Fully Secure DKG Protocols for Discrete Logarithm Revisited

Karim Baghery and Hossein Moghaddas

COSIC, KU Leuven, Leuven, Belgium
firstname.lastname@kuleuven.be

Abstract. In EUROCRYPT 1999, Gennaro, Jarecki, Krawczyk, and Rabin (GJKR) showed that in the well-known Pedersen robust Distributed Key Generation (DKG) protocol for the Discrete Logarithm (DL), an adversary can bias the distribution of the resulting public key. To address this issue, they proposed a fully secure, statistically unbiased variant of the Pedersen DKG protocol. The GJKR protocol achieves robustness and guarantees a statistically uniform final public key, at the cost of $O(n^2)$ exponentiations per party, where n denotes the number of parties. In this paper, we revisit fully secure robust DKG protocols for the DL setting and propose three more efficient alternatives, each using $O(n)$ exponentiations per party in the happy path while offering different trade-offs in security, efficiency, and round complexity. Our first protocol, like the GJKR scheme, guarantees that the final public key is indistinguishable from a uniformly random group element even to a computationally unbounded distinguisher. The second protocol is concretely more efficient and ensures that the public key distribution is *computationally* indistinguishable from uniform. In our third construction, we focus on minimizing the worst-case number of rounds in the second protocol and present a variant using three rounds in the happy path and one additional reconstruction round in the unhappy path. Our third scheme can be viewed as a fully secure and round-reduced variant of the biased construction by Atapoor et al. (ASIACRYPT 2023). In comparison with the most recent low-round fully secure DKG protocols by Katz (CRYPTO 2024), Cascudo-David (EUROCRYPT 2024), Kate et al. (CCS 2024), and Boneh et al. (EUROCRYPT 2025)–all of which achieve three rounds via two online rounds and one preprocessing round (or vice versa)–our third DKG protocol requires only $O(n)$ exponentiations per party in the happy path, as opposed to at least $O(n^2)$.

Keywords: Fully Secure Distributed Key Generation, Discrete Logarithm, Pre-Constructed VSS, Shamir Secret Sharing, Robust DKG.

1 Introduction

During the last two decades, significant strides have been made in the development of Distributed Key Generation (DKG) protocols for Discrete Logarithm

(DL) instances. These protocols are crucial components in threshold cryptography, enabling secure key generation, signing, and decryption across distributed parties. In a DKG protocol, each party uses a secret sharing scheme to share its secret among the other parties, allowing only qualified sets of them to reconstruct the shared secret.

Traditional secret sharing schemes, like Shamir's protocol [Sha79], assume an honest dealer and honest parties. To relax this assumption, Verifiable Secret Sharing (VSS) schemes [CGMA85, Fel87] have been developed that withstand various attacks, including incorrect share distribution and share reconstruction. Numerous VSS schemes are built on regular secret sharing protocols, adding verifiability features on top [Fel87, Ped92, ABCP23, Bag25]. Many known VSS schemes like Feldman [Fel87] and Pedersen [Ped92] use Shamir secret sharing and exploit the homomorphic property of the Discrete Logarithm (DL) and Pedersen commitment to achieve verifiability. In [ABCP23], Atapoor, Baghery, Cozzo, and Pedersen (ABCP) introduced the first practical hash-based VSS scheme in the honest majority setting. Their scheme uses a quantum Random Oracle (RO) and a (collapsing) hash-based commitment. In [CCG24], Cascudo, Cozzo, and Giunta demonstrated that, by allowing a logarithmic increase in private communication, the broadcast overhead in the ABCP scheme can be reduced in the optimistic case. More recently, Baghery [Bag25] generalized the ABCP [ABCP23] scheme and proposed a unified framework, dubbed Π, enabling the construction of Shamir-based VSS schemes using both homomorphic and non-homomorphic commitments in both classic and quantum RO models. Instantiating Π with different commitment schemes yields several practical VSS schemes [Bag25]. In particular, the hash-based instantiation yields the state-of-the-art VSS scheme, with a consistent improvement across all metrics compared with the hash-based VSS scheme proposed by ABCP [ABCP23]. Recently, Baghery, Bardeh, Khazaei, and Rahimi (BBKR) [BBKR25] presented a round-optimal (i.e., 2-round) variant of Π [Bag25], called 2R-Π (2-Round Π), which allows one to construct practical round-optimal VSS schemes in the honest majority setting.

In a DKG protocol for DL, n parties jointly generate a public key $y = g^x$, where g is a fixed generator of a cyclic group $\mathbb{G}$. Any $t+1$ parties can reconstruct the secret key x, while a coalition of at most t parties learns no additional information about x or the honest parties' shares beyond its own shares and the public key. In [Ped91], Pedersen proposed his well-known robust DKG protocol for a DL instance. In [GJKR99, GJKR07], Gennaro, Jarecki, Krawczyk, and Rabin (GJKR) demonstrated a vulnerability in the Pedersen DKG protocol [Ped91], wherein an adversary could bias the distribution of the generated public key y. Subsequently, they used both Pedersen [Ped92] and Feldman [Fel87] VSS schemes and proposed a fully secure version of the Pedersen robust DKG protocol, guaranteeing a statistically uniform final public key y. Robustness ensures that honest parties obtain correct outputs despite malicious behavior within the corruption threshold. More recently, ABCP [ABCP23, Sec. 4.1] leveraged their new hash-based VSS scheme to introduce a new variant of the Pedersen robust DKG protocol proposed by GJKR [GJKR99, GJKR07]. This variant [ABCP23]

reduces the number of exponentiations per party by almost a factor of n. In their proposed DKG protocol, they use a commit-and-open approach to prevent possible attacks by a rushing adversary. However, in this work, we show that their protocol is still not fully secure and a rushing adversary can bias the distribution of the generated public key. In a recent work [Kat24], Katz studied the round complexity of fully secure DKG protocols in the DL setting and presented several round-optimal (i.e., 3-round) DKG protocols, each offering different trade-offs in terms of security, efficiency, and setup requirements. Cascudo and David [CD24], as well as Kate et al. [KMM+24], also proposed 3-round, fully secure DKG protocols for the DL setting based on class groups, achieving one offline round (for public-key registration) followed by two online rounds. The protocol of Cascudo and David [CD24] has $O(n^2)$ computational complexity and is claimed to be concretely more efficient than the protocols introduced in [Kat24]. More recently, Boneh, Haitner, Lindell, and Segev [BHLS25] introduced an exponent verifiable random function (eVRF) and used it, together with Feldman VSS, to construct a DKG protocol in the DL setting. Similar to Katz's construction [Kat24], their scheme requires two rounds of preprocessing and one online round; however, due to the use of Feldman VSS, parties must perform $O(n^2)$ exponentiations during the verification phase.

Our Contributions. This work presents several fully secure DKG protocols for the DL setting in the synchronous model, each using $O(n)$ exponentiations per party in the happy path while offering different trade-offs among security, efficiency, and round complexity. Each party outputs the common public key and its secret-key share. Our contributions are as follows:

A Unified Framework for DL-Oriented Pre-Constructed VSS. As a foundation for our new DKG protocols, we formalize Pre-Constructed VSS (PC-VSS) and present PC-Π (Pre-Constructed Π), a variant of the general VSS protocol Π [Bag25]. In PC-Π, the dealer additionally publishes a commitment to the shared secret and proves its consistency with the distributed shares. This supports *optimistic reconstruction*, adapted from pre-constructed publicly verifiable secret sharing [BKNR25]: the dealer opens the commitment, while threshold reconstruction remains available if the dealer withholds a valid opening. We give two concrete instantiations of PC-Π and use them in the construction of our proposed DKG protocols. Threshold reconstruction recovers a qualified dealer's contribution if the dealer fails to provide a valid proof. To construct PC-Π we adapt the DL-oriented Non-Interactive Threshold Zero-Knowledge (NI-TZK) [BBC+19] proof of ABCP [ABCP23] to Π, which uses one commitment per coordinate. The resulting NI-TZK proof covers the following distributed relations,

$$R_j = \{(y_0, f_j, f(X)) \mid y_0 = g^{f(0)} \wedge f(j) = f_j\}, \quad j = 1, \ldots, n$$

where $f(X) \in \mathbb{Z}_q[X]_t$ is a secret polynomial in X of degree (at most) t with coefficients defined over the field $\mathbb{Z}_q$. This NI-TZK proof enables a prover to convince n verifiers that the public key y_0 has been correctly generated from the

secret key $f(0)$, and that each verifier has received a valid Shamir share of this secret, i.e., $f_j = f(j)$. We believe that PC-Π, together with its underlying NI-TZK proof, can serve as a more practical building block for distributed protocols than the original Π, and that the underlying ideas are general enough to extend naturally to other domains such as lattice- and isogeny-based cryptography.

Statistically Unbiased, Fully Secure DKG Protocol for DL. In our first proposed DKG protocol, DKG_1, we combine the perfectly simulatable VSS scheme $\Pi_{\mathsf{P}+}$ from [ABN+25] with a new hash-based PC-VSS scheme which is built using our proposed PC-Π framework. To construct DKG_1, we revisit the GJKR variant [GJKR99, GJKR07] of the Pedersen DKG protocol and propose a more efficient alternative that securely realizes our DKG functionality. Similar to the original GJKR scheme, our protocol DKG_1 ensures that the final public key is indistinguishable from a uniformly random group element to a computationally unbounded distinguisher, even when up to t parties are corrupted.

Computationally Unbiased, Fully Secure DKG Protocol for DL. We combine the hash-based VSS scheme Π_{LA} [Bag25] with our new PC-VSS scheme $\mathsf{PC}\text{-}\Pi_{\mathsf{LA}}$ to revisit the GJKR DKG protocol and design DKG_2, a practical, fully

Table 1. Asymptotic computational and communication costs per party in the fully-secure GJKR [GJKR99, GJKR07] variant of Pedersen DKG [Ped92], biased ABCP DKG [ABCP23], and our proposed DKG protocols. Silent verification is not counted as a round when no complaint is sent; paired counts are happy-path/worst-case. For our protocols, costs are for the happy path; their outputs omit public verification values for the final shares. DL: Discrete Logarithm, RO: Random Oracle, stst.: statistically, uni,: uniform, rnd: Rounds, PP: pre-processing, BC: Broadcast, n: Number of parties, $t \approx n/2$: Threshold value, $E_{\mathbb{G}}$: Exponentiation in group $\mathbb{G}$, PE: degree-t Polynomial Evaluation, H: Hashing, $|\mathbb{G}|$: $\mathbb{G}$ element size, $|\mathbb{Z}_q|$: $\mathbb{Z}_q$ element size, $|H|$: hash output size (equal to $|\mathbb{Z}_q|$ for our protocols).

DKG Scheme	Security, Round	Computation	Communication
GJKR [GJKR07] (Pedersen [Ped92])	DL, fully-secure stat. uniform, 2/6 rnd	$(2nt+n)$ $E_{\mathbb{G}}$+ $2n$ PE + 1 H	PV: $2n$ $\lvert\mathbb{Z}_q\rvert$, BC: $2t \approx n$ $\lvert\mathbb{G}\rvert$
This Work Section 4.2	DL, RO, fully-secure stat. uniform, 2/6 rnd	$6n$ $E_{\mathbb{G}}$ + $5n$ PE + $6n$ H	PV: $2n$ $\lvert\mathbb{Z}_q\rvert$, BC: n $\lvert\mathbb{G}\rvert$ + n $\lvert H\rvert$ + $2t$ $\lvert\mathbb{Z}_q\rvert$
ABCP [ABCP23]	DL, RO, biased, 2/4 rnd	$3n$ $E_{\mathbb{G}}$ + $3n$ PE + $5n$ H	PV: n $\lvert\mathbb{Z}_q\rvert$, BC: $2n$ $\lvert H\rvert$ + t $\lvert\mathbb{Z}_q\rvert$
This Work Section 5	DL, RO, fully-secure comp. uniform, 2/6 rnd	$2n$ $E_{\mathbb{G}}$ + $5n$ PE + $6n$ H	PV: n $\lvert\mathbb{Z}_q\rvert$, BC: $2n$ $\lvert H\rvert$ + $2t$ $\lvert\mathbb{Z}_q\rvert$ + $O(1)$ $\lvert\mathbb{G}\rvert$
Katz [Kat24]	DL, RO, fully-secure comp. uni., 3 rnd (2 PP)	$O(\binom{n}{t})$ $E_{\mathbb{G}}$	PV: — BC: $O(n)$
Cas-Dav [CD24], KMMST [KMM+24]	DL, RO, fully-secure comp. uni., 3 rnd (1 PP)	$O(n^2)$ $E_{\mathbb{G}}$	PV: — BC: $O(n)$
BHLS [BHLS25]	DL, RO, fully-secure comp. uni., 3 rnd (2 PP)	$O(n^2)$ $E_{\mathbb{G}}$	PV: $O(n)$ BC: $O(n)$
This Work Section 6	DL, RO, fully-secure comp. uni.m, 3/4 rnd	$O(n)$ $E_{\mathbb{G}}$	PV: $O(n)$ BC: $O(n)$

secure DKG protocol for the DL setting. This construction yields the first hash-based *fully secure* DKG protocol for DL, offering significantly higher efficiency than the GJKR protocol and, by a constant factor, fewer exponentiations than our first DKG protocol DKG_1. While DKG_2 achieves full robustness and security against up to t malicious parties, it guarantees that the distribution of the generated public key is only computationally (rather than statistically) indistinguishable from uniform. Overall, DKG_2 can be viewed as a *fully secure* counterpart of the state-of-the-art *biased* DKG protocol proposed by ABCP [ABCP23].

Low-Round Computationally Unbiased, Fully Secure DKG Protocol for DL. Our third contribution is DKG_3, a round-reduced variant of DKG_2 in the worst case. Its public key computation phase follows Canetti et al. [CGJ+99] and the later formulation of Abe and Fehr [AF04]; we combine this approach with the complaint-resolution mechanism of 2R-Π [BBKR25]. Compared to DKG_2, the new protocol requires three rounds in the happy path and at most four rounds, including reconstruction after a failed public-key proof, making it a practical, fully secure, and low-round DKG protocol for DL-based primitives. Notably, in comparison with the most recent round-optimal fully secure DKG protocols of Katz [Kat24], Cascudo and David [CD24], Kate et al. [KMM+24], and Boneh, Haitner, Lindell, and Segev [BHLS25]—all of which achieve three rounds via one online round plus two preprocessing rounds (or vice versa)—our DKG_3 protocol uses $O(n)$ exponentiations per party in the happy path and operates without preprocessing.

Table 1 compares the asymptotic costs of our proposed protocols and the most relevant robust DKG protocols. Protocols with similar profiles and properties are grouped together in the table.

Outline. In Section 2, we review the preliminaries used throughout the paper. In Section 3, we introduce a DL-oriented pre-constructed variant of the unified VSS protocol Π [Bag25] and provide two concrete instantiations. In Section 4, we revisit the GJKR variant of Pedersen's DKG protocol [Ped92, GJKR07] and present a more efficient construction with the same statistical security guarantee. In Section 5, we develop a second DKG protocol that improves concrete efficiency while providing computational, rather than statistical, security. In Section 6, we present a round-reduced variant of our second protocol. Finally, in Section 7, we conclude the paper.

2 Preliminaries

Notation. We let $\mathbb{G}$ denote a cyclic group of prime order q. We let λ denote a security parameter and assume that $1/q$ is negligible in λ. A function f is called *negligible in* λ, written $\mathsf{negl}(\lambda)$, if for any constant c, there exists some λ_0, such that $f(\lambda) < \lambda^{-c}$ for $\lambda > \lambda_0$. A function that is negligible in the security parameter λ is simply called negligible. In algorithms, $\coloneqq$ denotes deterministic assignment and $=$ equality checks; $\leftarrow$ denotes uniform sampling from a set or recording an algorithm's output. We let $\mathbb{Z}_q$ denote $\{0, \ldots, q-1\}$

and $\mathbb{Z}_q[X]_t$ denote polynomials of degree at most t in the variable X and with coefficients in $\mathbb{Z}_q$. For $n \in \mathbb{N}$, we write $[n] = \{1, \ldots, n\}$. Finally all logarithms are in base 2. We model all hashes as domain-separated random oracles. We use $H : \{0,1\}^* \to \mathbb{Z}_q$ for hash commitments and for hashing the pairs committed in $\Pi_{\mathsf{P}+}$, and $H_{\mathsf{FS}} : \{0,1\}^* \to \mathbb{Z}_q$ for Fiat–Shamir challenges. Distinct purposes, protocol instances, phases, dealers, and recipient coordinates use independent domains, whose identifiers are omitted from the formulas.

Commitments. We use two randomized commitments to pairs $(a, b) \in \mathbb{Z}_q^2$, with fresh randomness for each commitment. The Pedersen commitment is

$$\mathcal{C}(a, b; \omega) = g_1^a g_2^b g_3^\omega, \qquad \omega \leftarrow \mathbb{Z}_q,$$

where g_1, g_2, g_3 are independently sampled group generators, and no discrete logarithm relation among them is given to the adversary. It is perfectly hiding and computationally binding under DL hardness. The hash commitment is

$$\mathcal{C}(a, b; \omega) = H(a, b, \omega), \qquad \omega \leftarrow \{0,1\}^\lambda.$$

This commitment is computationally hiding and binding in the random oracle model. The hiding argument is given in Appendix C.1. For high-entropy values, we also use the deterministic hash commitment $\mathcal{C}(a, b) = H(a, b)$, which omits ω; it is binding in the random oracle model but not hiding in general.

2.1 Zero-Knowledge Proofs on Secret Shared Data

We recall the definition of NI-TZK proofs [BBC+19, ABCP23] in Appendix 2.1. These proofs are used in several recent VSS schemes [ABCP23, CCG24, Bag25], including the unified framework Π [Bag25], and are also employed in our proposed PC-VSS protocols.

