KU LEUVEN

ACNS 2025 – Munich, Germany
23-27 June 2025

Pre-Constructed Publicly Verifiable Secret Sharing and Applications

ia.cr/2025/576

Karim Baghery, Noah Knapen, Georgio Nicolas, Mahdi Rahimi

COSIC

KU Leuven, Belgium

Outline:

- Introduction and Background
  - Shamir Secret Sharing
  - Publicly Verifiable Secret Sharing (PVSS) in the Synchronous Setting
- Pre-Constructed Publicly Verifiable Secret Sharing (PPVSS)
  - Definition and Syntax
  - Schoenmakers' PVSS [Sch99] is a Special PPVSS
  - A General Construciton of PPVSS from a Shamir-based PVSS
  - Two Practical PPVSS Schemes in the Random Oracle and Plain Models
- Applications and Implementation Results
  - Emprical Performance of New RO-Based PPVSS Scheme
  - A Practical Universally Verifiable E-Voting Protocol
  - Threshold Binding Elgamal Encryption, …
  - Emprical Performance of New E-Voting Protocol
- Conclusions and Future Directions

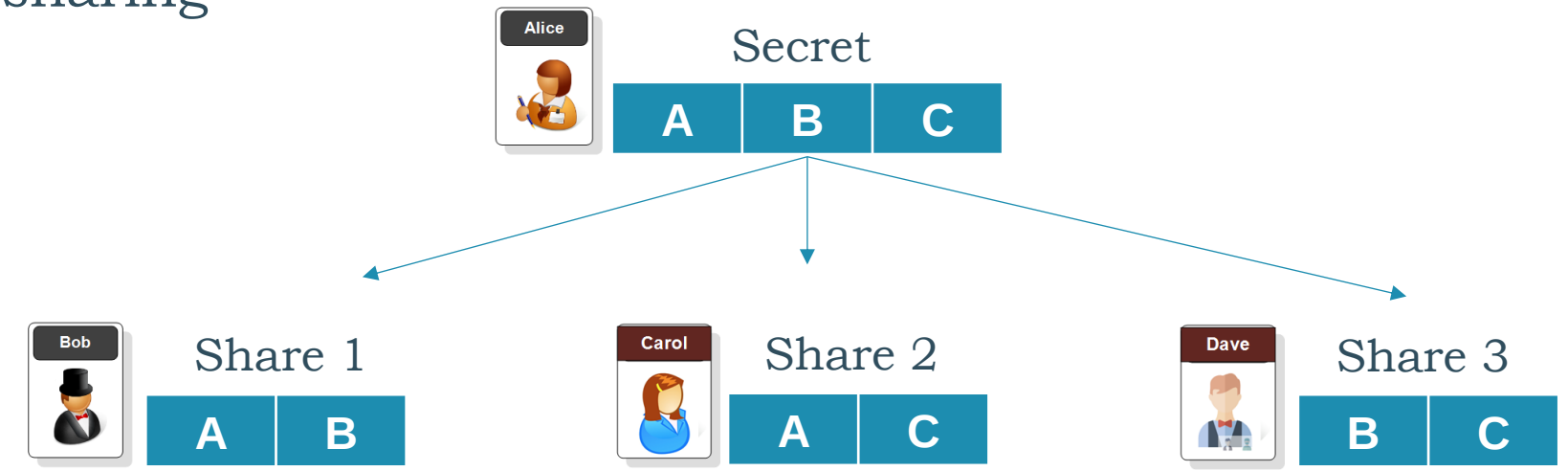
Secret Sharing: Motivation

- 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!**

- Replicated secret sharing

  - Initial
  - Share
  - Reconstruct



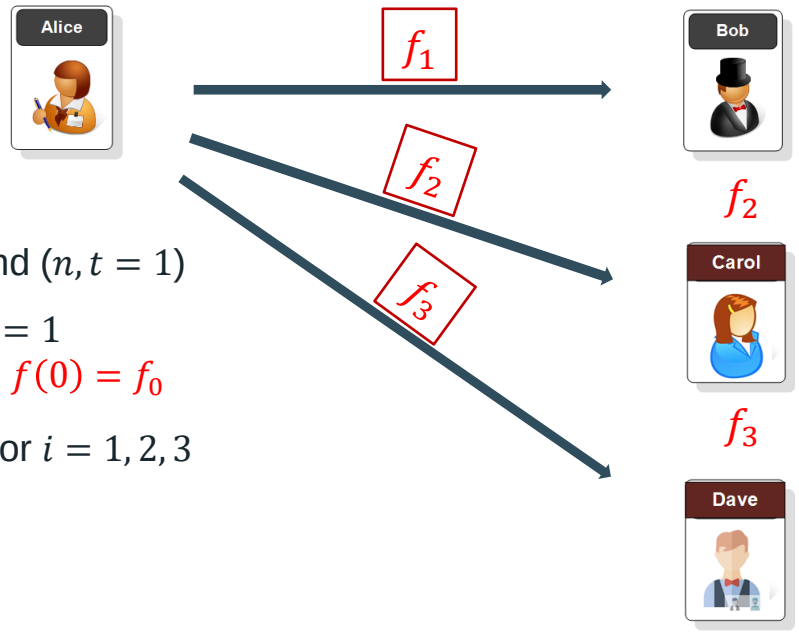
Shamir Secret Sharing: A Toy Example

- 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

Not Verifiable!

Share



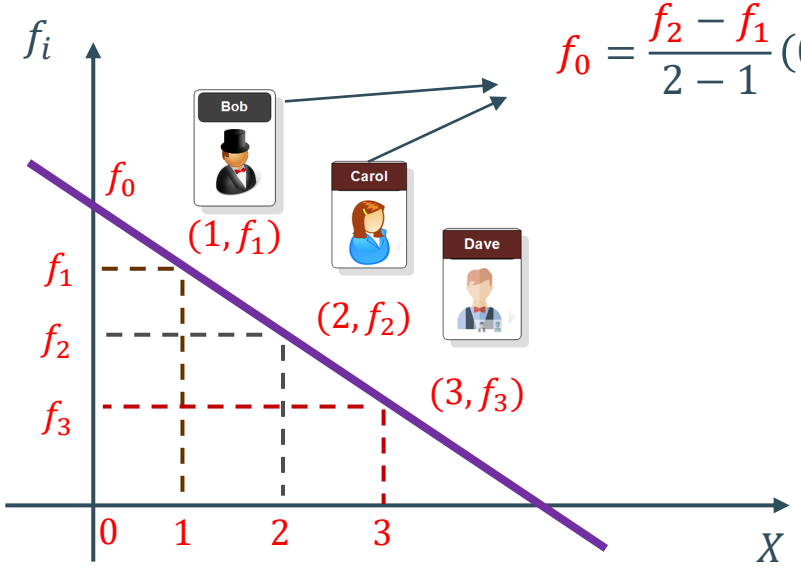
