

PiVer: Π Verifiable Secret Sharing Framework

Shahla Atapoor¹

Hossein Moghaddas¹

Jannik Spiessens¹

Karim Baghery¹

Georgio Nicolas¹

Barry Van Leeuwen¹

Daniele Cozzo²

Robi Pedersen⁴

Robin Jadoul³

Mahdi Rahimi¹



✉ piver@esat.kuleuven.be

✉ karim.baghery@kuleuven.be

1



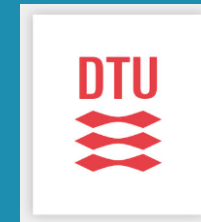
2



3

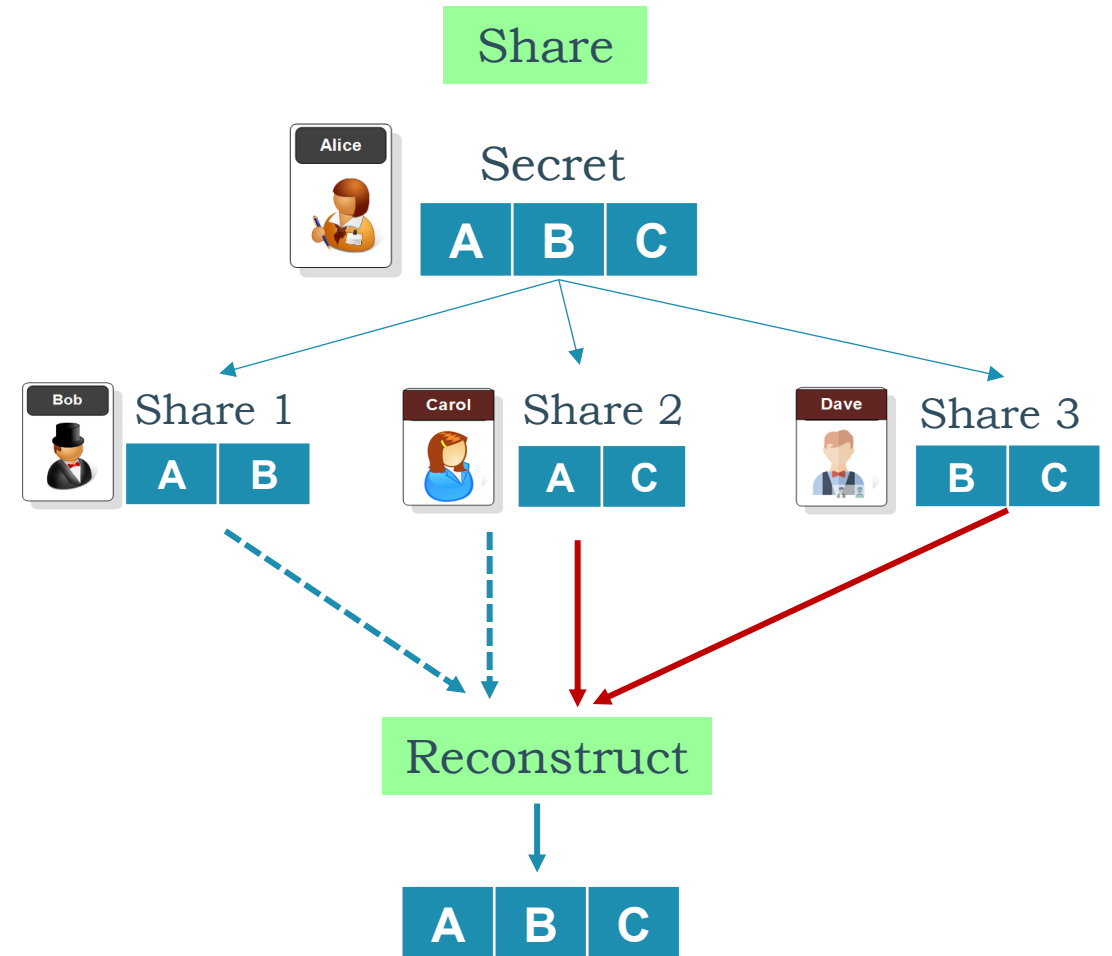


4



Motivation: Why Secret Sharing?

- Single secret → Single point of failure
- Threshold cryptography
 - Distribute trust among multiple parties
 - Robust storage: loss of individual shares does not prevent recovery
- Applications
 - Distributed Key Generation (DKG)
 - Threshold Signatures
 - Threshold Decryption
 - Multi-Party Computation (MPC)
 - Randomness Beacons
 - ...

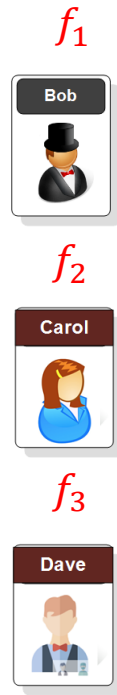
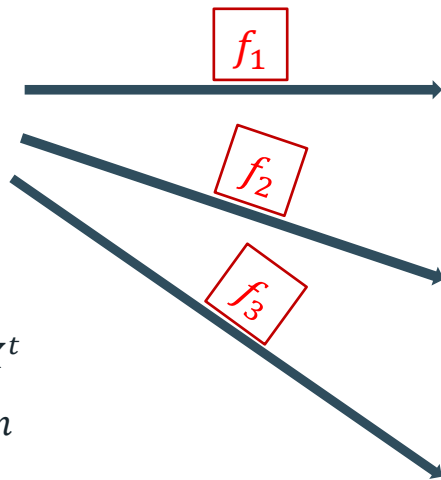


Shamir Secret Sharing: A Toy Example & Trust Issue

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

Not Verifiable!

