n\lvert\mathcal{H}\rvert + 0.5n\lvert\mathbb{Z}_N\rvert$	$1\ \mathcal{PE}$ $+\ 3\ \mathcal{H}$	$t\ \mathcal{PE}$ $+\ 3t\ \mathcal{H}$
Sec. 4.3, $\mathbf{\Pi_{LA}}$ (Hash, RO, PQ)	$1n\ \mathcal{H}$ $2n\ \mathcal{PE}$	Private: $1n\lvert\mathbb{Z}_N\rvert$ BC: $n\lvert\mathcal{H}\rvert + 0.5n\lvert\mathbb{Z}_N\rvert$	$1\ \mathcal{PE}\ +$ $+\ 2\ \mathcal{H}$	$t\ \mathcal{PE}\ +$ $+\ 2t\ \mathcal{H}$
Sch. [24]-PVSS (DDH, RO, Classic)	$4.5n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	Private: — BC: $1.5n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$	$nt + 4n\ E_{\mathbb{G}}$ $+\ 2.5n\ M_{\mathbb{G}}$	$5t\ E_{\mathbb{G}}$ $+\ t\ M_{\mathbb{G}}$
Cas-Dav [9]-PVSS (DBS, Plain, Classic)	$2n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	Private: — BC: $2n\lvert\mathbb{G}\rvert$	$2n\ P_{\mathbb{G}}\ +$ $n\ E_{\mathbb{G}} + n\ M_{\mathbb{G}}$	$2t\ P_{\mathbb{G}}\ +$ $t\ E_{\mathbb{G}} + t\ M_{\mathbb{G}}$
Cas-Dav [9]-PVSS (DDH, RO, Classic)	$4n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	Private: — BC: $2n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$	$5n\ E_{\mathbb{G}}$ $+\ 3n\ M_{\mathbb{G}}$	$5t\ E_{\mathbb{G}}$ $+\ t\ M_{\mathbb{G}}$
Sec. 6, $\mathbf{\Pi_S}$ & [10] (PDL, DDH, RO, Classic)	$2n\ E_{\mathbb{G}}$ $2n\ \mathcal{PE}$	Private: — BC: $n\lvert\mathbb{G}\rvert + 0.5n\lvert\mathbb{Z}_q\rvert$	$2n\ E_{\mathbb{G}}\ +$ $n\ \mathcal{PE}\ +n\ M_{\mathbb{G}}$	$5t\ E_{\mathbb{G}}\ +$ $t\ M_{\mathbb{G}}$

can have $O(\lambda)$ online communication size. However, their scheme relies on strong assumptions in bilinear groups and requires a trusted CRS of size $O(n\lambda)$.

1.2 Implications of New Results

The new framework for building VSS schemes can lead to new directions in construction of VSS schemes and threshold cryptographic protocols, with considerable implications. We expect any cryptographic construction that uses either of the VSS schemes of Feldman [15], Pedersen [22], ABCP [1], or PVSS schemes [9, 10, 20, 24], or a variation of them, can be potentially affected by the new results. Considering NIST's Threshold Cryptography project[3], which seeks to standardize threshold schemes for cryptographic primitives, we think the implications of our results can extend beyond theoretical implications, offering practical promise for improving real-world (threshold) cryptographic systems. Delving into the details of revisiting concrete threshold protocols lies beyond the scope of this paper. It is worth noting that the practical usage of new VSS schemes (i.e., $\mathbf{\Pi_F}$, $\mathbf{\Pi_P}$, and $\mathbf{\Pi_{LA}}$) differs significantly from previous schemes. In practice, utilizing the new VSS schemes involves constructing a NI-TZK proof for specific languages tailored for a target application. An example construction for such NI-TZK proofs can be found in [1, Sec. 4.1].

In this vein, our generalized variant of Schnorr's NIZK PoK scheme to the PDL relation (defined in Eq. (3)) can be a useful tool for constructing threshold cryptographic protocols based on Shamir secret sharing. Notably, the idea behind it is general enough for versatile deployment across PQ secure contexts.

1.3 Outline

In Sec. 2, we present an overview of some preliminary concepts. In Sec. 3, we introduce the new framework $\mathbf{\Pi}$ devised for constructing VSS schemes. Leveraging $\mathbf{\Pi}$, in Sec. 4 we present several new VSS schemes, with different features. In Sec. 5, we generalizing DL problem and Schnorr protocol over polynomials and present an efficient NIZK PoK scheme, that can be a useful tool for $\mathbf{\Pi}$, while also preserving potential interest for different purposes. In Sec. 6, we present an efficient PVSS scheme. Finally, we conclude the paper in Sec. 7.

2 Preliminaries

2.1 Notation, Fields, Groups, Exceptional Sets

We let λ denote a security parameter. We use the assignment operator $\leftarrow$ to denote uniform sampling from a set $\varXi$, e.g. $x \leftarrow \varXi$. Throughout this paper p and q denote two large primes such that q divides $p-1$, $\mathbb{G}$ is the unique subgroup of $\mathbb{Z}_p^\star$ of order q, and g is a generator of cyclic group $\mathbb{G}$ of prime order q. One can test if an element $a \in \mathbb{Z}_p^\star$ is in $\mathbb{G}$, by checking if $a^q = 1$. The group $\mathbb{G}$ is chosen

[3] More on https://csrc.nist.gov/Projects/Threshold-Cryptography/.

such that computing DL of $h \in \mathbb{G}$, i.e., $\log_g h$, is hard in this group. We write $\mathbb{Z}_q$ and $\mathbb{Z}_q[X]_t$ for polynomials of degree t in the variable X and with coefficients in finite field $\mathbb{Z}_q$, with known prime q. When we refer to groups we assume they have known prime order and efficient algorithms to compute group operations. It will be assumed that all parties know p, q, g, and N.

2.2 Shamir and Verifiable Secret Sharing

A $(t+1, n)$-Shamir secret sharing scheme [25] allows n parties to individually hold a share f_i of a secret f_0, such that any subset of t parties or less are unable to learn any information about the secret f_0, while any subset of at least $t{+}1$ parties are able to efficiently reconstruct the secret f_0. In more detail, this is achieved via polynomial interpolation over the ring $\mathbb{Z}_N$ (or field $\mathbb{Z}_q$). A secret polynomial $f(x) \in \mathbb{Z}_N[x]_t$ is chosen and its free term is set to be the secret f_0, namely $f(0) = f_0$. Each party P_i for $i \in \{1, \cdots, n\}$ is assigned the secret share $f_i = f(i)$. Then any subset $Q \subseteq \{1, \ldots, n\}$ of at least $t{+}1$ parties can reconstruct the secret f_0 via Lagrange interpolation by computing $f_0 = f(0) = \sum_{i \in Q} f_i \cdot L_{0,i}^Q$, where $L_{0,i}^Q := \prod_{j \in Q \setminus \{i\}} \frac{j}{j-i} \pmod N$, are the Lagrange basis polynomials evaluated at 0. Any subset of less than $t+1$ parties are unable to find $f_0 = f(0)$, as this is information theoretically hidden from the other shares.

If $\mathbb{Z}_N$ is a ring, the difference of any elements in $\{1, \ldots, n\}$ must be invertible modulo N, thus $\{1, \ldots, n\}$ must be an exceptional set (defined in Def. 2.1). This is the case if n is less than the smallest prime divisor q of N. In the case where more than q parties want to participate in the protocol, we would have to work in a subgroup $\mathbb{Z}_{N'} \subset \mathbb{Z}_N$ such that the smallest divisor of N' is larger than q. Next we recall the definition of *(super)exceptional sets*.

Definition 2.1 (Exceptional set [3, 5, 14]). *An* exceptional set (modulo N) *is a set $\Xi_k = \{c_1, \ldots, c_k\} \subseteq \mathbb{Z}_N$, where the pairwise difference of all distinct elements is invertible modulo N. If further the pairwise sum of* all *elements is invertible modulo N, Ξ_k is called a* superexceptional set (modulo N).

Verifiable Secret Sharing. Standard secret sharing schemes are secure against passive attacks. In many applications, a secret sharing scheme needs to be secure against the malicious dealer or parties with active attacks. This is achieved through VSS schemes, which allow a dealer to share a secret among a group of individuals in a verifiable manner [13]. VSS schemes allow a dealer to distribute the secret in a *verifiable* manner, so that the shareholders can verify the validity of the shares and only a specific number of them can access the secret.

2.3 Sigma Protocols

Next, we recall the definition of sigma protocols (Σ-protocols). Here the algorithms are Probabilistic Polynomial-Time (PPT), unless mentioned. Let $X = X(\lambda)$ and $W = W(\lambda)$ be sets. Let R be a relation on $X \times W$ that defines a language $L = \{x \in X : \exists w \in W, R(x, w) = 1\}$. Given $x \in L$, an element $w \in W$

such that $R(x, w) = 1$ is called a witness. Let relation generator $\mathcal{R}$ be a PPT algorithm such that $\mathcal{R}(1^\lambda)$ outputs pairs (x, w) such that $R(x, w) = 1$.

A sigma-protocol (Σ-protocol) for the relation R is a 3-round interactive protocol between two PPT algorithms: a prover P and a verifier V. P holds a witness w for $x \in L$ and V is given x. In first round, P sends a commitment value a to V, and then in second round, V answers with a randomly sample challenge value d. Finally, P answers with a response z, and V verifies the proof and outputs `true` or `false`. The triple $\mathsf{trans} := (a, d, z)$ is called a transcript of the Σ-protocol. A Σ-protocol is supposed to satisfy *Completeness*, *Honest Verifier Zero-Knowledge* (HVZK), and *Special Soundness* defined below.

Definition 2.2 (Completeness). *A Σ-protocol with parties (P, V) is* complete *for $\mathcal{R}$, if for all $(x, w) \in R$, the honest V will always accept the honest P.*

Definition 2.3 (HVZK). *A Σ-protocol with parties (P, V) satisfies* HVZK *for $\mathcal{R}$, if there exists a PPT algorithm $\mathcal{S}$ that given $x \in X$, can simulate the* trans *of the scheme, s.t. for all $x \in L$, $(x, w) \in R$,*

$$\mathsf{trans}(P(x, w) \leftrightarrow V(x)) \approx \mathsf{trans}(\mathcal{S}(x) \leftrightarrow V(x))$$

where $\mathsf{trans}(P(\cdot) \leftrightarrow V(\cdot))$ indicates the transcript of the Σ-protocol with (P, V), and $\approx$ denotes the indistinguishability of transcripts.

Definition 2.4 (Special Soundness). *A Σ-protocol with parties (P, V) is* special sound *for $\mathcal{R}$, if there exists a PPT extractor $\mathcal{E}$, such that for any $x \in L$, given two valid transcripts (a, d, z) and (a, d', z') for the same message a but $d \neq d'$, then $\mathcal{E}(a, d, z, d', z')$ outputs a witness w for the relation R.*

Withing the Random Oracle (RO) model, using Fiat-Shamir transform [16], a public-coin, complete, HVZK, and special soundness Σ-protocol can be turned into a Non-Interactive Zero-Knowledge (NIZK) proof or argument of knowledge.

2.4 Chaum-Pedersen Protocol for DL Equality

Let $\mathbb{G}$ be a group with hard DL, and g, h be two group elements, where g is the group generator. Let a prover aim to convince a verifier that for the public statement g, h, a, b, he knows a witness x which holds in the following relation,

$$R_{DLEQ} = \{(g, h, a, b), x \mid a = g^x \wedge b = h^x\}. \tag{2}$$

This relation is known as DL EQuality (DLEQ). In [12], Chaum and Pedersen introduced an efficient NIZK proof of knowledge for DLEQ, as summarized in Fig. 1. This protocol is widely employed in various cryptographic protocols (e.g., threshold decryption, e-voting systems, PVSS schemes, etc.).

Prover: Given the statement $(g, h, a, b) \in \mathbb{G}$ and the witness value $x \in \mathbb{Z}_q$, proceed as follows and output a proof π.

1. Sample $r \leftarrow\!\!\$\ \mathbb{Z}_q$ uniformly at random; and set $c_1 = g^r$ and $c_2 = h^r$.
2. Set $d \leftarrow \mathcal{H}(a, b, c_1, c_2)$, where $\mathcal{H}$ is a random oracle.
3. Set $z = r + d \cdot x \mod q$; and Return $\pi := (d, z)$

Verifier: Given the statement $(g, h, a, b) \in \mathbb{G}$ and the proof $\pi = (d, z)$, checks if $d = \mathcal{H}(a, b, \frac{g^z}{a^d}, \frac{h^z}{b^d})$ and outputs `true` or `false`.

