KU LEUVEN

The IMDEA Software Institute
(08 Nov 2023)

VSS from Distributed ZK Proofs and Applications

Karim Baghery
Postdoctoral Researcher, COSIC, KU Leuven, Belgium
karim.baghery@kuleuven.be

with Shahla Atapoor, Daniele Cozzo, Robi Pedersen

Asiacrypt'23
ia.cr/2023/992



Outline:

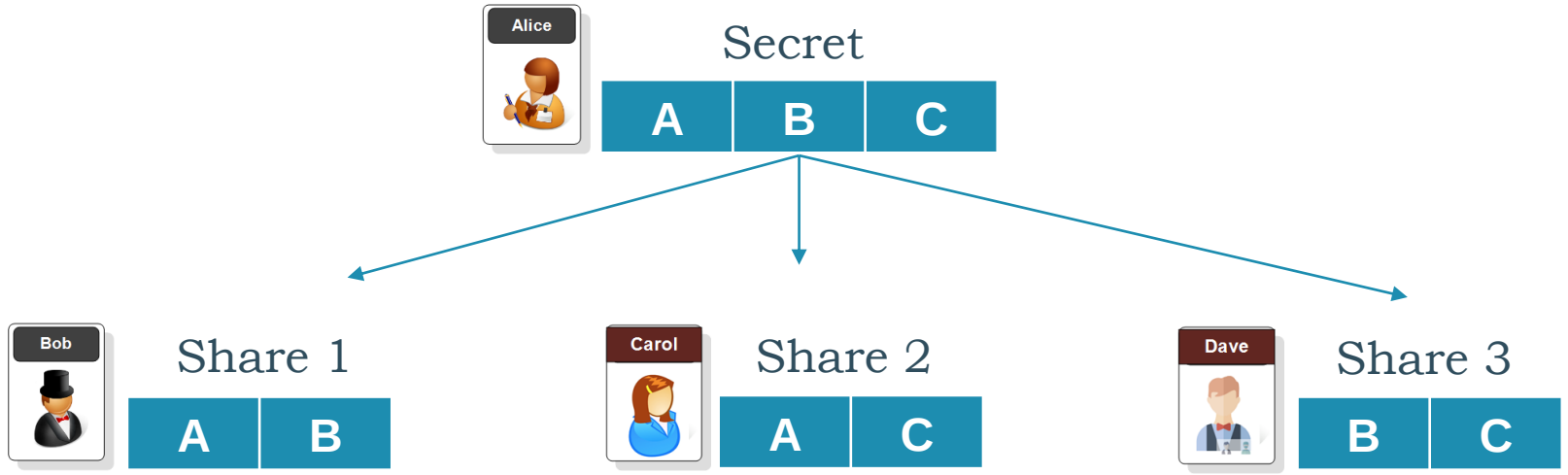
- Introduction
  - Shamir Secret Sharing
  - Verifiable Secret Sharing (VSS) from Homomorphic Commitments
  - Threshold ZK Proofs Over Secret Shared Data
- Our Contribution
  - VSS from ZK Proofs Over Shared Data
    - An Efficient NI-TZK Proof Scheme for Shamir Relation (in the QROM)
    - A (PQ-Secure) VSS from the Proposed NI-TZK Proof Scheme
    - Empricial Performance and Comparisons
  - More Efficient Threshold Protocols in the DL Setting
    - An Efficient Variant of Pedersen DKG and GJKR Threshold Signature
  - More Efficient Threshold Protocols from Isogenies
    - An Efficient Variant of ABCP DKG and Campos-Muth Threshold Signature



Secret Sharing: Motivation

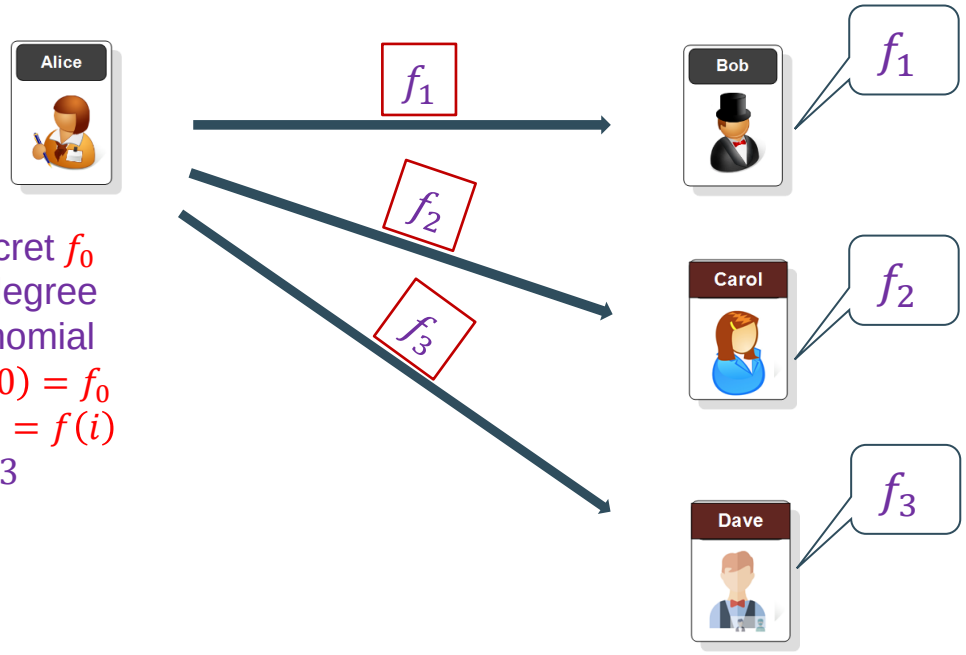
- In 2019, two different exchanges in Canada and Korea lost their secret keys and lost $150 million and 2.5 billion Won worth of cryptocurrency forever!