- Given a secret f_0 and $(n, t = 1)$
- Sample a degree t = 1 polynomial $f(X)$ s.t. $f(0) = f_0$
- Compute $f_i = f(i)$ for $i = 1, 2, 3$

Not Verifiable!

Reconstruction

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

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



Shamir-based Publicly Verifiable Secret Sharing (PVSS):

- **Initial:** Parties register their public keys $\{pk_i\}_{i=1}^n$ and store the secret keys for the reconstruction phase.
- **Share**: Given $\{pk_i\}_{i=1}^n$, the Dealer of Shamir-based PVSS encrypts the shares $\{f_i = f(i)\}_{i=1}^n$ and proves that the sharing & encryption are done correctly. I.e., it generates a *publicly verifiable* NIZK proof for

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

- **Verify**: Anyone can verify the validity of all the encrypted shares $\{C_i\}_{i=1}^n$ & return $1/0$.
- **Reconstruction**: A set of $|Q| > t$ parties decrypt $\{C_i\}_{i \in Q}$, publish shares, & a NIZK proof of correct decryption. Then $t+1$ valid ones are used to reconstruct the secret.

✓ In PVSS schemes usually the secret is set to be either f_0 or g^{f_0}, where g is the group generator.

✓ We focus on PVSS schemes where the secret is g^{f_0}, but our results can also be extended to other PVSS schemes, e.g. [Gro21, CD24]