Share



- Given a secret f_0 and (n, t)
- Sample a degree t polynomial

$$f(X) = f_0 + a_1X + \dots + a_tX^t$$
- Compute $f_i = f(i)$ for $i = 1, \dots, n$

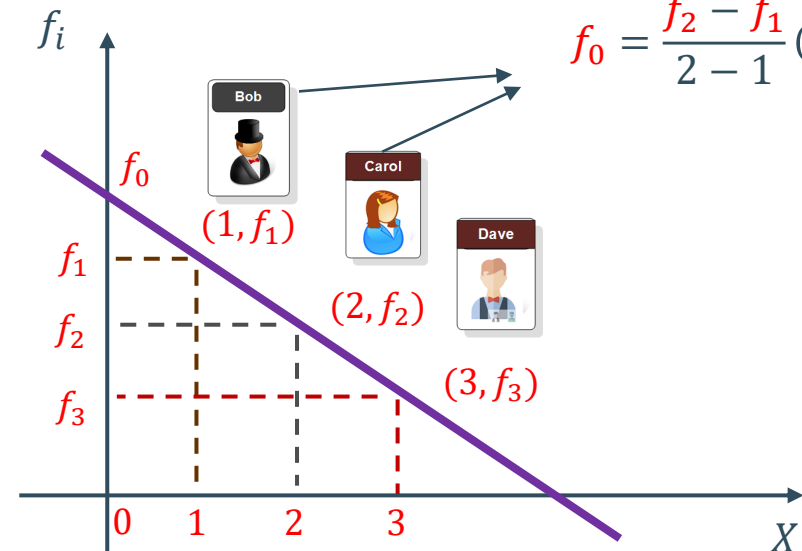
➤ E.g., for $(n, t) = (3, 1)$

Not Verifiable!