✓ **If the exchange's private key had been backed up or recovered, this disaster would have been prevented!**



Shamir Secret Sharing: An Example

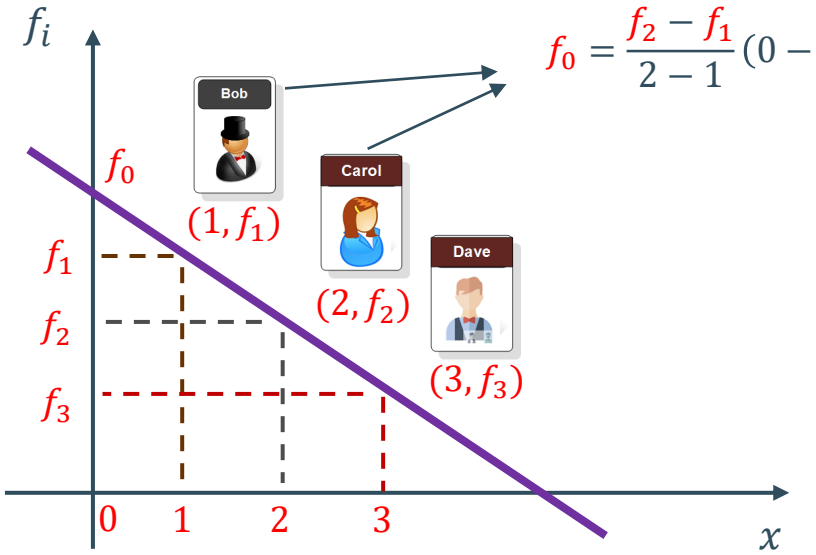
- Given a secret f_0 a trusted dealer P_0 shares it with $n \geq 2t+1$ parties $\{P_i\}_{i=1}^{n}$, s.t.,
  - Any set of up to $t+1$ shareholders can reconstruct the secret f_0
  - Any set of up to t shareholders cannot reconstruct the secret f_0



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

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

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

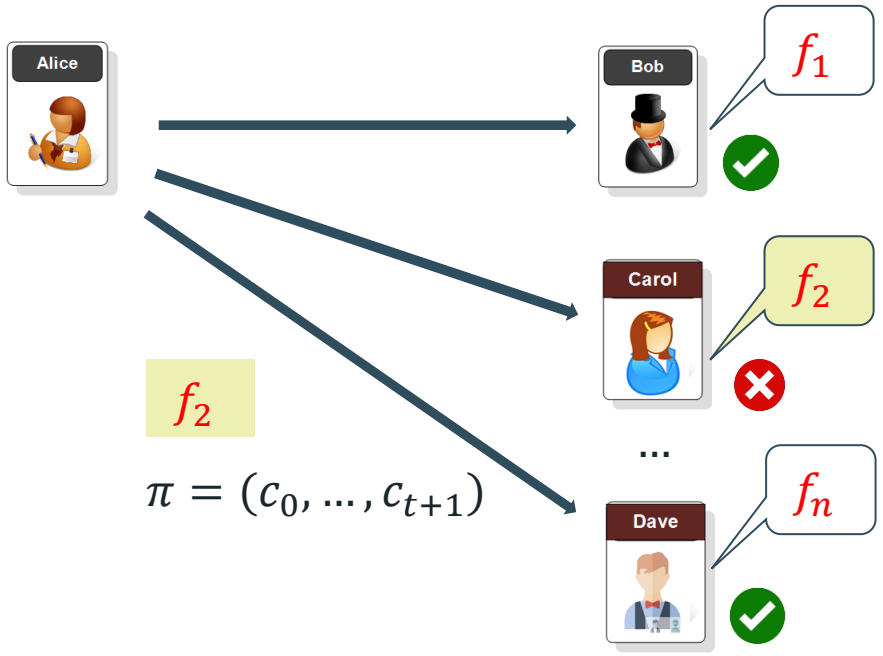


VSS from Homomorphic Commitments: Feldman [Fel87]

- 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.,
  - Any up to $t+1$ of shareholders can reconstruct the secret f_0
  - Any up to t of shareholders cannot reconstruct the secret f_0

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

- Given a secret f_0
- 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 $\pi = (c_0, \ldots, c_{t+1})$



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

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

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

Homomorphic Commitments:

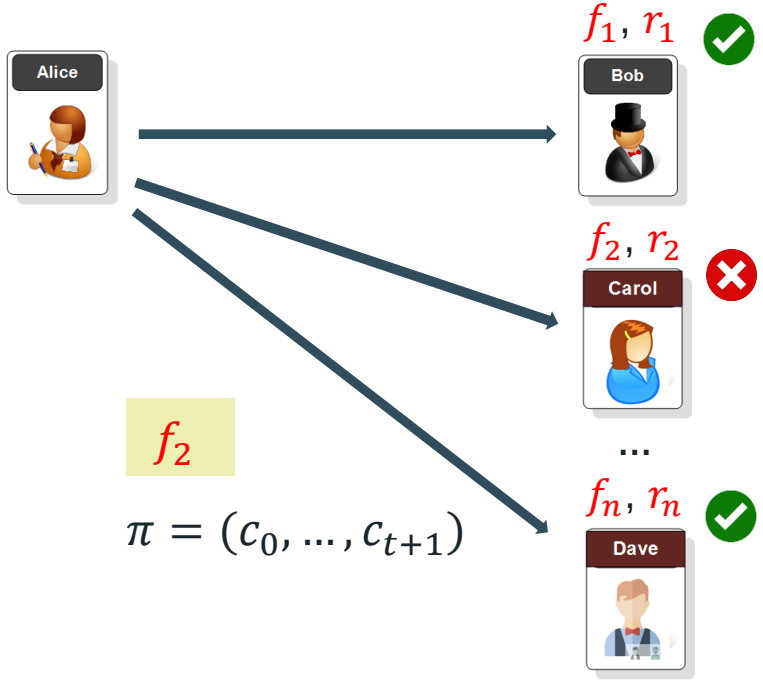
$com(a_1) \times com(a_2) = com(a_1 + a_2)$

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

VSS from Homomorphic Commitments: Pedersen [Ped91]

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

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



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

$$g^{f_1} h^{r_1} = c_0^{1^0} \times c_1^{1^1} \times \cdots \times c_t^{1^t}$$
$$= g^{f_0} h^{r_0} \times g^{a_1} h^{b_1} \times \cdots \times g^{a_t} h^{b_t} = g^{f(1)} h^{r(1)}$$