In typical NIZK (non-interactive zero-knowledge) arguments for NP languages, there is a single prover P and a single verifier V, where P knows both a statement $\mathbb{x}$ and witness for the statement $\mathbb{w}$, while V only knows the statement $\mathbb{x}$. In a threshold ZK proof, a single prover interacts with several verifiers $\{V_i\}_{i=1}^n$ over a network that includes secure point-to-point channels, and each verifier V_j holds a share $\mathbb{x}^{(j)} \in \mathbb{F}^{l_j}$ of the main statement $\mathbb{x}$. The single prover aims to convince multiple verifiers that the main input $\mathbb{x}$ is in some language $L \subseteq \mathbb{F}^l$. Next, we recall the formal definitions of distributed inputs (or statements), languages, relations, and threshold ZK proofs given by Boneh et al. [BBC+19].

Definition 1 (Distributed Inputs, Languages, and Relations [BBC+19]). *Let n be the number of parties, $\mathbb{F}$ be a finite field, and $l, l_1, l_2, \cdots, l_n \in \mathbb{N}$ be length parameters, where $l = l_1 + l_2 + \cdots + l_n$. An n-distributed input over $\mathbb{F}$ (or just distributed input/statement) is a vector $\mathbb{x} = \mathbb{x}^{(1)} \parallel \mathbb{x}^{(2)} \parallel \cdots \parallel \mathbb{x}^{(n)} \in \mathbb{F}^l$ where $\mathbb{x}^{(i)} \in \mathbb{F}^{l_i}$ is a piece (or share) of $\mathbb{x}$. An n-distributed language L is a set of n-distributed inputs. A distributed NP relation with witness length h is a binary*

relation $R(\mathbb{x}, \mathbb{w})$ *where* $\mathbb{x}$ *is an* n*-distributed input and* $\mathbb{w} \in \mathbb{F}^h$*. We assume that all* $\mathbb{x}$ *in* L *and* $(\mathbb{x}, \mathbb{w}) \in R$ *share the same length parameters. Finally, we let* $L_R = \{\mathbb{x} : \exists \mathbb{w}(\mathbb{x}, \mathbb{w}) \in R\}$*.*

Definition 2 (n-Verifier Interactive Proofs [BBC+19]). *An* n-Verifier Interactive Proof *scheme over* $\mathbb{F}$ *is an interactive protocol* $\Gamma = (P, V_1, V_2, \cdots, V_n)$ *involving a prover* P *and* n *verifiers* $\{V_i\}_{i=1}^n$*. The protocol proceeds as follows.*

- *The prover* P *holds an* n*-distributed input* $\mathbb{x} = \mathbb{x}^{(1)} \parallel \mathbb{x}^{(2)} \parallel \cdots \parallel \mathbb{x}^{(n)} \in \mathbb{F}^l$*, a witness* $\mathbb{w} \in \mathbb{F}^h$*, and each verifier* V_j *holds an input share* $\mathbb{x}^{(j)}$*.*
- *Parties can communicate in synchronous rounds over secure point-to-point channels. While honest parties follow the protocol, malicious parties (i.e., parties controlled by the adversary) can send arbitrary messages.*
- *At the end, each verifier outputs either* 1 *(accept) or* 0 *(reject) based on its view, where the view of* V_j *consists of its input piece* $\mathbb{x}^{(j)}$*, random coins* $r^{(j)}$*, and messages it received during the execution of* Γ*.*

$\Gamma(\mathbb{x}, \mathbb{w})$ denotes running Γ on a secret-shared input $\mathbb{x}$ and witness $\mathbb{w}$. We say that $\Gamma(\mathbb{x}, \mathbb{w})$ accepts (respectively, rejects) if at the end all verifiers output 1 (resp., 0). $View_{\Gamma,\mathcal{T}}(\mathbb{x}, \mathbb{w})$ denotes the (joint distribution of) views of verifiers $\{V_j\}_{j\in\mathcal{T}}$ in the execution of Γ on the distributed statement $\mathbb{x}$ and witness $\mathbb{w}$. Let $R(\mathbb{x}, \mathbb{w})$ be an n-distributed relation over a finite field $\mathbb{F}$. We say that an n-verifier interactive proof scheme $\Gamma = (P, V_1, \cdots, V_n)$ is a distributed threshold ZK proof protocol for R with t-security against a malicious prover *and* malicious verifiers, and with soundness error ϵ, if Γ satisfies the following properties [BBC+19, ABCP23]:

Definition 3 (Completeness). *For every* n*-distributed input* $\mathbb{x} = \mathbb{x}^{(1)} \parallel \mathbb{x}^{(2)} \parallel \cdots \parallel \mathbb{x}^{(n)} \in \mathbb{F}^l$*, and witness* $\mathbb{w} \in \mathbb{F}^h$*, such that* $(\mathbb{x}, \mathbb{w}) \in R$*, the execution of* $\Gamma(\mathbb{x}^{(1)} \parallel \mathbb{x}^{(2)} \parallel \cdots \parallel \mathbb{x}^{(n)}, \mathbb{w})$ *accepts with probability 1.*

Definition 4 (Soundness Against Prover and t Verifiers.). *For every* $\mathcal{T} \subseteq [n]$ *of size* $|\mathcal{T}| \leq t$*, every adversary* $\mathcal{A}$ *controlling the prover* P *and verifiers* $\{V_j\}_{j\in\mathcal{T}}$*, every* n*-distributed input* $\mathbb{x} = \mathbb{x}^{(1)} \parallel \mathbb{x}^{(2)} \parallel \cdots \parallel \mathbb{x}^{(n)} \in \mathbb{F}^l$*, and every witness* $\mathbb{w} \in \mathbb{F}^h$*, the following holds. If there is no* n*-distributed input* $\mathbb{x}' \in L_R$ *such that* $\mathbb{x}'_{\mathcal{H}} = \mathbb{x}_{\mathcal{H}}$*, where* $\mathcal{H} = [n] \setminus \mathcal{T}$*, the execution of* $\Gamma^\star(\mathbb{x}, \mathbb{w})$ *rejects except with probability at most* ϵ*, where* $\Gamma^\star$ *denotes the interaction of* $\mathcal{A}$ *with the honest verifiers.*

Definition 5 (Threshold ZK). *For every* $\mathcal{T} \subseteq [n]$ *of size* $|\mathcal{T}| \leq t$ *and every adversary* $\mathcal{A}$ *controlling* $\{V_j\}_{j\in\mathcal{T}}$*, there exists a simulator* $\mathcal{S}$ *such that for every* n*-distributed input* $\mathbb{x} = \mathbb{x}^{(1)} \parallel \mathbb{x}^{(2)} \parallel \cdots \parallel \mathbb{x}^{(n)} \in \mathbb{F}^l$*, and witness* $\mathbb{w} \in \mathbb{F}^h$ *such that* $(\mathbb{x}, \mathbb{w}) \in R$*, we have* $\mathcal{S}(\mathbb{x}_{\mathsf{pub}}, (\mathbb{x}^{(j)})_{j\in\mathcal{T}}) \approx_c View_{\Gamma^\star,\mathcal{T}}(\mathbb{x}, \mathbb{w})$*, where* $\mathbb{x}_{\mathsf{pub}}$ *is the public part of the statement, or empty if there is none. Here,* $\Gamma^\star$ *denotes the interaction of adversary* $\mathcal{A}$ *with the honest prover* P *and the honest verifiers* $\{V_j\}_{j\notin\mathcal{T}}$*. Statistical and perfect threshold ZK replace computational indistinguishability by negligible statistical distance and equality of distributions, respectively.*

Fiat–Shamir Extraction. We use adaptive Fiat–Shamir extraction for the 2-special-sound protocols below [AFK23]; Appendix C.2 gives the extraction guarantees and describes the verifier used to check private recipient openings.

2.2 Verifiable Secret Sharing

Our protocols use Shamir secret sharing [Sha79], which we recall below.

Shamir Secret Sharing. A $(t+1, n)$-Shamir secret sharing scheme [Sha79] allows n parties to individually hold a share f_j of a common secret s. No subset of at most t parties can learn any information about s, while any subset of at least $t+1$ parties can efficiently reconstruct it. This is achieved via polynomial interpolation over the field $\mathbb{Z}_q$. A polynomial $f(X) \in \mathbb{Z}_q[X]_t$ is chosen with constant term $s = f(0)$. Each party P_j for $j \in \{1, \cdots, n\}$ is assigned the secret share $f_j = f(j)$. Then any subset $Q \subseteq \{1, \ldots, n\}$ of at least $t+1$ parties can reconstruct the secret s via Lagrange interpolation by computing $s = f(0) = \sum_{j \in Q} f_j \cdot L_{0,j}^Q$, where $L_{0,j}^Q$ are the Lagrange basis polynomials evaluated at 0:

$$L_{0,j}^Q := \prod_{\ell \in Q \setminus \{j\}} \frac{\ell}{\ell - j} \pmod q.$$

Verifiable Secret Sharing (VSS). A typical secret sharing scheme is designed to withstand passive attacks. VSS schemes, first introduced in 1985 [CGMA85], also withstand active attacks by a malicious dealer or shareholders. Next, we recall the definition of computational VSS schemes, adapted from [Ped92,Bag25], and give a brief overview of several constructions most relevant to this work. In these definitions, the dealer employs NI-TZK proofs [BBC+19, ABCP23] to demonstrate the correctness of the distributed shares, while verification requires the participation of at least $t+1$ honest parties.

Definition 6. *An (n, t, s)-VSS consists of four polynomial-time algorithms (Initial, Share, Verify, Reconstruct) as follows:*

1. $\mathsf{Initial}(1^\lambda) \to pp$*: Given 1^λ, the public parameters pp are sampled and shared with the parties.*
2. $\mathsf{Share}(n, t, s, pp) \to (\{f_j\}_{j=1}^n, \pi^{\mathsf{sh}})$*: Given a secret s, it computes the shares $\{f_j\}_{j=1}^n$ and outputs them together with a (non-interactive) threshold proof π^{sh} of their validity.*
3. $\mathsf{Verify}(n, t, \{f_j\}_{j=1}^n, \pi^{\mathsf{sh}}) \to$ `true/false`*: Given (n, t), the shares $\{f_j\}_{j=1}^n$, and the proof π^{sh}, the algorithm outputs either* `true` *or* `false`*.*
4. $\mathsf{Reconstruct}(\{f_j\}_{j \in Q, |Q| \geq t+1}, \pi^{\mathsf{sh}}) \to \{s, \mathsf{false}\}$*: Given any $t+1$ or more of the shares and the proof π^{sh}, it returns either the secret s, or* `false`*.*

A secure VSS scheme needs to satisfy the following properties.

- **Correctness**: For any integers $n > 1$ and $t < n$, every secret $s \in \mathbb{Z}_q$, and every $Q \subseteq [n]$ with $|Q| \geq t+1$, a VSS is called correct if

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Initial}(1^\lambda), (\{f_j\}_{j=1}^n, \pi^{\mathsf{sh}}) \leftarrow \mathsf{Share}(n,t,s,pp) : \\ \mathsf{Verify}(n,t,\{f_j\}_{j=1}^n, \pi^{\mathsf{sh}}) = \mathsf{true} \end{array}\right] = 1,$$

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Initial}(1^\lambda), (\{f_j\}_{j=1}^n, \pi^{\mathsf{sh}}) \leftarrow \mathsf{Share}(n,t,s,pp), \\ s' \leftarrow \mathsf{Reconstruct}(\{f_j\}_{j\in Q}, \pi^{\mathsf{sh}}) : s' = s \end{array}\right] = 1 \ .$$

– **Verifiability**: A shareholder must be able to verify the validity of the received share and if the Verify algorithm returns true, then any qualified set of honest parties can reconstruct a unique secret s^*, except with negligible probability. More formally, given λ, for any integers $n \geq 2t+1$ and $t \geq 0$, any fixed corrupted set $\mathcal{T} \subseteq [n]$ with $|\mathcal{T}| \leq t$, and any PPT adversary $\mathcal{A}$ controlling the dealer and the shareholders in $\mathcal{T}$, let C denote the event that there exists $s^* \in \mathbb{Z}_q$ such that $\mathsf{Reconstruct}(\{f_j\}_{j\in Q}, \pi^{\mathsf{sh}}) = s^*$ for every $Q \subseteq [n] \setminus \mathcal{T}$ with $|Q| \geq t+1$. The shares in C are those retained by the honest party after verification and complaint resolution. We require

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Initial}(1^\lambda), (\{f_j\}_{j=1}^n, \pi^{\mathsf{sh}}) \leftarrow \mathcal{A}(n,t,pp), \\ b \leftarrow \mathsf{Verify}(n,t,\{f_j\}_{j=1}^n, \pi^{\mathsf{sh}}) : b = \mathsf{true} \ \wedge \ \neg C \end{array}\right] \leq \mathsf{negl}(\lambda) \ .$$

– **Unpredictability**: For any integers $n > 1$ and $t < n$, a VSS is called unpredictable if for every PPT adversary $\mathcal{A}$ that has non-adaptively corrupted up to t parties, for all λ, and a random secret s, we have:

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Initial}(1^\lambda), (\{f_j\}_{j=1}^n, \pi^{\mathsf{sh}}) \leftarrow \mathsf{Share}(n,t,s,pp); \\ \mathcal{T} \subset [n]; |\mathcal{T}| \leq t; s' \leftarrow \mathcal{A}(n,t,\{f_j\}_{j\in\mathcal{T}}, \pi^{\mathsf{sh}}) : s' = s \end{array}\right] \leq \mathsf{negl}(\lambda) \ .$$

– **Simulatability:** In some cases, the definition of unpredictability can be strengthened by requiring that the view of corrupted parties can be simulated. Intuitively, this means that even if an adversary corrupts up to t parties, its joint view can be reproduced without knowing the secret.

Feldman and Pedersen VSS Protocols. For brevity, we refer to Appendix A for a summary of the classic Feldman [Fel87] and Pedersen [Ped92] VSS schemes.

Perfectly Simulatable VSS Scheme $\Pi_{\mathsf{P}+}$. Atapoor et al. [ABN+25] proposed $\Pi_{\mathsf{P}+}$, a variant of Baghery's Π_{P} VSS scheme [Bag25] that improves performance by a constant factor. We recall Π_{P} in Appendix B.1 and present $\Pi_{\mathsf{P}+}$ directly below with $z = r - df$. Given (n,t,s) and random group generators $pp = (g_1, g_2)$, the Share, Verify, and Reconstruct algorithms act as follows:

– $\mathsf{Share}(n,t,s,pp) \to (\{f_j, \gamma_j\}_{j=1}^n, \pi^{\mathsf{sh}})$: Sample uniformly random polynomials $f(X), r(X) \in \mathbb{Z}_q[X]_t$, subject to $f(0) = s$. For $j \in [n]$, set $f_j := f(j)$, $r_j := r(j)$, and $c_j := g_1^{H(f_j, r_j)} g_2^{\gamma_j}$ for independent $\gamma_j \leftarrow \mathbb{Z}_q$. Compute $d := H_{\mathsf{FS}}(c_1, \ldots, c_n)$ and $z(X) := r(X) - df(X)$. Send (f_j, γ_j) securely to P_j and publish $\pi^{\mathsf{sh}} := (c_1, \ldots, c_n, z(X))$. No challenge rejection is needed.

– $\mathsf{Verify}(n, t, \{f_j, \gamma_j\}_{j=1}^n, \pi^{\mathsf{sh}}) \to$ `true/false`: Each P_j checks $\deg(z) \le t$, computes $d := H_{\mathsf{FS}}(c_1, \ldots, c_n)$, and checks $c_j = g_1^{H(f_j, z(j)+df_j)} g_2^{\gamma_j}$. A missing message or failed check triggers a complaint. The dealer answers by broadcasting (f_j, γ_j); a missing or invalid answer, or more than t complaints, disqualifies the dealer. Otherwise, the recipient replaces its private data with the valid broadcast answer.
– $\mathsf{Reconstruct}(\{f_j, \gamma_j\}_{j \in Q, |Q| \ge t+1}) \to \{s, \mathsf{false}\}$: Each party P_j broadcasts f_j and γ_j. Confirm an opening if $c_j = g_1^{H(f_j, z(j)+df_j)} g_2^{\gamma_j}$, where $d := H_{\mathsf{FS}}(c_1, \ldots, c_n)$. Interpolate $f(X)$ from any $t+1$ confirmed shares and output $f(0)$, or `false` if fewer than $t+1$ valid shares are available.

This variant preserves perfect simulatability and uses two exponentiations and one hash evaluation per commitment or verification. These commitments are perfectly hiding because each γ_j is sampled independently and uniformly. In the random oracle model, they are computationally binding under DL hardness: two openings to different pairs either give a collision in H or reveal $\log_{g_2} g_1$.

Computationally Simulatable Hash-Based VSS Scheme Π_{LA}. In [Bag25], Baghery presented Π_{LA}, an optimized version of the VSS scheme of Atapoor, Baghery, Cozzo, and Pedersen [ABCP23]. We recall its randomized hash-based instantiation [Bag25, Section 4.3], with the hiding and binding properties in Section 2. The form used in DKG_2 omits these additional randomizers; its security is proved directly in Theorem 5. Given H, random oracle H_{FS}, and parameters n and t, the algorithms act as follows. We write the response as $z(X) = r(X) - df(X)$, as in the VSS stage of DKG_2.