Scheme	Share	Broadcast & Dow.	Verification	Recon.
[Sch99] (DDH, RO)	$4.5n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	BC: $1.5n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$ Dow: $2.5n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$	$nt + 4n\ E_{\mathbb{G}}$ $+ nt + 2n\ M_{\mathbb{G}}$	$5t\ E_{\mathbb{G}}$ $+ t\ M_{\mathbb{G}}$
[HV08] (DDH, Plain)	$1.5n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	BC: $1.5n\lvert\mathbb{G}\rvert$ Dow: $2.5n\lvert\mathbb{G}\rvert$	$nt\ E_{\mathbb{G}} + 2nP_{\mathbb{G}}$ $+ nt\ M_{\mathbb{G}}$	$5t\ E_{\mathbb{G}}$ $+ t\ M_{\mathbb{G}}$
[CD17] (DBS, Plain)	$2n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	BC: $2n\lvert\mathbb{G}\rvert$ Dow: $3n\lvert\mathbb{G}\rvert$	$2n\ P_{\mathbb{G}} +$ $n\ E_{\mathbb{G}} + n\ M_{\mathbb{G}}$	$2t\ P_{\mathbb{G}} +$ $t\ E_{\mathbb{G}} + t\ M_{\mathbb{G}}$
[CD17] (DDH, RO)	$4n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	BC: $2n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$ Dow: $3n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$	$5n\ E_{\mathbb{G}}$ $+ 3n\ M_{\mathbb{G}}$	$5t\ E_{\mathbb{G}}$ $+ t\ M_{\mathbb{G}}$
[CD22] (DDH, RO)	$4n\ E_{\mathbb{G}}$ $2n\ \mathcal{PE} + 3n\ M_{\mathbb{G}}$	BC: $n\lvert\mathbb{G}\rvert + 2\lvert\mathbb{Z}_q\rvert$ Dow: $2n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$	$2n\ E_{\mathbb{G}} +$ $n\ \mathcal{PE} + 2n\ M_{\mathbb{G}}$	$5t\ E_{\mathbb{G}} +$ $2t\ M_{\mathbb{G}}$
[CD20, Bag25] (DDH, RO)	$2n\ E_{\mathbb{G}}$ $2n\ \mathcal{PE}$	BC: $n\lvert\mathbb{G}\rvert + 0.5n\lvert\mathbb{Z}_q\rvert$ Dow: $2n\lvert\mathbb{G}\rvert + 0.5n\lvert\mathbb{Z}_q\rvert$	$2n\ E_{\mathbb{G}} +$ $n\ \mathcal{PE} + n\ M_{\mathbb{G}}$	$5t\ E_{\mathbb{G}} +$ $t\ M_{\mathbb{G}}$

Our Contribution:

Pre-Constructed Publicly Verifiable Secret Sharing (PPVSS)

Pre-Constructed PVSS (PPVSS): Definition & Syntax

✓ We introduce Pre-Constructed PVSS as an extension of PVSS with enhanced utility and efficiency.

- **Initial:** Parties register their public keys $\{pk_i\}_{i=1}^n$ and agree on a commitment/public key pk_0.
- **Share**: Given $\{pk_i\}_{i=1}^n$ and pk_0, the Dealer of Shamir-based PPVSS first encrypts the shares $\{f_i = f(i)\}_{i=1}^n$ and commits to (or encrypts) the main secret f_0 using pk_0. Then, it proves that the sharing, encrypting, and committing to (or encryption of) the main secret are done correctly. I.e., it generates a NIZK proof for

$$R_{PPVSS} = \{(C_0, C_i, pk_i, f(X)) \mid C_0 = Com/Enc(f(0), pk_0) \wedge C_i = Enc(f(i), pk_i) \text{ for } i = 1, \dots, n\}.$$

- **Verify**: Anyone can verify the validity of the commitment C_0 and all the encrypted shares $\{C_i\}_{i=1}^n$ & return 1/0.
- **(Optimistic) Reconstruction**: The dealer opens the commitment C_0 and parties verify the opening & return the main secret. Alternatively, the party with knowledge of secret key sk_0 decrypts C_0 and publishes a NIZK proof.
- **(Pessimistic) Reconstruction**: A set of $|Q| > t$ parties decrypt $\{C_i\}_{i \in Q}$, publish shares & a NIZK proof of correct decryption. Then $t + 1$ valid ones are used to reconstruct the secret.

PPVSS: Security Properties & Constructions

➢ **A Secure PPVSS Scheme Requires to Satisfy:**

- **Correctness:** If the dealer and parties follow the protocol, then the Verify algorithm will return true and the Reconstruction by both approaches will return f_0.
- **Verifiability:** If Verify algorithm returns 1, then with high probability the values $\{C_i\}_{i=1}^{n}$ are encryptions of a valid sharing of some (unique) secret, and C_0 is a commitment (or a ciphertext under pk_0) to the same secret.
- **IND1-Secrecy (Indistinguishability of Secrets):** Prior to the reconstruction phase, the public information together with the secret keys sk_i of any set of at most t parties, excluding sk_0, gives no information about the shared secret f_0.

➢ **PPVSS Schemes:** Schoenmakers' PVSS scheme [Sch99] and its paring-based variant [HV08] happen to also publish a commitment to the secret as well so they are a special case of a PPVSS.