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

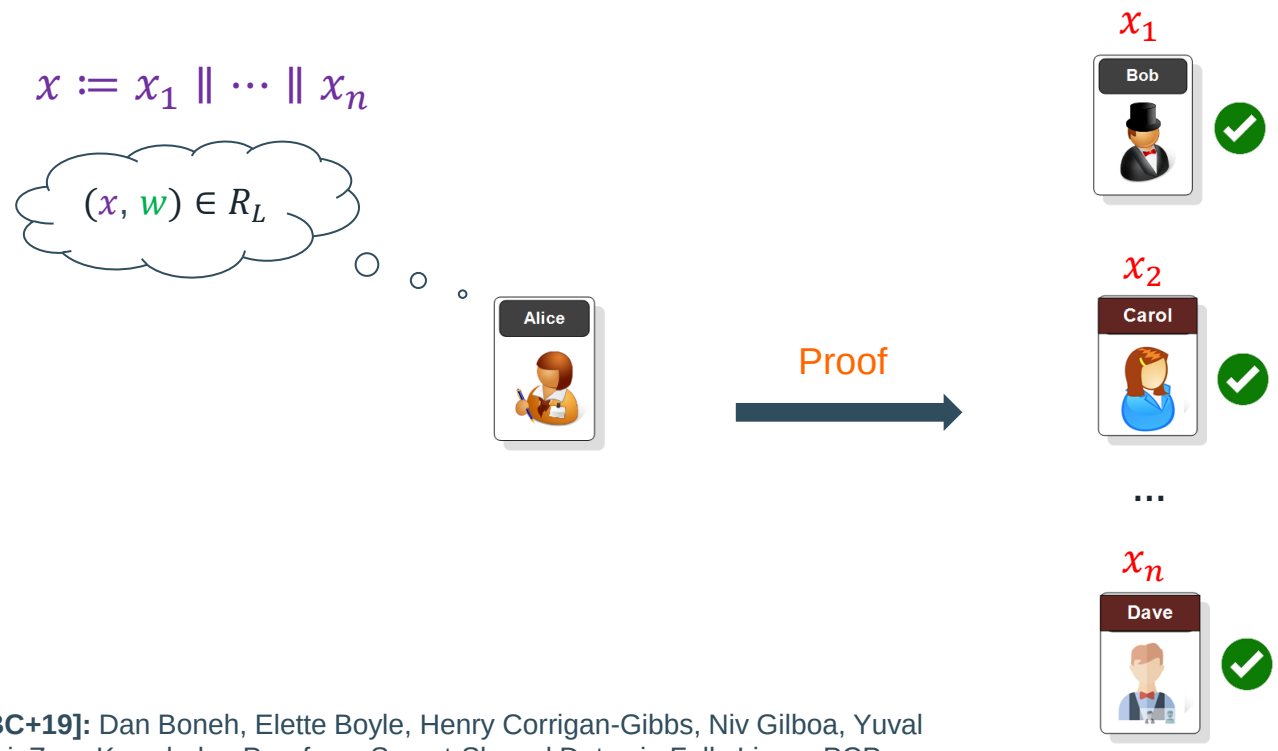
...

Pedersen Commitment:

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

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

- Involves a prover and n verifiers, which allows the parties to communicate in synchronous rounds.
- The prover holds an n-distributed input $x := x_1 \| \cdots \| x_n$ and a witness w, and each verifier V_i holds an input piece x_j.



- **Completeness:** If $x \in L$, an 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 will be malicious and collude with the prover.
- **Threshold Zero-Knowledge (TZK):** any subset of the verifiers up to a certain bound, i.e., t, should learn no additional information about statement x, beyond their own shares and the fact that $x \in L$.

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

Outline:

- Introduction
  - Shamir Secret Sharing
  - Verifiable Secret Sharing (VSS) from Homomorphic Commitments
  - Threshold ZK Proofs Over Secret Shared Data
- Our Contribution
  - VSS from ZK Proofs Over Shared Data
    - An Efficient NI-TZK Proof Scheme for Shamir Relation (in the QROM)
    - A (PQ-Secure) VSS from the Proposed NI-TZK Proof Scheme
    - Empricial Performance and Comparisons
  - More Efficient Threshold Protocols in the DL Setting
    - An Efficient Variant of Pedersen DKG and GJKR Threshold Signature
  - More Efficient Threshold Protocols from Isogenies
    - An Efficient Variant of ABCP DKG and Campos-Muth Threshold Signature



Our Contribution: A TZK Proof for Shamir Relation

- We present an NI-TZK proof scheme for the following n-distributed relation,

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

$(x, w) \in R_L$

$x := f_1 \| \cdots \| f_n$

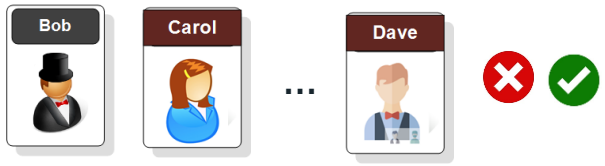


1) Sample a degree t polynomial $r(x)$
2) For $i = 1, \dots, n$: sample y_i and y_i' and set $C_i = Com(r(i), y_i)$ and $C_i' = Com(f_i, y_i')$
3) Set $d = Hash(C_1, \dots, C_n, C_1', \dots, C_n)$
4) Set $z(X) = r(X) + df(X)$
5) Publish $\pi = (C_1, \dots, C_n, C_1', \dots, C_n', z(X))$ and send privatly (f_i, y_i, y_i') to verifier V_i.

$\pi = (C_1, \dots, C_n, C_1', \dots, C_n', z(X))$

- ✓ Completeness
- ✓ Soundness against prover and t verifiers
- ✓ Threshold ZK

f_i, y_i, y_i'



Verifier V_i act as follows and outputs accept/reject:

1) If $C_i' \neq Com(f_i, y_i')$ outputs reject
2) Set $d = Hash(C_1, \dots, C_n, C_1', \dots, C_n)$
3) If $C_i == Com(z(i) - df_i, y_i)$ outputs accept, otherwise reject.

