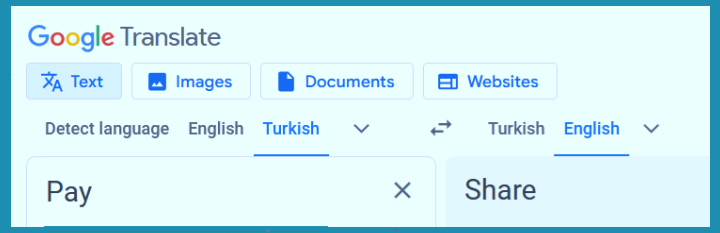
KU LEUVEN

Workshop on Secret Sharing Schemes
12-14 May 2025 – Tarragona, Spain



Π: A Unified Framework for Computational Verifiable Secret Sharing

ia.cr/2023/1669

Karim Baghery

COSIC

KU Leuven, Belgium

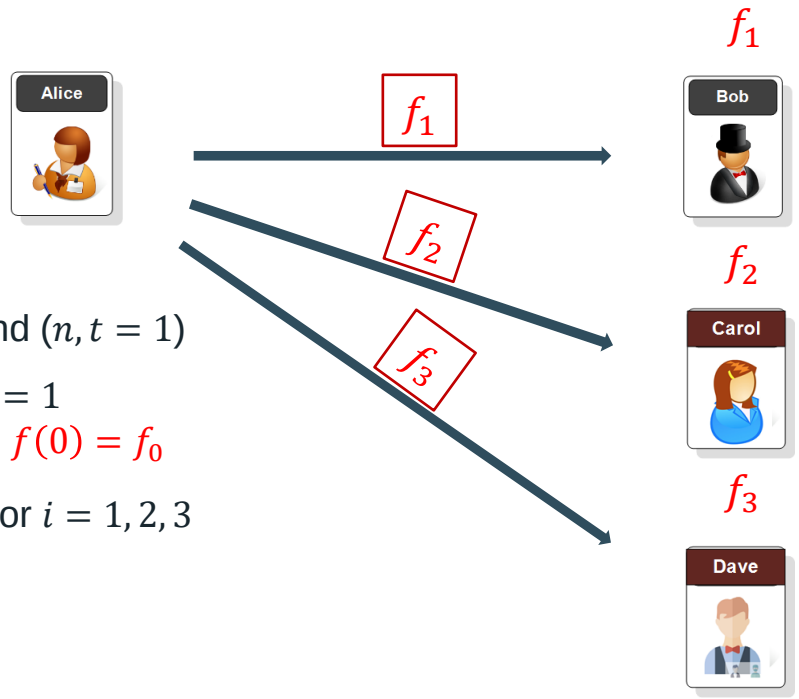
karim.baghery@kuleuven.be

Shamir Secret Sharing: A Toy Example

- Initial
- Share
- Reconstruct

- Given a secret f_0 a **trusted** dealer P_0 shares it with $n \geq 2t+1$ parties $\{P_i\}_{i=1}^{n}$, s.t.,
  - $t+1$ shareholders can reconstruct the secret f_0
  - t shareholders cannot reconstruct the secret f_0

Share

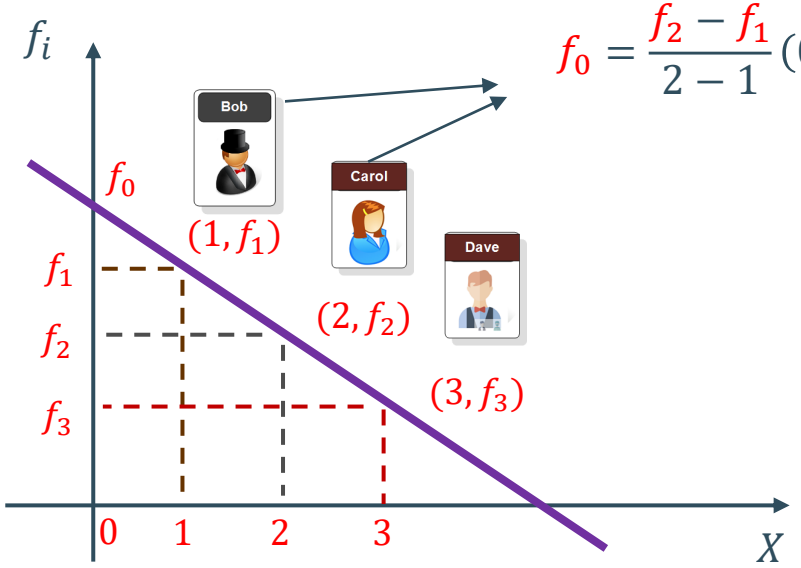


- Given a secret f_0 and $(n, t = 1)$
- Sample a degree t = 1 polynomial $f(X)$ s.t. $f(0) = f_0$
- Compute $f_i = f(i)$ for $i = 1, 2, 3$

Reconstruction

$$f_X - f_1 = \frac{f_2 - f_1}{2 - 1}(X - 1)$$

$$f_0 = \frac{f_2 - f_1}{2 - 1}(0 - 1) + f_1$$



Computational Verifiable Secret Sharing (VSS)

Based on Homomorphic Commitments

VSS from Homomorphic Commitments: Feldman [Fel87]

- Given a secret f_0 a ~~**trusted**~~ dealer P_0 shares it with $n \geq 2t+1$ parties $\{P_i\}_{i=1}^{n}$, s.t.,
  - $t+1$ shareholders can reconstruct the secret f_0
  - t shareholders cannot reconstruct the secret f_0