Scheme	Share	Broadcast & Dow.	Verification	Recon.
[Sch99] (DDH, RO)	$4.5n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	BC: $1.5n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$ Dow: $2.5n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$	$nt + 4n\ E_{\mathbb{G}}$ $+ nt + 2n\ M_{\mathbb{G}}$	$5t\ E_{\mathbb{G}}$ $+ t\ M_{\mathbb{G}}$
[HV08] (DDH, Plain)	$1.5n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	BC: $1.5n\lvert\mathbb{G}\rvert$ Dow: $2.5n\lvert\mathbb{G}\rvert$	$nt\ E_{\mathbb{G}} + 2nP_{\mathbb{G}}$ $+ nt\ M_{\mathbb{G}}$	$5t\ E_{\mathbb{G}}$ $+ t\ M_{\mathbb{G}}$

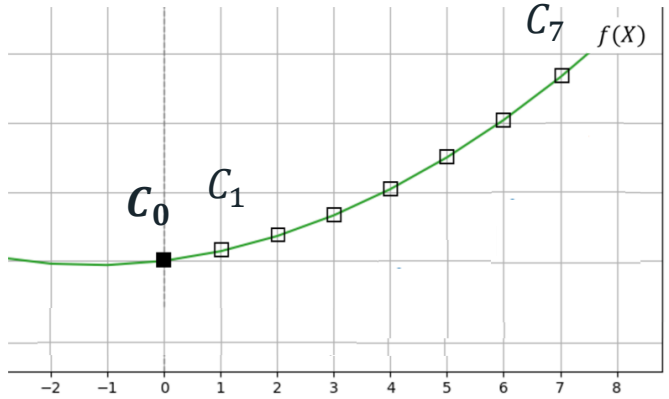
PPVSS from a Shamir-based PVSS: General Strategy

- **We propose a general strategy to efficiently turn**

A Shamir-based PVSS ⇒ A Shamir-based PPVSS

- **In Shamir-Based PVSS:** Given $\{pk_i\}_{i=1}^{n}$, the Dealer proves that he knows a $f(X)$ such that

$$C_i = Enc(f(i), pk_i) \quad \text{for } i = 1, \dots, n$$



- **In Shamir-Based PPVSS:** Given $\{pk_i\}_{i=0}^{n}$, the Dealer proves that he knows a $f(X)$ such that

$$C_i = Enc(f(i), pk_i) \quad \text{for } i = 0, 1, \dots, n$$

- **Efficiency:** The efficiency of the new PPVSS scheme closely aligns with that of the original PVSS protocol, with only a marginal increase, i.e., $\frac{1}{n}\times$ inreace

Practical PPVSS in the RO and Plain Models:

- **Instantiations**

A Shamir-based PVSS ⇒ A Shamir-based PPVSS

- **A PPVSS in the RO Model:** We instantiate the general protocol using the RO-based PVSS scheme proposed in [CD20, Bag25] and obtain an efficient PPVSS scheme.
- **A PPVSS in the Plain Model:** Next, we instantiate the general construction using the paring-based PVSS scheme proposed in [CD17] and obtain an efficient PPVSS scheme in the plain model.

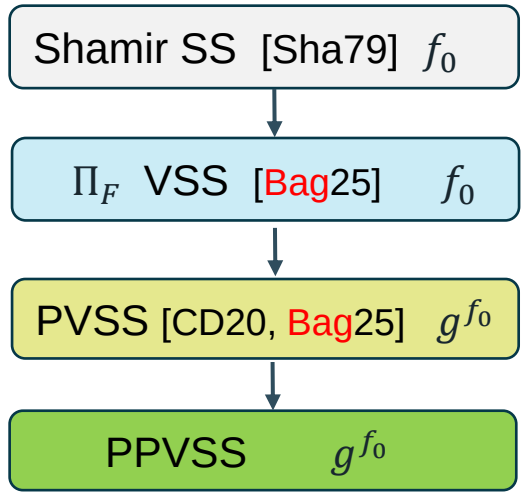
We show that both the PPVSS schemes satisfy **Completeness**, **Verifiability** and **IND1-Secrecy**.