Return accept, if all parties return accept.

Our Contribution: A VSS from the Proposed TZK Proof

- Given the proposed TZK proof scheme for the Shamir relation, we construct a post-quantum secure NI-VSS based on Shamir secret sharing.

$x := f_1 \parallel \cdots \parallel f_n$



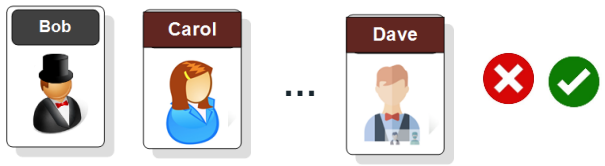
1) Given n and t, to share f_0 the dealer does:
2) Sample a degree t polynomial $f(x)$ s.t., $f(0) = f_0$ and sets $f_i = f(i)$ for $i = 1, \dots, n$
3) Using $x := f_1 \parallel \cdots \parallel f_n$ and polynomial $f(x)$ run the prover of new NI-TZK proof and obtains $\pi = (C_1, \dots, C_n, C'_1, \dots, C'_n,\ z(X))$ and (f_i, y_i, y'_i).
4) Publish $\pi = (C_1, \dots, C_n, C'_1, \dots, C'_n,\ z(X))$ and send privatly (f_i, y_i, y'_i) to participant V_i.

$\pi = (C_1, \dots, C_n, C'_1, \dots, C'_n, z(X))$

- ✓ Completeness
- ✓ Reconstructability (soundness)
- ✓ Unpredictable (TZK)

- ❑ In practice we can omit y_i *and* y'_i if f_0 has enough entropy.

f_i, y_i, y'_i



Party V_i runs the verifier of the proof scheme and outputs accept/reject:
1) If $C'_i \neq Com(f_i, y_i')$ outputs reject
2) Set $d = Hash(C_1, \dots, C_n, C'_1, \dots, C_n)$
3) If $C_i == Com(z(i) - df_i, y_i)$ outputs accept, otherwise, reject.
4) If party V_j complains aggainst the dealer, then dealer publishesh (f_i, y_i, y'_i), s.t., other parties can check and decide.
5) At the end, the honest parties either reject, or get sure that have received a valid share.

Our Contribution: (New) Reconstruction Approach

- Given $t+1$ valid shares, parties can reconstruct the secret. W.l.o.g., given valid shares $f_1, \dots, f_{t+1}$, we can interpolate polynomial $\hat{f}(X)$ of degree t that passes through those points. Compute $\hat{r}(X) \coloneqq z(X) - d\hat{f}(X)$ and verify that $C_i' = Com(\hat{f}(i))$ and $C_i = Com(\hat{r}(i))$ for all i. If yes, output $\hat{f}(0)$, else output reject.

- Alternatively, the dealer can calculate and publish the reconstructed secret $f_0 = f(0)$ (or a function thereof) along with a NI-TZK proof for the distributed relation,

$$R_i = \{(f_0, f_i, f(X)) \mid f(0) = f_0 \wedge f(i) = f_i\} \text{ for } i = 1, \dots, n$$

where the proof is ZK for t parties, but can be verified by any set of $t+1$ valid shares.

✓ One may notice that using the new approach, one can perform both sharing and reconstruction at the same time, and save on the number of rounds.

Performance of New VSS: Asymptotic Costs

- New VSS Scheme vs. Pedersen [Ped91]

Table 2. Asymptotic costs in Pedersen [28] and our proposed NI-VSS schemes. DL: Discrete Logarithm, BC: Broadcast, n: Number of parties, $E_{\mathbb{G}}$: Exponentiation in group $\mathbb{G}$, $M_{\mathbb{G}}$: Multiplication in group $\mathbb{G}$, $\mathcal{PE}$: degree-t Polynomial Evaluation, $\mathcal{H}$: Hashing, $|\mathbb{G}|$: $\mathbb{G}$ element size, $|\mathbb{Z}_q|$: $\mathbb{Z}_q$ element size, $|\mathbb{Z}_N|$: $\mathbb{Z}_N$ element size, $|\mathcal{H}|$: Output size of $\mathcal{H}$, DV: Designated Verifier, FS: Fiat-Shamir.

VSS Scheme	Assumption	Sharing	Dealer's Communication	Verification (DV)
Pedersen [28]	DL-based (IT & Classic)	$1n\ E_{\mathbb{G}}$ $2n\ \mathcal{PE}$	Private: $2n\ \|\mathbb{Z}_q\|$ BC: $t \approx 0.5n\ \|\mathbb{G}\|$	$t \approx 0.5n\ E_{\mathbb{G}}$
This work Sec. 3.2	Hash-based (PQ)	$2n\ \mathcal{H}$ $2n\ \mathcal{PE}$	Private: $1n\ \|\mathbb{Z}_N\|$ BC: $2n\ \|\mathcal{H}\| + 0.5n\ \|\mathbb{Z}_N\|$	$1\ \mathcal{PE} + 3\ \mathcal{H}$ ($1\ \mathcal{H}$ is for FS)