– $\mathsf{Share}(n, t, s, pp) \to (\{f_j, \omega_j\}_{j=1}^n, \pi^{\mathsf{sh}})$: Sample uniformly random polynomials $f(X), r(X) \in \mathbb{Z}_q[X]_t$, subject to $f(0) = s$. For $j \in [n]$, set $f_j := f(j)$, $r_j := r(j)$, and $c_j := H(f_j, r_j, \omega_j)$ for independent $\omega_j \leftarrow \{0,1\}^\lambda$. Compute $d := H_{\mathsf{FS}}(c_1, \ldots, c_n)$ and $z(X) := r(X) - df(X)$. Send (f_j, ω_j) securely to P_j and publish $\pi^{\mathsf{sh}} := (c_1, \ldots, c_n, z(X))$.
– $\mathsf{Verify}(n, t, \{f_j, \omega_j\}_{j=1}^n, \pi^{\mathsf{sh}}) \to$ `true/false`: Each P_j checks $\deg(z) \le t$, computes $d := H_{\mathsf{FS}}(c_1, \ldots, c_n)$, and checks $c_j = H(f_j, z(j) + df_j, \omega_j)$. A missing message or failed check triggers a complaint. The dealer answers by broadcasting (f_j, ω_j); a missing or invalid answer, or more than t complaints, disqualifies the dealer. Otherwise, the recipient replaces its private data with the valid broadcast answer.
– $\mathsf{Reconstruct}(\{f_j, \omega_j\}_{j \in Q, |Q| \ge t+1}) \to \{s, \mathsf{false}\}$: Each P_j broadcasts (f_j, ω_j). An opening is confirmed if $c_j = H(f_j, z(j) + df_j, \omega_j)$. Interpolate from any $t+1$ confirmed openings and output $f(0)$, or return `false` if fewer than $t+1$ valid openings are available.

Lemma 1 (Consistency of accepted openings). *Let $n \ge 2t+1$, and consider an accepted $\Pi_{\mathsf{P}+}$ or Π_{LA} sharing, with or without the additional hash randomizers. Except with negligible probability, the honest shares lie on a unique*

$f \in \mathbb{Z}_q[X]_t$. *Every opening accepted in a polynomial-time continuation, including a complaint answer or reconstruction submission at a corrupted index j, has first component $f(j)$.*

Proof. Fix the honest recipient set $\mathcal{H}$. For two accepting transcripts with the same commitments and distinct challenges d, d', either the transcripts open a commitment to different pairs, violating binding, or they open it to the same pair (a, b). With the convention $z = r - df$, this gives $z(j) + da = z'(j) + d'a$, hence

$$\widehat{f} = \tfrac{z'-z}{d-d'}, \qquad a = \widehat{f}(j), \qquad b = z(j) + d\widehat{f}(j).$$

Thus $\widehat{f}$ and $\widehat{r} = z + d\widehat{f}$ have degree at most t and open the commitments at these indices with the extracted opening data. For $\Pi_{\mathsf{P}+}$, binding is as stated in Section 2.2.

Apply AFK as in Section 2.1, first with the coordinates in $\mathcal{H}$. Except for the knowledge error or a binding violation, extraction gives a polynomial whose openings agree with the original accepted shares. Since $|\mathcal{H}| \geq t+1$, these identify a unique f. Suppose a later accepted opening at some index j disagrees with $f(j)$ with non-negligible probability. Apply the same extraction to $\mathcal{H} \cup \{j\}$. The modified verifier also checks that opening. The modified prover returns a rejecting response unless an accepted later opening at j disagrees with $f(j)$. The extracted polynomial must agree with the $t+1$ honest shares and with the alleged opening, contradicting its inconsistency unless binding is broken. AFK preserves the original execution and its continuation, so both comparisons concern that execution. A union bound covers all polynomially many submitted openings. □

Round-Optimal Variant of the Π VSS Scheme. Baghery, Bardeh, Khazaei, and Rahimi [BBKR25] introduced 2R-Π, a two-round variant of the unified Π framework [Bag25] for the honest-majority setting, with instantiations for Π_{P}, Π_{F}, and Π_{LA}. In the first round of 2R-Π, each recipient commits to a one-time secret key and privately sends the key to the dealer. In the second round, the dealer broadcasts masked shares, while dissatisfied recipients disclose their secret keys, enabling public complaint verification. We use a pre-constructed variant of 2R-Π_{LA}, together with its complaint-resolution mechanism, to construct DKG_3. The full 2R-Π_{LA} protocol is recalled in Figure 1.

2.3 A Σ-Protocol for Pedersen Commitment and DL Equality

Let g_1, g_2 be independently sampled group generators, and let g_3 be any group generator, possibly g_1. A prover aims to convince a verifier that, for the public statement $(g_1, g_2, g_3, a, b) \in \mathbb{G}^5$, it knows a witness $(s, \gamma) \in \mathbb{Z}_q^2$ satisfying the following relation:

$$R_{\mathsf{Ped\text{-}DLEQ}} = \left\{ \big((g_1, g_2, g_3, a, b), (s, \gamma)\big) \;\middle|\; a = g_1^s g_2^\gamma \wedge b = g_3^s \right\}. \tag{1}$$

In Fig. 2 we recall an efficient Non-Interactive Zero-Knowledge (NIZK) argument of knowledge in the random oracle model for this relation. It is obtained

All parties have access to commitment scheme $H(.)$ and a random oracle $H_{\mathsf{FS}} : \{0,1\}^* \to \mathbb{Z}_q$. There are n parties $\{P_j\}_{j=1}^n$, of which at most t may be malicious, with $n \geq 2t+1$. A dealer P_i, for a publicly known index $i \in [n]$, shares a high-entropy secret $s \in \mathbb{Z}_q$ among the verifier parties $\{P_j\}_{j \in \mathcal{V}}$, where $\mathcal{V} := [n] \setminus \{i\}$. The dealer also retains its own evaluation and opening at i.

Sharing

Round 1:

- **Dealer P_i:**
  1. Sample $f(X) \leftarrow \mathbb{Z}_q[X]_t$ uniformly with $f(0) = s$, and sample $r(X) \leftarrow \mathbb{Z}_q[X]_t$ uniformly.
  2. For each $j \in [n]$, compute the commitments $c_j := H(f_j, r_j)$, where $f_j := f(j)$ and $r_j := r(j)$. Then set $d := H_{\mathsf{FS}}\big((c_j)_{j\in[n]}\big)$, $z(X) := r(X) + d \cdot f(X)$, and $\pi^{\mathsf{sh}} := \big((c_j)_{j\in[n]},\ z(X)\big)$.
  3. Securely send f_j to party P_j for all $j \in \mathcal{V}$ and broadcast π^{sh}.
- **For each $j \in \mathcal{V}$, party P_j:** Sample $k_j \leftarrow \mathbb{Z}_q$, compute $K_j := H(k_j)$, send k_j to P_i, and broadcast commitment K_j.

Round 2:

- **Dealer P_i:** $\forall j \in \mathcal{V}$, if $H(k_j) = K_j$, then broadcast $a_j := f_j + k_j$; otherwise, broadcast f_j.
- **For each $j \in \mathcal{V}$, party P_j:** Verify that $\deg(z(X)) \leq t$ and $c_j = H(f_j, z(j) - d \cdot f_j)$. If the verification succeeds, P_j is happy and broadcasts nothing; otherwise, P_j is unhappy and broadcasts k_j.

Local Computation: For each $j \in \mathcal{V}$, party P_j:

1. Disqualify P_i and halt if, for some $\ell \in \mathcal{V}$, either of the following holds:
   (a) P_i broadcast f_ℓ such that $c_\ell \neq H(f_\ell, z(\ell) - d \cdot f_\ell)$.
   (b) P_i broadcast a_ℓ, and P_ℓ broadcast k_ℓ such that $K_\ell = H(k_\ell)$ and $c_\ell \neq H(a_\ell - k_\ell, z(\ell) - d \cdot (a_\ell - k_\ell))$.
2. Disqualify any unhappy party P_ℓ who broadcast k_ℓ such that $K_\ell \neq H(k_\ell)$. Let $\mathcal{Q} \subseteq \mathcal{V}$ denote the set of non-disqualified parties; halt if $j \notin \mathcal{Q}$.
3. Return f_j as follows: if P_i broadcast f_j in Round 2, use that value; else, if happy, use the value received in Round 1; else, compute $f_j := a_j - k_j$.

Reconstruction

Round 1: Each party P_j with $j \in \mathcal{Q} \cup \{i\}$ broadcasts its alleged share f_j.

Local Computation: For each $\ell \in \mathcal{V}$, party P_ℓ:

1. For each $j \in \mathcal{Q} \cup \{i\}$, mark party P_j as *confirmed* if $c_j = H(f_j, z(j) - d \cdot f_j)$.
2. Interpolate $f(X)$ from the f_j values of any $t+1$ *confirmed* parties. Output $f(0)$.

Fig. 1. 2-Round Π_{LA}: a round-optimal variant of Π_{LA} in the honest majority setting [BBKR25].

by applying the Fiat–Shamir transformation to the standard Σ-protocol for a preimage of the homomorphism $(s, \gamma) \mapsto (g_1^s g_2^\gamma, g_3^s)$ [Mau09]. This NIZK argument is later used as a building block in the construction of the DKG protocol DKG_3 in Section 6.

Theorem 1 (NIZK Argument for the Relation in Eq. (1)). *The protocol given in Fig. 2 is a secure NIZK argument of knowledge for relation $R_{\mathsf{Ped\text{-}DLEQ}}$ (given in Eq. (1)) that satisfies completeness, zero-knowledge, and knowledge soundness in the RO model.*

The proof is given in Appendix C.3.

3 Pre-Constructed VSS

Inspired by pre-constructed PVSS [BKNR25], we define *Pre-Constructed VSS* (PC-VSS), in which the dealer publishes a commitment to the shared secret and

Statement: $(\mathbb{x}, \mathbb{w}) \in R_{\mathsf{Ped\text{-}DLEQ}}$, where $\mathbb{x} := (g_1, g_2, g_3, a, b) \in \mathbb{G}^5$ is the public instance and $\mathbb{w} := (s, \gamma) \in \mathbb{Z}_q^2$ is the witness.

Prover: On input $(\mathbb{x}, \mathbb{w})$, proceed as follows:

1. Sample $r_1, r_2 \leftarrow \mathbb{Z}_q$ and set $c_1 := g_1^{r_1} g_2^{r_2}$, $c_2 := g_3^{r_1}$.
2. Set $e := H_{\mathsf{FS}}(g_1, g_2, g_3, a, b, c_1, c_2)$, where H_{FS} is modeled as a random oracle.
3. Set $\zeta_1 := r_1 - es \bmod q$ and $\zeta_2 := r_2 - e\gamma \bmod q$.
4. Output $\pi := (e, \zeta_1, \zeta_2)$.

Verifier: On input $\mathbb{x}$ and proof $\pi := (e, \zeta_1, \zeta_2)$, output `true` if

$$e = H_{\mathsf{FS}}\left(g_1, g_2, g_3, a, b, g_1^{\zeta_1} g_2^{\zeta_2} a^e, g_3^{\zeta_1} b^e\right),$$

and output `false` otherwise.

Fig. 2. An efficient NIZK argument of knowledge for the relation in Eq. (1).

proves that the privately distributed shares reconstruct to that secret. This commitment supports *optimistic reconstruction*: the dealer reveals the secret, and each party checks the opening locally. If the dealer withholds a valid opening, the shareholders can instead reconstruct the same secret from their shares. The commitment to the secret follows the idea of committed VSS [AF04]; optimistic reconstruction is adapted from pre-constructed PVSS [BKNR25]. Verification here uses the shareholders' private inputs, as in VSS, rather than public verification as in PVSS.

3.1 Definition of PC-VSS

We begin by formalizing PC-VSS, extending the VSS syntax with a commitment to the shared secret and optimistic reconstruction [Ped92, ABCP23, Bag25, BKNR25].

Definition 7 (Pre-Constructed VSS). *A* Pre-Constructed VSS *(PC-VSS) scheme consists of four algorithms or protocols,* $(\mathsf{Initial}, \mathsf{Share}, \mathsf{Verify}, \mathsf{Reconstruct})$*:*

1. $\mathsf{Initial}(1^\lambda)$ *outputs public parameters* pp*, including a commitment key* h_0 *for committing to the shared secret.*
2. $\mathsf{Share}(\mathsf{pp}, n, t, s)$ *distributes shares* $\{f_j\}_{j=1}^n$ *of* s *and any private opening data* ω_j *securely to their recipients, and publishes a commitment* c_0 *to* s *and a threshold proof* π^{sh} *of their consistency. The dealer retains any randomness* ω_0 *needed to open* c_0*.*
3. $\mathsf{Verify}(\mathsf{pp}, n, t, \{f_j\}_{j=1}^n, c_0, \pi^{\mathsf{sh}})$ *is the distributed verification protocol. Each shareholder uses its own share, its private opening data, and the public transcript; after any complaints are resolved, the honest shareholders agree to accept or reject the sharing.*
4. $\mathsf{Reconstruct}$ *supports two approaches:*
   (a) *(Optimistic)* $\mathsf{Rec}^{\mathsf{opt}}(h_0, c_0, s, \omega_0)$ *checks whether the claimed opening* (s, ω_0) *is valid for* c_0 *under* h_0*. It returns* s *if the check succeeds, and* false *otherwise. The randomness argument is omitted when the commitment does not require it.*

(b) *(Pessimistic)* $\mathsf{Rec}^{\mathsf{pes}}(\mathsf{pp}, n, t, \{f_j\}_{j\in Q}, c_0, \pi^{\mathsf{sh}})$ *checks the submitted shares using their opening data and reconstructs the secret from at least* $t+1$ *valid shares; otherwise it returns* false.

A secure PC-VSS scheme satisfies the following properties:

Correctness. *For honestly generated parameters and an honest sharing of any secret* s*, verification accepts with probability one. Optimistic reconstruction returns* s *on the dealer's honest opening, and pessimistic reconstruction returns* s *from the shares of any set* $Q \subseteq [n]$ *with* $|Q| \geq t+1$.

Verifiability. *For* $n \geq 2t+1$*, against any PPT adversary controlling the dealer and up to* t *shareholders, an accepted sharing fixes a unique reconstructed secret* s^**, except with negligible probability. The commitment* c_0 *admits a valid opening to* s^**, and*

$$\mathsf{Rec}^{\mathsf{pes}}(\mathsf{pp}, n, t, \{f_j\}_{j\in Q}, c_0, \pi^{\mathsf{sh}}) = s^*$$

for every set Q *of at least* $t+1$ *honest shareholders. The probability that* $\mathsf{Rec}^{\mathsf{opt}}$ *accepts an adversarial opening to a secret other than* s^* *is negligible. The dealer need not cooperate in reconstruction.*

Unpredictability. *For every PPT adversary* $\mathcal{A}$ *statically corrupting a set* $\mathcal{T} \subseteq [n]$ *with* $|\mathcal{T}| \leq t$*, and a secret* s *sampled uniformly from the secret space,*

$$\Pr\left[\mathcal{A}(\mathsf{pp}, n, t, \mathcal{T}, \{f_j, \omega_j\}_{j\in\mathcal{T}}, c_0, \pi^{\mathsf{sh}}) = s\right] \leq \mathsf{negl}(\lambda),$$

where the parameters and sharing are generated honestly and the probability is over their randomness and that of $\mathcal{A}$.

Simulatability. *For every PPT adversary* $\mathcal{A}$ *statically corrupting a set* $\mathcal{T}$ *of at most* t *shareholders, there exists a PPT simulator* $\mathcal{S}$ *s.t., for every secret* s*,*

$$\mathcal{S}\big(\mathsf{pp}, \tau, n, t, \mathcal{T}, c_0, \{f_j\}_{j\in\mathcal{T}}\big) \approx_c \mathsf{View}_{\mathsf{PC},\mathcal{A}}(s).$$

The real view consists of the corrupted shareholders' shares, their private opening data, and the public sharing and verification transcript. In both the real and simulated distributions, the public inputs and corrupted shareholders' shares are distributed as in an honest sharing. Here, τ *is empty unless simulation uses a setup trapdoor; in that case, a setup simulator generates* (pp, τ) *with the same distribution of public parameters as* $\mathsf{Initial}$*, retaining* τ *privately. Thus, the sharing reveals no information beyond the corrupted shareholders' own shares and the public commitment* c_0.

We next construct PC-Π (Pre-Constructed Π), the DL-oriented PC-VSS variant of the Π framework [Bag25]. In this instantiation, the commitment key is $h_0 = g$, and the commitment c_0 from the definition is denoted by $y_0 = g^s$. An optimistic opening consists of the scalar s alone. The shares are Shamir evaluations $f_j = f(j)$ of a polynomial of degree at most t with $f(0) = s$, where $q > n$.

Prover: Given $f(X) \in \mathbb{Z}_q[X]_t$ and the input $\mathbb{x} = (g, y_0 := g^{f(0)}, f_1, \cdots, f_n)$, proceed as follows and output a proof π.

1. Sample $r(X) \leftarrow \mathbb{Z}_q[X]_t$ uniformly at random;
2. For $j = 1, \ldots, n$: sample independent opening randomness ω_j for $\mathcal{C}$, set $\mathsf{C}_j := \mathcal{C}(f_j, r(j); \omega_j)$, and send ω_j privately to V_j;
3. Set $\mathsf{C}_0 := g^{r(0)}$ and $d := H_{\mathsf{FS}}(g, y_0, \mathsf{C}_0, \mathsf{C}_1, \ldots, \mathsf{C}_n)$, where H_{FS} is modeled as a random oracle;
4. Set $z(X) := r(X) - d \cdot f(X) \mod q$;
5. Return $\pi := (g, y_0, \mathsf{C}_0, \mathsf{C}_1, \ldots, \mathsf{C}_n, z(X))$;

Verification: Given $\pi := (g, y_0, \mathsf{C}_0, \mathsf{C}_1, \ldots, \mathsf{C}_n, z(X))$, the verifiers $\{V_j\}_{j=1}^n$ use their individual shares f_j and opening data ω_j and engage in the verification process as follows:

1. Verifier j proceeds as follows and outputs `true` or `false`.
   (a) Set $d := H_{\mathsf{FS}}(g, y_0, \mathsf{C}_0, \mathsf{C}_1, \ldots, \mathsf{C}_n)$;
   (b) If the required opening data is present, $\deg(z(X)) \leq t$, and $\mathsf{C}_j = \mathcal{C}(f_j, z(j) + d \cdot f_j; \omega_j) \ \wedge \ \mathsf{C}_0 = g^{z(0)} y_0^d$, return `true`; otherwise `false`;
2. Return `true` if *all* the verifiers return `true`; otherwise return `false`.

Fig. 3. The DL oriented NI-TZK proof for the n-distributed relation defined in Eq. (2).