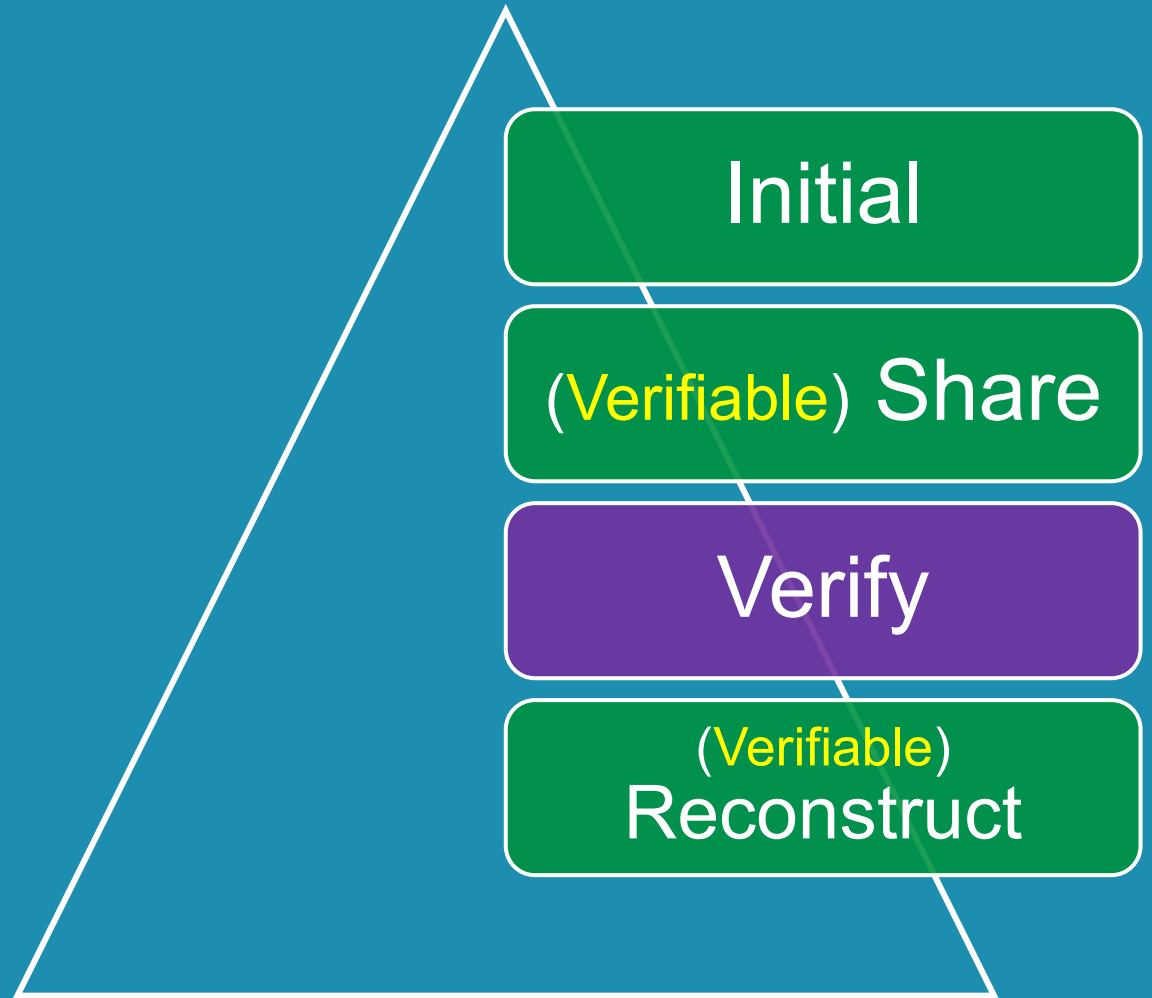
Reconstruction

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

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



Verifiable Secret Sharing



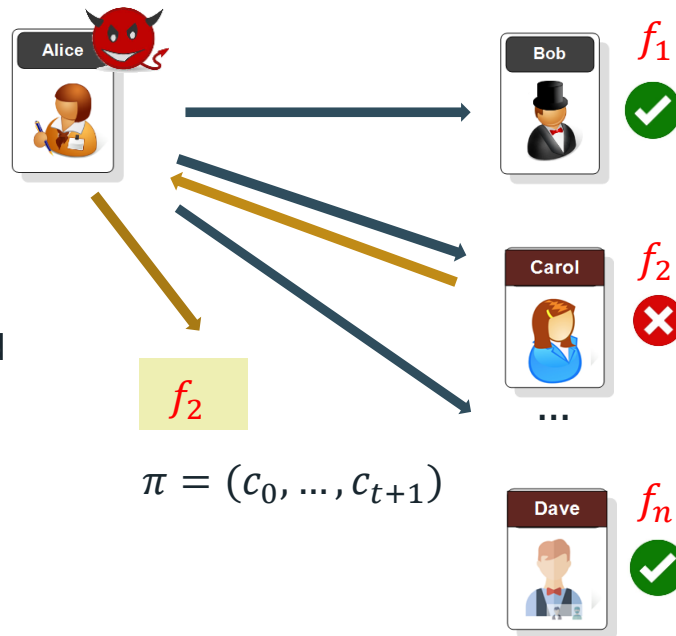
VSS from Homomorphic Commitments: Feldman [Fel87]

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

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

Share

- Given a secret f_0 and (n, t)
- Sample a degree t polynomial
 $f(X) = f_0 + a_1X + \dots + a_tX^t$
- Compute $f_i = f(i)$ for $i = 1, \dots, n$
- Set $c_0 = g^{f_0}, c_1 = g^{a_1}, \dots, c_t = g^{a_t}$
- Broadcast $\pi = (c_0, \dots, c_{t+1})$ and send privately f_i to party P_i



Verify

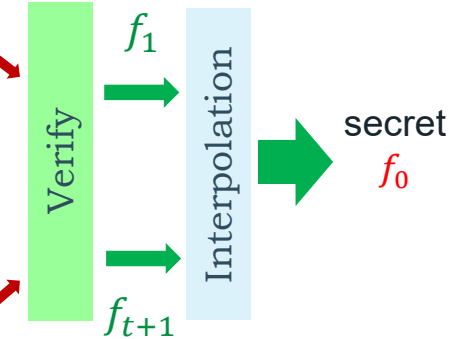
$$g^{f_i} = \prod_{j=0}^t c_j^{i^j}$$

$$\begin{aligned} g^{f_1} &= c_0^{1^0} \times c_1^{1^1} \times \dots \times c_t^{1^t} \\ &= g^{f_0} \times g^{a_1} \times \dots \times g^{a_t} \\ &= g^{f(1)} \end{aligned}$$

$$\begin{aligned} g^{f_2} &= c_0^{2^0} \times c_1^{2^1} \times \dots \times c_t^{2^t} \\ &= g^{f_0} \times g^{2a_1} \times \dots \times g^{2^t a_t} \\ &= g^{f(2)} \end{aligned}$$

$$\begin{aligned} g^{f_n} &= c_0^{n^0} \times c_1^{n^1} \times \dots \times c_t^{n^t} \\ &= g^{f_0} \times g^{na_1} \times \dots \times g^{n^t a_t} \\ &= g^{f(n)} \end{aligned}$$

Reconstruct



Add. Homomorphic Commitments:
 $com(a_1) \times com(a_2) = com(a_1 + a_2)$
 e.g., $g^{a_1} \times g^{a_2} = g^{a_1+a_2}$

VSS from Homomorphic Commitments: Pedersen [Ped91]

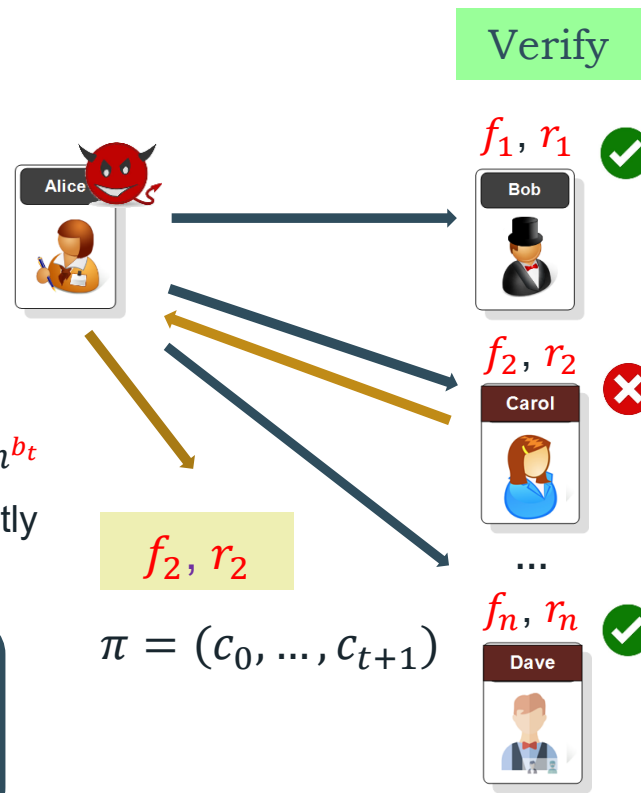
- Pedersen VSS: t of shareholders (information theoretically) learns nothing about f_0
- Pedersen Commitment: $com(f, r) = g^f h^r$ (Perfectly Hiding + Computationally Binding)

Share

- Given a secret f_0 and (n, t)
- Sample a degree t polynomial $f(X) = f_0 + a_1X + \dots + a_tX^t$
- Sample another degree t polynomial $r(X) = r_0 + b_1X + \dots + b_tX^t$
- Set $f_i = f(i)$ and $r_i = r(i)$ for $i = 1, \dots, n$
- Sets $c_0 = g^{f_0} h^{r_0}, c_1 = g^{a_1} h^{b_1}, \dots, c_t = g^{a_t} h^{b_t}$
- Broadcast $\pi = (c_0, \dots, c_{t+1})$ and send privately (f_i, r_i) to party P_i