Fig. 1. Chaum-Pedersen NIZK proof of knowledge for DLEQ [12].

3 A Unified Framework for VSS Schemes

The GRR simple VSS scheme [18] allows a dealer to perform Shamir secret sharing and convince n verifiers that any $t+1$ of them can reconstruct a secret [18]. In their simple scheme, to share f_0, the dealer first does Shamir secret sharing and obtains the shares $\{f_i\}_{i=1}^n$. Then, it samples another degree-t polynomial $r(X)$ and sets $r_i = r(i)$ for $i = 1, \ldots, n$. After that, it commits to $\{f_i\}_{i=1}^n$ with $\{r_i\}_{i=1}^n$, by setting $c_i = \mathcal{C}(f_i, r_i)$, where $\mathcal{C}$ can be any commitment scheme. At the end, it securely sends (f_i, r_i) to P_i, and broadcasts $\{c_i\}_{i=1}^n$. Although their scheme is highly efficient, it lacks the guarantee of a *unique* reconstructed secret. In certain scenarios, such as robust cloud storage, this lack of uniqueness might not be a concern since computations on the shares are not required. However, when parties aim to perform computations on a unique value $f(0)$, they must get sure that they all possess distinct evaluations of a *unique* degree-t polynomial $f(X)$. This condition ensures that Lagrange interpolation with any of $t+1$ points will lead to a *unique* secret $f(0)$. This property, termed *verifiable secret and polynomial sharing*, is described by GRR [18]. To achieve *verifiable secret and polynomial sharing*, the recent PQ-secure VSS scheme by ABCP [1] leverages a hash-based NI-TZK proof scheme for the Shamir relation, which is proven to satisfy *computational* TZK and computational soundness in the quantum RO model. In a different setting, the recent lightweight asynchronous VSS scheme by Shoup and Smart [26] also relies on hash-based (thus non-homomorphic) commitments and either a random beacon or a random oracle to achieve the mentioned property.

In this section, we introduce $\mathbf{\Pi}$, designed for constructing VSS schemes with the flexibility to use both non-homomorphic and homomorphic commitments. It is based on Shamir secret sharing, works in the honest majority setting and in certain cases (i.e., in case of PVSS scheme) operates on the assumption that each shareholder has registered his/her Public Key (PK), that can facilitate secure communications. $\mathbf{\Pi}$ combines the strengths of the *simple* VSS scheme from [18], and the standard VSS scheme from [1], to achieve the best of both. Alternatively, it can be viewed as an optimized and generalized version of the ABCP VSS scheme [1]. With $\mathbf{\Pi}$, we achieve an efficient approach to building VSS schemes with various properties, like IT unpredictability and public verifiability.

3.1 Our Definitions

Before going through the construction of $\mathbf{\Pi}$, we summarize our definition of VSS schemes, which are adapted from [1, 22, 24].

Definition 3.1. *An (n, t, f_0)-VSS consists of four PPT algorithms of (Initialization, Share, Verification, Reconstruction) as follows:*

1. Initialization*: In this phase, the public keys of parties are registered, public parameters are sampled and all shared with the parties.*
2. $\mathsf{Share}(n, t, f_0) \rightarrow (\{f_i\}_{i=1}^{n}, \pi_{Share})$*: It secret shares f_0 and outputs the shares $\{f_1, \cdots, f_n\}$, and a (non-interactive) threshold proof π_{Share} to prove the validity of the shares. Note that, π_{Share} can only be verified by at least $t + 1$ of the shares (or commitments/encryption of the shares).*
3. $\mathsf{Verification}(n, t, \{f_i\}_{i=1}^{n}, \pi_{Share}) \rightarrow \texttt{true/false}$*: Given n, threshold value t, the shares $\{f_i\}_{i=1}^{n}$ (or commitments/encryptions of them), and the threshold proof π_{Share}, generated by* Share*, the algorithm outputs either* `true`/`false`*.*
4. $\mathsf{Reconstruction}(\{f_i\}_{i \in Q, |Q|=t+1}) \rightarrow \{f_0, \mathsf{false}\}$*: Given any $t + 1$ of the shares, e.g., $\{f_i\}_{i=1}^{t+1}$, it returns either the main secret f_0, or* `false`*.*

A VSS further has two requirements, defined as follows [1, 24].

- **Verifiability constraint**: A shareholder must be able to verify the validity of the received share. If they all are valid, then Reconstruction should produce a *unique secret* f_0 when run on any $t + 1$ distinct valid shares.
- **Unpredictability**: The protocol must be unpredictable, meaning that there is no strategy for selecting t shares of the secret that would enable someone to predict the secret f_0 with a significant advantage.

These definitions use TZK proofs over shared data [7] to prove the validity of the distributed shares, which their verification requires at least $t + 1$ honest parties. Similar to the definition of threshold ZK [1, 7], in some cases, the definition of unpredictability can be strengthened by requiring that given the individual statements (i.e., shares) of the t corrupted parties, the view of the adversary can be simulated. This means that the adversary gains no knowledge more than what publicly can be computed from the execution of the VSS protocol.

3.2 Construction of $\mathbf{\Pi}$ and Security Proofs

Let, D be a dealer and $P_1, \ldots, P_n$ are n participants of a VSS scheme. Let $\mathcal{C}$ be a hiding and binding commitment scheme that is verifiable. Namely,

1. given $c = \mathcal{C}(m, \gamma)$, it is hard to learn any information about (m, γ),
2. it is infeasible (or computationally hard) to find two pairs (m, γ) and (m', γ') s.t., $\mathcal{C}(m, \gamma) = \mathcal{C}(m', \gamma')$,
3. given (c, m, γ) anyone can efficiently verify if $c = \mathcal{C}(m, \gamma)$.

The general construction of $\mathbf{\Pi}$ appears in Fig. 2, which uses a computationally/perfectly hiding commitment scheme $\mathcal{C}$ and a (classic or quantum) random oracle $\mathcal{H}$. Intuitively, the general construction employs an efficient and general NI-TZK proof scheme for the Shamir relation, made non-interactive using the Fiat-Shamir transform [16]. It is also important to note that in $\mathbf{\Pi}$, instead of

Initialization: Parties $P_1, \cdots, P_n$ generate parameters for $\mathcal{C}$ and each one registers a PK to facilitate secure communications. For the sake of simplicity, we presume the existence of a dealer D and $P_1, \cdots, P_n$ parties who will receive the shares.

Share: Given (n, t), random oracle $\mathcal{H}$, to share f_0, the dealer D proceeds as follows:

1. Sample a uniformly random polynomial $f(X)$ and $r(X)$ of degree t with coefficients in a ring $\mathbb{Z}_N$ (or a field $\mathbb{Z}_q$), subject to $f(0) = f_0$.
2. For $i = 1, 2, \cdots, n$: set $f_i := f(i)$, and $r_i := r(i)$.
3. For $i = 1, 2, \cdots, n$: set $c_i = \mathcal{C}(f_i, r_i)$ (or set $c_i = \mathcal{C}((f_i, r_i), \gamma_i)$, where $\gamma_i = \gamma(i)$ are obtained by evaluating a new random degree-t polynomial $\gamma(X)$ in point i).
4. Set $z(X) = r(X) + d \cdot f(X)$ and $\pi_{Share} := (c_1, \ldots, c_n, z(X))$, where d is the challenge value obtained from the random oracle, i.e., $d := \mathcal{H}(c_1, \ldots, c_n)$;
5. Send share f_i (and γ_i if applicable) securely to P_i and broadcast π_{Share}.

Verification: Given $\pi_{Share} := (c_1, \ldots, c_n, z(X))$, and the individual shares:

1. Each party P_i first checks if $z(X)$ is a degree t polynomial, and then computes $d := \mathcal{H}(c_1, \ldots, c_n)$ and uses his/her share f_i (and γ_i if applicable) and checks if $c_i = \mathcal{C}(f_i, z(i) - d \cdot f_i)$ (or $c_i = \mathcal{C}((f_i, z(i) - d \cdot f_i), \gamma_i)$). If the verification of P_i fails, then P_i broadcasts a complain against the dealer.
2. If the number of shareholders complaining against the dealer exceeds a threshold value t, the dealer will be disqualified, and the verification process will result in a `false` outcome.
3. In case a shareholder P_i raises a complaint about the verification of their part, the dealer will broadcast $f_i = f(i)$ (and γ_i if applicable) to enable everyone to verify it using the verification equation. If the verification passes, the protocol continues as usual. However, if it fails, the dealer will be disqualified, leading to a `false` verification outcome. Since the disqualification decision is solely based on the information broadcasted, all honest shareholders will ultimately reach a consensus either on a set of qualified parties $Q \subseteq \{P_1, P_2, \cdots, P_n\}$ or on rejecting the final verification.

Reconstruction: Each party P_i broadcasts the secret value f_i (and γ_i if applicable). A party P_i is said to be confirmed if $c_i = \mathcal{C}(f_i, z(i) - d \cdot f_i)$ (or $c_i = \mathcal{C}((f_i, z(i) - d \cdot f_i), \gamma_i)$), where $d := \mathcal{H}(c_1, \ldots, c_n)$. Consider f_i values of any $t + 1$ confirmed parties and interpolate $f(X)$ of degree t that pass through those points.Finally, the output is $f_0 = f(0)$ or `false` (if $t + 1$ valid shares were not obtained).

Fig. 2. $\mathbf{\Pi}$: A Unified Framework for Building VSS Schemes. In the general construction, $\mathcal{H}$ represents an instantiation for the quantum/classic random oracle, and $\mathcal{C}((\cdot,\cdot),\cdot)$ (resp. $\mathcal{C}(\cdot,\cdot)$) is a perfectly (resp. computationally) hiding and computationally (resp. perfectly) binding commitment scheme.

having separate commitments to the (secret) shares and the randomizers used in the first round of the interactive (threshold) proof schemes, e.g., as in Schnorr protocol (or in ABCP PQ-secure VSS scheme [1]), we use a single polynomial commitment to commit to both (secret) shares (i.e., individual private statements) and the randomizers simultaneously.

Security. We prove the security of $\mathbf{\Pi}$ in the following theorem.

Theorem 3.1 (A Unified Framework for VSS Schemes). *If the commitment scheme $\mathcal{C}$ is computationally (resp. perfectly) hiding and perfectly (resp.*

computationally) binding and $\mathcal{H}$ is a (classic/quantum) random oracle, then the generic construction given in Fig. 2 is a secure VSS scheme. That is, (i) the Reconstruction protocol results in a unique secret distributed by the dealer for any qualified set of shareholders, (ii) any non-qualified set of shareholders is unable to recover (or learn anything about) the secret.

Proof. The proof of this theorem can be regarded as an extension of the proof found in [18, Theorem 1]. Furthermore, we demonstrate that our proposed construction can also satisfy the *verifiable secret and polynomial sharing* property. On another front, this proof can also be seen as a slightly simplified version of the proof in [1, Theorem 3.1], where the commitments are merged and in some instantiations the quantum RO is relaxed to classic RO. It's worth noting that following a valid sharing phase, any coalition of $t+1$ honest parties is capable of reconstructing the witness polynomial $f(X)$.

As mentioned in the Verification algorithm, since the disqualification decision is solely based on public (broadcast) information, all honest shareholders ultimately reach the same decision. Moreover, if the dealer will be honest and follow the Share algorithm, then the Verification algorithm will return `true`, and all the honest shareholders will get a valid and distinct share of a *unique* secret.

Verifiability. In the majority-honest scenario where $t+1$ of the parties are honest, with $n \geq 2t+1$, this property is achieved via the random oracle $\mathcal{H}$ and the binding property of the commitment scheme $\mathcal{C}$. Assume w.l.o.g. that at least $P_1, \ldots, P_{t+1}$ parties are honest. Let $f(X)$, $r(X)$ and $z(X) := r(X) + d \cdot f(X)$ be the polynomials of degree t determined by values f_i, r_i, and $d \cdot f_i + r_i$, for $1 \leq i \leq t+1$, where d is obtained from the random oracle, i.e., $d = \mathcal{H}(c_1, \ldots, c_n)$. If $c_i = \mathcal{C}(f_i, z(i) - d \cdot f(i))$ for all $i = 1, \cdots, t+1$, then define $f_i := f(i)$. Otherwise, set $f_i := 0$.