3.2 A DL-Oriented NI-TZK Protocol for PC-Π

The DL-oriented NI-TZK protocol for PC-Π links a public commitment $y_0 = g^{f(0)}$ to privately distributed Shamir shares $f_j = f(j)$ of the same secret. For a polynomial $f(X) \in \mathbb{Z}_q[X]_t$, it proves the following n-distributed relations:

$$R_j = \{(y_0, f_j, f(X)) \mid y_0 = g^{f(0)} \wedge f_j = f(j)\}, \quad j \in [n], \tag{2}$$

where $n \geq 2t+1$ and $f(X)$ has degree at most t. The protocol is given in Fig. 3 for evaluation points $1, \ldots, n$; replacing j by any distinct nonzero public evaluation point α_j gives the same construction on another Shamir evaluation domain. We use either randomized commitment scheme from Section 2. For the Pedersen instantiation, set $g_1 = g$. The randomizers are sampled independently for each recipient and sent privately. The DL-oriented NI-TZK scheme (Fig. 3) is a variant of ABCP's proof for the same distributed relation [ABCP23]. We merge the two commitments at each coordinate as in Π [Bag25], publish the Schnorr first message directly instead of committing to it, and simplify the challenge input. As in [ABCP23], the construction can be seen as the conjunction of: (i) the NI-TZK protocol used in the general Π VSS protocol [Bag25], and (ii) an execution of the classic Schnorr identification protocol [Sch90]. The shares held by any $t+1$ accepting honest verifiers determine the polynomial extracted in the soundness proof. An accepting check by any further honest recipient guarantees that its share agrees with the extracted polynomial, except with negligible probability. The security of this protocol is established in the following theorem.

Theorem 2 (DL-Oriented NI-TZK Protocol for PC-Π). *Let L be an n-distributed language for the list of relations given in Eq. (2), and let $t \geq 1$ be a security threshold such that $n \geq 2t+1$. Assuming that $\mathcal{C}$ is instantiated with*

either randomized commitment scheme specified in Section 2 and the DL problem is computationally hard, the protocol described in Fig. 3 is an NI-TZK proof for L that satisfies completeness, threshold ZK, and (special) soundness against the prover and t malicious verifiers in the RO model.

Proof. Completeness is straightforward, as for $j = 1, \ldots, n$,

$$\mathcal{C}(f_j, z(j) + d \cdot f_j; \omega_j) = \mathcal{C}(f_j, r(j); \omega_j) = \mathsf{C}_j, \quad \text{and}$$

$$g^{z(0)} y_0^d = g^{r(0) - df(0)} g^{df(0)} = g^{r(0)} = \mathsf{C}_0 \ .$$

Special soundness uses the same two-transcript calculation as Π [Bag25, Theorem 3.1]. Consider accepting transcripts with the same public commitments, distinct challenges d, d', and responses z, z'. Unless binding is violated, they open each honest coordinate to the same pair, even if the opening randomness differs. Thus $z(j) + df_j = z'(j) + d' f_j$, and the public check gives $g^{z(0)} y_0^d = g^{z'(0)} y_0^{d'}$. The extracted polynomial

$$\widehat{f}(X) := \tfrac{z'(X) - z(X)}{d - d'}$$

has degree at most t and satisfies $g^{\widehat{f}(0)} = y_0$. These equations give special soundness for the relation defined by the public commitments: extraction returns $\widehat{f}$, its masking polynomial $z + d\widehat{f}$, and valid opening data, or a binding violation. The result of Attema, Fehr and Klooß (AFK), applied to the verifier that checks the honest recipients' openings as described in Section 2.1, gives expected polynomial-time knowledge extraction in the original execution with the stated knowledge error. Binding identifies the extracted evaluations with the original private inputs. Since $n - t \geq t + 1$, the honest evaluations uniquely determine this polynomial, proving the required soundness.
For threshold ZK, we use the same simulator for both commitment schemes. Its inputs are the public statement (g, y_0) and the private inputs $\{f_j\}_{j \in \mathcal{T}}$ of the corrupted set $\mathcal{T} \subseteq [n]$, $|\mathcal{T}| \leq t$. We first consider an honestly sampled challenge d. The simulator proceeds as follows:

1. Sample a random degree-t polynomial $z'(X)$; then define $\mathsf{C}'_0 := g^{z'(0)} y_0^d$.
2. Sample another random polynomial $f'(X)$ of degree at most t such that $f'(j) = f_j$ for all $j \in \mathcal{T}$. Then, define $r'(X) := z'(X) + df'(X)$.
3. For all $j = 1, \ldots, n$, sample independent opening randomness ω_j for $\mathcal{C}$ and set $\mathsf{C}'_j := \mathcal{C}(f'(j), r'(j); \omega_j)$.

The simulator outputs the public transcript together with $\{\omega_j\}_{j \in \mathcal{T}}$ and records acceptance for the simulated honest verifiers. It needs neither a setup trapdoor nor a polynomial whose constant term is the discrete logarithm of y_0.

In both executions, z' is uniform and the committed pairs at corrupted coordinates are $(f_j, z'(j) + df_j)$, with honestly distributed openings. At honest coordinates the pairs may differ, but their independent randomizers remain private. The hiding argument allows (y_0, d, z') and the corrupted openings as auxiliary

Initial: Parties agree on pp, including $(\mathbb{G}, q, g)$, a random oracle $H_{\mathsf{FS}} : \{0,1\}^* \to \mathbb{Z}_q$, and the parameters for $\mathcal{C}$.

Share: Given (pp, n, t) and a secret $s \in \mathbb{Z}_q$, the dealer D proceeds as follows:

1. Sample $f(X) \leftarrow \mathbb{Z}_q[X]_t$ uniformly, subject to $f(0) = s$. Set $f_j \coloneqq f(j)$ for $j \in [n]$ and $y_0 \coloneqq g^s$.
2. Run the prover of Fig. 3 on f and $(g, y_0, f_1, \ldots, f_n)$. Send f_j securely to P_j together with the proof's opening data ω_j, and broadcast the resulting proof π^{sh}.

Verify: Given pp, the public proof π^{sh}, and its private pair (f_j, ω_j), each party P_j proceeds as follows:

1. Apply verifier V_j's check in Fig. 3 with opening data ω_j. Broadcast a complaint against D if the check fails or the private pair is missing or malformed.
2. Reject the sharing if more than t parties complain. Otherwise, D broadcasts (f_j, ω_j) for each complaining party P_j.
3. Reject if any required response is missing or fails the corresponding verifier's check; otherwise accept, using the corrected shares where necessary. All honest parties make the same decision from the public complaint transcript.

Reconstruct:

1. (Optimistic) Given a claimed opening $s \in \mathbb{Z}_q$, check $y_0 = g^s$. Return s if the check succeeds, and false otherwise.
2. (Pessimistic) Each participating party P_j broadcasts (f_j, ω_j). Confirm an opening if it passes verifier V_j's check in Fig. 3, using the broadcast values as its private input. Interpolate $f(X)$ from any $t+1$ confirmed shares and return $f(0)$. Return false if too few valid shares are available or $g^{f(0)} \neq y_0$.

Fig. 4. PC-Π: a DL-oriented PC-VSS variant of the Π framework. The commitment $y_0 = g^{f(0)}$ and the sharing proof bind optimistic and pessimistic reconstruction to the same secret.

information. It is perfect for the Pedersen commitments and computational for the hash commitments, by Section 2.

In the non-interactive setting, the simulator samples d uniformly and programs $H_{\mathsf{FS}}(g, y_0, \mathsf{C}'_0, \ldots, \mathsf{C}'_n) = d$. It aborts if this input was already queried with a different answer. The fresh uniformly distributed C'_0 makes this event negligible for a polynomial number of prior queries. Together with the AFK soundness argument above, this proves the claimed non-interactive threshold ZK security in the random oracle model. □

3.3 Construction of PC-Π

The protocol PC-Π employs the DL-oriented NI-TZK proof from Fig. 3 to link the public commitment $y_0 = g^{f(0)}$ to the distributed shares $f_j = f(j)$. As in the original Π framework [Bag25], the construction uses a commitment scheme $\mathcal{C}$ for the shares and a random oracle H_{FS}. The commitment y_0 additionally permits optimistic reconstruction. The construction is given in Fig. 4.

Theorem 3 (A Framework for DL-Oriented PC-VSS). *Under the assumptions of Theorem 2, for either of the commitment instantiations considered there, the construction in Fig. 4 satisfies correctness, verifiability, unpre-*

dictability, and simulatability as defined in Section 3.1, against a computationally bounded adversary corrupting at most t shareholders, where $n \geq 2t + 1$.

Proof. The argument extends that for Π [Bag25, Theorem 3.1] using the DL-oriented proof of Theorem 2.

- *Correctness* follows from completeness of the proof, polynomial interpolation, and the identity $y_0 = g^{f(0)}$.
- *Verifiability* follows from soundness: an accepted sharing fixes a polynomial of degree at most t consistent with the honest shares and with $y_0 = g^{f(0)}$. The commitment checks reject inconsistent reconstruction shares, except with negligible probability. Every set of at least $t+1$ honest shareholders therefore reconstructs the same $f(0)$, and the optimistic check accepts exactly that value in $\mathbb{Z}_q$.
- *Simulatability* follows from threshold ZK, given y_0 and the corrupted shareholders' shares. An honest dealer answers complaints with the complainant's share and opening data, both available to the simulator. Both commitment instantiations use the simulator described in Theorem 2.
- *Unpredictability* for uniformly sampled $f(0)$ follows from simulatability and DL hardness. Up to t shares at distinct nonzero points are uniform and independent of $f(0)$; a predictor given the simulated view would therefore recover the discrete logarithm of y_0. □

3.4 Instantiations and Random Shamir Sharings

We instantiate PC-Π with the randomized Pedersen and hash commitments from Section 2, naming the schemes PC-Π_{P} and PC-Π_{LA}, respectively. These are pre-constructed variants of Π_{P} and Π_{LA} [Bag25]. The general threshold-ZK guarantee uses independent opening randomness for each recipient. The subscripts identify the share commitments; the secret commitment is $y_0 = g^{f(0)}$ throughout this construction. We also consider the two-generator specialization $\mathcal{C}(a, b) = g_1^a g_2^b$, denoted by PC-$\Pi_{\mathsf{F}}$, which is the pre-constructed variant of Π_{F}. The following lemma gives standalone simulation guarantees for the forms without additional randomizers: it simulates the sharing alone, rather than jointly with the honest outputs of the DKG functionality defined in Section 4.1.

Lemma 2 (Simulation for random Shamir sharings). *Fix any s, and sample f uniformly from $\mathbb{Z}_q[X]_t$ subject to $f(0) = s$. Let $\mathcal{T} \subseteq [n]$ with $|\mathcal{T}| \leq t$. Given $y_0 = g^s$ and the evaluations at $\mathcal{T}$, the sharing proof remains simulatable when the additional randomizers are omitted. This holds for $\mathcal{C}(a, b) = g_1^a g_2^b$, with $g_1 = g$ and g_2 generated by the simulator, and for $\mathcal{C}(a, b) = H(a, b)$ in the programmable ROM. Completeness and soundness continue to hold for arbitrary dealer inputs.*

Proof. Given y_0 and $\{f(j)\}_{j \in \mathcal{T}}$, sample independent uniform evaluations at $t - |\mathcal{T}|$ fixed distinct indices outside $\mathcal{T}$. Together with y_0, these t scalar evaluations determine by interpolation in the exponent all $e_j = g^{f(j)}$. They have exactly

the distribution induced by an honest sharing, conditional on the given inputs. Sample uniform $d \in \mathbb{Z}_q$ and $z \in \mathbb{Z}_q[X]_t$, and set $\mathsf{C}_0 = g^{z(0)} y_0^d$.

For Pedersen commitments, the setup simulator samples $g_2 = g_1^\alpha$ with uniform $\alpha \in \mathbb{Z}_q^*$, retaining α. Set $\mathsf{C}_j = e_j^{1+\alpha d} g_1^{\alpha z(j)}$. These are the genuine commitments for the sampled sharing and the masking polynomial $r = z + df$, without requiring the unknown scalar evaluations.

For hash commitments, evaluate $H(f(j), z(j) + df(j))$ normally at corrupted indices and sample independent uniform elements $\mathsf{C}_j \in \mathbb{Z}_q$ at honest indices. In the domain for each honest j, recognize the corresponding oracle input (a, b) by the tests $g^a = e_j$ and $b = z(j) + da$, and answer it with C_j. All other fresh queries receive independent uniform answers. Abort if a corresponding input was already queried; the fresh uniform $z(j)$ bounds this probability by $O(nQ/q)$ for Q prior queries. The simulator answers subsequent queries in the same way, including after the transcript is delivered.

Finally, program the Fiat–Shamir challenge as in Theorem 2. The fresh uniform C_0 makes a conflict negligible. This gives the required distribution of the sharing view, up to the stated negligible programming errors. Honest dealers answer only corrupted recipients' complaints, using their known evaluations. The completeness and soundness arguments are unchanged, since both commitments remain binding. □

The lemma concerns a random sharing of any fixed secret, with simulation given y_0 and the corrupted evaluations. The threshold-ZK claim for every fixed polynomial in Theorem 2 uses independent randomizers. For random sharings, the lemma instead uses a setup trapdoor or a programmable commitment oracle. This allows the randomizers used for simulatability in Π [Bag25] to be omitted for the guarantee above.

The standalone simulator does not automatically simulate jointly with the functionality's honest outputs: its sampled missing evaluations need not agree with the polynomial chosen by the functionality. In our DKGs the secret is also uniform, and we use the deterministic hash commitment $H(a, b)$. Its binding suffices for the same soundness argument; simulation with the required joint distribution is proved directly in Theorems 4 to 6.

4 The GJKR Fully Secure DKG Protocol Revisited

4.1 DKG Model and Security Definitions

The following model, security definitions, and conventions apply to all three DKG constructions.

A Distributed Key Generation (DKG) protocol for DL lets n parties generate a public key $y = g^x$ and private Shamir shares $x_j = F(j)$, where $F \in \mathbb{Z}_q[X]_t$ and $F(0) = x$. We assume $q > n$ and $t < n/2$. Parties communicate synchronously over authenticated private channels and a broadcast channel. A static PPT adversary corrupts a set $\mathcal{T} \subseteq [n]$ of at most t parties and may rush: within each

Ideal functionality $\mathcal{F}_{\mathsf{DKG}}^{t,n}$

The public parameters specify $(\mathbb{G}, q, g, n, t)$ and the construction's auxiliary generators and random-oracle domains. Auxiliary generators are sampled independently and uniformly from $\mathbb{G} \setminus \{1\}$, and the adversary is not given their discrete logarithm relations. Let $\mathcal{T}$ be the static corrupted set.

1. Receive $x_j \in \mathbb{Z}_q$ from the ideal adversary for each $j \in \mathcal{T}$. Set any missing or invalid input to 0 at the prescribed deadline.
2. Sample $x \leftarrow \mathbb{Z}_q$ uniformly. Sample $F \in \mathbb{Z}_q[X]_t$ uniformly subject to
$$F(0) = x \quad \text{and} \quad F(j) = x_j \quad (j \in \mathcal{T}).$$
3. Set $y \coloneqq g^x$. Send $(y, F(j))$ privately to each party P_j, and send y to the ideal adversary, including when $\mathcal{T} = \varnothing$.

The ideal adversary receives every message addressed to a corrupted party and cannot prevent delivery to honest parties.

Fig. 5. Robust DKG with a DL public key and private scalar shares.

round it sees honest messages addressed to corrupted parties and all honest broadcasts before sending its messages.

We define the ideal functionality in Figure 5 for our DKG protocols, which generates a public key for the DL setting and distributes a share of the corresponding secret key to each party. The functionality captures the security properties achieved by the alternative DKG protocols of Pedersen [Ped91] and GJKR [GJKR99,GJKR07]. The corrupted shares are chosen before y is revealed; the remaining polynomial is uniform subject to those choices. The functionality guarantees a uniformly distributed ideal secret and guaranteed output delivery.

Definition 8 (Fully secure DKG). *Let $\mathsf{REAL}_{\Pi,\mathcal{A}}(1^\lambda)$ be the joint distribution of $\mathcal{T}$, the honest parties' outputs, and the adversary's view in an execution of Π. The view includes the public transcript, the corrupted parties' private incoming messages and opening data, and the adversary's random coins and oracle query history. Let $\mathsf{IDEAL}_{\mathcal{F}_{\mathsf{DKG}}^{t,n},\mathcal{S}}(1^\lambda)$ contain $\mathcal{T}$, the honest parties' outputs from $\mathcal{F}_{\mathsf{DKG}}^{t,n}$, and the output of an ideal adversary $\mathcal{S}$. Protocol Π securely realizes $\mathcal{F}_{\mathsf{DKG}}^{t,n}$ if, for every static PPT rushing adversary $\mathcal{A}$ corrupting at most t parties, there is a PPT simulator $\mathcal{S}$ such that*

$$\mathsf{REAL}_{\Pi,\mathcal{A}}(1^\lambda) \approx_c \mathsf{IDEAL}_{\mathcal{F}_{\mathsf{DKG}}^{t,n},\mathcal{S}}(1^\lambda).$$

The comparison includes the specified public setup and oracle access. If simulation uses a setup trapdoor, the simulator must generate public parameters with the prescribed distribution. In the random oracle model, the simulator may program the challenge oracle H_{FS}. Let $\mathcal{D}$ denote a PPT distinguisher for this comparison.

The honest output vector is used to compare the two experiments; it is not delivered to the adversary. This comparison ensures that the simulated view is consistent with the actual honest outputs. All oracles remain available after the outputs are delivered. In the simulation proofs, Q bounds all oracle queries in the experiment, including those of $\mathcal{D}$.

Under DL hardness, computational realization also implies that no PPT adversary can recover x from its view, including y, except with negligible probability. This consequence applies to all three constructions.