Pedersen Commitment:

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



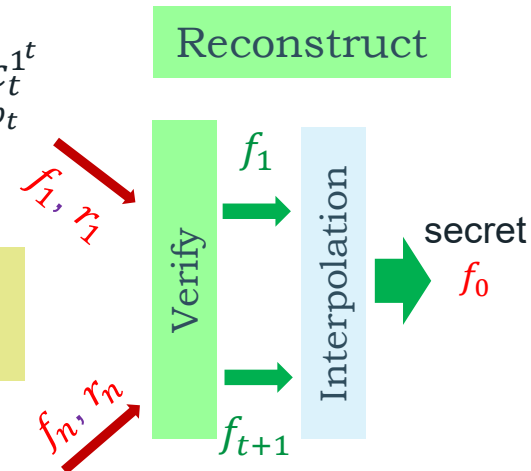
Verify

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

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

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

Reconstruct



Limitations of VSS from Homomorphic Commitment (HC)

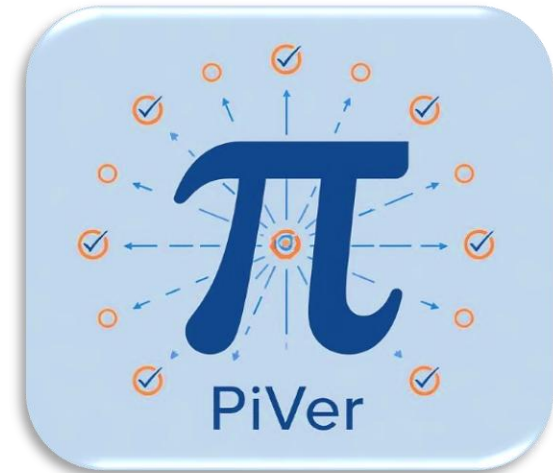
&



Limitations of HC-Based VSS and PiVer:

- Tied to homomorphic commitments
 - Strong algebraic assumptions, limited flexibility
- Hard to extend to Post-Quantum (PQ) settings
 - Heavy reliance on discrete-log-style structure
- Inefficient Verification
 - Single share verification requires $O(t)$ exp. per party
 - Reconstruction & Complaint handling requires $O(t^2)$ exp.
- Poor scalability
 - Not efficient for batching or high-throughput settings
- Often ad-hoc / non-modular
 - Difficult to adapt, reuse, or compose cleanly

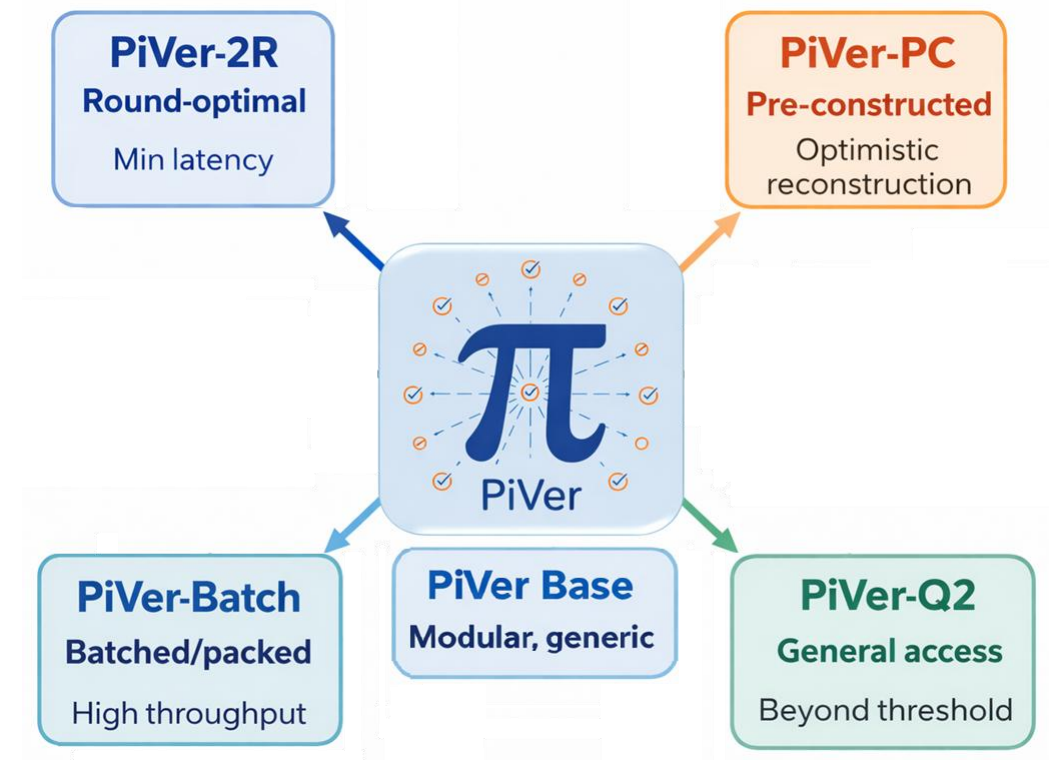
PiVer: Π Verifiable Secret Sharing Framework



A unified and modular framework for building computational VSS protocols in the synchronous setting.