The dealer has committed (in a distributed fashion) himself to the distinct values $c_1, \ldots, c_n$ by broadcasting $\pi_{Share} := (c_1, \ldots, c_n, z(X))$ and sending $\{f_i\}_{i=1}^{n}$ to $n \geq 2t+1$ parties, where at least $t+1$ of them are honest. Therefore, the values f_i and $r_i := z(i) - d \cdot f_i$ for $1 \leq i \leq t+1$ are set at the end of the sharing phase, and consequently the polynomial $f(x)$ is set. Then, the value of f_0 is well-defined at the end of the sharing phase, and given a degree-t polynomial $z(X)$, and opening of $t+1$ commitments with points $r_i := z(i) - d \cdot f_i$, enables the reconstruction of degree-t polynomials $r(X) := z(X) - d \cdot f(X)$ and $f(X)$ using Lagrange interpolation. Consequently, any $t+1$ honest parties will be able to collectively reconstruct the secret $f(0)$. Looking from a different perspective, one may notice that the general construction uses a special sound sigma protocol with designated verification. Given two acceptable transcripts $(c_1, \ldots, c_n, d, z(X))$ and $(c_1, \ldots, c_n, d' \neq d, z'(X))$, obtained by rewiring the prover (i.e., the dealer), from the verification equation, we can write $\mathcal{C}(f_i, z(i) - d \cdot f_i) = \mathcal{C}(f_i, z'(i) - d' \cdot f_i)$ for $i = 1, \ldots, n$, where $n \geq 2t+1$. Then, relying on the binding property of the commitment scheme $\mathcal{C}$, we can conclude that $z(i) - d \cdot f_i = z'(i) - d' \cdot f_i$, and therefore $f_i = \dfrac{z(i) - z'(i)}{d - d'}$ for $i = 1, \ldots, n$. Assuming that $d - d'$ is invertible modulo N (or q), given any set of $t+1$ valid shares, an extractor can extract a *unique* degree-t polynomial from the dealer.

As a result, similarly, at the end of the Reconstruction phase, any set of $t+1$ honest parties can reconstruct a *unique* degree-t polynomial and output a unique value f_0. Assume by contradiction that they reconstruct $f'_0 \neq f_0$ by choosing $t+1$ values $f'_1, \ldots, f'_{t+1}$, such that $c_i = \mathcal{C}(f'_i, z(i) - d \cdot f'_i)$. This means that the t-degree polynomials $f'(X)$, and $r'(X)$ interpolated by the points f'_i and $z(i) - d \cdot f'_i$ (resp.) have the property that $\mathcal{C}(f'_i, z(i) - d \cdot f'(i)) = \mathcal{C}(f'_i, r'(i)) = c_i$ for $i = 1, \ldots, t+1$, but $f'(X) \neq f(X)$ (as they differ in the free term), thus there must be an index j such that $f'(j) \neq f(j)$. Since each degree-t polynomial gets unique with its $t+1$ distinct evaluations, then the values $(z(j) - d \cdot f'(j))$ and $(z(j) - d \cdot f(j))$ are a double opening for the commitment scheme $\mathcal{C}$, which is known to either the dealer or P_j, which contradicts the hypothesis (the binding property of $\mathcal{C}$).

Unpredictability. If the dealer is honest in the sharing phase, then the adversary sees t points on a polynomial of degree t (i.e., $f(X)$) plus a masked degree-t polynomial $z(X) := r(X) + d \cdot f(X)$ and all the commitment values $c_i := \mathcal{C}(f_i, r_i)$ for $i = 1, \ldots, n$. But as we assume that given d and $z_i = r_i + d \cdot f_i$ obtaining (random values) f_i and r_i from c_i is hard (or infeasible in some cases), then from commitments $\{c_i\}_{i=1}^{n}$ and the masked degree-d polynomial $z(X)$, obtaining the values of $f(X)$ or $r(X)$ in other points is computationally hard (or infeasible). Note that t evaluations of a degree-t polynomial, information theoretically does not reveal any information about the target polynomial. Hence, the adversary cannot recover (or learn any information about) other points, including the secret value $f(0)$, from $(c_1, \ldots, c_n, d, z(X))$. In other words, given the individual shares (i.e., statements) of t (corrupted) parties, it is possible to simulate the view of the adversary. To this end, w.l.o.g., given the shares $\{f_i\}_{i=1}^{t}$, the simulator samples two random degree-t polynomials $r'(X)$ and $f'(X)$, such that $f'(i) = f_i$ for $i = 1, \ldots, t$. Then, the simulator sets $c'_i = \mathcal{C}(f'(i), r'(i))$ for $i = 1, \ldots, n$ and $z'(X) := r'(X) + d \cdot f'(X)$, where $d = \mathcal{H}(c'_1, \ldots, c'_n)$. At the end, the simulator returns $(c'_1, \ldots, c'_n, d, z'(X))$ as the simulated transcript. It's worth noting that the proof can naturally be extended to the scenario where the dealer commits to (f_i, r_i) using a perfectly hiding commitment scheme, such as a variant of Pedersen's scheme with three random generators [8,22]. In that case, the simulation can be perfect and the resulting VSS scheme can achieve IT unpredictability. □

Efficiency. As in Shamir secret sharing, the process of sharing f_0 among n parties with a threshold of t requires the dealer to compute n evaluations of a degree-t polynomial $f(X)$. Subsequently, to generate π_{Share}, the dealer needs to compute an additional set of n evaluations for $r(X)$ (and $\gamma(X)$ if applicable). This process also involves generating n commitments and performing t subtractions between the coefficients of $f(X)$ and $r(X)$, and a single query to the random oracle $\mathcal{H}$, which should ideally be highly efficient in practice. During the verification phase, parties take part in the verification of $\mathbf{\Pi}$ (outlined in Fig. 2) and disseminate the final output to the network. As part of this procedure, each party needs to evaluate a degree-t polynomial $z(X)$ and compute a query to the random oracle and a single commitment. Regarding communication, the dealer broadcasts $(c_1, \ldots, c_n, z(X))$, which consists of n commitments and $t+1$ polynomial coefficients. The dealer also securely sends a share to each participant.

4 Constructing VSS Schemes Via $\mathbf{\Pi}$

The strength of $\mathbf{\Pi}$ lies in its generality, simplicity, and efficiency, as it only requires a secure commitment scheme $\mathcal{C}$ and a (quantum or classic) random oracle $\mathcal{H}$. Commitment schemes are one of the core primitives in cryptography, and can be built efficiently. A true random oracle might not exist in real life, but with some estimations, they usually are built using cryptographic hash functions.

In this section, we employ different commitment schemes for $\mathcal{C}$ and utilize $\mathbf{\Pi}$ to build several VSS schemes. The proposed schemes exhibit various trade-offs in terms of efficiency and security. To achieve this goal, we commence by revisiting established constructions from the existing literature. Subsequently, leveraging $\mathbf{\Pi}$, we introduce an alternative scheme for each VSS scheme.

4.1 $\mathbf{\Pi_F}$: A Novel VSS Scheme from Pedersen Commitment

Overview of Feldman VSS Scheme. One of the primary computationally secure VSS schemes is Feldman's scheme, which is based on Shamir and was proposed by Feldman in [15]. In Feldman's scheme, given (n, t) and group generator g_1, to share a *high-entropy* secret f_0, the dealer proceeds as follows:

1. Sample a uniformly random degree-t polynomial $f(X) := f_0 + a_1 X + \cdots + a_t X^t$ with coefficients in $\mathbb{Z}_q$, subject to $f(0) = f_0$.
2. For $i = 1, 2, \cdots, n$: set $f_i := f(i)$.
3. Compute $c_0 = g_1^{f_0}$ and $c_j = g_1^{a_j}$ for $j = 1, 2, \cdots, t$.
4. Set $\pi_{Share} := (c_0, c_1, \ldots, c_t)$; Sends share f_i securely to party P_i and broadcast π_{Share} as the proof.

Then, to verify their received shares, given $\pi_{Share} := (c_0, c_1, \ldots, c_t)$, and the individual shares $\{f_i\}_{i=1}^n$: each party P_i uses his/her share f_i and checks if $g_1^{f_i} = \prod_{j=0}^{t} c_j^{i^j}$ and outputs either `true` or `false`. For $n \geq 2t+1$, if *all* n parties return `true`, then the final Verification will return `true`. Otherwise, any possible conflict between the dealer and the parties will be solved using a known conflict resolution approach (also used in $\mathbf{\Pi}$).

$\mathbf{\Pi_F}$: An Efficient Alternative to Feldman Scheme. By instantiating $\mathbf{\Pi}$ with a Pedersen commitment [22], with two random group generators $(g_1, g_2) \in \mathbb{G}$, i.e. by setting $c_i := g_1^{f_i} g_2^{r_i}$, we obtain a novel VSS scheme, referred to as $\mathbf{\Pi_F}$. This scheme provides an alternative construction to the Feldman scheme [15]. In Fig. 3, we provide a concise overview of $\mathbf{\Pi_F}$, focusing solely on the steps that deviate from our general construction $\mathbf{\Pi}$.

Under DL assumption, Theorem 3.1 and its security proof can be adapted for $\mathbf{\Pi_F}$. Note that, as in the Feldman scheme, in $\mathbf{\Pi_F}$, f_0 should contain sufficient entropy, and the new VSS scheme is, at best, secure against computationally

Share: Given two random group generators g_1 and g_2, the parameters n and t, to share f_0, the dealer follows the steps outlined in Fig. 2, specifically, with the following deviations:

3. For $i = 1, 2, \cdots, n$: Compute $c_i = g_1^{f_i} g_2^{r_i}$.

Verification: Given g_1, g_2, $\pi_{Share} := (c_1, \ldots, c_n, z(X))$, and the shares $\{f_i\}_{i=1}^{n}$:

1. Each party P_i first checks if $z(X)$ is a degree t polynomial, and then computes $d = \mathcal{H}(c_1, \ldots, c_n)$ and uses his/her share f_i and checks if $c_i = g_1^{f_i} g_2^{z(i)-d\cdot f_i}$. If the verification of P_i fails, then P_i broadcasts a complain against the dealer.

Reconstruction: Using the reconstruction approach outlined in Fig. 2, each party P_i broadcasts the secret value f_i. A party P_i is said to be confirmed if $c_i = g_1^{f_i} g_2^{z(i)-d\cdot f_i}$, where $d := \mathcal{H}(c_1, \ldots, c_n)$. Consider f_i values of any $t+1$ confirmed parties and interpolate $f(X)$ of degree t that pass through those points. Finally, the output is $f_0 = f(0)$ or `false` (if $t+1$ valid shares were not obtained).

Fig. 3. $\mathbf{\Pi_F}$: A novel VSS scheme based on discrete logarithm.

bounded (classical) adversaries. In terms of efficiency, $\mathbf{\Pi_F}$ can have considerably faster verification and reconstruction compared to the Feldman scheme. However, this advantage comes at the cost of approximately 2.5-3 × slower sharing and 2× increased communication. Please refer Tab. 1 for the details.

4.2 $\mathbf{\Pi_P}$: A Novel VSS Scheme from Pedersen Commitment

Overview of Pedersen VSS Scheme. The Pedersen VSS scheme is a variation of Feldman's scheme [15] that uses a perfectly hiding commitment scheme. In Pedersen VSS scheme, the commitment takes the form of a Pedersen commitment, denoted as $c_i = g_1^{a_i} g_2^{b_i}$, where a_i, b_i are the coefficients of two degree-t polynomials. This approach ensures that fewer than t parties receive no information about the secret, thereby achieving information-theoretical unpredictability. In the Pedersen scheme, given two random group generators g_1 and g_2, and parameters n and t, the process of sharing f_0 is done as follows:

1. Sample two random degree-t polynomials $f(X) := f_0 + a_1 X + \cdots + a_t X^t$ and $r(X) := r_0 + b_1 X + \cdots + b_t X^t$ with coefficients in $\mathbb{Z}_q$, subject to $f(0) = f_0$.
2. For $i = 1, 2, \cdots, n$: set $f_i := f(i)$ and $r_i := r(i)$.
3. Compute $c_0 = g_1^{f_0} g_2^{r_0}$ and $c_j = g_1^{a_j} g_2^{b_j}$ for $j = 1, 2, \cdots, t$.
4. Set $\pi_{Share} := (c_0, c_1, \ldots, c_t)$; Sends share (f_i, r_i) securely to party P_i and broadcast π_{Share} as the proof.