Statistical unbiasedness. Computational realization of $\mathcal{F}_{\mathsf{DKG}}^{t,n}$ implies computational indistinguishability of the real key from uniform. To retain GJKR's stronger distributional guarantee [GJKR07, Section 4.1], we additionally call a protocol *statistically unbiased* if, for every adversary allowed above, its real secret key x satisfies

$$\Delta\big(x, U_{\mathbb{Z}_q}\big) \leq \mathrm{negl}(\lambda),$$

where Δ denotes statistical distance and x is the unique secret determined by the honest output shares, with an arbitrary default on a correctness failure. Since $x \mapsto g^x$ is a bijection, the same bound holds for g^x and the uniform distribution on $\mathbb{G}$; correctness extends it to the actual public key up to a negligible error. Thus, even an unbounded distinguisher given the public key has negligible advantage over uniform.

The active adversary remains PPT, as required for binding and soundness, and simulation of the full view is computational. Statistical real/ideal indistinguishability would imply unbiasedness directly, but computational realization alone does not. We prove both computational realization and statistical unbiasedness for DKG_1 within the same security proof. For DKG_2 and DKG_3, we prove computational unbiasedness.

In our VSS-based constructions, dealer P_i uses $f^{(i)}(X)$, with $f_{ij} = f^{(i)}(j)$ and contribution $s_i = f^{(i)}(0)$. For a qualified corrupted dealer, the accepted sharing determines this polynomial except with negligible probability. The initial qualified set $\mathcal{Q}_0$ fixes $F(X) = \sum_{i\in\mathcal{Q}_0} f^{(i)}(X)$, with $x = \sum_{i\in\mathcal{Q}_0} s_i$ and $x_j = \sum_{i\in\mathcal{Q}_0} f_{ij}$. The final qualified set $\mathcal{Q}_1 \subseteq \mathcal{Q}_0$ contains the dealers whose public-key proofs are accepted; the contributions of the other dealers in $\mathcal{Q}_0$ are recovered and remain in F. We write π_i^{sh} for an initial VSS proof and π_i^{pk} for a public-key proof. In DKG_1 and DKG_2, the latter is a PC-VSS proof; in DKG_3, it is the equality argument.

Common conventions. We use the random-oracle conventions of Section 2, with $1 \leq t < n/2$ and $g = g_1$. Every sharing includes the dealer's own coordinate, which it retains with its opening data; thus every qualified dealer has at least $n - |\mathcal{T}| \geq t + 1$ honest recipients for reconstruction. For the Π_{LA} and $\mathsf{PC}\text{-}\Pi_{\mathsf{LA}}$ share commitments in our DKG schemes, we use the deterministic commitment $H(a, b)$. Round labels follow the worst-case schedule. In DKG_1 and DKG_2, Round 3 is skipped when Round 2 contains no complaints. Silent verification is

not counted as a round when no complaint is sent. Each sender has one complaint entry per dealer in each complaint round, indexed by its authenticated identity; malformed entries are ignored.

4.2 Construction of DKG_1

In this subsection, we introduce our first distributed key generation protocol, denoted by DKG_1, which securely realizes $\mathcal{F}_{\mathsf{DKG}}^{t,n}$ and has a statistically uniform public key against up to t malicious parties. The design of DKG_1 is inspired by the well-known GJKR DKG protocol [GJKR99, GJKR07], but requires asymptotically fewer exponentiations per party in the happy path. In the GJKR protocol, the first phase relies on Pedersen's perfectly simulatable VSS scheme [Ped92] to distribute a secret, while in the second phase the qualified parties employ Feldman VSS [Fel87] together with their initial shares to derive the final public key. Following this blueprint, in the initial phase of DKG_1, we employ the perfectly simulatable $\Pi_{\mathsf{P}+}$ VSS scheme [ABN+25] (reviewed in Section 2.2). In the second phase, the qualified parties use the sharing proof of $\mathsf{PC}\text{-}\Pi_{\mathsf{LA}}$ with deterministic hash commitments (from Section 3.4), along with their initial shares, to efficiently extract the final public key. At the end of DKG_1, each party P_j outputs the common public key y and its private share x_j of the secret key.

Technical overview. In the first phase of DKG_1, each party P_i secret-shares a value $s_i = f^{(i)}(0)$ using the $\Pi_{\mathsf{P}+}$ VSS scheme (reviewed in Section 2.2). Thanks to the perfect simulatability of $\Pi_{\mathsf{P}+}$, an adversary cannot make the secret-shared value of a corrupted party depend on the values shared by any honest parties. At the end of this stage, each honest party P_j holds the share $f^{(i)}(j)$ of every dealer $P_i \in \mathcal{Q}_0$. Thus the secret key $x = \sum_{i \in \mathcal{Q}_0} s_i$ is fixed before public key computation begins.

Once x is fixed, the qualified parties proceed to compute g_1^x. Each party $P_i \in \mathcal{Q}_0$ publishes $y_i = g_1^{s_i}$ and, using our pre-constructed VSS scheme $\mathsf{PC}\text{-}\Pi_{\mathsf{LA}}$, generates a proof π_i^{pk} from its original sharing polynomial and a freshly sampled masking polynomial. Verification uses the shares already distributed in the VSS stage, ensuring that y_i corresponds to the contribution fixed in that stage. A complaining party broadcasts its initial share and opening, so everyone can check the complaint without another response from the dealer.

If a party P_i acts maliciously—for instance, by refusing to publish π_i^{pk} or publishing an invalid one—the honest parties can fall back on the reconstruction procedure of $\Pi_{\mathsf{P}+}$ to recover $s_i = f^{(i)}(0)$. This ensures that P_i's contribution is still included in the final key x, since excluding it could allow the adversary to bias its distribution.

Theorem 4. *Assume DL hardness for the prescribed random generators and the random oracle model of Section 2. For $1 \le t < n/2$, DKG_1 computationally realizes $\mathcal{F}_{\mathsf{DKG}}^{t,n}$ against static PPT malicious rushing adversaries. Its initial VSS phase has information-theoretic privacy. Its real secret and public keys are statistically indistinguishable from uniform, even to a computationally unbounded distinguisher given the key alone.*

DKG_1

VSS Stage:

1. Given group generators (g_1, g_2) and parameters (n, t), each party P_i performs a $\Pi_{\mathsf{P}+}$ VSS as in Section 2.2:
   (a) **Round 1:** Sample $s_i \leftarrow \mathbb{Z}_q$ and run Share, writing $f^{(i)}$ for the sharing polynomial, (f_{ij}, γ_{ij}) for the private openings, and $\pi_i^{\mathsf{sh}} = (c_1^{(i)}, \ldots, c_n^{(i)}, z_i)$ for the public proof.
   (b) **Rounds 2–3:** Run Verify, broadcasting complaints in Round 2 and the dealer's responses in Round 3. Its opening check, with $d_i := H_{\mathsf{FS}}(c_1^{(i)}, \ldots, c_n^{(i)})$, is
   $$c_j^{(i)} = g_1^{H(f_{ij}, z_i(j) + d_i \cdot f_{ij})} g_2^{\gamma_{ij}}. \tag{3}$$
2. Each party builds the set of qualified dealers $\mathcal{Q}_0$ and uses any validated complaint response to correct its corresponding share.
3. The distributed secret value x is not explicitly computed by any party, but it equals $x = \sum_{i \in \mathcal{Q}_0} s_i$. Each party P_j sets its share of the secret as $x_j := \sum_{i \in \mathcal{Q}_0} f_{ij}$.

Public Key Computation (Extracting $y = g_1^x$):

4. Each party P_i, $i \in \mathcal{Q}_0$, reuses $f^{(i)}(X)$ and the shares and openings from the VSS stage.
   (a) **Round 4:** Set $y_i := g_1^{s_i}$ and run the prover of Fig. 3 on the same $f^{(i)}$, using $g = g_1$, $\mathcal{C}(a, b) = H(a, b)$, and a fresh masking polynomial $\rho^{(i)}$, writing $\rho^{(i)}, \delta_i, \zeta_i$ for the prover's r, d, z. Publish $\pi_i^{\mathsf{pk}} := (g_1, y_i, \mathsf{C}_0^{(i)}, \ldots, \mathsf{C}_n^{(i)}, \zeta_i)$.
   (b) **Round 5:** Each P_j marks a dealer P_i, $i \in \mathcal{Q}_0$, for reconstruction if its proof is missing, malformed, or has $\deg(\zeta_i(X)) > t$. For every remaining proof, P_j sets $\delta_i := H_{\mathsf{FS}}(g_1, y_i, \mathsf{C}_0^{(i)}, \mathsf{C}_1^{(i)}, \ldots, \mathsf{C}_n^{(i)})$ and checks if
   $$\mathsf{C}_j^{(i)} = H(f_{ij}, \zeta_i(j) + \delta_i \cdot f_{ij}) \ \wedge \ \mathsf{C}_0^{(i)} = g_1^{\zeta_i(0)} y_i^{\delta_i}. \tag{4}$$
   A failure of the public $\mathsf{C}_0^{(i)}$ check marks P_i for reconstruction. If the share check fails, P_j broadcasts a complaint (i, f_{ij}, γ_{ij}) against P_i.
   (c) **Local computation:** For a complaint (i, v, γ) from P_j, mark P_i if the supplied (v, γ) passes Eq. (3) but v fails the share check in Eq. (4). Ignore all other complaints. Let $\mathcal{Q}_1 \subseteq \mathcal{Q}_0$ consist of the unmarked dealers.
   (d) **Round 6 (if needed):** For every $i \in \mathcal{Q}_0 \setminus \mathcal{Q}_1$, each party P_j broadcasts its retained (f_{ij}, γ_{ij}). Confirm openings satisfying Eq. (3) and recover s_i from any $t + 1$ confirmed openings. Set $y_i := g_1^{s_i}$.
   (e) Retain the accepted public contribution for each $i \in \mathcal{Q}_1$ and compute $y := \prod_{i \in \mathcal{Q}_0} y_i$. Each party P_j outputs (y, x_j).

Fig. 6. DKG_1: a fully secure distributed key generation protocol with statistically uniform DL public key in the honest majority setting.

Proof strategy. The initial VSS fixes a uniform secret, and reconstruction preserves every qualified contribution. For simulation, we replace honest dealers' second-phase commitments to honest recipients by random field elements. We show that a query to a replaced input would yield a DL solver. The reduction puts the unknown scalar into two honest polynomials with opposite signs, so it can still compute their sum and all honest outputs. Finally, the privacy of the initial sharing lets us replace the aggregate polynomial by the functionality's choice.

The full proof is in Appendix C.4.

Efficiency of DKG_1. Compared with the fully secure variant of Pedersen DKG presented in [GJKR99, GJKR07], DKG_1 requires asymptotically fewer exponen-

tiations per party in the happy path, where all parties follow the protocol. In this case, each party performs approximately $6n$ exponentiations within the group, $5n$ evaluations of degree-t polynomials, and $6n$ hash evaluations, including the commitment and Fiat–Shamir hashes. In the GJKR DKG, each party performs approximately $(2tn+n)$ exponentiations and $2n$ degree-t polynomial evaluations. The linear bound refers to exponentiations; polynomial evaluation and hashing costs are counted separately. Complaint handling and reconstruction add to these costs. For example, in the initial phase, t corrupted recipients can each complain against every honest dealer, requiring each honest party to verify $t(n-t)$ public responses. Thus adversarial executions can require $\Theta(nt)$ verification equations per party. In the happy path, each party privately sends approximately $2n$ field elements, all during the initial sharing. It broadcasts approximately n hash outputs, n group elements, and $2t$ field elements for the polynomial coefficients. In comparison, each party in GJKR privately sends $2n$ field elements and broadcasts approximately $2t$ group elements. Complaint responses and reconstruction openings require additional communication. The protocol uses two rounds in the happy path and at most six in the worst case: three for the initial VSS, one for the public-key proofs, one for complaints, and one for reconstruction if needed. Table 1 compares the asymptotic costs of the two fully secure protocols.

5 Fully Secure Computationally Uniform DKG

Next, we introduce another fully secure and robust DKG protocol for sampling a DL instance, $y = g_1^x$, which we call DKG_2. Compared to DKG_1 (from Section 4.2), DKG_2 requires fewer exponentiations in the happy path. Its public key, however, is only *computationally indistinguishable* from uniform in the presence of up to t malicious parties. The design of DKG_2 follows the blueprint of DKG_1:

1. In the first phase, we replace the perfectly simulatable VSS scheme $\Pi_{\mathsf{P}+}$ with the hash-based VSS scheme Π_{LA} [Bag25] (reviewed in Section 2.2), omitting its additional randomizers for random contributions.
2. In the second phase, we use the sharing proof of $\mathsf{PC}\text{-}\Pi_{\mathsf{LA}}$ PC-VSS scheme with deterministic commitments from Section 3.4.

Privacy during the initial sharing is therefore computational, whereas in DKG_1 it is information-theoretic. As in DKG_1, at the end of the VSS stage each honest party P_j holds $f^{(i)}(j)$ for every $P_i \in \mathcal{Q}_0$. Verifiability ensures that each qualified dealer has shared a unique recoverable polynomial $f^{(i)}(X)$. Hence the secret $x = \sum_{i\in\mathcal{Q}_0} s_i$ is fixed before public key computation begins.

The parties then compute $y = g_1^x$ as in DKG_1 (see Fig. 6). Each $P_i \in \mathcal{Q}_0$ publishes $y_i = g_1^{s_i}$ and a $\mathsf{PC}\text{-}\Pi_{\mathsf{LA}}$ proof π_i^{pk} using its original polynomial and a fresh masking polynomial. The other parties verify the proof with their shares from the VSS stage. This enforces consistency across both the VSS and public key computation phases, preventing adversaries from altering their contributions midway. A complaining party broadcasts its initial share, so everyone can check the complaint without another response from the dealer. Binding prevents a

VSS Stage:

1. Given group generator g_1 and parameters (n, t), each party P_i performs Π_{LA} VSS as in Section 2.2, omitting the hash randomizers:
   (a) **Round 1:** Sample $s_i \leftarrow \mathbb{Z}_q$ and run Share, writing $f^{(i)}$ for the sharing polynomial, f_{ij} for the private shares, and $\pi_i^{\mathsf{sh}} = (c_1^{(i)}, \ldots, c_n^{(i)}, z_i)$ for the public proof.
   (b) **Rounds 2–3:** Run Verify, broadcasting complaints in Round 2 and the dealer's responses in Round 3. Its opening check, with $d_i := H_{\mathsf{FS}}(c_1^{(i)}, \ldots, c_n^{(i)})$, is

$$c_j^{(i)} = H(f_{ij}, z_i(j) + d_i \cdot f_{ij}) \tag{5}$$

2. Each party builds the set of qualified dealers $\mathcal{Q}_0$ and uses any validated complaint response to correct its corresponding share.
3. The distributed secret value x is not explicitly computed by any party, but it equals $x = \sum_{i \in \mathcal{Q}_0} s_i$. Each party P_j sets its share of the secret as $x_j := \sum_{i \in \mathcal{Q}_0} f_{ij}$.

Public Key Computation (Extracting $y = g_1^x$):

4. Each $P_i \in \mathcal{Q}_0$ reuses $f^{(i)}$ and the scalar shares from the VSS stage. Run the public key computation phase of Fig. 6 (Rounds 4–6), with the following changes: an honest P_j sends (i, f_{ij}); a received complaint (i, v) from P_j is valid if v passes Eq. (5) but fails the share check in Eq. (4); reconstruction uses the retained scalar openings checked against Eq. (5). Each P_j outputs (y, x_j).

Fig. 7. DKG_2: a hash-based, fully secure distributed key generation protocol for DL.

false complaint from forcing reconstruction of an honest contribution. Finally, if a party P_i misbehaves during this stage—for instance, by refusing to publish π_i^{pk} or publishing an invalid one—the honest parties can invoke the reconstruction procedure of Π_{LA} to recover $s_i = f^{(i)}(0)$. This ensures that P_i's contribution is still included in the final key x, since excluding it could allow an adversary to bias the resulting distribution.

The protocol is given in Fig. 7, and its security is established in the following theorem.

Theorem 5. *Assume DL hardness and the random oracle model of Section 2. Then* DKG_2 *computationally realizes* $\mathcal{F}_{\mathsf{DKG}}^{t,n}$ *against static PPT malicious rushing adversaries corrupting at most* t *parties, for* $1 \le t < n/2$*. Its secret and public keys are computationally indistinguishable from uniform.*

Proof strategy. We replace honest dealers' commitments at honest recipients in both phases, extending the DL reduction used for DKG_1 to the initial sharing. The reduction bounds the probability that an oracle query reaches a replaced input, allowing simulation consistent with the honest outputs. The full proof is in Appendix C.5.

Efficiency of DKG_2. As in DKG_1, the main improvement over the fully secure GJKR protocol [GJKR99, GJKR07] is the number of exponentiations per party. In the happy path, each party performs approximately $2n$ exponentiations, $5n$ evaluations of degree-t polynomials, and $6n$ hash evaluations, including challenge queries. Both the initial VSS sharing and its reconstruction use only hashing and field operations. Each party privately sends approximately n field elements, and broadcasts approximately $2n$ hash outputs, $2t$ field elements, and a constant

number of group elements. Complaint and reconstruction costs are additional, as described for DKG_1. The protocol uses two rounds in the happy path and at most six in the worst case: three for initial sharing, one for the public-key proofs, one for complaints, and one for reconstruction if needed. When reconstruction is needed, the recovered constants are summed before computing their joint contribution with a single exponentiation.

Our DKG_2 as a Fully Secure Alternative to the ABCP DKG. In [ABCP23], Atapoor, Baghery, Cozzo, and Pedersen introduced the first practical hash-based VSS protocol in the synchronous setting, which they then used to build an efficient DKG protocol for the DL setting with $O(n)$ exponentiations per party in the happy path. Our DKG_2 can be viewed as a fully secure alternative: as we now explain, the ABCP DKG protocol is not fully secure, since an adversary can bias the distribution of the final public key.