- Initial - Share
- **Verify** - Reconstruct

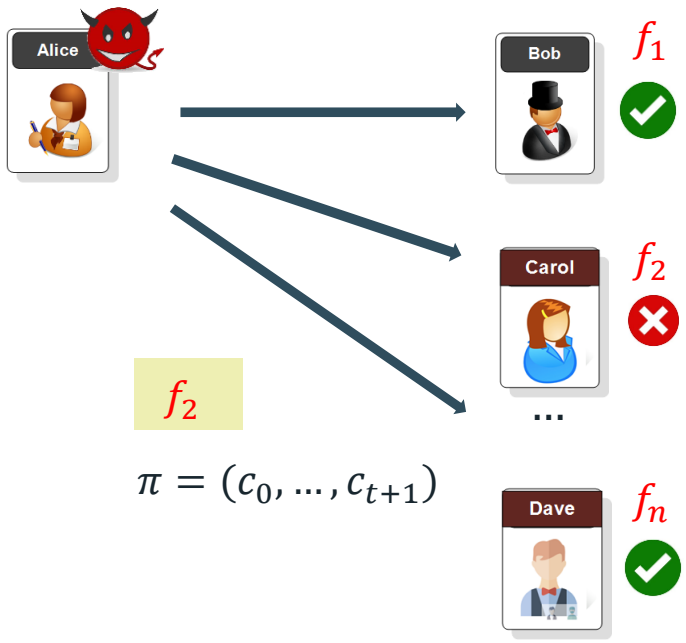
Share

- Given a secret f_0 and (n, t)
- Sample a degree t polynomial $f(X) = f_0 + a_1 X + \ldots + a_t X^t$
- Compute $f_i = f(i)$ for $i = 1, \ldots, n$
- Sets $c_0 = g^{f_0},\ c_1 = g^{a_1}, \ldots, c_t = g^{a_t}$
- Publish $\pi = (c_0, \ldots, c_{t+1})$ and send privatly f_i to party P_i

Add. Homomorphic Commitments:

$com(a_1) \times com(a_2) = com(a_1 + a_2)$

e.g., $g^{a_1} \times g^{a_2} = g^{a_1 + a_2}$



Verify

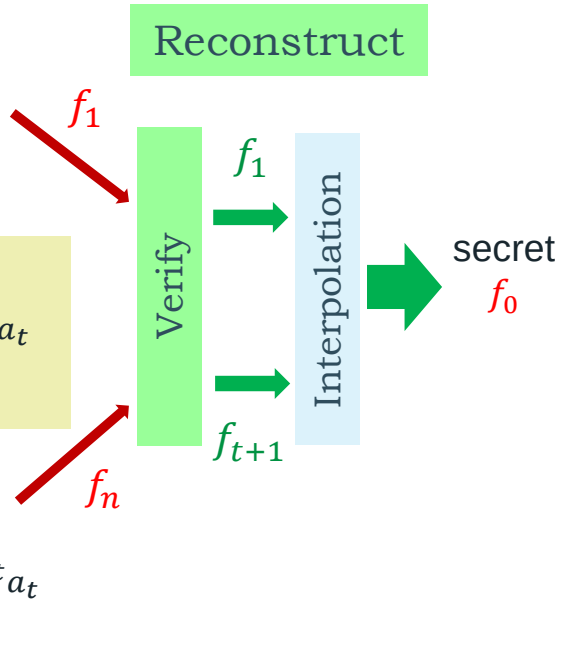
$$g^{f_i} = \prod_{j=0}^{t} c_j^{i^j}$$

$$g^{f_1} = c_0^{1^0} \times c_1^{1^1} \times \cdots \times c_t^{1^t} = g^{f_0} \times g^{a_1} \times \cdots \times g^{a_t} = g^{f(1)}$$

$$g^{f_2} = c_0^{2^0} \times c_1^{2^1} \times \cdots \times c_t^{2^t} = g^{f_0} \times g^{2a_1} \times \cdots \times g^{2^t a_t} = g^{f(2)}$$

$$g^{f_n} = c_0^{n^0} \times c_1^{n^1} \times \cdots \times c_t^{n^t} = g^{f_0} \times g^{na_1} \times \cdots \times g^{n^t a_t} = g^{f(n)}$$

Reconstruct



[Fel87] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In 28th Annual Symposium on Foundations of Computer Science, 1987.

VSS from Homomorphic Commitments: Pedersen [Ped91]

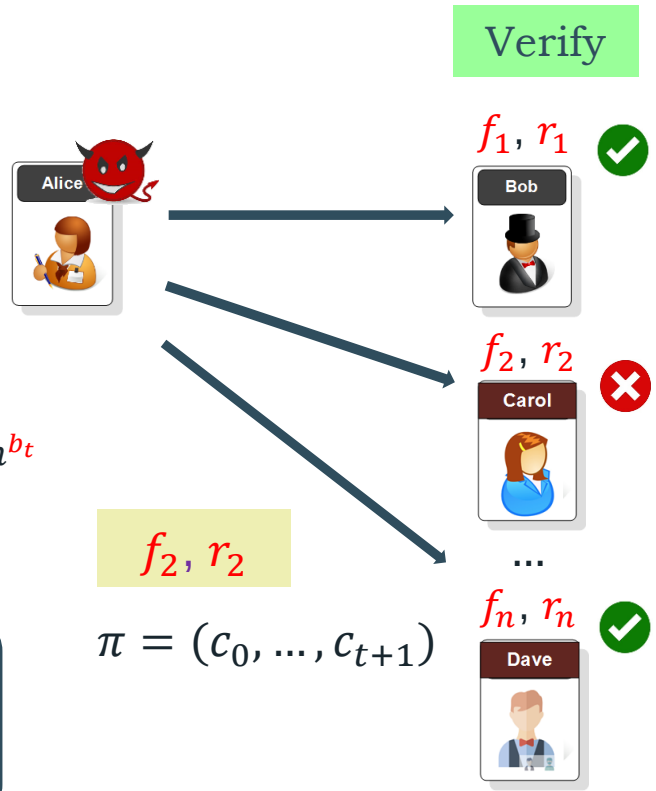
- Pedersen VSS: t of shareholders (information theoretically) learns nothing about f_0
- Pedersen Commitment: $com(f,r) = g^f h^r$ (Perfectly Hiding + Computationally Binding)