Then, to verify their received shares, given $\pi_{Share} := (c_0, c_1, \ldots, c_t)$, and the individual shares $\{f_i, r_i\}_{i=1}^{n}$: each party P_i uses his/her share (f_i, r_i) and checks if $g_1^{f_i} g_2^{r_i} = \prod_{j=0}^{t} c_j^{i^j}$ and outputs either `true` or `false`. The rest of the verification is similar to the Feldman scheme.

$\mathbf{\Pi_P}$: An Efficient Alternative to Pedersen Scheme. Instantiating $\mathbf{\Pi}$ with an extended variant of the Pedersen commitment scheme from [8], specifically by employing *three* randomly chosen group generators $(g_1, g_2, g_3) \in \mathbb{G}$, leads to the

Share: Given three random group generators (g_1, g_2, g_3), the parameters (n, t), to share f_0, the dealer follows the steps outlined in Fig. 2, but, with the following deviations:

3. Sample a new degree-t polynomial $\gamma(X)$ with coefficients in $\mathbb{Z}_q$. For $i = 1, 2, \cdots, n$, compute $\gamma_i = \gamma(i)$ and set $c_i = g_1^{f_i} g_2^{r_i} g_3^{\gamma_i}$.
4. Compute the challenge value $d := \mathcal{H}(c_1, \ldots, c_n)$; Check if $g_1 \neq g_2^d$ and if the check passed, set $z(X) = r(X) + d \cdot f(X)$ and $\pi_{Share} := (c_1, \ldots, c_n, z(X))$. Otherwise, restart the protocol from step 3;
5. Send shares (f_i, γ_i) securely to party P_i and broadcast π_{Share} as the proof.

Verification: Given random group generators (g_1, g_2, g_3), a proof $\pi_{Share} := (c_1, \ldots, c_n, z(X))$, and the shares $\{f_i, \gamma_i\}_{i=1}^n$:

1. Each party P_i first checks if $z(X)$ is a degree t polynomial, computes challenge value $d = \mathcal{H}(c_1, \ldots, c_n)$ and then uses his/her shares (f_i, γ_i) and checks if $c_i = g_1^{f_i} g_2^{z(i) - d \cdot f_i} g_3^{\gamma_i}$. If the verification of P_i fails, then P_i broadcasts a complain against the dealer.

Reconstruction: Using the reconstruction approach outlined in Fig. 2, each party P_i broadcasts the secret values f_i and γ_i. A party P_i is said to be confirmed if $c_i = g_1^{f_i} g_2^{z(i) - d \cdot f_i} g_3^{\gamma_i}$, where $d := \mathcal{H}(c_1, \ldots, c_n)$. Consider f_i values of any $t+1$ confirmed parties and interpolate $f(X)$ of degree t that pass through those points. Finally, the output is $f_0 = f(0)$ or `false` (if $t+1$ valid shares were not obtained).

Fig. 4. $\mathbf{\Pi_P}$: A novel IT-secure VSS scheme from Pedersen commitments.

construction of a novel DL-based VSS scheme denoted as $\mathbf{\Pi_P}$. This scheme can be considered as an alternative to the Pedersen VSS scheme. In $\mathbf{\Pi_P}$, commitments c_i are computed using a variant of the Pedersen commitment, i.e., $c_i = g_1^{f_i} g_2^{r_i} g_3^{\gamma_i}$ for $i = 1, \ldots, n$, where $\gamma_i = \gamma(i)$ are new randomizers obtained by evaluating a new random degree-t polynomial $\gamma(X)$ for $i = 1, \ldots, n$. The purpose of the new randomizer γ_i is to achieve IT unpredictability. Thus, in order to ensure its effect is not canceled, before computing the value $z(X) = r(X) + d \cdot f(X)$, the dealer additionally checks if $g_1 \neq g_2^d$ and continues if the check passes. This check is necessary to ensure that the simulation of the transcript is perfect, and the protocol satisfies IT unpredictability against up to t shareholders. At the end, along with the share f_i, the dealer also sends the randomizer γ_i to party P_i. Then, given public values $(g_1, g_2, g_3, c_i, z(X))$ and secret values (f_i, γ_i), party P_i first checks if $z(X)$ is a degree t polynomial, and then computes $d = \mathcal{H}(c_1, \ldots, c_n)$ and verifies if $c_i = g_1^{f_i} g_2^{z(i) - d \cdot f_i} g_3^{\gamma_i}$ and outputs either `true` or `false`. The description of $\mathbf{\Pi_P}$ is summarized in Fig. 4.

Under DL assumption, Theorem 3.1 and its security proof can be adapted for $\mathbf{\Pi_P}$. Notably in this case, since the commitment to (f_i, r_i) is perfectly hiding and $g_1 \neq g_2^d$, the simulation of transcript is perfect and the resulting VSS scheme can achieve IT unpredictability, and a computationally unbounded adversary $\mathcal{A}$ who controls up to t parties, cannot learn anything about the other shares and the main secret from the transcript of the protocol, i.e., $(c_1, \ldots, c_n, d, z(X))$. Note that in this case, there is a negligible probability that the dealer may need to repeat step 3 in the sharing phase. The resulting VSS scheme $\mathbf{\Pi_P}$ surpasses

Pedersen VSS scheme [22] in the verification and reconstruction phases. Please refer to Tab. 1 for detailed comparisons.

4.3 $\mathbf{\Pi_{LA}}$: A Novel VSS Scheme from Hash Functions

Overview of the RO-based VSS Scheme of ABCP. Recently, Atapoor, Baghery, Cozzo, and Pedersen [1], proposed a general construction and showed that given a NI-TZK proof scheme for the Shamir relation, given in eq. (1), one can build a VSS scheme based on Shamir secret sharing. Following their initial result, they built a NI-TZK proof scheme for the Shamir relation, and then used it to construct a novel PQ-secure VSS scheme. Their resulting VSS scheme uses NI-TZK proofs, which use a quantum RO and a quantum computationally hiding commitment scheme, which both are built from hash functions. Their construction is extremely efficient and outperforms the prior computationally secure VSS schemes from the literature. In their scheme, given n and t, to share the secret f_0, the dealer proceeds as follows:

1. Sample two random degree-t polynomials $r(X) := r_0 + b_1X + \cdots + b_tX^t$ and $f(X) := f_0 + a_1X + \cdots + a_tX^t$ with coefficients in $\mathbb{Z}_N$, subject to $f(0) = f_0$.
2. For $i = 1, 2, \cdots, n$: set $f_i := f(i)$ and $r_i := r(i)$, and also samples two vectors of randomnesses y_i, y'_i.
3. Compute $c_i = \mathcal{C}(f(i), y_i)$ and $c'_i = \mathcal{C}(r(i), y'_i)$ for $i = 1, 2, \cdots, n$, where $\mathcal{C}$ is a quantum computationally hiding commitment scheme.
4. Set the challenge value $d = \mathcal{H}(c_1, \ldots, c_n, c'_1, \ldots, c'_n)$, where $\mathcal{H}$ is an RO.
5. Set the response $z(X) = r(X) - d \cdot f(X)$;
6. Finally, set $\pi_{Share} := (c_1, \ldots, c_n, c'_1, \ldots, c'_n, z(X))$; Sends share f_i and the randomnesses (y_i, y'_i) securely to party P_i and broadcast π_{Share} as the proof.

Verification. To verify their received shares, given $\pi_{Share} := (c_1, \ldots, c_n, c'_1, \ldots, c'_n, z(X))$, and the individual shares $\{f_i\}_{i=1}^n$ and randomnesses $\{y_i, y'_i\}_{i=1}^n$: each party P_i uses his/her share (f_i, y_i, y'_i) and proceeds as follows: 1) checks if $\mathcal{C}(f_i, y_i) = c_i$; 2) computes the challenge value $d = \mathcal{H}(c_1, \ldots, c_n, c'_1, \ldots, c'_n)$; 3) checks if $\mathcal{C}(z(i) + df_i, y'_i) = c'_i$; and outputs either `true` or `false`. The rest of the verification, i.e., conflict resolution, is the same as in $\mathbf{\Pi}$ (given in Fig. 2).

Reconstruction. The reconstruction phase can be done in a way similar to that of $\mathbf{\Pi}$. In [1], authors also introduced and employed a novel reconstruction approach based on NI-TZK proofs, where the dealer discloses the main secret, and the parties subsequently utilize their shares to confirm the authenticity of the revealed secret $\hat{f}_0$. Intuitively, in this approach, the dealer employs the VSS scheme as a distributed commitment to prove the authenticity of the disclosed secret $\hat{f}_0$.

$\mathbf{\Pi_{LA}}$*: More Efficient VSS Scheme from Hash Functions.* By instantiating $\mathbf{\Pi}$ with a non-homomorphic commitment scheme, like those based on hash functions, we obtain a novel PQ-secure VSS scheme in the quantum random oracle model, named $\mathbf{\Pi_{LA}}$. In $\mathbf{\Pi_{LA}}$, in case f_0 and f_i for $i = 1, \ldots, n$ have enough entropy, commitments c_i are computed as $c_i = \mathsf{H}(f_i, r_i)$ for $i = 1, \ldots, n$, where H

Share: Given a secure hash function H, quantum random oracle $\mathcal{H}$, the parameters n and t, to share f_0, the dealer follows the steps outlined in Fig. 2, but, with the following deviations:

3. For $i = 1, 2, \cdots, n$: Compute commitments $c_i = \mathsf{H}(f_i, r_i)$ (or set $c_i = \mathsf{H}(f_i, r_i, \gamma_i)$, where $\gamma_i = \gamma(i)$ are obtained by evaluating a new random degree-t polynomial $\gamma(X)$ in point i).

Verification: Given H and $\mathcal{H}$, $\pi_{Share} := (c_1, \ldots, c_n, z(X))$, and the shares $\{f_i\}_{i=1}^{n}$:

1. Each party P_i first checks if $z(X)$ is a degree t polynomial, and if so, computes $d = \mathcal{H}(c_1, \ldots, c_n)$ and then uses his/her share f_i and checks if $c_i = \mathsf{H}(f_i, z(i) - d \cdot f_i)$ (or $c_i = \mathsf{H}((f_i, z(i) - d \cdot f_i), \gamma_i)$). If the verification of P_i fails, then P_i broadcasts a complain against the dealer.

Reconstruction: Using the reconstruction approach outlined in Fig. 2, each party P_i broadcasts the secret value f_i (and γ_i if applicable). A party P_i is said to be confirmed if $c_i = \mathsf{H}(f_i, z(i) - d \cdot f_i)$ (or $c_i = \mathsf{H}((f_i, z(i) - d \cdot f_i), \gamma_i)$), where $d := \mathcal{H}(c_1, \ldots, c_n)$. Consider f_i values of any $t+1$ confirmed parties and interpolate $f(X)$ of degree t that pass through those points. Finally, the output is $f_0 = f(0)$ or `false` (if $t+1$ valid shares were not obtained).

Fig. 5. $\mathbf{\Pi_{LA}}$: A novel PQ-secure VSS scheme from hash functions.

is a well-defined secure hash function and the coefficients of $f(X)$ and $r(X)$ are sampled randomly from ring $\mathbb{Z}_N$ [4]. Then, given $(f_i, c_1, \ldots, c_n, z(X))$, party P_i first checks if $z(X)$ is a degree t polynomial, and then computes $d = \mathcal{H}(c_1, \ldots, c_n)$ and verifies if $c_i = \mathsf{H}(f_i, z(i) - d \cdot f_i)$ and outputs either `true` or `false`. In the case f_0 and $\{f_i\}_{i=1}^{n}$ lack enough entropy, the dealer can act as in Fig. 4 and use an additional randomizer $\gamma_i := \gamma(i)$ in the commitments, i.e., $c_i = \mathsf{H}(f_i, r_i, \gamma_i)$ for $i = 1, \ldots, n$. Accordingly, to verify their shares, for the last check party P_i uses (f_i, γ_i) and verifies if $c_i = \mathsf{H}(f_i, z(i) - d \cdot f_i, \gamma_i)$ and outputs either `true` or `false`. The description of $\mathbf{\Pi_{LA}}$ is summarized in Fig. 5.