The DKG protocol of [ABCP23], presented in their Fig. 4, proceeds in two phases: the *VSS and Committing* phase and the *Opening, Verification, and Public Key Computation* phase. In the first phase, each party samples a random secret, shares it using their VSS scheme, and commits to its partial public key. In the second phase, each party opens its commitment and privately sends the corresponding shares to the others. Upon receiving the openings and proofs, parties verify the validity of the received shares. If a party refuses to open its commitment at the start of the second phase, it is disqualified from the protocol.

This design, however, creates an opportunity for a rushing adversary. By waiting to see the openings of honest parties before deciding whether to open its own commitment, the adversary can selectively abort and thereby bias the distribution of the final public key. The GJKR countermeasure [GJKR07], which we also adopt in our proposed protocols, is to fix every contribution in recoverable form before any partial public key is revealed, so that a later abort cannot remove it. If a dealer later withholds a valid public-key proof, the honest parties reconstruct its contribution and include it in the final key.

Table 1 summarizes the asymptotic costs of our fully secure hash-based protocol DKG_2 compared to the biased ABCP DKG protocol [ABCP23].

6 DKG_3: A Round-Reduced Variant of DKG_2

In this section, we present DKG_3, a variant of DKG_2 (from Section 5) that reduces the worst-case number of rounds. The protocol uses three rounds in the happy path and a fourth round if a qualified dealer fails to provide a valid public-key proof. As in DKG_2, the public key is computationally indistinguishable from uniform in the presence of up to t malicious parties. Here we adapt a two-round variant of $\mathsf{PC}\text{-}\Pi_{\mathsf{LA}}$ with deterministic commitments from Section 3.4. Sharing uses the Pedersen secret commitment $A_i = g_1^{s_i} g_2^{\gamma_i}$ in place of g^{s_i}. Only after qualification is $y_i = g_1^{s_i}$ revealed with an equality proof.

Technical overview. To construct DKG_3, we combine a two-round variant of $\mathsf{PC}\text{-}\Pi_{\mathsf{LA}}$ in the VSS stage with a NIZK argument for the relation in Eq. (1) dur-

ing the public-key extraction stage. In two-round $\mathsf{PC}\text{-}\Pi_{\mathsf{LA}}$, we resolve complaints using the two-round mechanism of 2R-Π [BBKR25], recalled in Section 2.2. Recipients commit to one-time secret keys and send their secret keys privately to the dealer in the first round; the dealer broadcasts masked shares using the same secret keys in the second round. A recipient with invalid private data can disclose its secret key, allowing everyone, including all parties, to check the dealer's answer without another round.

The adapted sharing proof links A_i to the constant term of the shared polynomial. The Pedersen commitment hides that contribution while qualification is completed; its binding ensures that the sharing proof and the later public-key proof refer to the same contribution. The equations are given in Fig. 8.

The public key computation phase follows the approach of Canetti et al. [CGJ+99], also used in a later formulation by Abe and Fehr [AF04]: split the secret commitment, prove knowledge of the two discrete logarithms, and reconstruct failed contributions. With $g_3 = g_1$, our equality relation expresses this split as $y_i = g_1^{s_i}$ and $A_i/y_i = g_2^{\gamma_i}$. Their sharing proofs and security models differ; in particular, Canetti et al. use a jointly generated interactive challenge and erasures for adaptive security. Our construction combines this phase with the hash-based sharing proof and two-round complaint resolution above. After two rounds, the set $\mathcal{Q}_0$ and its contributions are fixed. In the third round, each $P_i \in \mathcal{Q}_0$ publishes $y_i = g_1^{s_i}$ and uses the NIZK argument to prove that its exponent is the value committed in A_i. If a dealer fails to provide a valid proof, the parties reconstruct its original contribution and include it in the final product.

Theorem 6. *Assume DL hardness for the prescribed random generators and the random oracle model of Section 2. Then* DKG_3 *computationally realizes* $\mathcal{F}_{\mathsf{DKG}}^{t,n}$ *against static PPT malicious rushing adversaries corrupting at most* t *parties, for* $1 \le t < n/2$. *Its secret and public keys are computationally indistinguishable from uniform.*

Proof strategy. Simulation must account for both the sharing commitments and the commitments to one-time pads. We extend DKG_1's DL reduction so that a query to either kind of replaced input yields a candidate hidden share, while the unknown polynomial terms cancel in the aggregate F, allowing the reduction to compute the honest outputs. The full proof is in Appendix C.6.

Efficiency of DKG_3. In the happy path, each party performs approximately $7n + O(1)$ exponentiations: three per dealer for the public sharing check and four per dealer for the equality-proof verification, where the common term $g_1^{\zeta_{1i}}$ is reused across the two verification equations. Generating its own commitments and proof adds only a constant number of exponentiations. Each party also performs approximately $3n$ degree-t polynomial evaluations and $6n$ hash evaluations, including challenge queries. Multi-exponentiation can further reduce the concrete group-operation cost. Each party privately sends approximately $2n$ field elements for shares and pads. It broadcasts $O(n)$ field elements and hash outputs, and a constant number of group elements in the happy path. Complaint and reconstruction costs are additional. As in DKG_2, recovered constants are summed

Parties have access to generators (g_1, g_2) and parameters (n, t) as in Section 4.1. Pads are uniform elements of $\mathbb{Z}_q$, and pad commitments use $H(k)$ in the separate domain specified in Section 2.

1. **Sharing and Qualification:** Party P_i samples $s_i \leftarrow \mathbb{Z}_q$ and performs the following two-round sharing with commitment A_i to s_i:

(a) **Round 1:**

i. Sample $f^{(i)}, r^{(i)} \leftarrow \mathbb{Z}_q[X]_t$, conditioned on $f^{(i)}(0) = s_i$, and independent $\gamma_i, \sigma_i \leftarrow \mathbb{Z}_q$. For $j \in [n]$, set

$$A_i \coloneqq g_1^{s_i} g_2^{\gamma_i}, \qquad B_i \coloneqq g_1^{r^{(i)}(0)} g_2^{\sigma_i}, \qquad c_j^{(i)} \coloneqq H(f_{ij}, r^{(i)}(j)),$$
$$d_i \coloneqq H_{\mathsf{FS}}(g_1, g_2, A_i, B_i, c_1^{(i)}, \ldots, c_n^{(i)}), \qquad z_i \coloneqq r^{(i)} - d_i f^{(i)}, \qquad w_i \coloneqq \sigma_i - d_i \gamma_i.$$

The sharing proof is $\pi_i^{\mathsf{sh}} \coloneqq (A_i, B_i, (c_j^{(i)})_{j\in[n]}, z_i, w_i)$.

ii. As a recipient, for each dealer $P_j \neq P_i$, sample $k_{ji} \leftarrow \mathbb{Z}_q$ and set $K_{ji} \coloneqq H(k_{ji})$.

iii. Send f_{ij} securely to each $P_j \neq P_i$, retaining f_{ii}. Send k_{ji} securely to each dealer $P_j \neq P_i$. Broadcast π_i^{sh} and $(K_{ji})_{j\neq i}$.

(b) **Round 2 (dealer):** For $j \in [n] \setminus \{i\}$, check the received pad opening against $K_{ij} = H(k_{ij})$. Broadcast $a_{ij} \coloneqq f_{ij} + k_{ij} \pmod q$ if the check succeeds, and broadcast f_{ij} otherwise. Each answer specifies whether it is masked or unmasked.

(c) **Round 2 (recipient):** For each dealer $P_i \neq P_j$, party P_j parses π_i^{sh}, checks $\deg(z_i) \leq t$, computes its challenge d_i, and verifies

$$c_j^{(i)} = H(f_{ij}, z_i(j) + d_i f_{ij}) \quad \wedge \quad B_i = g_1^{z_i(0)} g_2^{w_i} A_i^{d_i}. \tag{6}$$

A missing or malformed public proof, excessive degree, or failed public check disqualifies the dealer. Otherwise, P_j broadcasts nothing if the private check succeeds, and broadcasts k_{ij} if it fails or the private data are missing. Party P_j is happy with this sharing in the former case, $\mathsf{unhappy}$ in the latter.

2. **Local Computations:**

(a) Reject a dealer if any required answer in Round 2 is missing or malformed, any unmasked answer fails Eq. (6), or a complaint with a valid pad opening decrypts a masked answer to a share that fails that equation.

(b) Ignore complaints with invalid pad openings and complaints about unmasked answers.

(c) Determine the common set $\mathcal{Q}_0$.

(d) For each qualified dealer, use a valid unmasked answer when present, the original private data if happy, and otherwise the answer decrypted with the recipient's pad.

(e) The secret is $x = \sum_{i\in\mathcal{Q}_0} s_i$, and each party P_j sets $x_j \coloneqq \sum_{i\in\mathcal{Q}_0} f_{ij}$.

3. **Public Key Computation (Extracting $y = g_1^x$):**

(a) **Round 3:** Each $P_i \in \mathcal{Q}_0$ sets $y_i \coloneqq g_1^{s_i}$ and runs the NIZK scheme in Fig. 2 with $g_3 = g_1$, statement $(g_1, g_2, g_1, A_i, y_i)$ and witness (s_i, γ_i), writing δ_i for the challenge e. Publish y_i and $\pi_i^{\mathsf{pk}} \coloneqq (\delta_i, \zeta_{1i}, \zeta_{2i})$.

(b) For each $j \in \mathcal{Q}_0$, reject missing or malformed group elements or proofs; otherwise apply the verifier of Fig. 2 to $(g_1, g_2, g_1, A_j, y_j)$ and π_j^{pk}. Let $\mathcal{Q}_1 \subseteq \mathcal{Q}_0$ consist of the dealers with valid proofs.

(c) **Round 4 (if needed):** For each $i \in \mathcal{Q}_0 \setminus \mathcal{Q}_1$, every party P_j broadcasts its retained f_{ij}. Verify these openings using Eq. (6) and reconstruct s_i from any $t+1$ confirmed openings.

(d) Every party locally computes the final public key

$$y \coloneqq g_1^{\sum_{i\in\mathcal{Q}_0\setminus\mathcal{Q}_1} s_i} \prod_{i\in\mathcal{Q}_1} y_i.$$

If $\mathcal{Q}_1 = \mathcal{Q}_0$, omit Round 4 and use the product alone. Each party P_j outputs (y, x_j).

Fig. 8. DKG_3: an efficient, low-round fully secure DKG protocol for DL.

before exponentiating. The linear bound concerns group exponentiations; polynomial evaluations and hashing are counted separately. The parties know which dealers require reconstruction only after receiving the proofs in Round 3. Recon-

struction consequently requires a fourth round, since the masked answers from the sharing phase cannot serve as public reconstruction shares.

7 Conclusion

We revisited the fully secure and robust variant of the Pedersen DKG protocol proposed in [GJKR07], and presented several new variants that achieve improved efficiency while offering different trade-offs in terms of communication rounds, public-key distribution, and computational and communication costs.

As in the original GJKR protocol, our first fully secure and robust DKG construction guarantees that the final public key is statistically close to uniform, even when up to t parties behave maliciously.

We further show that the recent practical hash-based robust DKG protocol of ABCP [ABCP23] does not achieve full security, as an adversary can bias the distribution of the resulting public key.

Our DKG_3 requires three rounds in the happy path and one additional round when reconstruction is needed. Compared to recent alternatives [Kat24, CD24, KMM+24, BHLS25], it offers a different trade-off between round complexity and group operations. In particular, each party performs $O(n)$ exponentiations in the happy path, rather than the $O(n^2)$ exponentiations required by the most efficient existing schemes.

Future work could reduce the number of rounds in DKG_3, or shift some of its interaction into an offline or preprocessing phase, without sacrificing its asymptotic efficiency or strong security guarantees. We also believe that most of these techniques can be adapted to post-quantum secure DKG protocols.

Acknowledgments

This work was supported by the Flemish Government through the Cybersecurity Research Program with grant number VOEWICS02.

References

ABCP23. Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. VSS from distributed ZK proofs and applications. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part I*, volume 14438 of *Lecture Notes in Computer Science*, pages 405–440, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore.

ABN+25. Shahla Atapoor, Karim Baghery, Georgio Nicolas, Robi Pedersen, and Jannik Spiessens. Batched and packed (publicly) verifiable secret sharing: A unified framework and applications. Cryptology ePrint Archive, Report 2025/2018, 2025.

AF04. Masayuki Abe and Serge Fehr. Adaptively secure feldman VSS and applications to universally-composable threshold cryptography. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152

of *Lecture Notes in Computer Science*, pages 317–334, Santa Barbara, CA, USA, August 15–19, 2004. Springer Berlin Heidelberg, Germany.

AFK23. Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-Shamir transformation of multi-round interactive proofs (extended version). *Journal of Cryptology*, 36(4):36, October 2023.

Bag25. Karim Baghery. Π: A unified framework for computational verifiable secret sharing. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025: 28th International Conference on Theory and Practice of Public Key Cryptography, Part IV*, volume 15677 of *Lecture Notes in Computer Science*, pages 110–142, Røros, Norway, May 12–15, 2025. Springer, Cham, Switzerland.

BBC+19. Dan Boneh, Elette Boyle, Henry Corrigan-Gibbs, Niv Gilboa, and Yuval Ishai. Zero-knowledge proofs on secret-shared data via fully linear PCPs. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 67–97, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Cham, Switzerland.

BBKR25. Karim Baghery, Navid Ghaedi Bardeh, Shahram Khazaei, and Mahdi Rahimi. On round-optimal computational VSS. *IACR Communications in Cryptology (CiC)*, 2(2):13, 2025.

BHLS25. Dan Boneh, Iftach Haitner, Yehuda Lindell, and Gil Segev. Exponent-VRFs and their applications. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part VII*, volume 15607 of *Lecture Notes in Computer Science*, pages 195–224, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland.

BKNR25. Karim Baghery, Noah Knapen, Georgio Nicolas, and Mahdi Rahimi. Preconstructed publicly verifiable secret sharing and applications. In Marc Fischlin and Veelasha Moonsamy, editors, *ACNS 2025: 23rd International Conference on Applied Cryptography and Network Security, Part I*, volume 15825 of *Lecture Notes in Computer Science*, pages 89–119, Munich, Germany, June 23–26, 2025. Springer, Cham, Switzerland.

CCG24. Ignacio Cascudo, Daniele Cozzo, and Emanuele Giunta. Verifiable secret sharing from symmetric key cryptography with improved optimistic complexity. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology – ASIACRYPT 2024, Part VII*, volume 15490 of *Lecture Notes in Computer Science*, pages 100–128, Kolkata, India, December 9–13, 2024. Springer, Singapore, Singapore.

CD24. Ignacio Cascudo and Bernardo David. Publicly verifiable secret sharing over class groups and applications to DKG and YOSO. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part V*, volume 14655 of *Lecture Notes in Computer Science*, pages 216–248, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland.

CGJ+99. Ran Canetti, Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Adaptive security for threshold cryptosystems. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 98–115, Santa Barbara, CA, USA, August 15–19, 1999. Springer Berlin Heidelberg, Germany.

CGMA85. Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th Annual Symposium on Foundations of Computer*

Science, pages 383–395, Portland, Oregon, October 21–23, 1985. IEEE Computer Society Press.

CM22. Fabio Campos and Philipp Muth. On actively secure fine-grained access structures from isogeny assumptions. In Jung Hee Cheon and Thomas Johansson, editors, *Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022*, pages 375–398, Virtual Event, September 28–30, 2022. Springer, Cham, Switzerland.

Fel87. Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *28th Annual Symposium on Foundations of Computer Science*, pages 427–437, Los Angeles, CA, USA, October 12–14, 1987. IEEE Computer Society Press.

GJKR99. Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 295–310, Prague, Czech Republic, May 2–6, 1999. Springer Berlin Heidelberg, Germany.

GJKR07. Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology*, 20(1):51–83, January 2007.

Kat24. Jonathan Katz. Round-optimal, fully secure distributed key generation. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part VII*, volume 14926 of *Lecture Notes in Computer Science*, pages 285–316, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

KMM+24. Aniket Kate, Easwar Vivek Mangipudi, Pratyay Mukherjee, Hamza Saleem, and Sri Aravinda Krishnan Thyagarajan. Non-interactive VSS using class groups and application to DKG. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024: 31st Conference on Computer and Communications Security*, pages 4286–4300, Salt Lake City, UT, USA, October 14–18, 2024. ACM Press.

Mau09. Ueli M. Maurer. Unifying zero-knowledge proofs of knowledge. In Bart Preneel, editor, *AFRICACRYPT 09: 2nd International Conference on Cryptology in Africa*, volume 5580 of *Lecture Notes in Computer Science*, pages 272–286, Gammarth, Tunisia, June 21–25, 2009. Springer Berlin Heidelberg, Germany.

Ped91. Torben P. Pedersen. A threshold cryptosystem without a trusted party (extended abstract) (rump session). In Donald W. Davies, editor, *Advances in Cryptology – EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 522–526, Brighton, UK, April 8–11, 1991. Springer Berlin Heidelberg, Germany.

Ped92. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140, Santa Barbara, CA, USA, August 11–15, 1992. Springer Berlin Heidelberg, Germany.

Sch90. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252, Santa Barbara, CA, USA, August 20–24, 1990. Springer, New York, USA.

Sha79. Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.

Appendix:

A Feldman and Pedersen VSS Schemes

Feldman VSS Scheme. One of the primary computationally secure VSS schemes is Feldman's scheme [Fel87], which is based on Shamir secret sharing. In Feldman's scheme, given (n, t, s) and group generator g_1, to share a *high-entropy* secret s, the Share and Verify algorithms are as follows:

- $\mathsf{Share}(n, t, s, g_1) \to (\{f_j\}_{j=1}^n, \pi^{\mathsf{sh}})$: Sample a uniformly random degree-t polynomial $f(X) := s + a_1 X + \cdots + a_t X^t$ with coefficients in $\mathbb{Z}_q$, subject to $f(0) = s$. For $j = 1, 2, \cdots, n$: set $f_j := f(j)$. Compute $c_0 := g_1^s$ and $c_\ell := g_1^{a_\ell}$ for $\ell = 1, 2, \cdots, t$. Set $\pi^{\mathsf{sh}} := (c_0, c_1, \ldots, c_t)$; send the share f_j securely to party P_j and return π^{sh} as the NI-TZK proof.
- $\mathsf{Verify}(n, t, \{f_j\}_{j=1}^n, \pi^{\mathsf{sh}}) \to$ `true/false`: Given n, threshold value t, the shares $\{f_j\}_{j=1}^n$, and the threshold proof $\pi^{\mathsf{sh}} := (c_0, c_1, \ldots, c_t)$: each party P_j uses its share f_j and checks if
$$g_1^{f_j} = \prod_{\ell=0}^t c_\ell^{j^\ell}$$
and outputs either `true` or `false`. For $n \geq 2t+1$, if *all* n parties return `true`, then the final Verify will return `true`. Otherwise, the dealer answers complaints by broadcasting the disputed shares, as in the complaint procedure recalled in Section 2.2.