Share

- Given a secret f_0 and (n,t)
- Sample a degree t polynomial $f(X) = f_0 + a_1 X + \ldots + a_t X^t$
- Sample another degree t polynomial $r(X) = r_0 + b_1 X + \ldots + b_t X^t$
- Set $f_i = f(i)$ and $r_i = r(i)$ for $i = 1, \ldots, n$
- Sets $c_0 = g^{f_0} h^{r_0}, c_1 = g^{a_1} h^{b_1}, \ldots, c_t = g^{a_t} h^{b_t}$
- Publish $\pi = (c_0, \ldots, c_{t+1})$ and send privatly (f_i, r_i) to party P_i

Pedersen Commitment:

$com(f,r) \times com(f',r') = g^f h^r \times g^{f'} h^{r'}$
$= g^{f+f'} h^{r+r'} = com(f+f', r+r')$



Verify

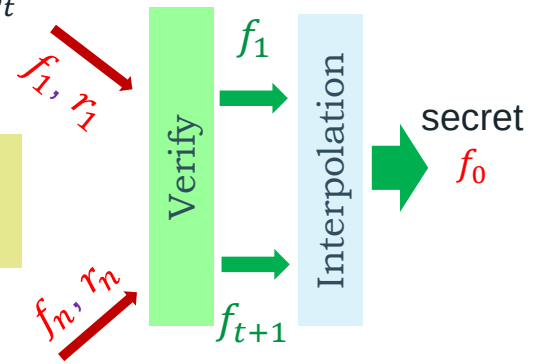
$$g^{f_i} h^{r_i} = \prod_{j=0}^{t} c_j^{i^j}$$

$$g^{f_1} h^{r_1} = c_0^{1^0} \times c_1^{1^1} \times \cdots \times c_t^{1^t}$$
$$= g^{f_0} h^{r_0} \times \cdots \times g^{a_t} h^{b_t}$$
$$= g^{f(1)} h^{r(1)}$$

$$g^{f_2} h^{r_2} = c_0^{2^0} \times c_1^{2^1} \times \cdots \times c_t^{2^t}$$
$$= \cdots = g^{f(2)} h^{r(2)}$$

...

Reconstruct



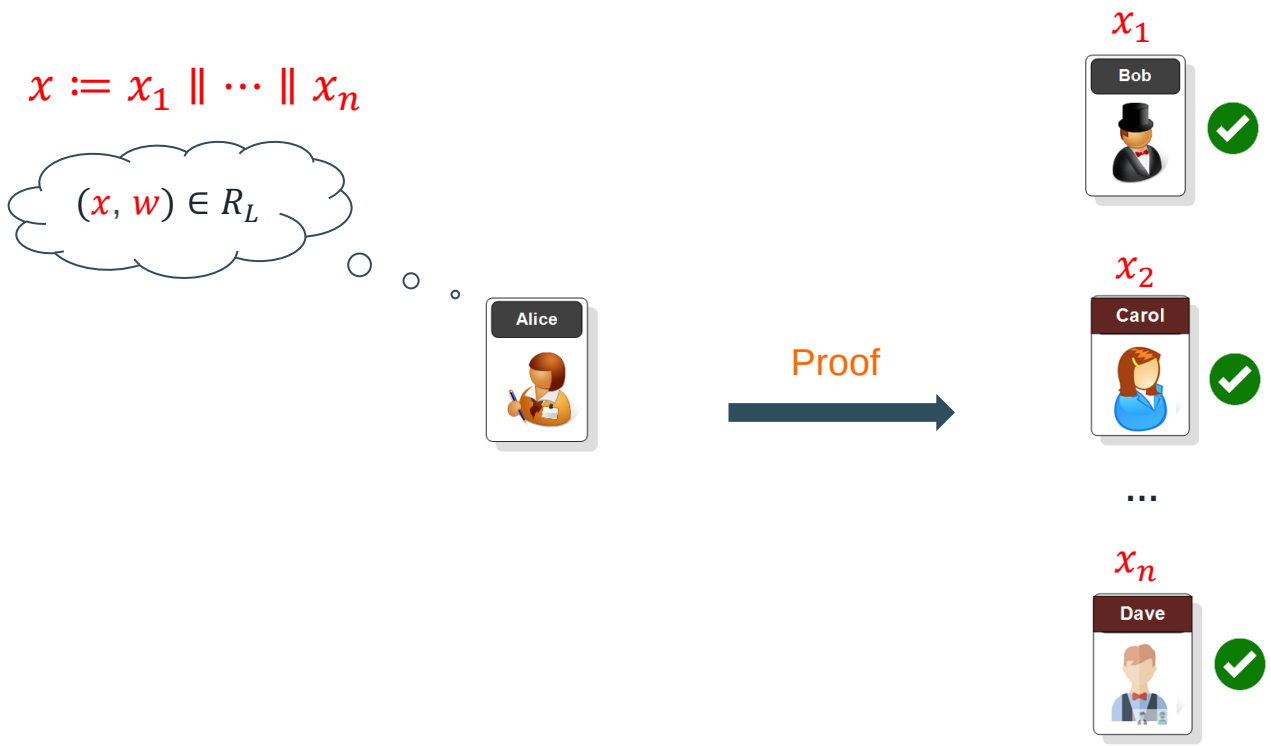
[Ped'91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. Advances in Cryptology – CRYPTO'91.

A PQ Secure Computational VSS

Based on Hash Functions

ZK Proofs Over Secret Shared Data: [BBC+19]

- Involves a prover and $n \geq 2t + 1$ verifiers, which allows the parties to communicate in synchronous rounds.
- The prover holds an n-distributed input $x \coloneqq x_1 \parallel \cdots \parallel x_n$ and a witness w, and each V_i holds an input piece x_j.



- **Completeness:** If $x \in L$, honest prover will convince honest verifiers.
- **Soundness against prover and t verifiers:** If $x \notin L$, honest verifiers will reject the proof, except for a negligible probability, even if up to t of verifiers collude with the prover.
- **Threshold Zero-Knowledge (TZK):** any subset of the verifiers up to t, should learn no additional information about x, beyond their own shares and the fact that $x \in L$.