Under pre-image resistance and collision resistance of hash function H and the security of quantum random oracle $\mathcal{H}$, Theorem 3.1 and its security proof can be extended for $\mathbf{\Pi_{LA}}$. The proof, can also be written similar to the proof of Theorem 2 in [1]. Note that in this case c_i hides f_i, r_i against computationally bounded (quantum) adversaries. $\mathbf{\Pi_{LA}}$ outperforms the ABCP scheme [1], by a constant factor in terms of all efficiency metrics (please refer to Tab. 1, and Tab. 2).

4.4 Efficiency Comparisons of New VSS Schemes

We conducted an analysis of the asymptotic costs for the proposed VSS schemes ($\mathbf{\Pi_F}, \mathbf{\Pi_P}, \mathbf{\Pi_{LA}}$) and compared them to relevant constructions from existing literature. A summary of the results can be found in Table 1.

[4] Note that, in this case the challenge value d is sampled from an exceptional set. Therefore, when $\mathbb{Z}_N$ is a cryptographically sized field, we can achieve a negligible error rate in $\mathbf{\Pi_{LA}}$, i.e. below $2^{-\lambda}$. In cases where $\mathbb{Z}_N$ is a ring, it's possible to encounter situations where the largest exceptional set has a size of $k < 2^{\lambda}$. In such scenarios, the dealer needs to amplify the soundness error rate in a standard manner by repeating the protocol $l = \lceil \lambda / \log k \rceil$ times.

Table 2. Implementation results of VSS schemes Pedersen [22], $\mathbf{\Pi_P}$, ABCP [1], and $\mathbf{\Pi_{LA}}$. n: Number of parties, t: Threshold value, $|\mathbb{Z}_q| = |\mathbb{Z}_N| = |\mathbb{G}| = |\mathcal{H}| = 256$ bits.

(n, t)	Metrics	Pedersen [22]	$\mathbf{\Pi_P}$	ABCP [1]	$\mathbf{\Pi_{LA}}$
(32, 15)	Sharing	74.8 msec	222.1 msec	2.2 msec	1.7 msec
	Verification	10.7 msec	7.1 msec	0.13 msec	0.10 msec
	Dealer's Broadcast	0.5 KB	1.5 KB	2.5 KB	1.5 KB
	Dealer's Private Com.	2.0 KB	2.0 KB	1.0 KB	1.0 KB
(128, 63)	Sharing	303 msec	897 msec	13.2 msec	11.4 msec
	Verification	99.2 msec	8.3 msec	0.37 msec	0.33 msec
	Dealer's Broadcast	2.0 KB	6.0 KB	10.0 KB	6.0 KB
	Dealer's Private Com.	8.0 KB	8.0 KB	4.0 KB	4.0 KB
(512, 255)	Sharing	1.29 sec	3.71 sec	0.13 sec	0.12 sec
	Verification	552 msec	12.5 msec	1.2 msec	1.1 msec
	Dealer's Broadcast	8.0 KB	24.0 KB	40.0 KB	24.0 KB
	Dealer's Private Com.	32.0 KB	32.0 KB	16.0 KB	16.0 KB
(2048, 1023)	Sharing	6.45 sec	16.77 sec	1.81 sec	1.78 sec
	Verification	2.32 sec	28.3 msec	4.9 msec	4.8 msec
	Dealer's Broadcast	32.0 KB	96.0 KB	160.0 KB	96.0 KB
	Dealer's Private Com.	128.0 KB	128.0 KB	64.0 KB	64.0 KB
(8192, 4095)	Sharing	47.1 sec	98.8 sec	28.6 sec	28.5 sec
	Verification	9.38 sec	0.092 sec	0.020 sec	0.018 sec
	Dealer's Broadcast	128 KB	384 KB	640 KB	384 KB
	Dealer's Private Com.	512 KB	512 KB	256 KB	256 KB
(16384, 8191)	Sharing	149.0 sec	279.8 sec	112.0 sec	111.5 sec
	Verification	18.7 sec	0.178 sec	0.070 sec	0.050 sec
	Dealer's Broadcast	256 KB	768 KB	1280 KB	768 KB
	Dealer's Private Com.	1024 KB	1024 KB	512 KB	512 KB

Empirical Performance of Pedersen, $\mathbf{\Pi_P}$, ABCP, and $\mathbf{\Pi_{LA}}$ Schemes. In addition, we assessed the practical performance of $\mathbf{\Pi_P}$ and $\mathbf{\Pi_{LA}}$ through a prototype implementation in SageMath and compared their performance with the Pedersen scheme and the recently proposed ABCP construction [1]. We used the source code implementations for the Pedersen and ABCP schemes from [1] [5]. Our experiments are done using the elliptic curve Ed25519 and the hash function SHA256 for both commitment and random oracle, on a laptop with Ubuntu 22.04 LTS, a 11th Gen Intel(R) Core(TM) i9-11950H at base frequency 2.60GHz, and 128GB of memory. Both the sharing and verification algorithms are executed in single-thread mode. We have summarized our implementation results for various parameter sets in Tab. 2.

Upon comparing the implementation outcomes of the Pedersen scheme with those of $\mathbf{\Pi_P}$, it becomes apparent that $\mathbf{\Pi_P}$ yields a remarkable acceleration in the verification phase. Notably, $\mathbf{\Pi_P}$ achieves verification times that are 12×, 82×, and 102× faster in comparison to the Pedersen scheme for the parameter pairs (n, t) equal to $(128, 63)$, $(2048, 1023)$, and $(16384, 8191)$, respectively. In terms of communication costs, $\mathbf{\Pi_P}$ demands a slightly larger data broadcast from the dealer, amounting to 3× compared to 1× in the Pedersen scheme. We highlight

[5] Available on `https://github.com/Baghery/VSS-ABCP23`.

that these achievements within $\mathbf{\Pi_P}$ are accompanied by a slightly slower sharing phase, resulting in speeds ranging from $1.88-3.0\times$ in comparison to the baseline of $1\times$ in Pedersen scheme. Moreover, it's worth noting that the disparity in costs becomes less pronounced as the values of (n, t) increase. For instance, in the case of $(n, t) = (16384, 8191)$, the sharing phase of $\mathbf{\Pi_P}$ is approximately 88% slower than the sharing phase of the Pedersen scheme. We believe that this gap and the verification time can be reduced through various optimization techniques. One optimization approach is to use improved algorithms for evaluating polynomials at multiple points. Another optimization can be the employment of a more efficient perfectly hiding commitment scheme.

Regarding, $\mathbf{\Pi_{LA}}$, we can see that it is slightly more efficient than the recent construction by ABCP [1] in terms of computation and communication costs. Notably, owing to their reliance on lightweight cryptography and polynomial evaluations exclusively, both $\mathbf{\Pi_{LA}}$ and ABCP [1] exhibit swifter performance than $\mathbf{\Pi_P}$ and the Pedersen scheme [22], which relies on asymmetric cryptography.

Regarding the efficiency of the reconstruction phase in all studied and new DL-based VSS schemes, as shown in Table 1, we observe that the new schemes require t times fewer exponentiations but at the cost of t polynomial evaluations. By a rough estimation, if we multiply the verification time of each VSS scheme by t, we can obtain a rough estimation of the time required for secret reconstruction. For instance, for $(n, t) = (16384, 8191)$, the secret reconstruction phase for $\mathbf{\Pi_P}$ could be approximately $8191\times$ faster compared to the Pedersen VSS. We note that in practice, this gap can be narrowed by employing multi-exponentiation techniques in both schemes, and we expect it can be more effective for the Pedersen VSS scheme.

It is also worth mentioning that our implementation is done using SageMath, and it remains relatively basic, functioning as a single-threaded process without specific optimizations. Given that our proposed schemes heavily rely on polynomial evaluations, an effective optimization is to employ more efficient algorithms for the evaluation of a polynomial at multiple points, as outlined in [28].

5 Generalizing DL and Schnorr Over Polynomials

This section introduces an efficient NIZK PoK scheme that can serve as a tool for $\mathbf{\Pi}$ while maintaining relevance for other threshold schemes and applications.

Let $\mathbb{G}$ be a group with hard DL, and g be the group generator. Let a prover aim to convince a verifier that for the public statement $F \in \mathbb{G}$, he knows a witness $f \in \mathbb{Z}_q$ which holds in relation $R_{DL} = \{(g, F), f \mid F = g^f\}$. Schnorr's known ID protocol [23] allows a prover to efficiently achieve this goal. In the NIZK version of Schnorr's ID protocol, given g, f, a prover samples a randomness $r \in \mathbb{Z}_q$, sets $\mathit{\Gamma} = g^r$, and publishes $\mathit{\Gamma}$ and $z = r + d \cdot f \mod q$ as the proof, where d is the challenge value obtained from the random oracle $\mathcal{H}$, i.e., $d := \mathcal{H}(F, \mathit{\Gamma})$. Then, given the statement (g, F) and the proof $(\mathit{\Gamma}, z)$, a verifier first sets $d = \mathcal{H}(F, \mathit{\Gamma})$, and then checks if $g^z = \mathit{\Gamma} F^d$, and returns `true` or `false`.

Next, we generalize Schnorr's ID protocol and present a NIZK PoK scheme for the Polynomial DL (PDL) relation R_{PDL}, defined as follows,

$$R_{PDL} = \{(g, x_i, F_i), f(X) \mid F_i = g^{f(x_i)}\}, \quad i = 1, 2, \ldots, n, \tag{3}$$

where $f(X) \in \mathbb{Z}_q[X]_t$ is a (at most) degree $t \leq n-1$ witness polynomial with coefficients defined over $\mathbb{Z}_q$, and $x_1, \ldots, x_n$ are n *distinct* elements from $\mathbb{Z}_q$. The R_{PDL} relation is base on the PDL problem defined as follows.

Definition 5.1 (Polynomial Discrete Logarithm Problem). *Let $\mathbb{G}$ be a finite cyclic group of order q generated by g. Given $F_1, \ldots, F_n$ from $\mathbb{G}$ and distinct elements $x_1, \ldots, x_n$ from $\mathbb{Z}_q$, find a polynomial $f(X) \in \mathbb{Z}_q[X]_t$ of (at most) degree t, where $0 \leq t \leq n-1$, such that $F_i = g^{f(x_i)}$ for all $i = 1, \ldots, n$.*

In other words, an algorithm $\mathcal{A}$ has advantage ϵ in solving PDL in $\mathbb{G}$ if

$$\Pr[\mathcal{A}(x_1, \ldots, x_n, g, g^{f(x_1)}, \ldots, g^{f(x_n)}) = f(X)] \geq \epsilon$$

where $f(X) \in \mathbb{Z}_q[X]_t$ is (at most) a degree-t polynomial with $0 \leq t \leq n-1$, and the probability is over the random choice of generator $g \in \mathbb{G}^*$ and the distinct choice of $x_1, \ldots, x_n$ in $\mathbb{Z}_q$. We say that the (t, ϵ)-PDL assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the PDL problem in $\mathbb{G}$.

Occasionally we drop the t and ϵ and refer to the PDL assumption in $\mathbb{G}$. It can be seen that the hardness of the PDL problem can be reduced to that of the DL problem. As an instance, let $\mathcal{A}$ be an adversary against the PDL problem, and $(g, h := g^f)$ be the challenge values for the DL problem. Then, one can construct an adversary $\mathcal{B}$ that acts as follows. $\mathcal{B}$ sets $F_1 := h, x_1 := 1, x_2 := 2$, and additionally samples another random element F_2 from $\mathbb{G}$, and sends the tuple (g, x_1, x_2, F_1, F_2) to the adversary $\mathcal{A}$. If $\mathcal{A}$ returns $f(X)$ such that $F_1 = g^{f(1)}$ and $F_2 = g^{f(2)}$, then $\mathcal{B}$ returns $f(1)$ as the answer to the DL challenge.

We assume $t + 1 \leq n$ in the PDL problem, however, it's worth noting that as we increase n, we add more evaluations of $f(X)$ into the statement. Thus, we anticipate the existence of an upper bound for n (for a specific t), and we leave it as an interesting feature research question.