Performance of New VSS: Empirical Performance

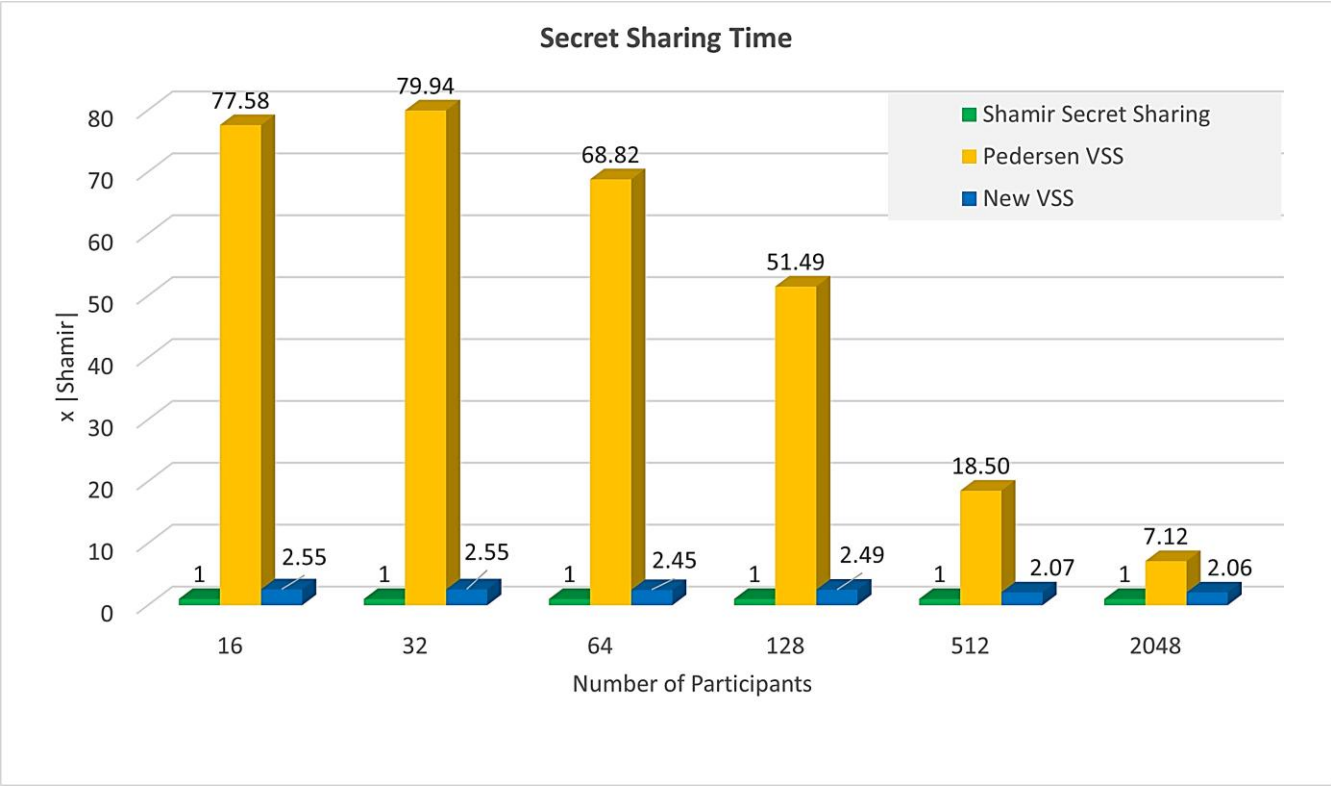
- For both commitment and random oracle, we used the SHA256 hash function.
- For the Pedersen scheme implementation, we utilized Curve 25519.
- Laptop: Ubuntu 22.04 LTS, a 11th Gen Intel(R) Core i9-11950H at base frequency 2.60GHz, & 64GB of memory.
- Our codes are publicly available on github.com/Baghery/VSS-ABCP23
- Can be run by scanning the QR-code

Table 3. Empirical performance of NI-VSS schemes Pedersen [28] and our proposed scheme for various numbers of parties and threshold values (n, t). n: Number of parties, t: Threshold value, BC: Broadcast, $|\mathbb{Z}_q| = |\mathbb{Z}_N| = |\mathbb{G}| = |\mathcal{H}| = 256$ bits.

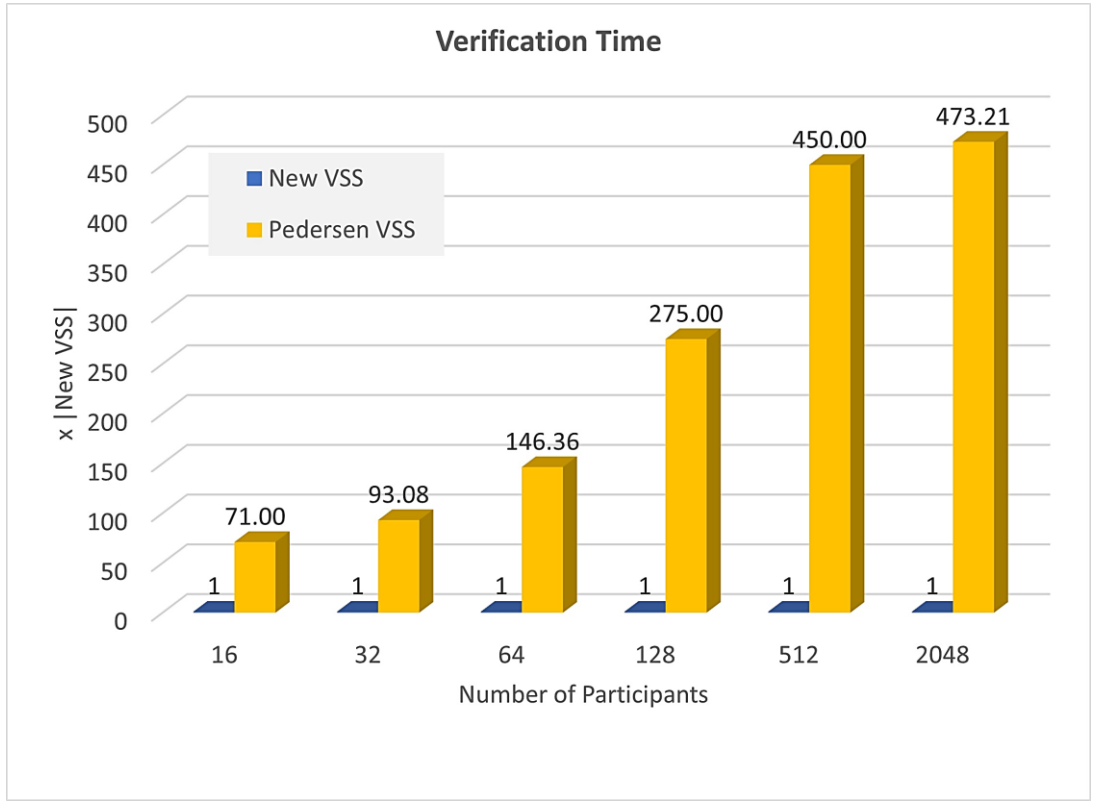
(n, t)	Scheme	Sharing	Dealer's Communication	Verification
(32, 15)	Pedersen [28]	78.1 msec	Private: 2.0 KB + BC: 0.5 KB	10.2 msec
	This Work	2.40 msec	Private: 1.0 KB + BC: 2.5 KB	0.12 msec
(128, 63)	Pedersen [28]	310 msec	Private: 8.0 KB + BC: 2.0 KB	97.6 msec
	This Work	13.7 msec	Private: 4.0 KB + BC: 10.0 KB	0.36 msec
(512, 255)	Pedersen [28]	1.310 sec	Private: 32.0 KB + BC: 8.0 KB	525 msec
	This Work	0.140 sec	Private: 16.0 KB + BC: 40.0 KB	1.20 msec
(2048, 1023)	Pedersen [28]	6.53 sec	Private: 128 KB + BC: 32 KB	2.35 sec
	This Work	2.00 sec	Private: 64 KB + BC: 160 KB	4.90 msec
(8192, 4095)	Pedersen [28]	48.1 sec	Private: 512 KB + BC: 128 KB	9.47 sec
	This Work	30.6 sec	Private: 256 KB + BC: 640 KB	0.019 sec
(16384, 8191)	Pedersen [28]	153.6 sec	Private: 1024 KB + BC: 256 KB	19.2 sec
	This Work	121.7 sec	Private: 512 KB + BC: 1280 KB	0.039 sec