✓ **Positive Result: We can build NI-TZK proof schemes for $n \geq 2t + 1$ [BBC+19].**

[BBC+19]: Dan Boneh, Elette Boyle, Henry Corrigan-Gibbs, Niv Gilboa, Yuval Ishai. Zero-Knowledge Proofs on Secret-Shared Data via Fully Linear PCPs.

[ABCP23]: A PQ-Secure NI-TZK Proof for Shamir Relation

- A hash-based NI-TZK proof scheme for Shamir n-distributed relation,

$$R_i = \{(f_i, f(X)) \mid f(i) = f_i\} \quad \text{for } i = 1, \dots, n.$$

- Using the proposed NI-TZK proof scheme, we construct a PQ secure VSS scheme.

DL: Discrete Logarithm, BC: Broadcast, PV: Private, n: Number of parties, t: Number of malicious parties, E_G: Exponentiation in G, PE: Degree-d polynomial evaluations, H: Hashing, $|X|$: X element size.

VSS Scheme	High-Entropy Secret?	Assumption	Sharing	Dealer's Communication	Veriy
Feldman [Fel87]	Yes	DL (Plain)	$0.5n\ E_G + n\ PE$	PV: $n\ \lvert Z_q \rvert$ & BC: t $\approx 0.5n\ \lvert G \rvert$	$t \approx 0.5n\ E_G$
Pedersen [Ped91]	No	DL (Plain)	$1n\ E_G + 2n\ PE$	PV: $2n\ \lvert Z_q \rvert$ & BC: t $\approx 0.5n\ \lvert G \rvert$	$t \approx 0.5n\ E_G$
[ABCP23]	No	Hash (QROM)	$2n\ H + 2n\ PE$	PV: $2n\ \lvert Z_q \rvert$ & BC: $2n\ \lvert H \rvert + 0.5n\ \lvert Z_q \rvert$	$1\ PE + 3\ H$

☺ Considerably faster Verify (& Reconstruct)
☺ Faster sharing (lightweight operations)
☺ PQ secure
😐 Slightly higher communication (~5x)
😐 Not IT-Simulatable (like [Ped91])

[ABCP23]: Shahla Atapoor, Karim Baghery, Daniele Cozzo, Robi Pedersen. VSS from Distributed ZK Proofs. ASIACRYPT 2023.

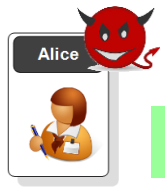
Our Construction From [ABCP23] Can be Optimized and Generalized

General Construction of Π: Share, Verify, Reconstruct

- Let $Com(.,.)$ be a (vector) commitment scheme, and $(Q)ROM$ be a (Quantum) ROM

Dealer

f_0



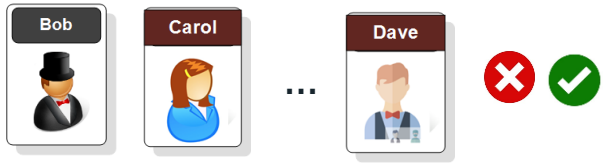
Share

1) Given (f_0, n, t): sample a degree-t $f(X)$ s.t., $f(0) = f_0$ & set $f_i = f(i)$ for $i = 1, \dots, n$
2) Sample a degree-t polynomial $r(X)$ & set $r_i = r(i)$ for $i = 1, \dots, n$
3) For $i = 1, \dots, n$: sample random values γ_i, and set $C_i = Com((f_i, r_i), \gamma_i)$
4) Set $d = (Q)ROM(C_1, \dots, C_n)$
5) Set $z(X) = r(X) + df(X)$ and $\pi = (z(X), C_1, \dots, C_n)$.
6) Publish $\pi = (C_1, \dots, C_n,\ z(X))$ and send privatly (f_i, γ_i) to participant V_i

(f_i, γ_i)

π

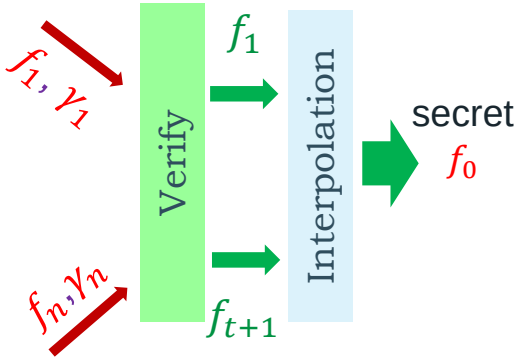
Shareholder V_i (f_i, γ_i) Verify



1) Party $V_i \rightarrow$ outputs accept/reject:
   - If $\deg(z(X)) \leq t$
   - Set $d = (Q)ROM(C_1, \dots, C_n)$
   - If $C_i == Com((f_i, z(i) - df_i), \gamma_i)$ outputs accept, otherwise, reject.
2) If V_j complains, dealer publish (f_j, γ_j), s.t., other parties can check them.
3) At the end, the honest parties either reject, or get sure that they have received a valid share.

✓ In practice we can omit γ_i if f_0 has enough entropy

Reconstruct



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

Security and Instantiations of Π:

Theorem

Let the (vector) commitment scheme $C((.,.),.)$ is perfectly (resp. computationally) hiding and computationally (resp. perfectly) binding, and RO is a random oracle. Then, the generic construction Π constitutes a secure VSS scheme in the honest-majority setting ($n \geq 2t+1$), that satisfies completeness, verifiability and perfect (resp. computational) simulatability.

Instantiations

- $C\big((m,r),\ \gamma\big) = g_1^m\ g_2^r\ g_3^\gamma \Rightarrow$ an alternative to Pedersen VSS [Ped91], named Π_P
  - Complete, Verifiable, and (Perfectly) Simulatable
- $C(m,r) = g_1^m\ g_2^r \Rightarrow$ an alternative to Feldman VSS [Fel87], named Π_F
  - Complete, Verifiable, and (Computationally) Unpredictable
  - Requires a high-entropy secret
- $C\big((m,r),\ \gamma\big) = Hash(m,r,\gamma) \Rightarrow$ an alternative to [ABCP23] VSS, named Π_{LA}
  - Complete, Verifiable, and Computationally Simulatable
  - In case of sharing a high-entropy secret, the randomizer γ can be omitted.
- ...