- **Efficiency:**

PPVSS & Security	Share	Broadcast & Dow	Verification	Opt. Reconst.	Pes. Reconst.
Schoenmakers [23] (DDH, RO)	$4.5n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	BC: $1.5n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$ Dow: $2.5n\lvert\mathbb{G}\rvert + n\lvert\mathbb{Z}_q\rvert$	$nt + 4n\ E_{\mathbb{G}} +$ $nt + 2n\ M_{\mathbb{G}}$	$1\ E_{\mathbb{G}}$	$5t\ E_{\mathbb{G}}$ $+ t\ M_{\mathbb{G}}$
Sec. 4.2 (PDL, DDH, RO)	$2n\ E_{\mathbb{G}}$ $2n\ \mathcal{PE}$	BC: $n\lvert\mathbb{G}\rvert + 0.5n\lvert\mathbb{Z}_q\rvert$ Dow: $2n\lvert\mathbb{G}\rvert + 0.5n\lvert\mathbb{Z}_q\rvert$	$2n\ E_{\mathbb{G}} +$ $n\ \mathcal{PE} + n\ M_{\mathbb{G}}$	$1\ E_{\mathbb{G}}$	$5t\ E_{\mathbb{G}} +$ $t\ M_{\mathbb{G}}$
Hei-Vil [20] (DBS, Plain)	$1.5n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	BC: $1.5n\lvert\mathbb{G}\rvert$ Dow: $2.5n\lvert\mathbb{G}\rvert$	$nt\ E_{\mathbb{G}} + 2n\ P_{\mathbb{G}}$ $+ nt\ M_{\mathbb{G}}$	$1\ E_{\mathbb{G}}$	$2t\ P_{\mathbb{G}} + t\ E_{\mathbb{G}}$ $+ t\ M_{\mathbb{G}}$
Sec. 4.3 (DBS, Plain)	$2n\ E_{\mathbb{G}}$ $1n\ \mathcal{PE}$	BC: $2n\lvert\mathbb{G}\rvert$ Dow: $3n\lvert\mathbb{G}\rvert$	$n\ E_{\mathbb{G}} + 2n\ P_{\mathbb{G}}$ $+ n\ M_{\mathbb{G}}$	$1\ E_{\mathbb{G}}$	$2t\ P_{\mathbb{G}} + t\ E_{\mathbb{G}}$ $+ t\ M_{\mathbb{G}}$

The Proposed RO-Based PPVSS Scheme:

- **Share**: Given $(n, t, pk_0 = g^{s_0}, pk_1 = g^{s_1}, \dots, pk_n = g^{s_n}, f_0)$, dealer acts as follows:
  - Does Shamir sharing using $f(X)$ and obtains the shares $\{f_i = f(i)\}_{i=1}^{n}$.
  - It sets $C_i = pk_i^{f_i}$ for $i = 0, 1, \dots, n$. Then, it generates a NIZK proof π_{share} as
    - Samples $r(X)$ and Sets $\Gamma_i = pk_i^{r(i)}$ for $i = 0, 1, \dots, n$.
    - 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 $C_i = \left\{pk_i^{f_i}\right\}_{i=0}^{n}$ as the encrypted shares.



- **Verify**: Given $(n, t, pk_0, pk_1, \dots, pk_n, C_1, \dots, C_n, z(X), d)$, a public verifier checks if the followings hold,

$$\deg\big(z(X)\big) \le t \quad \text{and} \quad d = RO\left(C_0, C_1, \dots, C_n, \frac{pk_0^{z(0)}}{C_0^d}, \frac{pk_1^{z(1)}}{C_1^d}, \dots, \frac{pk_n^{z(n)}}{C_n^d}\right)$$

- **(Optimistic) Reconstruction**: Given (g, f_0, pk_0, C_0) anyone checks if $C_0 = pk_0^{f_0}$ and return g^{f_0} or false.
- **(Pessimistic) Reconstruction**: As in the original PVSS scheme [CD20, Bag25].

The new PPVSS scheme satisfies **Completeness**, **Verifiability** and **Secrecy** under PDL and DDH assumptions in the RO model.

Applications and Implementation Results

Instantiation Parameters and Benchmarking Setup

- All schemes implemented in Rust.

Cryptographic Primitives/Libraries:

- DL: Ristretto Group under Curve25519 (curve25519-dalek crate)
- RO: Blake3 (blake3 crate)

Benchmarking Setup:

- MacBook Pro with an M4 Pro CPU 12 core (8p/4e), 24GB RAM.
- Restriction to 8 threads.
- All timing numbers are the averages over 100 executions.

- Open-Source Repository: https://github.com/KULeuven-cosic/ppvss-evoting

- Easy to reproduce benchmarks.
- Executable examples.

Empirical Performance: [Sch99] vs. Our RO-Based PPVSS

(n, t)	Metrics	Schoenmakers [22]	Λ_{RO} PPVSS (Fig. 3)	Verification Gain
(64, 31)	Sharing	1.29 ms	0.59 ms	
	Verification	8.30 ms	0.51 ms	16.2×
	Dealer's Broadcast	5 KB	3 KB	
(128, 63)	Sharing	2.39 ms	1.12 ms	
	Verification	31.60 ms	0.94 ms	33.6×
	Dealer's Broadcast	10 KB	6 KB	
(256, 127)	Sharing	4.87 ms	2.47 ms	
	Verification	123.58 ms	2.10 ms	58.8 ×
	Dealer's Broadcast	20 KB	12 KB	
(512, 255)	Sharing	10.51 ms	6.21 ms	
	Verification	482.70 ms	4.97 ms	97.1 ×
	Dealer's Broadcast	40 KB	24 KB	
(1024, 513)	Sharing	24.73 ms	17.96 ms	
	Verification	1.91 s	13.68 ms	139.6 ×
	Dealer's Broadcast	80 KB	48 KB	
(2048, 1023)	Sharing	64.56 ms	59.16 ms	
	Verification	7.72 s	42.97 ms	179.6 ×
	Dealer's Broadcast	160 KB	96 KB	

Repository:
https://github.com/KULeuven-cosic/ppvss-evoting

Applications: Practical Universally Verifiable E-Voting Scheme

- In [Sch99], Schoenmakers proposed a variety of applications based on his (P)PVSS scheme.