Performance of New VSS: Shamir vs. Pedersen, New VSS

Comparison of sharing time in standard Shamir secret sharing and VSS schemes.



Comparison of verification time for new VSS and Pedersen scheme



Outline:

- Introduction
  - Shamir Secret Sharing
  - Verifiable Secret Sharing (VSS) from Homomorphic Commitments
  - Threshold ZK Proofs Over Secret Shared Data
- Our Contribution
  - VSS from ZK Proofs Over Shared Data
    - An Efficient NI-TZK Proof Scheme for Shamir Relation (in the QROM)
    - A (PQ-Secure) VSS from the Proposed NI-TZK Proof Scheme
    - Empricial Performance and Comparisons
  - More Efficient Threshold Protocols in the DL Setting
    - An Efficient Variant of Pedersen DKG and GJKR Threshold Signature
  - More Efficient Threshold Protocols from Isogenies
    - An Efficient Variant of ABCP DKG and Campos-Muth Threshold Signature



More Efficient Threshold Protocols from DL: DKG & Thr. Sig.

- Using the new VSS scheme and new reconstruction approach we revisit
  - Pedersen DKG protocol [Ped91]
  - Threshold robust signature of Gennaro, Jarecki, Krawczyk, Rabin (GJKR) [GJKR07]

- To this end, we first construct a NI-TZK proof scheme for the following n-distributed relation,

$$R_i = \{(y, f_i, f(X)) \,|\, y = g^{f(0)} \wedge f(i) = f_i\} \text{ for } i = 1, \ldots, n.$$

✓ This allows a dealer to prove that he/she has generated y using the shared secret $f(0)$.

✓ It can be independently interesting in other DL-based threshold protocols.

An Efficient DKG Protocol for DL: Variant of Pedersen DKG

- **Round 1 (VSS and Committing):** Each party P_i proceed as follows:
  - Sample $f^{(i)}(X)$ subject to $f^{(i)}(0) = x_i$ and set $x_{ij} = f^{(i)}(j)$ for $j = 1, \dots, n$
  - Using $(g, h_i \coloneqq g^{x_i}, \{x_{ij}\}_{j=1}^{n})$ run the prover of proof scheme from previous slide, and obtain a proof $\pi \coloneqq (g, h_i, C_1, \dots, C_n, C'_1, \dots, C'_n, r(X))$ and shares $(y_{i0}, y'_{i0}, y_{i,j}, y'_{i,j})$ for $j = 1, \dots, n$.
  - Publishes $(C_1, \dots, C_n, C'_1, \dots, C'_n, r(X))$ and stores $\left(x_{ij}, h_i, y_{i0}, y'_{i0}, y_{i,j}, y'_{i,j}\right)$ for next round.
- **Round 2 (Opening, Verification, PK Computation):**
  - **Opening**: Each party broadcasts (h_i, y_{i0}, y'_{i0}) and sends $\left(x_{ij}, y_{i,j}, y'_{i,j}\right)$ to party P_j. If a party refuses to open a commitment, then that party is disqualified.
  - **Verification**: The parties verify the proofs and commitments, and in case of conflict, solve it as in the VSS scheme. In the end, all honest parties will agree on the same qualified set Q.
  - **PK Computation:** Parties compute the PK as $g^x = \prod_{i \in Q} g^{L_{0,i}^{Q} x_i}$, where $L_{0,i}^{Q} = \prod_{j \in Q, j \neq i} \frac{j}{j-1}$.

Performance of New DKG: Asymptotic Costs

- Pedersen DKG protocol [Ped91, GJKR07] vs. the new variant

Table 4. Asymptotic costs in the GJKR [21] variant of Pedersen DKG [28] and our proposed scheme. DL: Discrete Logarithm, BC: Broadcast, n: Number of parties, $t \approx n/2$: Threshold value, $E_{\mathbb{G}}$: Exponentiation in group $\mathbb{G}$, $\mathcal{PE}$: degree-t Polynomial Evaluation, $\mathcal{H}$: Hashing, $|\mathbb{G}|$: $\mathbb{G}$ element size, $|\mathbb{Z}_q|$: $\mathbb{Z}_q$ element size, $|\mathcal{H}|$: $\mathcal{H}$ image size.