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

Performance of New VSS Schemes: Asymptotic Costs

- Abbreviations: DL: Discrete Logarithm, (Q)ROM: (Quantum) Random Oracle Model, BC: Broadcast, PV: Private, n: # parties, E_G: Exponentiation in G, PE: Degree-t polynomial evaluations, H: Hashing, $|X|$: X element size.

VSS Scheme	High-Entropy Secret?	Assumption	Sharing	Dealer's Communication	Verification	Reconstruction
Feldman [Fel87]	Yes	DL (Plain)	$0.5n \approx t\ E_G$ $n\ PE$	PV: $n\ \|Z_q\|$ BC: t $\approx 0.5n\ \|G\|$	$t\ E_G + t\ M_G$	$t^2\ E_G + t^2\ M_G$
$\mathbf{\Pi_F}$ $(C_i = g_1^{f_i} g_2^{r_i})$	**Yes**	**DL (ROM)**	$\mathbf{2n\ E_G}$ $\mathbf{2n\ PE}$	**PV: $n\ \|Z_q\|$** **BC: $n\ \|G\| + 0.5n\ \|Z_q\|$**	$\mathbf{2\ E_G + 1\ PE + 1\ H}$	$\mathbf{2t\ E_G + t\ PE + t\ H}$
Pedersen [Ped91]	No	DL (Plain)	$1n \approx 2t\ E_G$ $2n\ PE$	PV: $2n\ \|Z_q\|$ BC: t $\approx 0.5n\ \|G\|$	$t\ E_G + t\ M_G$	$t^2\ E_G + t^2\ M_G$
$\mathbf{\Pi_P}$ $(C_i = g_1^{f_i} g_2^{r_i} g_3^{\gamma_i})$	**No**	**DL (ROM)**	$\mathbf{3n\ E_G}$ $\mathbf{2n\ PE}$	**PV: $2n\ \|Z_q\|$** **BC: $n\ \|G\| + 0.5n\ \|Z_q\|$**	$\mathbf{3\ E_G + 1\ PE + 1\ H}$	$\mathbf{3t\ E_G + t\ PE + t\ H}$
[ABCP23]	Yes (or No)	Hash function (QROM)	$2n\ H$ $2n\ PE$	PV: $1n\ \|Z_q\|$ (or $2n\ \|Z_q\|$) BC: $2n\ \|H\| + 0.5n\ \|Z_q\|$	$1\ PE + 3\ H$	$t\ PE + 3t\ H$
$\mathbf{\Pi_{LA}}$ $(C_i = Hash(f_i, r_i, ~~\gamma_i~~)$	**Yes (~~or No~~)**	**Hash function (QROM)**	$\mathbf{n\ H}$ $\mathbf{2n\ PE}$	**PV: $1n\ \|Z_q\|$ (or $2n\ \|Z_q\|$)** **BC: $n\ \|H\| + 0.5n\ \|Z_q\|$**	$\mathbf{1\ PE + 2\ H}$	$t\ \mathbf{PE} + 2t\ \mathbf{H}$

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

Empirical Performance: [Ped91, ABCP23] vs. Π_P & Π_{LA}

- For both commitment and (quantum) random oracle, we used the SHA256 hash function.
- For the implementation of Pedersen and Π_P VSS schemes, we have used Curve 25519.
- Laptop: Ubuntu 22.04 LTS, a 11th Gen Intel(R) Core i9-11950H at base frequency 2.60GHz, & 128GB of memory.
- Our codes are available on github.com/Baghery/VSS-Pi as well as github.com/KULeuven-COSIC/Pi-VSS
- Can be run by scanning the QR-code

Table 2. Implementation results of VSS schemes Pedersen [23], Π_P, ABCP [1], and Π_{LA}. n: Number of parties, t: Threshold value, $|\mathbb{Z}_q| = |\mathbb{Z}_N| = |\mathbb{G}| = |\mathcal{H}| = 256$ bits.

(n, t)	Metrics	Pedersen [23]	Π_P	ABCP [1]	Π_{LA}	**Shamir**
(32, 15)	Share	74.8 msec	222.1 msec	2.2 msec	1.7 msec	0.7 msec (2.4×)
	Verify	10.7 msec	7.1 msec	0.13 msec	0.10 msec	
	Dealer's Broadcast	0.5 KB	1.5 KB	2.5 KB	1.5 KB	
	Dealer's Private Com.	2.0 KB	2.0 KB	1.0 KB	1.0 KB	
(128, 63)	Share	303 msec	852 msec	13.2 msec	11.4 msec	5.2 msec (2×)
	Verify	99.2 msec	8.3 msec	0.37 msec	0.33 msec	
	Dealer's Broadcast	2.0 KB	6.0 KB	10.0 KB	6.0 KB	
	Dealer's Private Com.	8.0 KB	8.0 KB	4.0 KB	4.0 KB	
(512, 255)	Share	1.29 sec	3.63 sec	0.13 sec	0.12 sec	2×
	Verify	552 msec	12.5 msec	1.2 msec	1.1 msec	
	Dealer's Broadcast	8.0 KB	24.0 KB	40.0 KB	24.0 KB	
	Dealer's Private Com.	32.0 KB	32.0 KB	16.0 KB	16.0 KB	
(2048, 1023)	Share	6.45 sec	16.07 sec	1.81 sec	1.78 sec	2×
	Verify	2.32 sec	28.3 msec	4.9 msec	4.8 msec	
	Dealer's Broadcast	32.0 KB	96.0 KB	160.0 KB	96.0 KB	
	Dealer's Private Com.	128.0 KB	128.0 KB	64.0 KB	64.0 KB	
(8192, 4095)	Share	47.1 sec	84.6 sec	28.6 sec	28.5 sec	2×
	Verify	9.38 sec	0.092 sec	0.020 sec	0.018 sec	
	Dealer's Broadcast	128 KB	384 KB	640 KB	384 KB	
	Dealer's Private Com.	512 KB	512 KB	256 KB	256 KB	
(16384, 8191)	Share	149.0 sec	221.69 sec	112.0 sec	111.5 sec	2×
	Verify	18.7 sec	0.178 sec	0.070 sec	0.050 sec	
	Dealer's Broadcast	256 KB	768 KB	1280 KB	768 KB	
	Dealer's Private Com.	1024 KB	1024 KB	512 KB	512 KB	