- We revisited and improved the performance of all those applications. Namely,
  - **Universally Verifiable E-Voting Protocol**
  - Threshold Binding ElGamal Encryption
  - Threshold Software Key Escrow

- **New Universally Verifiable E-Voting Protocol**
  - In [Sch99] e-voting protocol we replace his PPVSS scheme with our proposed RO-based PPVSS scheme and adapt the protocol accordingly.
  - **This reduces the ballot verification cost to $O(mn)$ from $O(mn^2)$,** where n and m denote the number of talliers and voters, respectively.

Initialization Phase: In this phase, the initialization of the PPVSS scheme has been completed, and each tallier A_i has sampled a secret key $x_i \in \mathbb{Z}_q$ and registered public key $\mathsf{pk}_i = G^{x_i}$. They also agree on pk_0 as the commitment key.

Ballot Casting Phase: Given the public keys of tallies, to casts a vote $v \in \{0,1\}$, a voter acts as below:

1. Samples a random secret value $s \in \mathbb{Z}_q$.
2. Using the public keys of talliers $\{\mathsf{pk}_i\}_{i=0}^n$ and the secret value s, runs the distribution protocol for his PVSS (or our PPVSS) scheme and obtains the encrypted shares and the associated proof π_{PVSS} (or π_{PPVSS}).
3. Computes the value $U = G^{s+v}$ and using the protocol given in Sec. 2.1 constructs a well-formedness proof π_U, showing that indeed $v \in \{0,1\}$, without revealing any information on v. The proof π_U refers to the value of $C_0 = h^s$ (or $y_0 = \mathsf{pk}_0^s$) which is published as part of π_{PVSS} (or π_{PPVSS}) in the (P)PVSS (or PPVSS) distribution protocol, and shows that:

$$\log_G U = \log_h C_0 \vee \log_G U = 1 + \log_h C_0 \text{ (or } \log_G U = \log_{\mathsf{pk}_0} y_0 \vee \log_G U = 1 + \log_{\mathsf{pk}_0} y_0\text{)}.$$

The ballot for voter V consists of the output of the (P)PVSS (or PPVSS) distribution protocol (which consists of encryption of shares and the proof π_{PVSS} (or π_{PPVSS})), the value U (which can be seen as an encryption of the vote), and proof π_U (which proves the well-formedness of U).

Ballot Verification Phase: Once voters submitted their ballots, the ballots are checked by the bulletin board. Note that both proofs π_{PVSS} (or π_{PPVSS}) and π_U are publicly verifiable and can be checked by anyone.