DKG Scheme	Assumption	Parties' Computation	Communication
GJKR [21] (Pedersen [28])	DL-based	$(2nt + n)\ E_{\mathbb{G}} + 2n\ \mathcal{PE} + 1\ \mathcal{H}$ ($2nt\ E_{\mathbb{G}}$ is for verify)	Private: $2n\ \lvert\mathbb{Z}_q\rvert$ BC: $2t \approx n\ \lvert\mathbb{G}\rvert$
This work Sec. 4.1	DL & Hash-based	$2n\ E_{\mathbb{G}} + 3n\ \mathcal{PE} + 5n\ \mathcal{H}$ ($2n\ E_{\mathbb{G}} + n\ \mathcal{PE} + 3n\ \mathcal{H}$ is for verify)	Private: $1n\ \lvert\mathbb{Z}_q\rvert$ BC: $2n\ \lvert\mathcal{H}\rvert + t\ \lvert\mathbb{Z}_q\rvert$

✓ In new DKG protocol parties need to compute $O(n)$ Exp. Instead of $O(n^2)$.

More Efficient Robust Threshold Signature from Schnorr:

- Using the new DKG and new VSS, we can revisit GJKR [GJKR07] robust threshold signature
- **KeyGen:** This is done using new DKG protocol.
- **Signing Protocol:** To sign m, the parties in the qualified set Q proceed as follows:
  - They run the new DKG to sample g^b, and each obtains a share b_i of b.
  - They compute the challenge $\mathrm{d} = \mathrm{hash}(g^b \parallel m)$.
  - They perform the following steps
    - Each publishes $r^{(i)}(X) = b^{(i)}(X) - d\, f(X)$.
    - Using the shared values from VSS phase, each other party P_j verifies if $r^{(i)}(j) = b^{(i)}(j) - d\, f(j)$.
    - Whenever, one of these checks fails, P_j broadcasts a complain against P_i, and conflicts are solved as in the VSS scheme.
    - At the end, using the values $r^{(i)}(0)$ parties compute the response $\mathrm{r} = \sum_{i \in Q} r^{(i)}(0)$.
  - Finally, parties output the signature (r, d, m).
- **Verification:** A verifier runs the verification of the Schnorr signature.

Outline:

- Introduction
  - Shamir Secret Sharing
  - Verifiable Secret Sharing (VSS) from Homomorphic Commitments
  - Threshold ZK Proofs Over Secret Shared Data
- Our Contribution
  - VSS from ZK Proofs Over Shared Data
    - An Efficient NI-TZK Proof Scheme for Shamir Relation (in the QROM)
    - A (PQ-Secure) VSS from the Proposed NI-TZK Proof Scheme
    - Empricial Performance and Comparisons
  - More Efficient Threshold Protocols in the DL Setting
    - An Efficient Variant of Pedersen DKG and GJKR Threshold Signature
  - More Efficient Threshold Protocols from Isogenies
    - An Efficient Variant of ABCP DKG and Campos-Muth Threshold Signature



Introduction: Isogenies - CSIDH

- **Hard Homogeneous Space (HSS) [Cou06]**:
  - A set $\mathcal{E}$ equipped with a group action by a group G

$$\star : G \times \mathcal{E} \to \mathcal{E}$$
$$g \star E \to E'$$

  - If $G =< g >$ cyclic of order N, then we can define

$$[\,]: Z_N \times \mathcal{E} \to \mathcal{E}$$
$$[a]E = g^a \star E$$

Vectorization Problem (GAIP):
- Given $[a]E$ find a.

PK of many protocols in the CSIDH setting [bkv19]

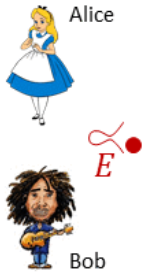
Multi Target GAIP (MT-GAIP):
- Given $[a_i]E$ find $a_i - a_j \; s.t. \; E_i = [a_i - a_j]E_j$.

PK of CSI-FiSh signature scheme [BKV19].

Vectorization Problem with Auxiliary Inputs:
- Given $[c_i a]E$ find a,
where $\mathbb{C}_k = \{c_0 = 0, c_1 = 1, c_2, \cdots, c_k\}$
is an exceptional set

PK of CSI-SharK signature scheme [ABCP23a].

- Symmetric HSS: HSS is symmetric around $E_0 \in \mathcal{E}$ if one can compute $[-a]E_0$ from $[a]E_0$
- CSIDH Key Exchange:
  - Alice: $E_A = [a]E \rightarrow [a]E_B = [a]([b]E) = [a+b]E = E_{AB}$
  - Bob: $E_B = [b]E \rightarrow [b]E_A = [b]([a]E) = [b+a]E = E_{AB}$



- There is no homomorphism:
  - **Such an operation does not exist in this setting!** $([a]E_0) \cdot ([b]E_0) = ?\,!?\,!?$
  - **Therefore, in case of threshold protocols this needs to be done in a round robin way,** $[a]([b]E) = [b]([a]E) = [a+b]E$

https://www.esat.kuleuven.be/cosic/blog/elliptic-curves-are-quantum-dead-long-live-elliptic-curves/

More Efficient DKG for CSIDH-based Protocols: High Level Idea

- **Our earlier DKG [ABCP23b]**
  - $E_i = [c_i x]E_0$

VSS Step (BGW88):
(by bivariate polynomials)

$f(i) = a_i$

(Structured) PK Computation Step:

$f(i) = a_i \cap F_i = [c_i f(0)]E_0$

- **New DKG**
  - $E_i = [c_i x]E_0$

VSS Step:
(by NI-SZK Proofs)

$f(i) = a_i$

(Structured) PK Computation Step:

$f(i) = a_i \cap F_i = [c_i f(0)]E_0$

Parties engage in a round-robin protocol and update the whole PK using their shares of the final secret key and give NI-TZK proof for the correctness of their update.