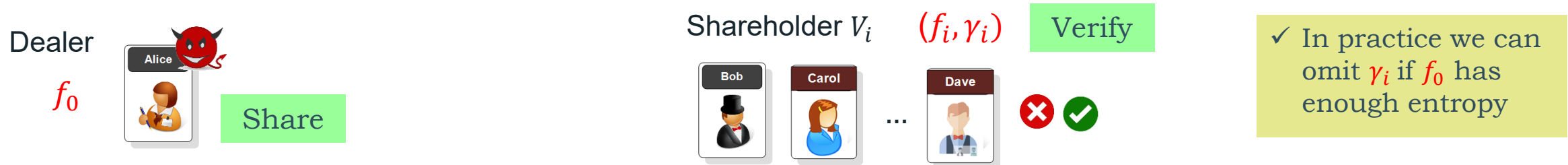
PiVer Submission:

- We aim to submit a unified and modular framework for constructing multiple variants of computationally secure VSS schemes in the synchronous setting (Category S7).
- Primitive Requirements
 - A secure commitment scheme (optionally homomorphic, non-malleable and/or *post-quantum* secure)
 - A (classic or quantum) random oracle
- Setting (as in [Fel87, Ped91])
 - Honest-majority ($n \geq 2t + 1$)
 - Synchronous network
 - Private channel between the dealer and parties
 - Common broadcast medium

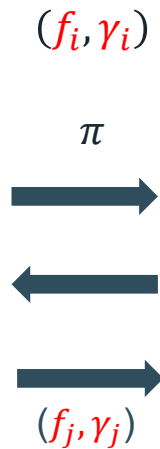


PiVer: Π Verifiable Secret Sharing Framework [ABCP23, Bag25]

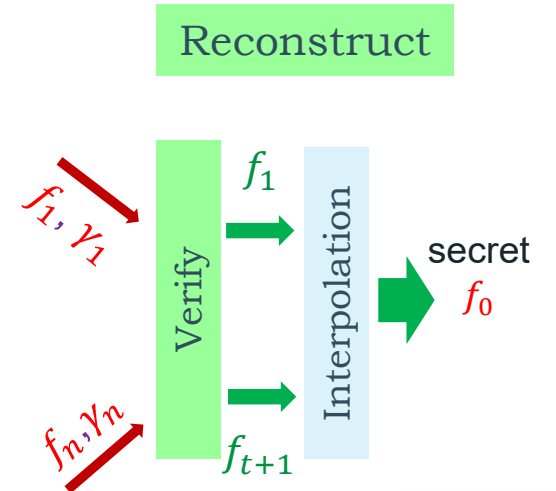
- Let $Com(.,.)$ be a (vector) commitment scheme, and $(Q)ROM$ be a (Quantum) ROM



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



- 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.
- If V_j complains, dealer publish (f_j, γ_j) , s.t., other parties can check them.
- At the end, the honest parties either reject, or get sure that they have received a valid share.



Security and Instantiations of base PiVer (Π):



Theorem [Bag25, ABCP23]

Let the (vector) commitment scheme $Com((\cdot, \cdot), \cdot)$ is perfectly (resp. computationally) hiding and computationally (resp. perfectly) binding, and ROM 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

- $Com(m, r) = g_1^m g_2^r \Rightarrow$ an alternative to Feldman VSS [Fel87], **named Π_F**
 - Complete, Verifiable, and (Computationally) Unpredictable
 - Requires a high-entropy secret
- $Com((m, r), \gamma) = g_1^m g_2^r g_3^\gamma \Rightarrow$ an alternative to Pedersen VSS [Ped91], **named Π_P**
 - Complete, Verifiable, and (Perfectly) Simulatable
 - Can be slightly improved by setting $Com((m, r), \gamma) = g_1^{Hash(m,r)} g_2^\gamma$, named Π_{P+}
- $Com((m, r), \gamma) = Hash(m, r, \gamma) \Rightarrow$ a synchronous alternative to [SS24] VSS, **named Π_{LA}**
 - Complete, Verifiable, and Computationally Simulatable
 - In case of sharing a high-entropy secret, the randomizer γ can be omitted.
- ...



Efficiency 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 $	$t E_G + t M_G$	$t^2 E_G + t^2 M_G$
Π_F ($C_i = g_1^{f_i} g_2^{r_i}$)	Yes	DL (ROM)	$2n E_G$ $2n PE$	PV: $n Z_q $ BC: $n G + 0.5n Z_q $	$2 E_G + 1 PE$ $+ 1 H$	$2t E_G + t PE + t H$
Pedersen [Ped91]	No	DL (Plain)	$1n \approx 2t E_G$ $2n PE$	PV: $2n Z_q $ BC: $t \approx 0.5n G $	$t E_G + t M_G$	$t^2 E_G + t^2 M_G$
Π_P ($C_i = g_1^{f_i} g_2^{r_i} g_3^{y_i}$)	No	DL (ROM)	$3n E_G$ $2n PE$	PV: $2n Z_q $ BC: $n G + 0.5n Z_q $	$3 E_G + 1 PE$ $+ 1 H$	$3t E_G + t PE + t H$
Π_{P+} ($C_i = g_1^{Hash(f_i, r_i)} g_2^{y_i}$)	No	DL (ROM)	$2n E_G, n H$ $2n PE$	PV: $2n Z_q $ BC: $n G + 0.5n Z_q $	$2 E_G + 1 PE$ $+ 2 H$	$2t E_G + t PE$ $+ 2t H$
Π_{LA} ($C_i = Hash(f_i, r_i, \gamma_i)$)	Yes (or No)	Hash function (QROM)	$n H$ $2n PE$	PV: $1n Z_q $ (or $2n Z_q $) BC: $n H + 0.5n Z_q $	$1 PE + 2 H$	$t PE + 2t H$

Variants of PiVer:

PiVer-2R
Round-optimal
Min latency

- PiVer-2R: Round-optimal variant [BBKR25]
 - Requires 2 rounds (instead of 3) in the general case
 - Provably optimal for computational VSS [BKP11]