Generalizing DL Assumption Over Polynomials
&
Π_S: An Efficient Publicly Verifiable Secret Sharing (PVSS) Protocol Based on Π_F VSS Scheme

Publicly Verifiable Secret Sharing (PVSS):

- Roughly speaking, in a Shamir-based VSS scheme a Prover (so called dealer) who knows a secret degree-t polynomial $f(X)$ generates a threshold ZK proof for the following n-distributed relation,

$$R_i = \{(f_i, f(X)) \mid f(i) = f_i\} \quad \text{for } i = 1, \dots, n.$$

The threshold ZK proof is designated and can only be verified by at least $t + 1$ distinct values of "f_i"s.

- Common VSS schemes → have happy and unhappy paths → 1 (happy) or 3 (unhappy) rounds
- [Fel87, Ped91, ABCP23] and all the schemes built by **Π** → 1 (happy) or 3 (unhappy) rounds

➢ In a Shamir-based PVSS scheme, given the public keys $\{pk_i\}_{i=1}^n$, the Prover encrypt the Shamir shares and proves that the sharing and encryption are done correctly. E.g., it generates a NIZK proof for

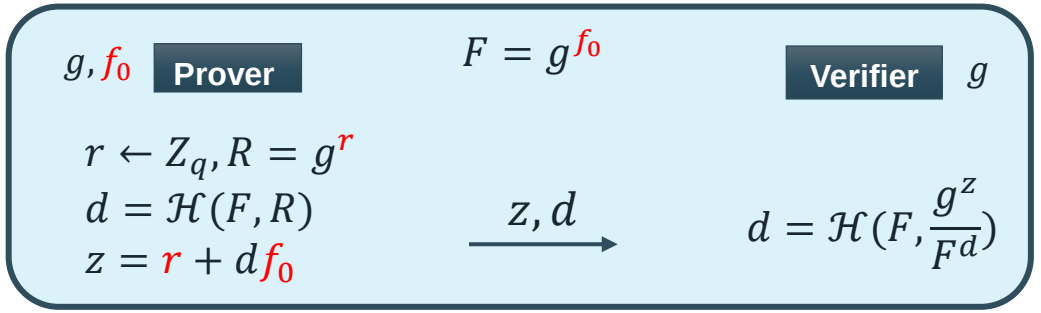
$$R_{PVSS} = \{(C_i, pk_i, f(X)) \mid C_i = Enc(f(i), pk_i)\} \text{ for } i = 1, \dots, n.$$

➢ PVSS schemes require a single (online) round and usually the secret is set to be either f_0 or g^{f_0}.

➢ Parties register their public keys $\{pk_i\}_{i=1}^n$ and store the secret keys for the reconstruction phase.

Generalizing DL Assumption Over Polynomials:

- **Discrete Logarithm (DL) Problem:** Given $(g, F = g^{f_0})$ its computationally hard to recover f_0.
- **Schnorr's $\sum$-Protocol:** Allows a prover to prove knowledge of f_0 for $R_{DL} = \{((g, F), f_0) \mid F = g^{f_0}\}$.



Polynomial Discrete Logarithm (PDL) Problem: Let (n, t) be integer values s.t., $n > t$, and $x_1, \ldots, x_n$ be public distinct elements from Z_q. Given $(g,\ x_i,\ F_i = g^{f(x_i)})$ for $i = 1, \ldots, n$, it's computationally hard to recover the secret polynomial $f(X) \in Z_q[X]_t$ of degree (up to) t.

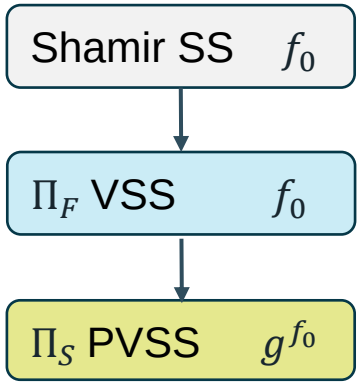
We show that the hardness of PDL problem can be reduced to that of DL problem.

DL ⇔ PDL

Generalizing Schnorr's $\sum$-Protocol: Allows a prover to prove knowledge of $f(X)$ for R_{PDL} defined as,

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

Π_S: An Efficient PVSS Scheme Based on Π_F VSS



- **Dealer**: In Π_S, given $(n, t, pk_1 = g^{s_1}, \dots, pk_n = g^{s_n}, f_0)$, the dealer acts as follows:
  - Does Shamir secret sharing using $f(X)$ and obtains the shares $\{f_i = f(i)\}_{i=1}^{n}$.
  - It sets $C_i = pk_i^{f_i}$ for $i = 1, \dots, n$. Then, it uses a variant of the generalized Schnorr protocol, and generates a NIZK proof π_{share}.
    - Samples $r(X)$ and Sets $\Gamma_i = pk_i^{r(i)}$ for $i = 1, \dots, n$.
    - Computes $d = RO(C_1, \dots, C_n, \Gamma_1, \dots, \Gamma_n)$ and $z(X) = r(X) + df(X)$.
    - Broadcasts $\pi_{share} := (z(X), d)$ as the proof and $C_i = pk_i^{f_i}$ for $i = 1, \dots, n$ as the encrypted shares.
- **Verifier**: Given $(n, t, pk_1, \dots, pk_n, C_1, \dots, C_n, z(X), d)$, a public verifier checks if the followings hold,

$$\deg\big(z(X)\big) \le t \quad \text{and} \quad d = RO\left(C_1, \dots, C_n, \frac{pk_1^{z(1)}}{C_1^d}, \dots, \frac{pk_n^{z(n)}}{C_n^d}\right)$$

- **Reconstruction**: As in [Sch99], a qualified set Q of parties decrypt their encrypted shares and obtain their shares $\left\{D_k := g^{f_k} = C_k^{1/s_k}\right\}_{k \in Q}$.
  They, also attach a proof for the correctness of decryption. Then, given any $t+1$ of valid shares $\{D_k = g^{f_k}\}_{k \in Q}$, one can obtain the secret g^{f_0} by Lagrange interpolation.