$$f^{(1)}(i) = a_i \cap F_i^{(1)} = [c_i f^{(1)}(0)]E_0$$

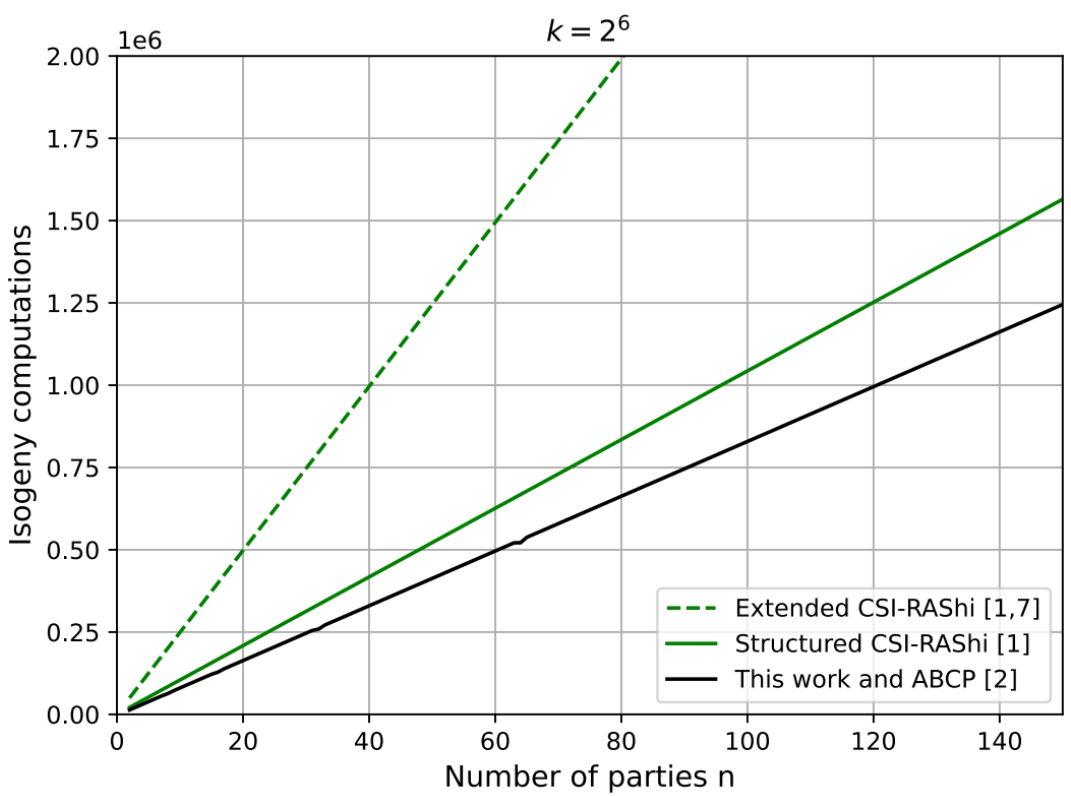
$$f^{(2)}(i) = a_i \cap F_i^{(2)} = [c_i f^{(2)}(0)]F_i^{(1)}$$

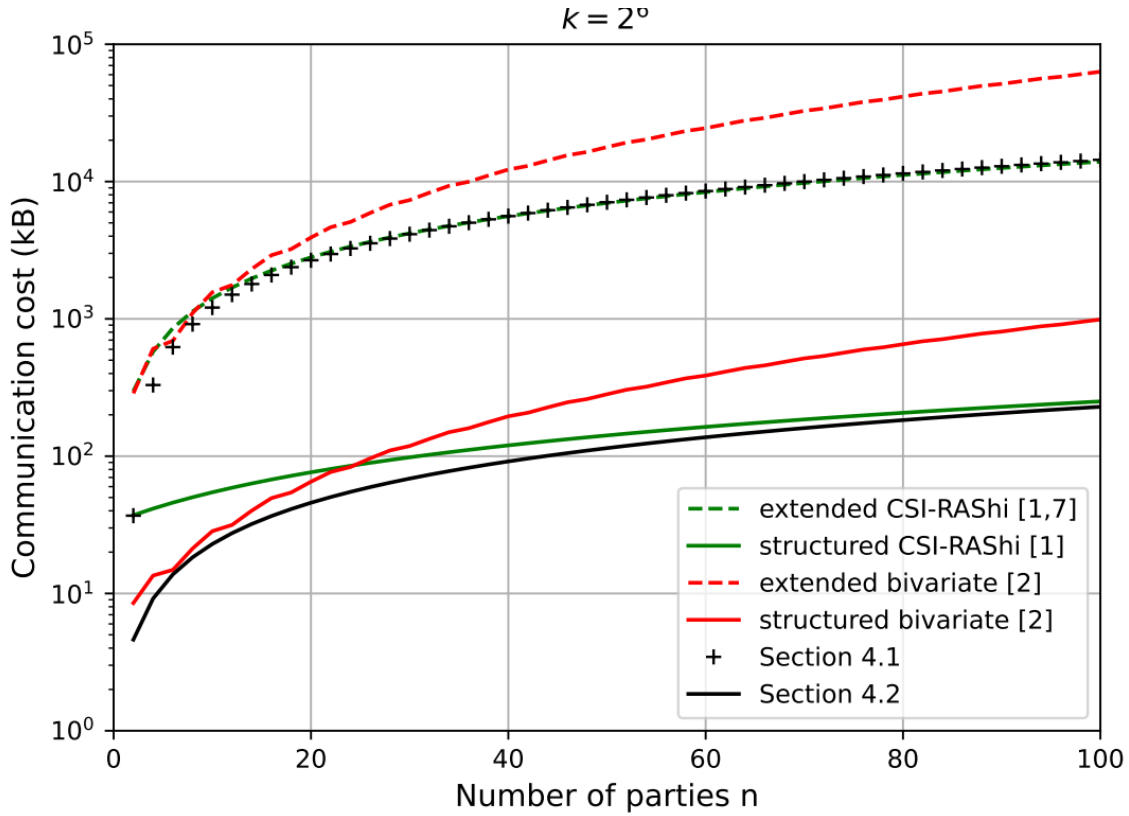
$$\backslash F_i^{(n)} = [c_i f^{(n)}(0)]F_i^{(n-1)}$$

- **[ABCP23b]:** Practical Robust DKG Protocols for CSIDH. ACNS23.

Performance of New DKG for CSIDH: Asymptotic Costs

- **Isogeny computation and communication cost in DKGs for CSIDH-based primitives**





KU LEUVEN

Thank You!



karim.baghery@kuleuven.be