PiVer-Batch
Batched/packed
High throughput

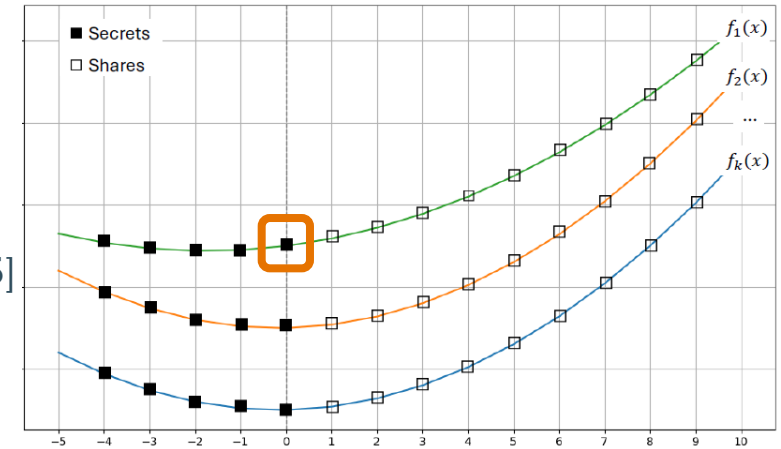
- PiVer-Batch: k -Batched & ℓ -Packed variant [ABNPS25]
 - Allows to share $k \times \ell$ secrets in a single execution
 - Significantly more efficient compared to the base

PiVer-PC
Pre-constructed
Optimistic reconstruction

- PiVer-PC: Pre-Constructed variant [BM26, BKNR25]
 - Dealer also publishes a commitment C_0 to the secret
 - Supports two reconstruction approaches

PiVer-Q2
General access
Beyond threshold

- PiVer-Q2: Generalized variant [ABJV25]
 - Supports arbitrary $Q2$ access structures
 - E.g., Replicated and Shamir secret sharing.



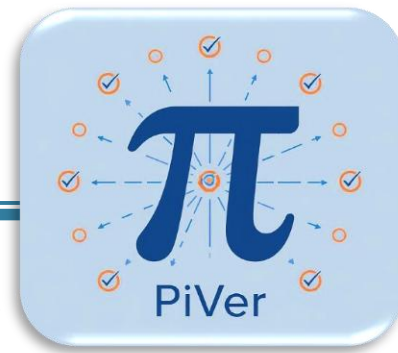
$$C_0 = \text{Com}(f_0) \cap f_i = f(i) \quad \text{for } i \in [1, n]$$

Optimistic Rec.

Pessimistic Rec.

- Recall that an access structure is said to be $Q2$ if no union of two unqualified sets covers the entire set of parties. (i.e., honest-majority in the threshold setting)

Instantiations of Different Variants of PiVer:



- We instantiate different variants of PiVer with DL- and Hash-based commitments.
 - In all variants we require only a hiding and binding commitment scheme.
 - The random oracle is instantiated with a secure hash function.

PiVer-2R
Round-optimal
Min latency

PiVer-Batch
Batched/packed
High throughput

PiVer-PC
Pre-constructed
Optimistic reconstruction

PiVer-Q2
General access
Beyond threshold

PiVer-DL

- Instantiations with DL-based commitments
 - $(\Pi_F, \Pi_P, \Pi_{P+}), (B\Pi_F, B\Pi_P, B\Pi_{P+}), (Q2\Pi_F, Q2\Pi_P, Q2\Pi_{P+}), (PC\Pi_F, PC\Pi_{LA}), (2R\Pi_F, 2R\Pi_P, 2R\Pi_{P+})$
 - Currently, PiVer-PC is constructed for $C_0 = g^{f_0}$

$$C_0 = g^{f_0} \cap f_i = f(i) \text{ for } i \in [1, n]$$

PiVer-PQ

- Instantiations with hash-based commitments
 - $\Pi_{LA}, B\Pi_{LA}, Q2\Pi_{LA}, 2R\Pi_{LA}$

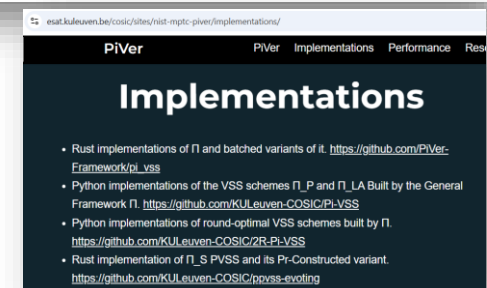
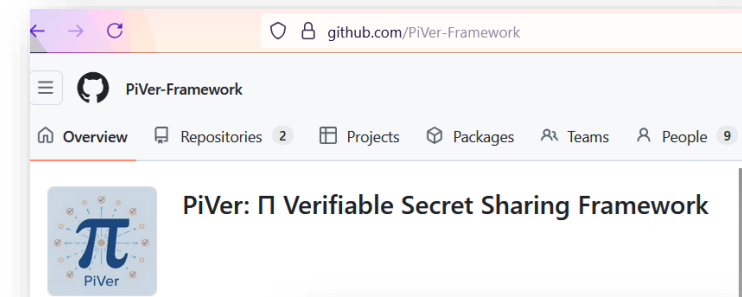
Package Specification and Implementation:



- Package Specification
 - Modular and formal specification
 - Security goals: Correctness, Reconstructability, Unpredictability or Simulatability
 - Supports both classical & PQ instantiations
 - Guidelines for parameters and deployments
- Implementation
 - Pure Rust implementation (with minimal dependency on external crates)
 - Currently, we assume reliable transmission as we run our tests on a single machine.
 - We will provide an open-source implementation on PiVer webpage as well as <https://github.com/PiVer-Framework>