Generalization of Schnorr Protocol. In Fig. 6, we introduce a generalized version of Schnorr's NIZK PoK protocol, which enables a prover to generate a NIZK proof of knowledge for the relation R_{PDL}, as defined in Eq. (3).

Theorem 5.1 (A NIZK Proof of Knowledge for R_{PDL}). *Let g be the generator of $\mathbb{G}$, $\{F_i\}_{i=1}^n \in \mathbb{G}$, $\{x_i\}_{i=1}^n$ be n distinct elements from $\mathbb{Z}_q$, and t be the (maximum) degree of witness polynomial $f(X)$. Assuming PDL is hard, for $0 \leq t < n$, the protocol π_{PDL} (described in Fig. 6) is a NIZK PoK for R_{PDL} in the RO model.*

Proof. We first prove the security of the interactive case, and then using standard Fiat-Shamir transform, extend it to the non-interactive case in the RO model.

Prover: Given the statement $(g, x_1, \ldots, x_n, F_1, \ldots, F_n)$ and the witness polynomial $f(X)$, proceed as follows and output a proof π.

1. Sample a degree-t polynomial $r(X) \in \mathbb{Z}_q[X]_t$; Set $\{\mathit{\Gamma}_i = g^{r(x_i)}\}_{i=1}^n$.
2. Set $d \leftarrow \mathcal{H}(F_1, \ldots, F_n, \mathit{\Gamma}_1, \ldots, \mathit{\Gamma}_n)$, where $\mathcal{H}$ is a random oracle.
3. Set $z(X) = r(X) + d \cdot f(X) \pmod{q}$;
4. Return $\pi := (\mathit{\Gamma}_1, \ldots, \mathit{\Gamma}_n, z(X))$

Verifier: Given statement $(g, \{x_i, F_i\}_{i=1}^n)$ and $\pi := (\mathit{\Gamma}_1, \cdots, \mathit{\Gamma}_n, z(X))$, the verifier first checks if $z(X)$ is a degree-t polynomial. If so, then sets $d \leftarrow \mathcal{H}(F_1, \ldots, F_n, \mathit{\Gamma}_1, \ldots, \mathit{\Gamma}_n)$ and checks if: $g^{z(x_i)} = \mathit{\Gamma}_i (F_i)^d$ for $i = 1, \ldots, n$, and outputs `true` or `false`. Note that to make the communication shorter, as in Schnorr's ID protocol, alternatively, the prover could publish $\pi := (d, z(X))$, and then the verifier would need to check if $z(X)$ is a degree-t polynomial and $d = \mathcal{H}(F_1, \ldots, F_n, \frac{g^{z(x_1)}}{F_1^d}, \ldots, \frac{g^{z(x_n)}}{F_n^d})$.

Fig. 6. π_{PDL}: An efficient NIZK proof of knowledge for R_{PDL}.

Completeness. If the prover and verifier honestly follow the protocol, for $i = 1, \ldots, n$, we have

$$g^{z(x_i)} = g^{r(x_i) + df(x_i)} = g^{r(x_i)} + \left(g^{f(x_i)}\right)^d = \mathit{\Gamma}_i F_i^d \ .$$

Special Soundness: Let $(\mathit{\Gamma}_i, d, z(X))$ and $(\mathit{\Gamma}_i, d', z'(X))$ be two acceptable transcripts with the same commitments and different challenge values, that are obtained by rewinding. Then, from the verification equation, we know that for $i = 1, \ldots, n$:

$$g^{z(x_i)} = \mathit{\Gamma}_i (F_i)^d \quad , \quad g^{z'(x_i)} = \mathit{\Gamma}_i (F_i)^{d'} \ .$$

This implies that, for $i = 1, \ldots, n$:

$$g^{z(x_i) - z'(x_i)} = F_i^{d - d'} \ \Rightarrow \ F_i = g^{\frac{z(x_i) - z'(x_i)}{d - d'}} \ .$$

Since $z(X)$ is a degree-t polynomial, therefore, if all the $n \geq t+1$ of the checks pass, from $f_i := \frac{z(x_i) - z'(x_i)}{d - d'}$ for $i = 1, \ldots, n$, we can obtain $n \geq t + 1$ *distinct* evaluations of a unique degree-t polynomial at points $x_1, \ldots, x_n$. Considering the fact that any degree-t polynomial can be determined from its $t + 1$ *distinct* evaluations, using Lagrange interpolation, w.l.o.g. an extractor can use $\{f_i\}_{i=1}^{t+1}$ and reconstruct (extract) a *unique* degree-t polynomial $f(X)$, which is a witness (resp. solution) for R_{PDL} relation (resp. PDL problem).

Honest Verifier Zero-Knowledge (HVZK): Next, we show that given the statement $(g, \{x_i, F_i\}_{i=1}^n)$ and the challenge value d, a simulator can simulate the transcript of the protocol. To this end, the simulator first randomly samples a degree-t polynomial $z'(X) \in \mathbb{Z}_q[X]_t$. Then, for $i = 1, \ldots, n$: sets $\mathit{\Gamma}'_i = \frac{g^{z'(x_i)}}{F_i^d}$. Finally, the simulator returns $(\{\mathit{\Gamma}'_i\}_{i=1}^n, z'(X))$ as the simulated proof. As it can be seen, since $z'(X)$ is sampled randomly, therefore $\{\mathit{\Gamma}'_i\}_{i=1}^n$ are also random, and the simulated proof is indistinguishable from the real one.

Since the interactive scheme is public coin, and satisfies completeness, (perfect) special soundness, and (computational) HVZK, then, in the random oracle model, using Fiat-Shamir transform [16], it can be turn into a NIZK proof of knowledge scheme for R_{PDL} (defined in eq. (3)). □

Efficiency of π_{PDL} and Related works. In π_{PDL}, a prover needs to evaluate a degree-t polynomial in n points, and compute n EXP in $\mathbb{G}$ and a single hash. Subsequently, the prover publishes a proof $\pi := (d, z(X))$, comprising $t+2$ field elements. On the other side, a verifier needs to evaluate a degree-t polynomial in n points, and compute $2n$ EXP, and 1 hashing operation.

In [6,11], authors introduced two different generalizations of the DL problem: the q-Diffie-Hellman Inverse [11] and the Generalized Diffie-Hellman [6]. In both cases, the adversary finally needs to compute a group element. In contrast, in the PDL problem, the adversary is tasked with computing a degree-t polynomial. To the best of our knowledge, this is the first time that the problem PDL (given in Eq. (3)) is explicitly defined and a NIZK PoK is presented for it. However, it is worth noting that it has been implicitly used in previous VSS schemes [9,10,15, 24]. In Feldman VSS scheme [15], given a set of commitments $\{c_j\}_{j=0}^{t}$, one can compute $g^{f(i)}$ for arbitrary value of i using the formula $g^{f(i)} = \prod_{j=0}^{t} c_j^{i^j}$. In [10], Cascudo and David also developed a sigma protocol for a variant of R_{PDL}. In this variation, they utilize different generators, specifically $F_i = g_i^{f(i)}$, instead of $F_i = g^{f(i)}$. However, when examining the proof of special soundness in their sigma protocol [10, Proposition 1], certain steps are unclear. Notably, there is an absence of a definitive statement and reduction to a hardness assumption. In other words, their proof of special soundness lacks an extraction algorithm. Furthermore, in their work [10], they introduce a probabilistic check protocol for R_{PDL} and specify that they have no prover. In a general sense, their check protocol uses locally computable checks based on [9]. In this approach, verifiers employ a random codeword from the dual code of the Reed-Solomon code, which was used in the statement. However, in essence, their check protocol can be seen as a non-interactive proof scheme that, in comparison to our NIZK *proof of knowledge* scheme π_{PDL}, achieves *soundness*. Tab. 3 compares the performance metrics for our proposed NIZK proof of knowledge π_{PDL} and compares it with the probabilistic check protocol from [10].

Table 3. A comparison of NIZK PoK π_{PDL} with Cascudo and David's probabilistic check protocol for R_{PDL} [10]. n: # Elements in the statement, t: degree of the witness polynomial, $E_{\mathbb{G}}$: Exponentiation in group $\mathbb{G}$, $M_{\mathbb{G}}$: Multiplication in $\mathbb{G}$, $\mathcal{PE}$: degree-t Polynomial Evaluation, $\mathcal{H}$: Hashing, $|\mathbb{Z}_q/\mathbb{G}|$: $\mathbb{Z}_q/\mathbb{G}$ element size, $|\pi|$: proof size, $|stat|$: Statement size.

Proof Schemes	Prover	$\|\pi\| + \|stat\|$	Verification
Check Protocol [10]	$n\ E_{\mathbb{G}} + n\ \mathcal{PE}$	$n\ \|\mathbb{G}\|$	$n\ E_{\mathbb{G}} + n\ \mathcal{PE} + n\ M_{\mathbb{G}}$
π_{PDL}, Fig. 6	$n\ E_{\mathbb{G}} + n\ \mathcal{PE} + 1\ \mathcal{H}$	$t\ \|\mathbb{Z}_q\| + n\ \|\mathbb{G}\|$	$2n\ E_{\mathbb{G}} + n\ \mathcal{PE} + 1\ \mathcal{H}$

6 $\mathbf{\Pi_S}$: A Novel PVSS Scheme from DL

Building upon the new VSS scheme $\mathbf{\Pi_F}$ (from Sec. 4.1) and leveraging the new NIZK PoK scheme π_{PDL} (from Sec. 5), we present a novel PVSS scheme in this section. It offers a more efficient alternative to Schoenmakers' scheme [24]. For further context, we have provided an overview of Schoenmakers' construction [24] in App. A, while here we introduce the new scheme.

$\mathbf{\Pi_S}$: An Efficient Alternative to Schoenmakers Scheme. Let g be a random generator of group $\mathbb{G}$. In the initialization step, a party P_i generates a secret key $s_i \leftarrow\!\!\$\ \mathbb{Z}_q$ and registers $h_i = g^{s_i}$, as its public key.

In a PVSS scheme, the dealer encrypts the shares under the public keys of the parties and subsequently proves the validity of these encrypted shares. This means ensuring that all the encrypted shares are distinct evaluations of a unique degree-t polynomial $f(X)$. Next, we show that building upon $\mathbf{\Pi_F}$ and incorporating the NIZK PoK π_{PDL} (shown in Fig. 6) for $n \geq 2t+1$, we can develop a more efficient PVSS scheme, designated as $\mathbf{\Pi_S}$. In $\mathbf{\Pi_S}$, instead of committing to the shares, the dealer initially encrypts the shares f_i under the public key h_i by computing $y_i = h_i^{f_i}$ for $i = 1, \ldots, n$. Subsequently, the dealer employs a minimally modified version of the prover from π_{PDL}. This modified prover operates with the inputs $(h_1, \ldots, h_n, 1, \ldots, n, y_1, \ldots, y_n)$ instead of $(g, 1, \ldots, n, y_1, \ldots, y_n)$, and generates a NIZK proof $(d, z(X) = r(X) + df(X))$. In this adapted version of π_{PDL}, the prover, for $i = 1, \ldots, n$, sets $c_i = h_i^{r(i)}$ instead of the original protocol's $c_i = g^{r(i)}$. At the end, the dealer discloses $\{y_i\}_{i=1}^n$ as the encryptions of shares and $\pi_{Share} := (d, z(X))$ as the proof [6]. Its important to note that, in $\mathbf{\Pi_S}$, similar to Schoenmakers' scheme [24], the secret is equal to $g^{f(0)}$.

Verification. Given $(\{h_i, y_i\}_{i=1}^n, d, z(X))$, to verify the shares, a *public* verifier first checks if $z(X)$ is a degree-t polynomial. If so, it checks if $d = \mathcal{H}(y_1, \ldots, y_n, \frac{h_1^{z(1)}}{y_1^d}, \ldots, \frac{h_n^{z(n)}}{y_n^d})$, and outputs either `true` or `false`.

Reconstruction. The reconstruction phase can be done in the same way as in Schoenmakers' PVSS scheme [24], which we summarized before. Note that, as in Schoenmakers' scheme, to reconstruct the secret $g^{f(0)}$, the parties do not learn and use the values of $f(i)$, rather than $F_i = g^{f(i)}$. Also, they do not expose their secret keys, and party P_i can reuse his/her key pair (h_i, s_i) in several runs of the PVSS scheme. The description of $\mathbf{\Pi_S}$ is summarized in Fig. 7.