We show that the new PVSS scheme satisfies **Completeness**, **Verifiability** and **Secrecy** under PDL and DDH assumptions in the RO model.

Performance of Π_S & Implementations by Dock Network:

PPVSS	Assumption	Sharing	Dealer's Com.	Verification	Reconstruction
[Sch99]	DDH, RO	$4.5n\ E_G + n\ PE$	$1.5n\ \|G\| + n\ \|Z_q\|$	$nt + 4n\ E_G + 2.5n\ M_G$	$5t\ E_G + t\ M_G$
[CD17]	DBS, Plain	$2n\ E_G + n\ PE$	$2n\ \|G\|$	$2n\ P_G + n\ E_G + n\ M_G$	$2t\ P_G + t\ E_G + t\ M_G$
[CD17]	DDH, RO	$4n\ E_G + n\ PE$	$2n\ \|G\| + n\ \|Z_q\|$	$5n\ E_G + 3n\ M_G$	$5t\ E_G + t\ M_G$
Π_S , [CD22]	DDH, PDL, RO	$2n\ E_G + 2n\ PE$	$n\ \|G\| + 0.5n\ \|Z_q\|$	$2n\ E_G + n\ PE + n\ M_G$	$5t\ E_G + t\ M_G$



✓ **The PVSS scheme Π_S and our proposed VSS scheme Π_{LA} was recently implemented in Rust by the developers of Dock Network, www.dock.io,**

✓ Their open-source implementations for Π_S are publicly available on:

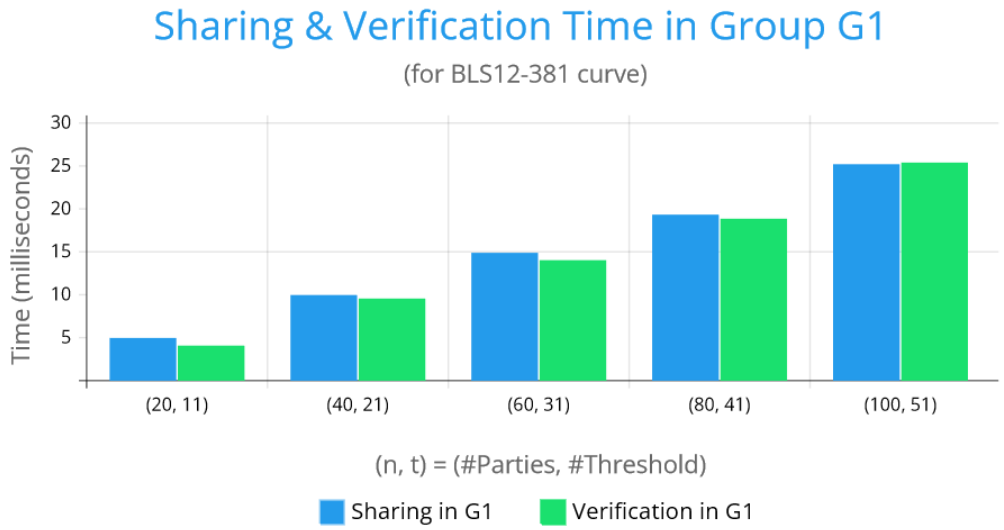
https://github.com/docknetwork/crypto/tree/main/secret_sharing_and_dkg/src/baghery_pvss

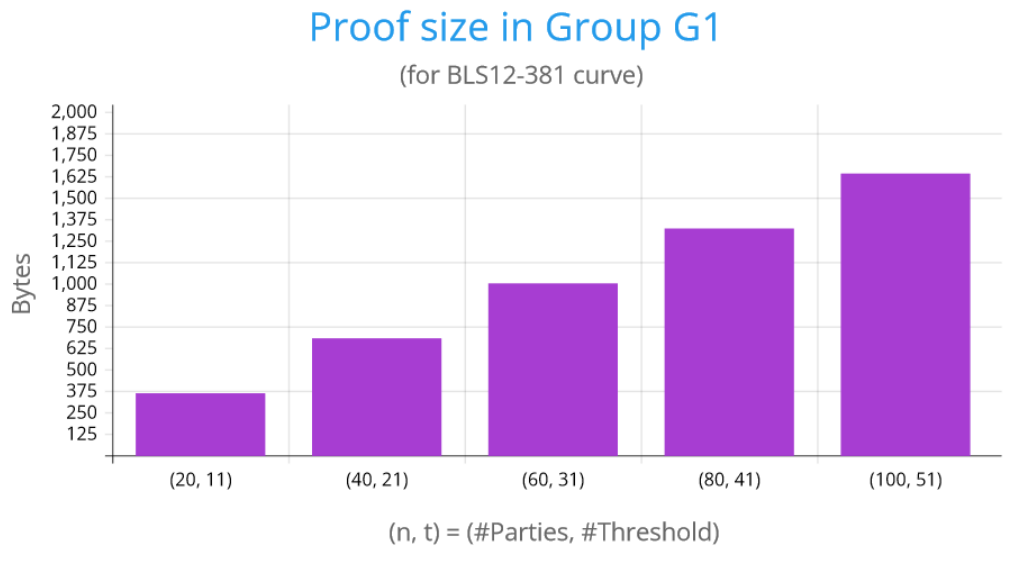
✓ Their open-source implementations for Π_{LA} are publicly available on:

https://github.com/docknetwork/crypto/blob/main/secret_sharing_and_dkg/src/baghery_vss.rs

Performance of Π_S: Run by the Developers of Dock Network

- ✓ **Performance of Π_S over BLS12-381 curve (by Dock Network, www.dock.io)**
  https://github.com/docknetwork/crypto/tree/main/secret_sharing_and_dkg/src/baghery_pvss





Made with Livegap Charts

"This primitive is useful for sharing secrets on a blockchain as the blockchain can verify the proof."