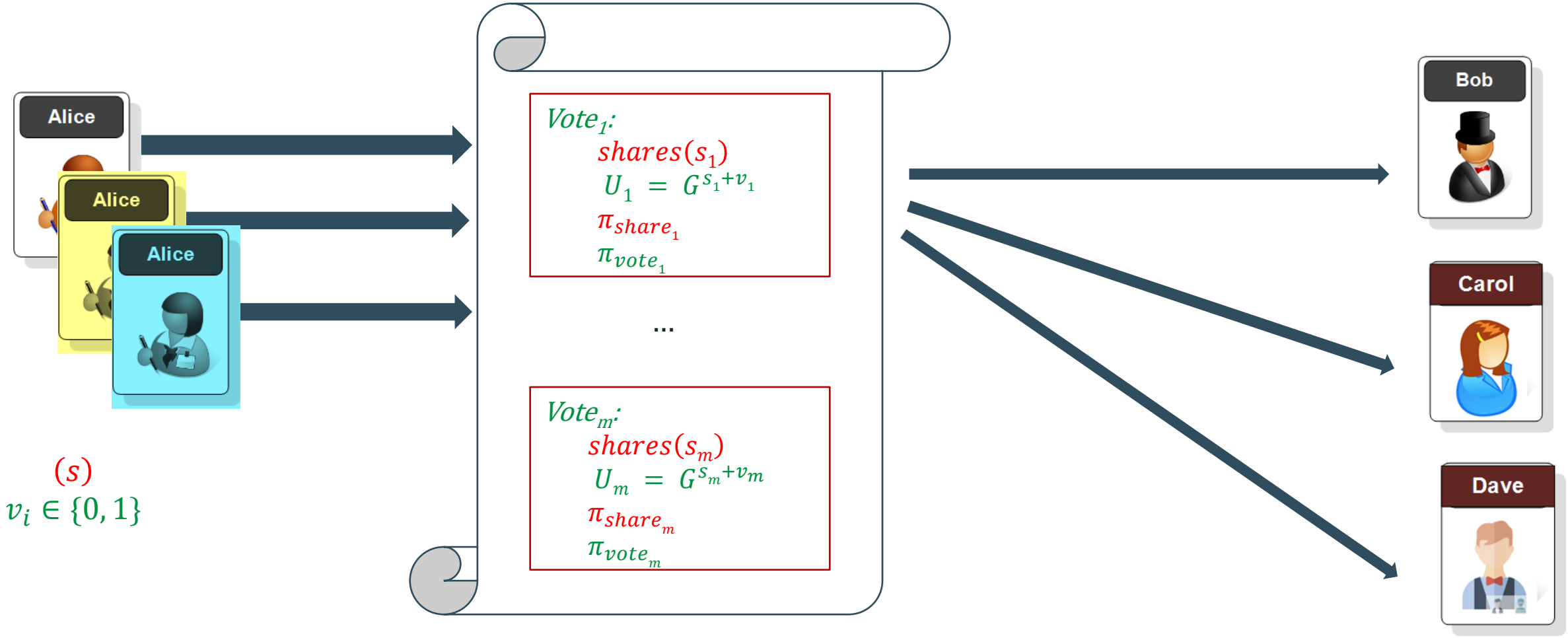
Tallying Phase: Suppose voters $\{V_j\}_{j=1}^m$, have all cast valid ballots. The tallying protocol consists of the following step:

1. Talliers first accumulate all the respective encrypted shares, under the same public keys, by computing the values $Y_i^\star$, where the index j ranges over all voters:

$$Y_i^\star = \prod_{j=1}^m Y_{ij} = y_i^{\sum_{j=1}^m p_j(i)}.$$

2. Next, each tallier A_i applies the reconstruction protocol of Schoenmakers' (P)PVSS (or our PPVSS) scheme to the value $Y_i^\star$, which will produce $S^\star = G^{\sum_{j=1}^m p_j(0)} = G^{\sum_{j=1}^m s_j}$, due to the homomorphic property of the (P)PVSS (or PPVSS) scheme.
3. Using $\{U_j\}_{j=1}^m$, talliers compute $U^\star = \prod_{j=1}^m U_j = G^{\sum_{j=1}^m s_j + v_j}$. Then, using $S^\star$ and $U^\star$ they compute $V^\star = \frac{U^\star}{S^\star} = G^{\sum_{j=1}^m v_j}$. Given $V^\star$, the tally $V = \sum_{j=1}^m v_j, 0 \le V \le m$, can be computed efficiently.

E-Voting Protocol



Tallying Procedure

$Vote_1$:

$shares(s_1)$

$U_1 = G^{s_1+v_1}$

π_{share_1}

π_{vote_1}

...

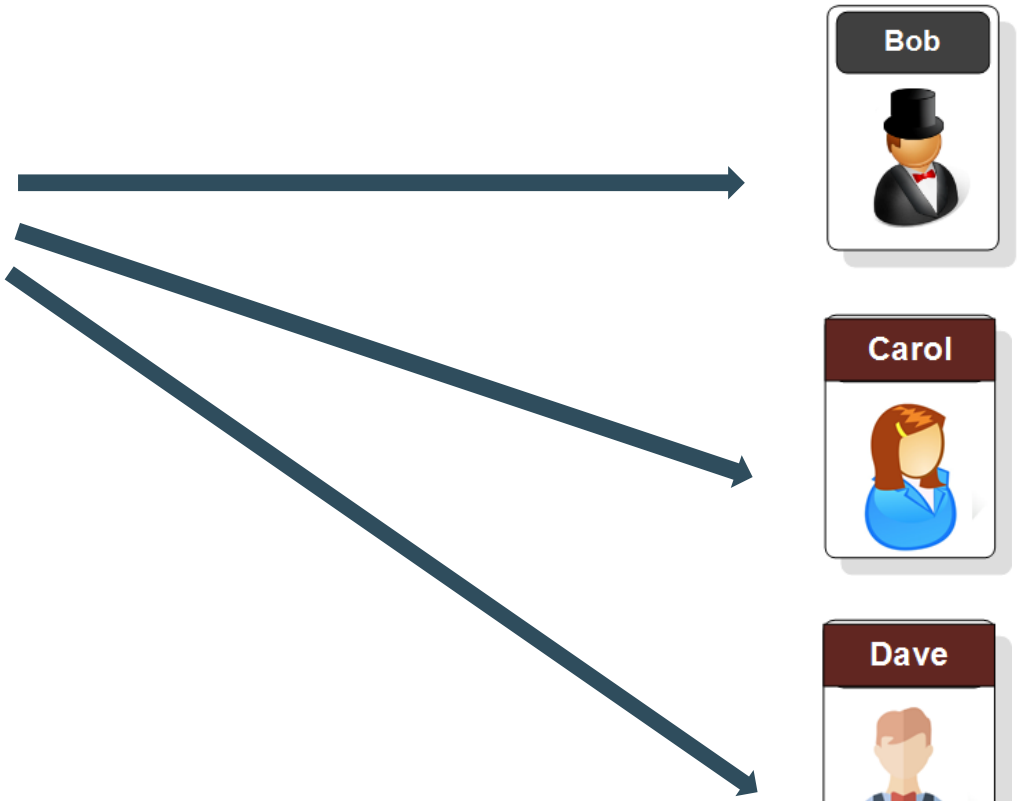
$Vote_m$:

$shares(s_m)$

$U_m = G^{s_m+v_m}$

π_{share_m}

π_{vote_m}



1. *Y* <- Add All Shares*
2. *Reconstruct S* using Y**
3. *U* <- Sum All U Values*
4. $V^* <- \frac{U^*}{S^*} = G^{\Sigma v}$

Empirical Performance: [Sch99] E-voting Protocol vs. Ours

- The new e-voting scheme achieves a remarkable speed up in the ballot verification phase, particularly for the large values of n.
  - **90 × for $n = 512$**
- Our implementation results shows that the new e-voting scheme is practical for even large-scale elections.

Repository: https://github.com/KULeuven-cosic/ppvss-evoting

Table 3. Implementation results of the universally verifiable e-voting protocols proposed in [23] and Section 5 for different types and sizes of elections. n: Number of talliers, t: Threshold value (number of malicious talliers), m: Number of voters, $|\mathbb{Z}_q| = |\mathbb{G}| = 256$ bits, s: seconds, ms: milliseconds, B: Byte, K: Kilo, M: Million.

Election, (n, t, m)	Metrics	Schoenmakers [23]	Our Scheme (Fig. 5)
Election 1 (board-room) (256, 127, 256)	Ballot Casting Time	4.65 ms	2.63 ms
	Voter's Communication	20.28 KB	12.28 KB
	Verification of One Ballot	121.15 ms	2.30 ms
	Tallying Time	11.18 ms	11.18 ms
Election 2 (board-room) (512, 255, 512)	Ballot Casting Time	9.92 ms	6.43 ms
	Voter's Communication	40.28 KB	24.28 KB
	Verification of One Ballot	481.19 ms	5.33 ms
	Tallying Time	27.91 ms	27.91 ms
Election 3, 4, 5 (large-scale) $(17, 8, m)$	Ballot Casting Time	0.56 ms	0.42 ms
	Voter's Communication	1648 B	1104 B
	Verification of One Ballot	1.24 ms	0.41 ms
$m = 50$K	Tallying Time	1.31 s	1.31 s
$m = 100$K	Tallying Time	2.61 s	2.61 s
$m = 1$M	Tallying Time	25.70 s	25.70 s

Conclusions and Future Research

Conclusions and Future Research:

- Introduced Pre-constructed PVSS (PPVSS), an extension of standard PVSS schemes that offers more applications and better efficiency with optimistic reconstruction.
- Proposed a general strategy for building a PPVSS scheme from a Shamir-based PVSS scheme.
- Presented two practical PPVSS schemes in the RO and plain models, based on the state-of-the-art PVSS protocols.
- Revisited a few applicaitons and impved their performance.
  - A novel and practical universally verifiable e-voting protocol.

➢ Future research can explore new applications and new protocols for PPVSS.

✓ We revisited a selected few applications → there are more that can be revisited.

✓ Only focused on DL-based protocol where the secret is g^{f_0}. We believe our results can be applied to other PVSS schemes where the secret is f_0, e.g. [Gro21, CD24] and/or are based on lattices e.g., [GHL22].

✓ We only used one positive feature of PPVSS in our revisited applications, i.e., commitment or encryption of the main secret. One can use the other positive feature of PPVSS, i.e., the optimistic reconstruction, to revisit more protocols.

KU LEUVEN

Thank You!

ia.cr/2025/576

✉ karim.baghery@kuleuven.be