Pedersen VSS Scheme. The Pedersen VSS scheme is a variation of Feldman's scheme [Fel87] that uses a perfectly hiding commitment scheme. Pedersen VSS ensures that any set of at most t parties learns no information about the secret, thereby achieving information-theoretic (IT) simulatability. In Pedersen's scheme, given (n, t, s) and two random group generators $pp := (g_1, g_2)$, to share a secret s, the Share and Verify algorithms are as follows:

- $\mathsf{Share}(n, t, s, pp) \to (\{f_j\}_{j=1}^n, \pi^{\mathsf{sh}})$: Sample two random degree-t polynomials $f(X) := s + a_1 X + \cdots + a_t X^t$ and $r(X) := r_0 + b_1 X + \cdots + b_t X^t$ with coefficients in $\mathbb{Z}_q$, subject to $f(0) = s$. For $j = 1, 2, \cdots, n$: set $f_j := f(j)$ and $r_j := r(j)$. Compute $c_0 := g_1^s g_2^{r_0}$ and $c_\ell := g_1^{a_\ell} g_2^{b_\ell}$ for $\ell = 1, 2, \cdots, t$. Set $\pi^{\mathsf{sh}} := (c_0, c_1, \ldots, c_t)$; send the share (f_j, r_j) securely to party P_j and return π^{sh} as the proof.
- $\mathsf{Verify}(n, t, \{f_j, r_j\}_{j=1}^n, \pi^{\mathsf{sh}}) \to$ `true/false`: Given n, threshold value t, the shares $\{f_j, r_j\}_{j=1}^n$, and the threshold proof $\pi^{\mathsf{sh}} := (c_0, c_1, \ldots, c_t)$: each party P_j uses its share (f_j, r_j) and checks if
$$g_1^{f_j} g_2^{r_j} = \prod_{\ell=0}^t c_\ell^{j^\ell}$$
and outputs either `true` or `false`. The rest of the verification is similar to the Feldman scheme.

B Additional VSS Constructions

B.1 The Π_{P} VSS Scheme

In [Bag25], Baghery presented Π_{P} as an efficient alternative to the Pedersen VSS scheme that can achieve perfect simulatability. In Π_{P}, given (n, t, s) and random group generators $pp := (g_1, g_2, g_3)$, to share a secret s, the Share, Verify and Reconstruct algorithms are as follows:

- $\mathsf{Share}(n, t, s, pp) \rightarrow (\{f_j\}_{j=1}^{n}, \pi^{\mathsf{sh}})$: Given $pp := (g_1, g_2, g_3)$, the parameters (n, t), random oracle H_{FS}, to share s, the dealer proceeds as follows: Sample two uniformly random polynomials $f(X)$ and $r(X)$ of degree t with coefficients in $\mathbb{Z}_q$, subject to $f(0) = s$. For $j = 1, 2, \cdots, n$: set $f_j := f(j)$, and $r_j := r(j)$. For $j = 1, 2, \cdots, n$: set $c_j := g_1^{f_j} g_2^{r_j} g_3^{\gamma_j}$, where γ_j are random elements from $\mathbb{Z}_q$. Compute the challenge value $d := H_{\mathsf{FS}}(c_1, \ldots, c_n)$; Check if $g_1 \neq g_2^d$ and if the check passes, set $z(X) := r(X) + d \cdot f(X)$ and $\pi^{\mathsf{sh}} := (c_1, \ldots, c_n, z(X))$. Otherwise, resample the γ_j values and recompute the commitments and challenge; Send shares (f_j, γ_j) securely to party P_j and return π^{sh}.
- $\mathsf{Verify}(n, t, \{f_j\}_{j=1}^{n}, \pi^{\mathsf{sh}}) \rightarrow$ `true/false`: Given n, threshold value t, group generators $pp := (g_1, g_2, g_3)$, a proof $\pi^{\mathsf{sh}} := (c_1, \ldots, c_n, z(X))$, and the shares $\{f_j, \gamma_j\}_{j=1}^{n}$: Each party P_j first checks that $\deg(z(X)) \leq t$, computes challenge value $d := H_{\mathsf{FS}}(c_1, \ldots, c_n)$ and then uses its shares (f_j, γ_j) and checks if $c_j = g_1^{f_j} g_2^{z(j) - d \cdot f_j} g_3^{\gamma_j}$. If the verification of P_j fails, then P_j broadcasts a complaint against the dealer. The dealer answers complaints by broadcasting the disputed share and opening data, as in the complaint procedure recalled in Section 2.2.
- $\mathsf{Reconstruct}(\{f_j\}_{j \in Q, |Q| = t+1}) \rightarrow \{s, \mathsf{false}\}$: Each party P_j broadcasts the secret values f_j and γ_j. A party P_j is said to be confirmed if $c_j = g_1^{f_j} g_2^{z(j) - d \cdot f_j} g_3^{\gamma_j}$, where $d := H_{\mathsf{FS}}(c_1, \ldots, c_n)$. Consider the f_j values of any $t + 1$ confirmed parties and interpolate the polynomial $f(X)$ of degree at most t that passes through those points. Finally, the output is $s = f(0)$ or `false` (if $t + 1$ valid shares were not obtained).

In [Bag25], Baghery also presented the Π_{F} VSS scheme, an efficient alternative to the Feldman VSS scheme. Unlike in Π_{P}, in Π_{F} the commitments are computed as $c_j = g_1^{f_j} g_2^{r_j}$ for $j = 1, \ldots, n$.

C Security Proofs

C.1 Hash Commitments

Binding. Among Q random-oracle queries, the probability of a collision in H is $O(Q^2/q)$.

Hiding. For the randomized hash commitment of Section 2, let a PPT algorithm with oracle access choose two messages m_0, m_1 and auxiliary information η. For a fresh independent $\omega \leftarrow \{0,1\}^\lambda$,

$$(\mathsf{pp}, \eta, H(m_0, \omega)) \approx_c (\mathsf{pp}, \eta, H(m_1, \omega)),$$

where $H(m, \omega)$ denotes $H(a, b, \omega)$ for $m = (a, b)$. The auxiliary information may include both messages, but the randomizer must remain private and independent. The two experiments are identical until a query reaches a committed input, which has probability $O(Q/2^\lambda)$ for Q queries.

C.2 Fiat–Shamir Extraction

We use the result of Attema, Fehr, and Klooß (AFK) on adaptive knowledge extraction [AFK23, Definition 10 and Theorem 4]. A continuation is the computation after the proof, including later oracle access and any auxiliary output. For a 2-special-sound protocol with one challenge in $\mathbb{Z}_q$, AFK gives an expected polynomial-time extractor with knowledge error $\kappa_{\mathsf{FS}} = (Q_{H_{\mathsf{FS}}} + 1)/q$, where $Q_{H_{\mathsf{FS}}}$ bounds the queries to the challenge oracle made in one prover execution, including its continuation, rather than the extractor's total queries across rewinds. Its extractor for protocols with one challenge [AFK23, Lemma 4 and Remark 5] retains the original execution and its auxiliary output. The probability that this execution accepts but extraction fails is at most κ_{FS}. When special soundness relies on computational binding, we use the formulation of [AFK23, Section 2.4]: extraction returns either a witness or a binding violation.

For a sharing proof, fix a set of recipient indices and regard the public commitment vector, together with any public secret commitment, as the statement. A verifier used only in the security proof checks the response polynomial and the private openings received at these indices. The openings are included in the response it verifies; they are not made public in the protocol. The challenge input includes the public statement. For the later-opening argument in **??**, this verifier also checks the later opening, and the protocol continuation is retained as auxiliary output. AFK then applies to the two-transcript extraction described there.

Expected polynomial-time extraction is used only in the soundness reductions for the real protocol; by Markov's inequality, a polynomial cutoff preserves non-negligible success in finding a binding violation, so PPT hardness suffices. The DKG simulators remain strictly polynomial time: they obtain corrupted dealers' polynomials by interpolation from the shares held by simulated honest recipients.

C.3 Proof of the Pedersen–DL Equality Argument

Proof (Proof of Theorem 1). The underlying interactive protocol is the standard proof of a preimage of $(s, \gamma) \mapsto (g_1^s g_2^\gamma, g_3^s)$ [Mau09]. Completeness follows by substitution. Two accepting transcripts with the same first message and $e \neq e'$ extract

$$\widehat{s} = \frac{\zeta_1' - \zeta_1}{e - e'}, \qquad \widehat{\gamma} = \frac{\zeta_2' - \zeta_2}{e - e'}.$$

Dividing the verification equations shows that these satisfy $a = g_1^{\widehat{s}} g_2^{\widehat{\gamma}}$ and $b = g_3^{\widehat{s}}$.

For any fixed challenge e, sample independent uniform $\zeta_1, \zeta_2 \in \mathbb{Z}_q$ and set

$$c_1 = g_1^{\zeta_1} g_2^{\zeta_2} a^e, \qquad c_2 = g_3^{\zeta_1} b^e.$$

The resulting transcript has exactly the real distribution, giving perfect honest-verifier zero knowledge. These properties do not require DL hardness.

For Fiat–Shamir zero knowledge, also sample e uniformly and program H_{FS} at the challenge input to return e; the fresh uniform first messages make a conflicting prior query negligible. Since the challenge hashes the full public statement, AFK gives adaptive knowledge extraction with the guarantees in Appendix C.2. □

C.4 Security of DKG_1

Proof (Proof of Theorem 4). Let $\mathcal{A}$ be a static PPT rushing adversary controlling $\mathcal{T} \subseteq [n]$, $|\mathcal{T}| \le t$. Write $\mathcal{H} = [n] \setminus \mathcal{T}$ and $m = |\mathcal{H}| \ge t + 1 \ge 2$. All oracle queries below include the domain identifiers specified in Section 2.

Correctness and robustness. By verifiability of $\Pi_{\mathsf{P}+}$ and Lemma 1, every dealer in $\mathcal{Q}_0$ has shared a unique polynomial of degree at most t, except with negligible probability. Binding follows from DL hardness and collision resistance of H in the ROM. Hence the honest parties hold shares of

$$F(X) = \sum_{i \in \mathcal{Q}_0} f^{(i)}(X), \qquad x = F(0), \qquad x_j = F(j).$$

The soundness argument of Theorem 2 uses only binding of the coordinate commitments, so it also applies to $H(f_{ij}, \rho^{(i)}(j))$. Thus an accepted public-key proof has $y_i = g_1^{s_i}$, except with negligible probability. If the proof is missing or rejected, the parties reconstruct the same contribution from its initial sharing.

An honest dealer belongs to $\mathcal{Q}_0$. Its genuine openings pass both phases, so a complaint that causes its reconstruction must supply a different accepted initial opening, violating binding. Missing or invalid public proofs and valid complaints are visible to everyone; hence all honest parties agree on $\mathcal{Q}_1$. The consistency of later openings ensures that reconstruction recovers the original polynomial. Consequently, all honest parties output $(g_1^{F(0)}, F(j))$, except with negligible probability. Their retained (f_{ij}, γ_{ij}) values also allow robust reconstruction of $F(0)$, since at least $n - |\mathcal{T}| \ge t + 1$ honest recipients hold valid openings for every qualified dealer.

Statistical unbiasedness. Define each qualified corrupted dealer's polynomial by interpolating its validated shares at a fixed set of $t + 1$ honest indices; on consistent sharings, this is its VSS polynomial. Fix an honest dealer P_h. For any $D \in \mathbb{Z}_q[X]_t$ vanishing on $\mathcal{T}$, change its sharing coins as follows, adjusting the mask to preserve z_h and the randomizers to preserve the commitments:

$$f^{(h)} \longmapsto f^{(h)} + D, \qquad r^{(h)} \longmapsto r^{(h)} + d_h D,$$

$$\gamma_{hj} \longmapsto \gamma_{hj} + \beta\big[H(f_{hj}, r_{hj}) - H(f_{hj} + D(j), r_{hj} + d_h D(j))\big],$$

where $\beta = \log_{g_2} g_1$ is used only in the analysis. For every fixed choice of the oracles, this transformation preserves the commitments, d_h, and z_h; its inverse uses $-D$. At corrupted indices, $D(j) = 0$, so both the hash inputs and γ_{hj} are unchanged. Thus all private messages and complaint responses seen by $\mathcal{A}$ remain the same, as do $\mathcal{Q}_0$ and the other dealers' polynomials.

Since D is arbitrary, F is uniform subject to its corrupted evaluations $x_j = F(j)$, even given the initial adversarial view and the other polynomials. Define a polynomial that equals 1 at 0 and vanishes on $\mathcal{T}$:

$$L_{\mathcal{T}}(X) = \prod_{j \in \mathcal{T}} \frac{X - j}{-j},$$

with empty product 1. Taking $D = \theta L_{\mathcal{T}}$ shifts $F(0)$ by any $\theta \in \mathbb{Z}_q$ without changing those evaluations. Hence $F(0)$ is exactly uniform. The actual outputs agree with this polynomial except on the negligible correctness failures above. This gives statistical indistinguishability of the real secret and public keys from uniform, including against an unbounded distinguisher given the key alone.

Computational simulation. Fix a PPT distinguisher $\mathcal{D}$ and query bound Q as in Definition 8. The simulator constructed below does not depend on $\mathcal{D}$.

Game 0. This is the real execution of DKG_1.

Game 1. After the VSS stage, interpolate each qualified corrupted dealer's polynomial from $t+1$ honest shares. Abort if it disagrees with another validated honest share or any later accepted reconstruction opening. Also abort if a corrupted dealer's public-key proof is accepted with $y_i \neq g_1^{s_i}$, or a complaint against an honest dealer supplies a different accepted initial opening. These checks use the shares received by simulated honest recipients from corrupted dealers and the openings sent by honest dealers to corrupted recipients. Let ε bound the total probability of these events in the real protocol; the correctness argument makes ε negligible. On an abort, the game returns a fixed failure symbol. Otherwise its honest outputs are $(g_1^{F(0)}, F(j))$.

Game 2. For every honest dealer $i \in \mathcal{H}$, generate the public-key proof by sampling independent uniform $\delta_i \leftarrow \mathbb{Z}_q$ and $\zeta_i \leftarrow \mathbb{Z}_q[X]_t$ and setting

$$\mathsf{C}_0^{(i)} = g_1^{\zeta_i(0)} y_i^{\delta_i}, \qquad \mathsf{C}_j^{(i)} = H\big(f_{ij}, \zeta_i(j) + \delta_i f_{ij}\big).$$

Program $H_{\mathsf{FS}}(g_1, y_i, \mathsf{C}_0^{(i)}, \ldots, \mathsf{C}_n^{(i)}) = \delta_i$, aborting if the input was queried before. The substitution $\rho^{(i)} = \zeta_i + \delta_i f^{(i)}$ gives the real distribution when the input is fresh. Since $\mathsf{C}_0^{(i)}$ is uniform, the total programming error is $O(mQ/q)$. Thus Games 1 and 2 are statistically close.

Game 3. For every $i, j \in \mathcal{H}$, sample $\mathsf{C}_j^{(i)}$ as an independent uniform element of $\mathbb{Z}_q$, omitting the corresponding internal H calls and letting honest recipients accept honest dealers' public-key proofs. At corrupted indices, keep the genuine oracle evaluations. Leave H itself unchanged, and retain the abort checks from Game 1.

The two games can be coupled until an external query reaches a replaced commitment input. Let E be the larger event that $\mathcal{A}$ or $\mathcal{D}$ queries the H domain for some $i, j \in \mathcal{H}$ with first scalar component f_{ij}, regardless of the second component. This includes queries made before the proof is published and after $\mathcal{D}$ receives the honest outputs. We show that $\Pr[E]$ in Game 3 is negligible under DL hardness.

Given $U = g_1^u$, a reduction $\mathcal{B}$ chooses a uniform pair of distinct honest dealers h, ℓ and independent uniform polynomials $p^{(i)} \in \mathbb{Z}_q[X]_t$, $i \in \mathcal{H}$. It interprets the honest polynomials as

$$f^{(h)} = p^{(h)} + uL_{\mathcal{T}}, \qquad f^{(\ell)} = p^{(\ell)} - uL_{\mathcal{T}}, \qquad f^{(i)} = p^{(i)} \quad (i \notin \{h, \ell\}).$$

The reduction generates the initial sharing using the known $p^{(i)}$. Applying the change of the sharing randomness with $D = uL_{\mathcal{T}}$ for P_h and $D = -uL_{\mathcal{T}}$ for P_ℓ gives exactly the same initial view for these $f^{(i)}$. The reduction need not compute the changed masks or openings at honest indices; the evaluations and openings at $\mathcal{T}$ are unchanged. It obtains the corrupted dealers' polynomials from simulated honest recipients and computes the aggregate polynomial as

$$F = \textstyle\sum_{i \in \mathcal{H}} p^{(i)} + \sum_{i \in \mathcal{Q}_0 \cap \mathcal{T}} f^{(i)}.$$

It can therefore provide every honest output to $\mathcal{D}$. It computes $y_h = g_1^{p^{(h)}(0)} U$ and $y_\ell = g_1^{p^{(\ell)}(0)} / U$, and generates all honest public-key proofs as in Game 3. The known values at $\mathcal{T}$ suffice for complaints against honest dealers; reconstruction of corrupted dealers uses the openings already received. If a complaint would require reconstruction of an honest dealer, the abort check triggers first. The reduction therefore needs no unknown opening. For a query with candidate v in the domain for (h, j), $j \in \mathcal{H}$, the reduction tests

$$g_1^v = g_1^{p^{(h)}(j)} U^{L_{\mathcal{T}}(j)}.$$

If it holds, then $u = (v - p^{(h)}(j))/L_{\mathcal{T}}(j)$; the analogous formula for ℓ has a minus sign. Since $L_{\mathcal{T}}(j) \neq 0$ at every honest index, extraction works for every corruption set $\mathcal{T}$, including $\mathcal{T} = \varnothing$ and $|\mathcal{T}| < t$. Earlier queries are retained and checked as well.