[6] It's important to note that this variant of π_{PDL} shares similarities with the sigma protocol proposed in [10], but with two key differences. In our case, we make an assumption that $n \geq 2t + 1$ (as opposed to their protocol where it's $n > t$) and crucially we require at least $t + 1$ of the public key owners to be honest and not collude with the dealer. Under these assumptions, we are able to prove the special soundness of this variant, and given the secret keys of $t+1$ honest parties, construct an efficient extraction algorithm that extracts a PDL witness from the prover (i.e., the dealer). For more details, please refer to the proof of Theorem 6.1.

Initialization: As in $\mathbf{\Pi}$, given the generator g for $\mathbb{G}$, parties register a PK $h_i := g^{s_i}$.

Share: Given $\{h_i\}_{i=1}^n$, the parameters n and t, to share f_0, the dealer acts as follows:

- Samples a uniformly random polynomial $f(X)$ of degree t with coefficients in $\mathbb{Z}_q$, subject to $f(0) = f_0$.
- For $i = 1, 2, \cdots, n$: sets $y_i := h_i^{f_i}$ as the encryption of $f_i = f(i)$.
- Runs a minimally modified version of the prover of π_{PDL} (outlined in Fig. 6) with $(h_i, i, y_i)_{i=1}^n$ and obtains $\pi_{Share} := (d, z(X) = r(X) + df(X))$.
- Broadcasts $\{y_i\}_{i=1}^n$ as the encryption of shares, and π_{Share} as the proof.

Verification: Given $\{h_i\}_{i=1}^n$, ciphertexts $\{y_i\}_{i=1}^n$, and the proof $\pi_{Share} := (d, z(X))$, a verifier first checks if $z(X)$ is a degree-t polynomial. If so, it checks if $d = \mathcal{H}(y_1, \ldots, y_n, \frac{h_1^{z(1)}}{y_1^d}, \ldots, \frac{h_n^{z(n)}}{y_n^d})$ and returns `true` or `false`.

Reconstruction: Based on the reconstruction approach outlined in Fig. 2 and [24] parties proceed as follows.

1. They first use their secret key s_i, and obtain their share $F_i := g^{f_i}$ from y_i by computing $F_i = y_i^{1/s_i}$. Then, they publish F_i plus a NIZK proof that the value F_i is a correct decryption of y_i. To this end, the party P_i needs to prove knowledge of an s_i such that, $(h_i = g^{s_i}) \wedge (y_i = F_i^{s_i})$, which can be done by the NIZK proof scheme for the DLEQ relation (given in Fig. 1).
2. Then, given any $t+1$ valid values of F_i, w.l.o.g. for $i = 1, \ldots, t+1$, the secret $g^{f(0)}$, can be obtained by Lagrange interpolation,

$$\prod_{i=1}^{t+1} F_i^{\lambda_i} = \prod_{i=1}^{t+1} (g^{f_i})^{\lambda_i} = g^{\sum_{i=1}^{t+1} f_i \lambda_i} = g^{f(0)} = g^{f_0},$$

where $\lambda_i = \prod_{j \neq i} \frac{j}{j-i}$ is a Lagrange coefficients.

Fig. 7. $\mathbf{\Pi_S}$: An efficient PVSS scheme from discrete logarithm.

Security. The security of $\mathbf{\Pi_S}$ can be proven in the random oracle model through some modifications in the proof of Theorem 5.1 and by referencing [24, Theorem 1, 2] under the PDL and Decisional Diffie-Hellman (DDH) assumptions.

Theorem 6.1 (Security of PVSS Scheme $\mathbf{\Pi_S}$). *Under the PDL and Decisional Diffie-Hellman (DDH) assumptions, the VSS scheme $\mathbf{\Pi_S}$ (outlined in Fig. 7), is a secure PVSS scheme against an static adversary in the random oracle model. That is, (i) the Reconstruction protocol results in the secret distributed by the dealer for any qualified set of shareholders, (ii) any non-qualified set of shareholders is unable to recover any (partial) information on the secret.*

Proof. We need to show that for any group of $t+1$ honest parties (referred to as a qualified set), the reconstruction protocol outlined in Fig. 7 results in a unique secret $g^{f(0)}$, distributed by the dealer. Additionally, we need to show that the new scheme satisfies unpredictability, meaning that, any subset of up to t parties is unable to recover any (partial) information on the secret.

To begin, akin to the proof of Theorem 5.1, for proving the special soundness of the interactive variant of the NIZK proof scheme employed during the sharing phase, we can argue as follows. Given two acceptable transcripts of the (interactive) protocol, denoted as $(c_i, d, z(X))$ and $(c_i, d', z'(X))$ for $i = 1, \ldots, n$, from the verification equation, we know that

$$h_i^{z(i)} = c_i(y_i)^d \quad , \quad h_i^{z'(i)} = c_i(y_i)^{d'} \quad \text{for } i = 1, \ldots, n \ .$$

This implies that,

$$h_i^{z(i)-z'(i)} = y_i^{d-d'} \ \Rightarrow \ y_i = h_i^{\frac{z(i)-z'(i)}{d-d'}} \qquad \text{for } i = 1, \ldots, n \ .$$

Then, if all $n \geq 2t+1$ of the checks in the verification process successfully, given the reconstruction protocol detailed in Fig. 7, any set of $t+1$ honest parties can decrypt $\{y_i\}_{i \in Q, |Q|=t+1}$, as $F_i := y_i^{1/s_i}$, and rewrite the last equation as below,

$$g^{z(i)-z'(i)} = F_i^{d-d'} \ \Rightarrow \ F_i = g^{\frac{z(i)-z'(i)}{d-d'}} \qquad \text{for } i \in Q, \ |Q| = t+1.$$

Now, since $z(X)$ is a degree-t polynomial, and since from $f_i := \frac{z(i)-z'(i)}{d-d'}$ for $i \in Q, |Q| = t+1$, we obtain $t+1$ *distinct* evaluations of a degree-t polynomial $\frac{z(X)-z'(X)}{d-d'}$, therefore an extractor can use $\{f_i\}_{i \in Q, |Q|=t+1}$ and reconstruct (extract) a *unique* degree-t polynomial $f(X)$, which is a witness for the R_{PDL} relation (or PDL problem). This implies that, any set of $t+1$ honest parties, can use their individual (decrypted) shares $F_i := y_i^{1/s_i}$, employ Lagrange interpolation (as in Fig. 7), and evaluate a unique degree-t polynomial $f(X)$ in the *exponent* for $i = 0, 1, \ldots, n$. By evaluating $g^{f(X)}$ at point 0, they can obtain a unique secret value $g^{f(0)}$.

Regarding unpredictability, it's important to first note that directly breaking the encryption used in the PVSS scheme implies breaking the Computational Diffie-Hellman (CDH) assumption. Because, given $g, h_i = g^{s_i}, y_i = h_i^{f_i} = g^{s_i f_i}$, an adversary would need to compute g^{f_i}. It is not a difficult task to show that if an adversary $\mathcal{A}$, manages to compute g^{f_i} with some success probability, we can construct another adversary $\mathcal{B}$ which employs $\mathcal{A}$ as a subroutine and breaks the CDH assumption with the same success probability. However, this alone does not show that parties cannot obtain partial information about the secret g^{f_0}. Furthermore, we show that the view of up to t parties is simulatable. To achieve this goal, a simulator proceeds as follows. W.l.o.g., it first samples $f(1), \ldots, f(t)$ randomly from $\mathbb{Z}_q$ and sets $F_1 = g^{f(1)}, \ldots, F_t = g^{f(t)}$, and $y_1 = h_1^{f(1)}, \ldots, y_t = h_t^{f(t)}$, where $h_1, \ldots, h_t$ are public keys of the t parties. Then, he samples $F_{t+1} = g^{f_{t+1}}$ randomly, without knowing f_{t+1}. Since the point $f_{t+1} = f(t+1)$ is only given implicitly, we cannot compute the point $f(t+2), \ldots, f(n)$. It suffices, however, that we can compute $F_{t+2} = g^{f_{t+2}}, \ldots, F_n = g^{f_n}$ by Lagrange interpolation, which also yields the remaining shares. The simulator, now deviates from the protocol by computing the public keys h_i of parties $\{P_i\}_{i=t+1}^n$ as $h_i := g^{w_i}$ for random $w_i \in \mathbb{Z}_q$. Then, the simulator sets $y_i = F_i^{w_i}$ for $i = t+1, \ldots, n$. This leads to obtain $h_1, \ldots, h_n$ and $y_1 = h_1^{f(1)}, \ldots, y_n = h_n^{f(n)}$, as required. Next, we note that the underlying proof scheme (i.e., a variant of π_{PDL} from Fig. 6) is honest-verifier zero-knowledge in the interactive case (and ZK in the non-interactive case). Akin to the proof of Theorem 5.1, given the (simulated) statement $\{h_i, y_i\}_{i=1}^n$ and the challenge value d, the simulator can sample a random degree-t polynomial $z'(X)$ and set $c'_i := h_i^{z'(i)}/y_i^d$ for $i = 1, \ldots, n$. This results in a simulated transcript which under Decisional Diffie-Hellman (DDH) assumption is indistinguishable from the real view of up to t parties.

Note that the statement that parties cannot get any partial information from $(h_i^{s_i} = g^{s_i f_i}, h_i = g^{s_i})$ about the random secret s_i and f_i holds under the assumption that ElGamal encryption is semantically secure, which is known to be equivalent to the DDL assumption. Recall that in ElGamal cryptosystem, given the public key $(g, h = g^f)$, an encryption of message $m = 1$ is equal to (h^s, g^s), where s is a random value from $\mathbb{Z}_q$. □

Efficiency. Compared to Schoenmakers' scheme [24] and its variants introduced in [9], $\mathbf{\Pi_S}$ offers a better efficiency in general. However, it's worth noting that by applying the same optimization as used in $\mathbf{\Pi_S}$ to reduce the proof length, the unpacked version of Cascudo and David's scheme [10] can achieve a performance level on par with $\mathbf{\Pi_S}$. For a detailed comparison, please refer to Table 1.

7 Conclusion

We introduced $\mathbf{\Pi}$, as a unified framework for building VSS protocols based on Shamir secret sharing [25] that works in the honest majority setting, achieves optimal resilience, and does not necessarily require a homomorphic commitment, rather than a secure commitment scheme and a random oracle.

Leveraging $\mathbf{\Pi}$, we proposed three VSS schemes, so-called $\mathbf{\Pi_F}, \mathbf{\Pi_P}$, $\mathbf{\Pi_{LA}}$, and a PVSS scheme, labeled $\mathbf{\Pi_S}$, which each satisfies different properties. $\mathbf{\Pi_F}$ and $\mathbf{\Pi_P}$ are two RO-based alternatives to the well-known VSS schemes proposed by Feldman [15] and Pedersen [22], while offering a faster verification and reconstruction. $\mathbf{\Pi_{LA}}$ is another instantiation of $\mathbf{\Pi}$ in the quantum random oracle model which compared to the recent VSS scheme proposed by Atapoor, Baghery, Cozzo, and Pedersen [1] it is slightly more efficient in terms of both computational and communication costs. $\mathbf{\Pi_S}$ is a variation of Schoenmakers' construction [24] and represents a highly efficient PVSS scheme.

We evaluated the empirical performance of our proposed VSS schemes $\mathbf{\Pi_P}$ and $\mathbf{\Pi_{LA}}$ via a prototype implementation, and compared them with Pedersen [22] and ABCP [1] constructions. Our asymptomatic comparisons and implementation results confirm that in general the proposed constructions, outperform the state-of-the-art VSS schemes in the majority-honest setting. Particularly, $\mathbf{\Pi_{LA}}$ can be an attractive scheme for post-quantum threshold cryptography.

We instantiated $\mathbf{\Pi}$ with DL-based and hash-based commitments. However, it is general enough to be instantiated with different commitment schemes, particularly those that are based on PQ-secure cryptographic assumptions.