— Git Page of DOCK Network

Conclusions and Future Research:

- We presented a simple and general construction to build computational VSS schemes in the honest-majority setting.
- It is general engough to be employed in various contexts such as lattices, isogenies, codes, and so on.
- New VSS schemes can be used in various threshold protocols
  - In many cases, Π_F and Π_{LA} can be used instead of Feldman VSS.
  - Similarly, Π_P can be used instead of Pedersen VSS.
  - Π_{LA} is PQ-secure and assomptetically only 1x slower than standard Shamir!
- New PVSS scheme Π_S can be used in various applicaitons.
  - Recently, we used a variant of it to revisit various applicaitons presented in [Sch99], and improved their performance. Our new results will be presented at ACNS 2025 (ia.cr/2025/576). New!
- In practice, when using the new VSS schemes, one may need to construct a variant of them that commits to the main secret as well, which can be achieved in all schemes with minimal modification. See section 4 of [ABCP23] as an example.



KU LEUVEN

Thank You!



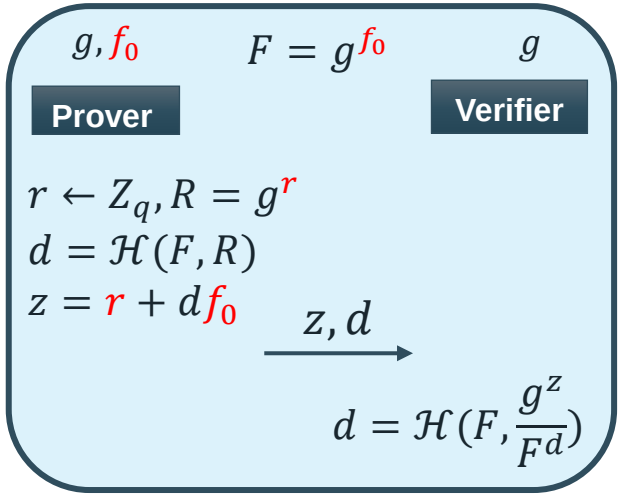
ia.cr/2023/1669

karim.baghery@kuleuven.be

An Efficient NIZK Proof of Knowledge for R_{PDL}:

Generalization of Schnorr's Protocol:

It allows a prover to prove knowledge of a witness polynomial $f(X)$ for the R_{PDL} relation:



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

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

1. Sample a degree-t polynomial $r(X) \in \mathbb{Z}_q[X]_t$; Set $\{\Gamma_i = g^{r(x_i)}\}_{i=1}^n$.
2. Set $d \leftarrow \mathcal{H}(F_1, \ldots, F_n, \Gamma_1, \ldots, \Gamma_n)$, where $\mathcal{H}$ is a random oracle.
3. Set $z(X) = r(X) + d \cdot f(X) \pmod q$;
4. Return $\pi := (\Gamma_1, \ldots, \Gamma_n, z(X))$

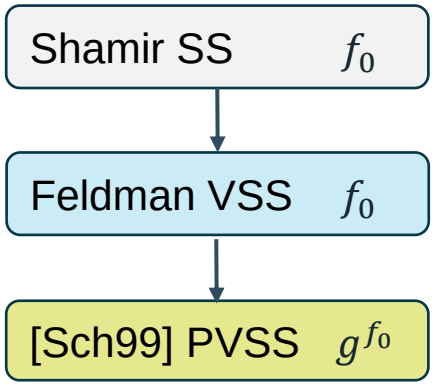
Verifier: Given statement $(g, \{x_i, F_i\}_{i=1}^n)$ and $\pi := (\Gamma_1, \cdots, \Gamma_n, z(X))$, the verifier first checks if $z(X)$ is a degree-t polynomial. If so, then sets $d \leftarrow \mathcal{H}(F_1, \ldots, F_n, \Gamma_1, \ldots, \Gamma_n)$ and checks if: $g^{z(x_i)} = \Gamma_i (F_i)^d$ for $i = 1, \ldots, n$, and outputs `true` or `false`. Note that to make the communication shorter, as in Schnorr's ID protocol, alternatively, the prover could publish $\pi := (d, z(X))$, and then the verifier would need to check if $z(X)$ is a degree-t polynomial and $d = \mathcal{H}(F_1, \ldots, F_n, \frac{g^{z(x_1)}}{F_1^d}, \ldots, \frac{g^{z(x_n)}}{F_n^d})$.

We prove that the generalized protocol satisfies **Completeness**, **Special Sound**, **HVZK** under PDL problem, and can be made non-interactive in the RO model.



Schoenmakers' PVSS Scheme:

- In some PVSS schemes the main secret is g^{f_0} [Sch99] and in some others it is $f_0 \in Z_q$ [Gro21].
- In CRYPTO'99, Schoenmakers [Sch99]:
  - Proposed an efficient PVSS protocol based on [Fel87] VSS scheme.
  - In his PVSS scheme, the secret is g^{f_0} and parties obtain g^{f_i} as a share of g^{f_0}.
  - In his scheme, given the public key $\{pk_i = g^{s_i}\}_{i=1}^{n}$ and the secret value f_0, the dealer runs Feldman VSS and obtains the shares $\{f_i\}_{i=1}^{n}$ and the proof $\pi_{Feldman}$. Then, the prover encrypts the shares as $C_i = pk_i^{f_i}$ and generates another NIZK proof that the encrypted values are valid Shamir shares.



- He also introduced several practical applications for his PVSS scheme, e.g., a universally verifiable e-voting protocol, threshold PKI, …

PVSS Scheme	Assumption	Sharing	Dealer's Communication	Verification	Reconstruction
[Sch99]	DDH & RO	$4.5n\ E_G + n\ PE$	PV: – BC: $1.5n\ \|G\| + n\ \|Z_q\|$	$nt + 4n\ E_G + 2.5n\ M_G$	$5t\ E_G + t\ M_G$

[Sch99] Berry Schoenmakers. A Simple Publicly Verifiable Secret Sharing Scheme and its Application to Electronic Voting. CRYPTO'99.