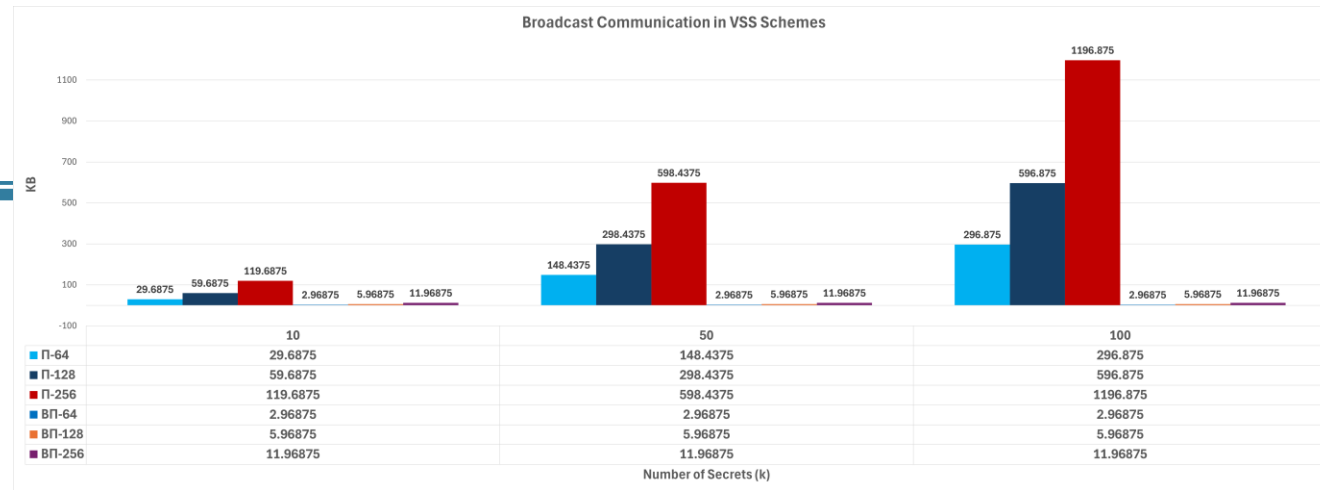


<https://www.esat.kuleuven.be/cosic/sites/nist-mptc-piver/>

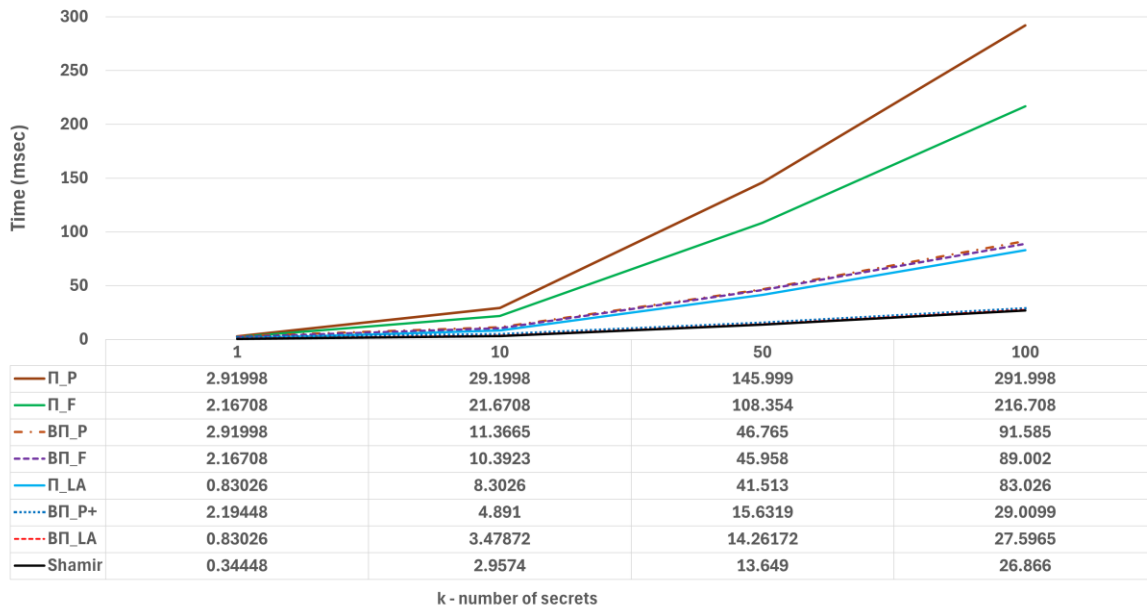


Performance:

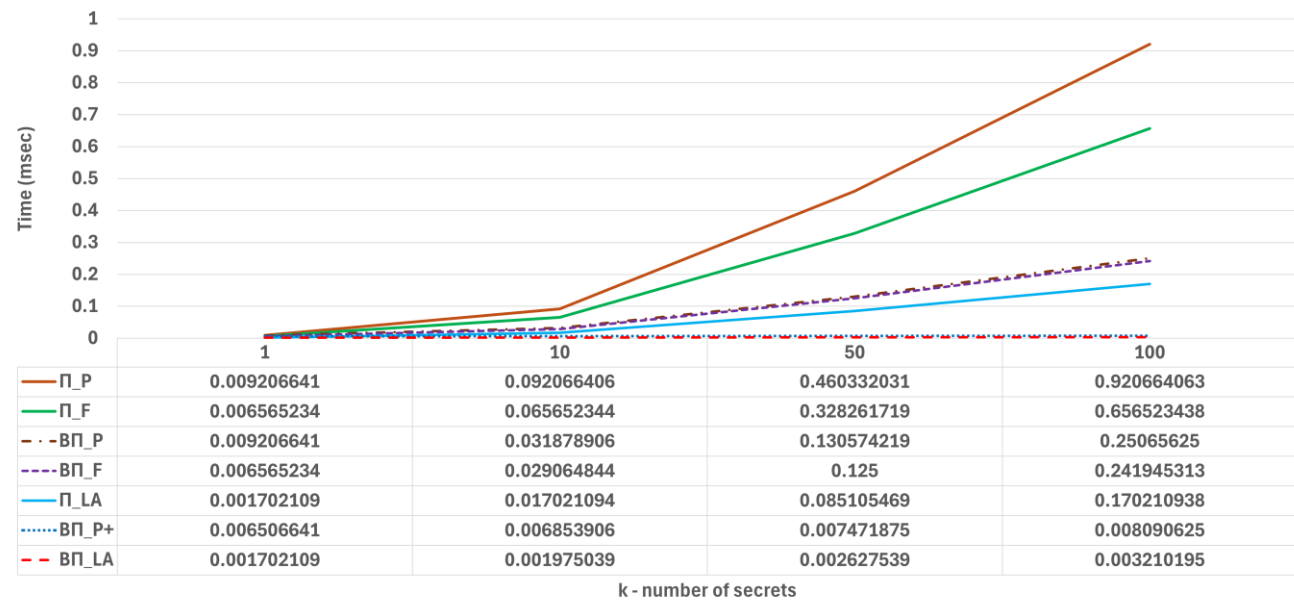
- Benchmark for $(n, t) = (256, 127)$
 - MacBook Pro, M4 Pro CPU, 24GB RAM, Single thread
 - BP_{LA} : Practical, Scalable, PQ-secure



Dealer's Sharing Time for $(n, t)=(256, 127)$

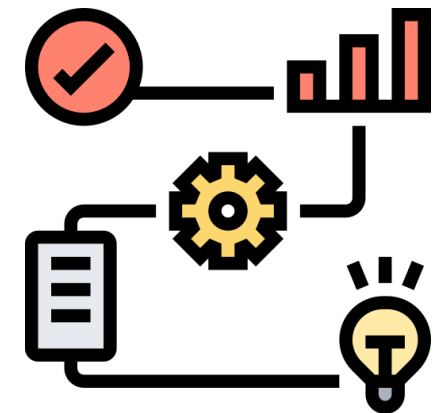
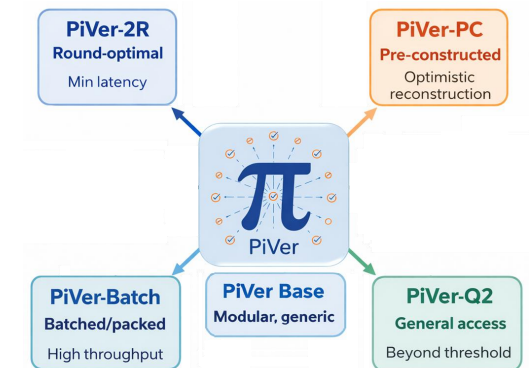


A Party's Verification Time for $(n, t)=(256, 127)$

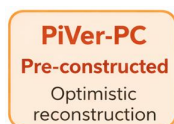
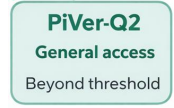


Conclusions and RoadMap:

- PiVer is a modular gadget for threshold cryptography (Cat. S7)
 - Can be integrated to various (PQ-secure) threshold schemes (DKG, Threshold Protocols, ...)
 - Designed for extensibility; supports rich variants out of the box:
 - Round optimal, batched-and-packed, pre-constructed, generalized access structures, publicly verifiable, ...
 - *Already adapted by some NIST submissions (e.g., isogeny- & lattice-based ones) and we expect to be reusable by more candidates*
- RoadMap & Plans
 - Prepare and submit the final PiVer package
 - Extend PiVer-PC to PQ-secure commitments
 - Optimize protocol for targeted use cases (e.g., DKGs, ...)
 - Expand implementations, benchmarks, and protocol integrations



Some of Key References:



- [ABCP23] Shahla Atapoor, Karim Baghery, Daniele Cozzo, Robi Pedersen. VSS from Distributed ZK Proofs and Applications. ASIACRYPT 2023. [PDF](#)
- [Bag25] Karim Baghery. Π : A Unified Framework for Computational Verifiable Secret Sharing. PKC 2025. [PDF](#)
- [BBKR25] Karim Baghery, Navid Ghaedi Bardeh, Shahram Khazaei and Mahdi Rahimi. On Round-Optimal Computational VSS. IACR Communications in Cryptology, vol. 2, no. 2, Jul 07, 2025. [PDF](#)
- [ABNPS25] Shahla Atapoor, Karim Baghery, Georgio Nicolas, Robi Pedersen, Jannik Spiessens. Batched and Packed (Publicly) Verifiable Secret Sharing: A Unified Framework and Applications. IACR Eprint. [PDF](#)
- [ABJV25] Shahla Atapoor, Karim Baghery, Robin Jadoul, Barry van Leeuwen. On Computational VSS for General Access Structures. IACR Eprint. [PDF](#)
- [BKNR25] Karim Baghery, Noah Knapen, Georgio Nicolas, Mahdi Rahimi. Pre-Constructed Publicly Verifiable Secret Sharing and Applications. ACNS 2025. [PDF](#)
- [BM26] Karim Baghery, Hossein Moghaddas. Fully Secure DKG Protocols for Discrete Logarithm Revisited. IACR Eprint. [PDF](#)
- [Ped92] Torben Prys Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. CRYPTO 91.
- [Fel87] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In 28th Annual Symposium on Foundations of Computer Science (SFCS 1987).
- [Sha79] Adi Shamir. How to share a secret. Communications of the ACM, 22(11):612–613, November 1979.

Thank You!



karim.baghery@kuleuven.be



piver@esat.kuleuven.be



<https://www.esat.kuleuven.be/cosic/sites/nist-mptc-piver/>



<https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/PiVer-PW01.pdf>