As a tool for $\mathbf{\Pi}$, we proposed a novel NIZK proof scheme that might be independently interesting. Specifically, we have defined an extended version of the discrete logarithm relation over *polynomials*, named R_{PDL}, and presented a new variant of Schnorr's NIZK proof of knowledge scheme for the R_{PDL} relation.

We think that this new NIZK Proof of Knowledge scheme for the R_{PDL} relation can be a useful tool for the development of more efficient threshold protocols based on Shamir secret sharing. As an example, we have already incorporated it within PVSS scheme $\mathbf{\Pi_S}$.

At the end, we highlight that the notable efficiency and general nature of the new framework makes it a valuable tool for constructing more efficient VSS schemes and revisiting a wide range of threshold protocols (e.g. DKG protocols, threshold signatures, threshold decryption, and more). Delving into the details of such protocols lies beyond the main scope of this paper. Future research can explore the further applications of $\mathbf{\Pi}$ and integration of new VSS schemes and the new NIZK proof scheme into various cryptographic protocols.

Acknowledgments

We would like to thank Ignacio Cascudo, Janno Siim and Daniele Cozzo for their valuable comments, and Shahram Khazaei, Hien Chu, Jannik Spiessens, Robi Pedersen, and Navid Ghaedi Bardeh for useful discussions and remarks about the earlier version of this work. This work has been supported in part by the CyberSecurity Research Flanders with reference number VR20192203.

References

1. Atapoor, S., Baghery, K., Cozzo, D., Pedersen, R.: VSS from distributed ZK proofs and applications. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings. Lecture Notes in Computer Science, Springer (2023), https://eprint.iacr.org/2023/992
2. Backes, M., Kate, A., Patra, A.: Computational verifiable secret sharing revisited. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 590–609. Springer, Heidelberg, Germany, Seoul, South Korea (Dec 4–8, 2011).
3. Baghery, K., Cozzo, D., Pedersen, R.: An isogeny-based ID protocol using structured public keys. In: Paterson, M. (ed.) Cryptography and Coding - 18th IMA International Conference, IMACC 2021, Oxford, UK, December 14-15, 2021, Proceedings. Lecture Notes in Computer Science, vol. 13129, pp. 179–197. Springer (2021). https://doi.org/10.1007/978-3-030-92641-0_9
4. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: 20th Annual ACM Symposium on Theory of Computing. pp. 1–10. ACM Press, Chicago, IL, USA (May 2–4, 1988).
5. Bishnoi, A., Clark, P.L., Potukuchi, A., Schmitt, J.R.: On zeros of a polynomial in a finite grid. Combinatorics, Probability and Computing **27**(3), 310–333 (2018)
6. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology – EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3027, pp. 223–238. Springer, Heidelberg, Germany, Interlaken, Switzerland (May 2–6, 2004).

7. Boneh, D., Boyle, E., Corrigan-Gibbs, H., Gilboa, N., Ishai, Y.: Zero-knowledge proofs on secret-shared data via fully linear PCPs. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019, Part III. Lecture Notes in Computer Science, vol. 11694, pp. 67–97. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2019).
8. Bootle, J., Groth, J.: Efficient batch zero-knowledge arguments for low degree polynomials. In: Abdalla, M., Dahab, R. (eds.) PKC 2018: 21st International Conference on Theory and Practice of Public Key Cryptography, Part II. Lecture Notes in Computer Science, vol. 10770, pp. 561–588. Springer, Heidelberg, Germany, Rio de Janeiro, Brazil (Mar 25–29, 2018).
9. Cascudo, I., David, B.: SCRAPE: Scalable randomness attested by public entities. In: Gollmann, D., Miyaji, A., Kikuchi, H. (eds.) ACNS 17: 15th International Conference on Applied Cryptography and Network Security. Lecture Notes in Computer Science, vol. 10355, pp. 537–556. Springer, Heidelberg, Germany, Kanazawa, Japan (Jul 10–12, 2017).
10. Cascudo, I., David, B.: ALBATROSS: Publicly AttestabLe BATched Randomness based On Secret Sharing. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020, Part III. Lecture Notes in Computer Science, vol. 12493, pp. 311–341. Springer, Heidelberg, Germany, Daejeon, South Korea (Dec 7–11, 2020).
11. Catalano, D., Fiore, D.: Practical homomorphic MACs for arithmetic circuits. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology – EUROCRYPT 2013. Lecture Notes in Computer Science, vol. 7881, pp. 336–352. Springer, Heidelberg, Germany, Athens, Greece (May 26–30, 2013).
12. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) Advances in Cryptology – CRYPTO'92. Lecture Notes in Computer Science, vol. 740, pp. 89–105. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 1993).
13. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In: 26th Annual Symposium on Foundations of Computer Science. pp. 383–395. IEEE Computer Society Press, Portland, Oregon (Oct 21–23, 1985).
14. Dalskov, A., Lee, E., Soria-Vazquez, E.: Circuit amortization friendly encodings and their application to statistically secure multiparty computation. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 213–243. Springer (2020)
15. Feldman, P.: A practical scheme for non-interactive verifiable secret sharing. In: 28th Annual Symposium on Foundations of Computer Science. pp. 427–437. IEEE Computer Society Press, Los Angeles, CA, USA (Oct 12–14, 1987).
16. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology – CRYPTO'86. Lecture Notes in Computer Science, vol. 263, pp. 186–194. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 1987).
17. Fujisaki, E., Okamoto, T.: A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In: Nyberg, K. (ed.) Advances in Cryptology – EUROCRYPT'98. Lecture Notes in Computer Science, vol. 1403, pp. 32–46. Springer, Heidelberg, Germany, Espoo, Finland (May 31 – Jun 4, 1998).
18. Gennaro, R., Rabin, M.O., Rabin, T.: Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In: Coan, B.A., Afek, Y. (eds.) 17th ACM Symposium Annual on Principles of Distributed Computing. pp. 101–111. Association for Computing Machinery, Puerto Vallarta, Mexico (Jun 28 – Jul 2, 1998).

19. Gentry, C., Halevi, S., Lyubashevsky, V.: Practical non-interactive publicly verifiable secret sharing with thousands of parties. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022, Part I. Lecture Notes in Computer Science, vol. 13275, pp. 458–487. Springer, Heidelberg, Germany, Trondheim, Norway (May 30 – Jun 3, 2022).
20. Groth, J.: Non-interactive distributed key generation and key resharing. Cryptology ePrint Archive, Report 2021/339 (2021), https://eprint.iacr.org/2021/339
21. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: Abe, M. (ed.) Advances in Cryptology – ASIACRYPT 2010. Lecture Notes in Computer Science, vol. 6477, pp. 177–194. Springer, Heidelberg, Germany, Singapore (Dec 5–9, 2010).
22. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) Advances in Cryptology – CRYPTO'91. Lecture Notes in Computer Science, vol. 576, pp. 129–140. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 11–15, 1992).
23. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) Advances in Cryptology – CRYPTO'89. Lecture Notes in Computer Science, vol. 435, pp. 239–252. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 1990).
24. Schoenmakers, B.: A simple publicly verifiable secret sharing scheme and its application to electronic. In: Wiener, M.J. (ed.) Advances in Cryptology – CRYPTO'99. Lecture Notes in Computer Science, vol. 1666, pp. 148–164. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 1999).
25. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979). https://doi.org/10.1145/359168.359176
26. Shoup, V., Smart, N.P.: Lightweight asynchronous verifiable secret sharing with optimal resilience. IACR Cryptol. ePrint Arch. p. 536 (2023), https://eprint.iacr.org/2023/536
27. Stadler, M.: Publicly verifiable secret sharing. In: Maurer, U.M. (ed.) Advances in Cryptology – EUROCRYPT'96. Lecture Notes in Computer Science, vol. 1070, pp. 190–199. Springer, Heidelberg, Germany, Saragossa, Spain (May 12–16, 1996).
28. Tomescu, A., Chen, R., Zheng, Y., Abraham, I., Pinkas, B., Golan-Gueta, G., Devadas, S.: Towards scalable threshold cryptosystems. In: 2020 IEEE Symposium on Security and Privacy. pp. 877–893. IEEE Computer Society Press, San Francisco, CA, USA (May 18–21, 2020).

A Overview of Schoenmakers PVSS Scheme

In Crypto 99, Schoenmakers [24] proposed a PVSS scheme, based on Feldman's scheme, which allows a dealer to encrypt the shares under the public key of the parties, and then generate a publicly-verifiable non-interactive ZK proof to show that the secret sharing and encryptions are done correctly.

Let g, h be two random generators of the group $\mathbb{G}$. In the initialization step, a party P_i generates a secret key $s_i \leftarrow\!\!\$\ \mathbb{Z}_q$ and registers $y_i = g^{s_i}$, as its public key. Then, given n and t, to share a *high-entropy* secret f_0, the dealer of Schoenmakers' construction proceeds as follows:

1. Sample a uniformly random degree-t polynomial $f(X) := f_0 + a_1 X + \cdots + a_t X^t$ with coefficients in $\mathbb{Z}_q$, subject to $f(0) = f_0$.
2. For $i = 1, 2, \cdots, n$: set $f_i := f(i)$ and $y'_i = y_i^{f(i)}$.
3. Set $c_0 = h^{f_0}$ and $c_j = h^{a_j}$ for $j = 1, 2, \cdots, t$.
4. Let $x_i = \prod_{j=0}^{t} c_j^{i^j}$, for $i = 1, 2, \cdots, n$. Then, the dealer shows that the encrypted shares y'_i are consistent by producing a proof of knowledge of the unique $f(X)$, $1 \leq i \leq n$, satisfying: $x_i = h^{f(i)} \ \wedge \ y'_i = y_i^{f(i)}$.
5. To generate the proof for above relation, the dealer uses an extended version of Chaum-Pedersen PoK scheme for DLEQ [12] and acts as follows:
   (a) For $i = 1, 2, \cdots, n$, it samples $r_i \leftarrow\!\!\$\ \mathbb{Z}_q$, and sets $a_i = h^{r_i}$ and $b_i = y_i^{r_i}$.
   (b) Using Fiat-Shamir transform, feeds $\{a_i, b_i, x_i, y'_i\}_{i=1}^{n}$ into the random oracle $\mathcal{H}$, an obtains a challenge value $d \in \mathbb{Z}_q$.
   (c) For $i = 1, 2, \cdots, n$: computes $z_i = r_i - d \cdot f_i \mod q$.
6. Publish $\pi_{Share} := (h, c_j, y_i, y'_i, d, z_i)$ for $0 \leq j \leq t$, and $1 \leq i \leq n$.

Verification. To verify the shares, given $\pi_{Share} := (h, c_j, y_i, y'_i, d, z_i)$ for $0 \leq j \leq t$, and $1 \leq i \leq n$, the verifier acts as follows:

- For $1 \leq i \leq n$: computes $x_i = \prod_{j=0}^{t} c_j^{i^j}$.
- For $1 \leq i \leq n$: using $(h, d, x_i, y_i, y'_i, z_i)$, computes a_i and b_i, as follows

$$a_i := h^{z_i} x_i^d \quad , \quad b_i := y_i^{z_i} (y'_i)^d$$

  and checks if the hash of $\{a_i, b_i, x_i, y'_i\}_{i=1}^{n}$ matches the challenge value d. If so returns `true`, otherwise returns `false`.

Reconstruction. To reconstruct the secret g^{f_0}, the parties proceed as follows.

1. They first use their secret key s_i, and obtain their share $F_i := g^{f_i}$ from y_i by computing $F_i = y_i^{1/s_i}$. Then, they publish F_i plus a NIZK proof that the value F_i is a correct decryption of y_i. To this end, the party P_i needs to prove knowledge of an s_i such that, $(h_i = g^{s_i}) \wedge (y_i = F_i^{s_i})$, which is done using Chaum-Pedersen [12] proof system for DLEQ (described in Fig. 1).
2. Then, given any $t+1$ valid values of F_i, w.l.o.g. for $i = 1, \ldots, t+1$, the secret $g^{f(0)}$, can be obtained by Lagrange interpolation,

$$\prod_{i=1}^{t+1} F_i^{\lambda_i} = \prod_{i=1}^{t+1} (g^{f_i})^{\lambda_i} = g^{\sum_{i=1}^{t+1} f_i \lambda_i} = g^{f(0)} = g^{f_0},$$

   where $\lambda_i = \prod_{j \neq i} \frac{j}{j-i}$ is a Lagrange coefficient.