For each fixed U and selected pair, the simulation has the same joint distribution of the Game 3 view, the honest polynomials $f^{(i)}$, and the honest outputs. The shifts $\pm uL_{\mathcal{T}}$ and the bijective changes of the initial sharing randomness preserve this distribution. Thus the pair is independent of the dealer involved in the first query witnessing E. That dealer lies in the pair with probability $2/m$, giving

$$\Pr_{\text{Game 3}}[E] \leq \frac{m}{2} \Pr[\mathcal{B}(g_1^u) = u].$$

This is negligible by DL hardness, establishing computational indistinguishability of Games 2 and 3.

Game 4. Let the functionality choose the key and, as in GJKR's simulator [GJKR07, Section 4.3, Fig. 3], adjust one honest contribution. Simulate the initial VSS stage as before, submit $x_j = \sum_{i \in \mathcal{Q}_0} f^{(i)}(j)$, $j \in \mathcal{T}$, and receive y. Fix an honest dealer P_h, keep the other dealers' prescribed contributions, and set its public contribution to make the product of all public contributions equal y:

$$y_h^* = y \Big/ \prod_{i \in \mathcal{Q}_0 \setminus \{h\}} g_1^{s_i}.$$

Generate every honest public-key proof as in Game 3, using y_h^* for P_h, and take the honest outputs from the functionality. To compare the joint distributions, let F_{ideal} be the ideal polynomial. The simulator never receives this polynomial; it is used only in the analysis. Define

$$\widetilde{f}^{(h)} = F_{\mathrm{ideal}} - \sum_{i \in \mathcal{Q}_0 \setminus \{h\}} f^{(i)}.$$

The real F has the same conditional distribution as the functionality's polynomial. To see this, fix the oracle functions before second-phase programming and condition on the initial adversarial view and the other dealers' polynomials. The initial change of the sharing randomness shows that F is uniform over all degree-at-most-t polynomials with evaluations x_j on $\mathcal{T}$, which is the distribution sampled by the functionality. Thus $\widetilde{f}^{(h)}$ has the required conditional distribution. It agrees with $f^{(h)}$ on $\mathcal{T}$ and has public contribution y_h^*. After the replacement in Game 3, these are the only values of this polynomial used in the remaining adversarial view. Consequently, Games 3 and 4 have the same distribution, including the full honest output vector and subsequent oracle access, with the same abort conditions.

Game 4 defines $\mathcal{S}_1$. It runs in polynomial time, using interpolation for corrupted dealers' polynomials and the retained openings for complaints and reconstruction. It programs only the second-phase domain of H_{FS}. On an internal abort, it returns an arbitrary view and lets the functionality deliver outputs, submitting default inputs if the ideal call has not yet occurred. To account for these aborts, put

$$\eta = \varepsilon + O(mQ/q) + \Pr_{\text{Game 3}}[E].$$

The comparisons bound both the distinguishing advantage between Games 0 and 3 and the abort probability in Game 3 by η. When the ideal failure symbol is replaced by the functionality's outputs, the real/ideal advantage is therefore at most 2η. Together with statistical unbiasedness, this proves the theorem. □

C.5 Security of DKG_2

Proof (Proof of Theorem 5). Let $\mathcal{A}$ be a static PPT rushing adversary controlling a set $\mathcal{T}$ of at most t parties, and write $\mathcal{H} = [n] \setminus \mathcal{T}$ and $m = |\mathcal{H}| \geq 2$.

Correctness and robustness. Apply the correctness and robustness argument of Theorem 4, replacing $\Pi_{\mathsf{P}+}$ by Π_{LA} and using scalar openings checked against Eq. (5). By Lemma 1, each qualified dealer's honest shares lie on a unique $f^{(i)} \in \mathbb{Z}_q[X]_t$, and every accepted complaint or reconstruction opening agrees with it, except with negligible probability. The public-key and complaint checks otherwise follow DKG_1: honest dealers remain qualified, a complaint causing an honest dealer's reconstruction violates binding, and all honest parties agree on $\mathcal{Q}_1$. Missing or invalid proofs lead to reconstruction of the same contribution. Consequently, every honest P_j outputs $(g_1^{F(0)}, F(j))$, where $F = \sum_{i \in \mathcal{Q}_0} f^{(i)}$. The retained openings also allow robust reconstruction of $F(0)$, since $n - |\mathcal{T}| \geq t + 1$.

Computational simulation. We use the experiment of Definition 8, including the honest outputs, and the bound Q on all oracle queries, including those of the distinguisher $\mathcal{D}$, as in Theorem 4.

Game 0. This is the real execution of DKG_2.

Game 1. Abort if a qualified corrupted sharing is inconsistent, an accepted reconstruction opening disagrees with its dealer's polynomial, a corrupted dealer's public-key proof is accepted with $y_i \neq g_1^{s_i}$, or a complaint against an honest dealer supplies a different accepted initial opening. The preceding argument bounds the probability of these events by a negligible function. Otherwise the honest outputs are $(g_1^{F(0)}, F(j))$, so Games 0 and 1 are statistically close.

Game 2. For every honest dealer $i \in \mathcal{H}$, generate both proofs by sampling the challenges and responses first: choose independent uniform $d_i, \delta_i \in \mathbb{Z}_q$ and $z_i, \zeta_i \in \mathbb{Z}_q[X]_t$, and set

$$c_j^{(i)} = H(f_{ij}, z_i(j) + d_i f_{ij}),$$
$$\mathsf{C}_0^{(i)} = g_1^{\zeta_i(0)} y_i^{\delta_i}, \qquad \mathsf{C}_j^{(i)} = H(f_{ij}, \zeta_i(j) + \delta_i f_{ij}).$$

Program the corresponding challenge inputs to d_i and δ_i, aborting if either was queried before. The substitutions $r^{(i)} = z_i + d_i f^{(i)}$ and $\rho^{(i)} = \zeta_i + \delta_i f^{(i)}$ give the real distribution when the inputs are fresh. In the initial proof, since the mask evaluation is fresh and uniform, the probability that its commitment input was queried earlier is at most Q/q; otherwise its fresh hash output makes the challenge input unpredictable, except with probability Q/q. For the public-key proof, $\mathsf{C}_0^{(i)}$ is uniform. Summing over the honest proofs gives negligible programming error, so Games 1 and 2 are statistically close.

Game 3. In both phases, for every $i, j \in \mathcal{H}$, replace the sharing commitment by an independent uniform element of $\mathbb{Z}_q$, omitting its internal H call and letting honest recipients accept honest dealers' proofs. At corrupted indices keep the genuine oracle evaluations. Leave H itself unchanged and retain the abort checks. The games can be coupled until an external query reaches a replaced input.

Let E be the larger event that $\mathcal{A}$ or $\mathcal{D}$ queries either sharing domain for some $i, j \in \mathcal{H}$ with first component f_{ij}, including before publication or after receiving the honest outputs.

Apply the reduction from Game 3 of Theorem 4, with the same $L_{\mathcal{T}}$ and opposite shifts $+uL_{\mathcal{T}}$ and $-uL_{\mathcal{T}}$ in two uniformly selected honest dealers' polynomials. Here both sharing phases are simulated using the known evaluations at $\mathcal{T}$ and the random commitments at honest indices; the initial $\Pi_{\mathsf{P}+}$ change of the sharing randomness is unnecessary. The public contributions are computable from $U = g_1^u$, and the two unknown terms cancel in F, allowing the reduction to supply the entire honest output vector. Complaints against honest dealers disclose only evaluations at $\mathcal{T}$, and honest dealers are never reconstructed.

Use the same query tests in both sharing domains, including for earlier queries. The polynomial shifts preserve the Game 3 distribution for every selected pair, so the first dealer witnessing E belongs to the pair with probability $2/m$. The same DL bound makes $\Pr[E]$ negligible, establishing computational indistinguishability of Games 2 and 3.

Game 4. Finally, after the VSS stage, submit $x_j = \sum_{i \in \mathcal{Q}_0} f^{(i)}(j)$, $j \in \mathcal{T}$, to $\mathcal{F}_{\mathsf{DKG}}^{t,n}$. Fix an honest dealer P_h. On receiving y, set

$$y_h^* = y / \prod_{i \in \mathcal{Q}_0 \setminus \{h\}} g_1^{s_i}$$

and use it in P_h's simulated public-key proof, taking the honest outputs from the functionality. In Game 3, the initial view and oracle state depend on $f^{(h)}$ only through its evaluations on $\mathcal{T}$. Conditional also on the other dealers' polynomials, F is uniform subject to $F(j) = x_j$ for $j \in \mathcal{T}$. The argument with F_{ideal} in Game 4 of Theorem 4 therefore applies: the substituted dealer polynomial has the required conditional distribution, unchanged evaluations on $\mathcal{T}$, and public contribution y_h^*. Games 3 and 4 have the same joint distribution, with the same abort conditions.

The simulator $\mathcal{S}_2$ implements Game 4, simulating every honest dealer's proofs and programming only H_{FS}. It obtains each qualified corrupted dealer's polynomial by interpolating $t+1$ honest recipients' shares. Its retained state supplies complaint responses and reconstruction messages for corrupted dealers. On an abort, it returns an arbitrary view and supplies default ideal inputs if needed. As in Theorem 4, the preceding game comparisons also bound the simulator's abort probability by a negligible function. All operations are polynomial time, and the functionality supplies the honest outputs. This proves computational realization of $\mathcal{F}_{\mathsf{DKG}}^{t,n}$ and, since the ideal key is uniform, computational unbiasedness. □

C.6 Security of DKG_3

Proof (Proof of Theorem 6). Let $\mathcal{A}$ be a static PPT rushing adversary controlling a set $\mathcal{T}$ of at most t parties, and write $\mathcal{H} = [n] \setminus \mathcal{T}$ and $m = |\mathcal{H}| \geq 2$.

Correctness and robustness. Complaint resolution determines a common qualified set and supplies a valid opening to every honest recipient of a qualified dealer. An honest dealer cannot be disqualified by a false complaint: unmasked answers are checked directly and complaints about them are ignored; for masked answers, an invalid pad opening is ignored, and a valid one recovers the dealer's original valid data. As in Lemma 1, sharing soundness and binding determine a unique $f^{(i)} \in \mathbb{Z}_q[X]_t$ for each $i \in \mathcal{Q}_0$, consistent with every accepted opening, except with negligible probability.

We must also show that an accepted public-key proof implies $y_i = g_1^{s_i}$, except with negligible probability. Two accepting sharing transcripts with the same commitments and distinct challenges give

$$\widehat{f}^{(i)} = \frac{z_i' - z_i}{d_i - d_i'}, \qquad \widehat{\gamma}_i^{\mathsf{sh}} = \frac{w_i' - w_i}{d_i - d_i'}, \qquad A_i = g_1^{\widehat{f}^{(i)}(0)} g_2^{\widehat{\gamma}_i^{\mathsf{sh}}}.$$

Together with the hash openings at the honest indices, these equations give a witness for the relation defined by the public commitments or a collision in H. Thus AFK gives expected polynomial-time extraction as in Section 2.1. Binding ensures that $\widehat{f}^{(i)}$ agrees with the original honest shares. Including a later reconstruction opening in the verifier's checks gives the same consistency guarantee as Lemma 1.

Apply this extractor and the extractor for the equality proof of Theorem 1 to the same original execution, retaining its continuation and oracle query history. Except for their negligible knowledge errors or a binding violation, they give openings $(\widehat{f}^{(i)}(0), \widehat{\gamma}_i^{\mathsf{sh}})$ and $(\widehat{s}_i, \widehat{\gamma}_i^{\mathsf{pk}})$ of the same A_i, with $y_i = g_1^{\widehat{s}_i}$. Pedersen binding under DL hardness identifies $\widehat{s}_i = \widehat{f}^{(i)}(0) = s_i$, as required.

If a qualified dealer fails to provide a valid proof, at least $t+1$ honest recipients supply openings that recover its original contribution. Hence every honest P_j outputs $(g_1^{F(0)}, F(j))$, where $F = \sum_{i \in \mathcal{Q}_0} f^{(i)}$. Every qualified contribution remains in the sum, and the retained openings also allow subsequent robust reconstruction.

Computational simulation. We use the experiment of Definition 8, including the honest outputs, and the bound Q on all oracle queries, including those of the distinguisher $\mathcal{D}$, as in Theorem 5.

Game 0. This is the real execution of DKG_3.

Game 1. Abort if a qualified corrupted sharing is inconsistent, an accepted reconstruction opening disagrees with its dealer's polynomial, a false complaint disqualifies an honest dealer, or a corrupted dealer's public-key proof is accepted with $y_i \neq g_1^{s_i}$. These events have negligible probability by the preceding argument. Otherwise the honest outputs are $(g_1^{F(0)}, F(j))$, so Games 0 and 1 are statistically close.

Game 2. For every honest dealer $i \in \mathcal{H}$, generate the sharing proof by sampling the challenge and responses first: choose independent uniform $A_i \in \mathbb{G}$, $d_i, w_i \in \mathbb{Z}_q$, and $z_i \in \mathbb{Z}_q[X]_t$, and set

$$B_i = g_1^{z_i(0)} g_2^{w_i} A_i^{d_i}, \qquad c_j^{(i)} = H(f_{ij}, z_i(j) + d_i f_{ij}).$$

Program the sharing challenge to d_i and simulate the equality proof for (A_i, y_i) as in Theorem 1. Perfect hiding of A_i, together with the substitutions $r^{(i)} = z_i + d_i f^{(i)}$ and $\sigma_i = w_i + d_i \gamma_i$, gives the real sharing distribution. The fresh uniform B_i and fresh first messages in the equality proof make the probability of an earlier query to these challenge inputs negligible. Thus Games 1 and 2 are statistically close.

Game 3. For every $i, j \in \mathcal{H}$, replace $c_j^{(i)}$ by an independent uniform hash output. For every such pair with $i \neq j$, also sample $a_{ij} \leftarrow \mathbb{Z}_q$ in advance and replace K_{ij} by an independent uniform hash output, interpreting the private pad as $k_{ij} = a_{ij} - f_{ij}$. Omit the corresponding internal commitment-oracle calls and let honest parties accept these honest sharings and pad openings. All other messages and the abort checks are unchanged; neither the sharing nor the pad commitment domains of H are programmed.

Uniform pads make this the same distribution as Game 2 until an external query reaches a replaced input. Let E be the event that $\mathcal{A}$ or $\mathcal{D}$ queries a sharing domain for $i, j \in \mathcal{H}$ with first component f_{ij}, or a pad domain for distinct $i, j \in \mathcal{H}$ at k_{ij}. The event includes queries made before publication and after the honest outputs are delivered. To bound E, use the reduction from Game 3 of Theorem 4, now with $U = g_1^u$ and the following simulation of the initial sharing. All honest sharing proofs use the replaced commitments and uniform A_i, and all public-key proofs are simulated from (A_i, y_i). Thus the reduction need not know the hidden shares or the openings of A_i. The unknown polynomial terms cancel in F, allowing it to supply every honest output. Messages involving a corrupted recipient use known shares and pads, so complaints require no unknown opening; an honest dealer is never reconstructed.

A sharing query gives a candidate $v = f_{ij}$ directly. A pad query b gives the candidate $v = a_{ij} - b$. In either case, for the selected dealer h test $g_1^v = g_1^{p^{(h)}(j)} U^{L_{\mathcal{T}}(j)}$ and extract $u = (v - p^{(h)}(j))/L_{\mathcal{T}}(j)$ on equality; use the opposite sign for the other selected dealer. Since a_{ij} is sampled in advance, the test applies to earlier pad queries as well. The pads $k_{ij} = a_{ij} - f_{ij}$ are independent uniform scalars, so the polynomial shifts preserve the joint distribution and conceal the selected pair. As before, the first query witnessing E yields a DL solution with probability $2/m$. Hence $\Pr[E]$ is negligible, proving computational indistinguishability of Games 2 and 3.

Game 4. After the second round, submit $x_j = \sum_{i \in \mathcal{Q}_0} f^{(i)}(j)$, $j \in \mathcal{T}$, to $\mathcal{F}_{\mathsf{DKG}}^{t,n}$. Fix an honest dealer P_h. On receiving y, set

$$y_h^* = y / \prod_{i \in \mathcal{Q}_0 \setminus \{h\}} g_1^{s_i}$$

and simulate P_h's equality proof for (A_h, y_h^*), taking the honest outputs from the functionality. In Game 3, the initial adversarial view and oracle state depend on $f^{(h)}$ only through its evaluations on $\mathcal{T}$: A_h is uniform, and the replaced commitments and masked answers are independent. Conditional also on the other dealers' polynomials, F is uniform subject to its corrupted evaluations. The same argument using the ideal polynomial as in Game 4 of Theorem 4 therefore gives identical joint distributions for Games 3 and 4, with the same abort conditions. The simulator $\mathcal{S}_3$ implements Game 4, simulating all honest sharing and equality proofs and programming only H_{FS}. It obtains each qualified corrupted dealer's polynomial by interpolating $t + 1$ honest recipients' shares. Its retained state supplies the reconstruction messages for corrupted dealers that fail to provide valid proofs. On an abort, it returns an arbitrary view and supplies default ideal inputs if needed. The game comparisons also bound its abort probability, as in Theorem 4. This gives a polynomial-time simulator whose view, jointly with the functionality's honest outputs, is computationally indistinguishable from the real execution. Uniformity of the ideal key then gives computational unbiasedness. □