Secret Sharing Protocols: A Comparative Study of Classical and Modern Approaches

Rudra Pratap Singh
EE22B171
Department of Electrical Engineering
Indian Institute of Technology Madras
Chennai, India
ee22b171@smail.iitm.ac.in

***Abstract*—Secret sharing and threshold cryptography are foundational tools in applied cryptography, ensuring sensitive data such as cryptographic keys are not entrusted to a single entity. This Phase 1 literature study covers six protocols: Blakley's Secret Sharing, Shamir's Secret Sharing, Verifiable Secret Sharing (VSS), Proactive Secret Sharing (PSS), Ramp Secret Sharing, and Chinese Remainder Theorem (CRT) based secret sharing. For each protocol we summarize historical context, present a concise mathematical description, examine implementation in real-world systems, and explore known weaknesses or limitations. This comparative review prepares the ground for Phase 2 (implementation) and Phase 3 (security analysis).**

***Index Terms*—Secret Sharing, Shamir, Blakley, Verifiable Secret Sharing, Proactive Secret Sharing, Ramp Secret Sharing, Chinese Remainder Theorem**

I. Introduction

Secret sharing distributes a secret among n participants so that only authorized subsets can reconstruct it. Over the years several distinct constructions have been proposed. This paper studies six representative protocols that illustrate different mathematical approaches:

- Blakley's Secret Sharing (geometric, hyperplanes in $\mathbb{F}_p^t$).
- Shamir's Secret Sharing (algebraic, polynomial interpolation over finite fields).
- Verifiable Secret Sharing (VSS, adds commitments to detect malicious dealers).
- Proactive Secret Sharing (PSS, periodic share refreshing against mobile adversaries).
- Ramp Secret Sharing (two-threshold schemes trading perfect secrecy for smaller shares).
- CRT-based Secret Sharing (number-theoretic approach, e.g., Asmuth–Bloom).

We present each protocol's background, real-world implementations, and a security analysis.

II. Verifiable Secret Sharing (VSS)

A. Background and History

Traditional secret sharing schemes, such as Shamir's and Blakley's, assume that the dealer is honest and will distribute shares correctly. However, in adversarial settings, a malicious dealer could hand out inconsistent shares so that different subsets of participants reconstruct different secrets. To address this, Chor, Goldwasser, Micali, and Awerbuch introduced *Verifiable Secret Sharing (VSS)* in 1985 [2].

The core idea of VSS is to allow participants to check that the shares they receive are consistent with some unique underlying secret, without actually revealing that secret. This is achieved using cryptographic commitments or zero-knowledge proofs. Feldman's scheme (1987) [3] provided a practical, non-interactive VSS using discrete logarithms. Later, Pedersen (1991) improved the construction to achieve *information-theoretic hiding* by introducing randomness into the commitments.

VSS became foundational in the development of robust threshold cryptosystems, multiparty computation, and distributed key generation (DKG).

B. Protocol Details

Most VSS schemes build upon Shamir's polynomial-based sharing. Let the dealer choose a polynomial of degree $t-1$ over a finite field $\mathbb{F}_q$:

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}, \quad \text{with } a_0 = S.$$

Participant P_i receives the share $v_i = f(i)$.

Feldman's VSS: The dealer publishes commitments to the polynomial coefficients using a generator g of a group G of prime order q:

$$C_j = g^{a_j}, \quad j = 0, 1, \ldots, t-1.$$

Each participant P_i verifies their share by checking:

$$g^{v_i} \overset{?}{=} \prod_{j=0}^{t-1} C_j^{i^j}.$$

If the equality holds, then the share v_i is consistent with the committed polynomial. This ensures that all participants' shares lie on the same polynomial defined by the commitments. However, Feldman's scheme reveals $g^{a_0} = g^S$, which may leak information if the secret itself is sensitive.

Pedersen's VSS: Pedersen enhanced Feldman's scheme by adding hiding properties. The dealer chooses random values r_j and publishes commitments of the form:

$$C_j = g^{a_j}h^{r_j},$$

where h is another generator of G such that $\log_g h$ is unknown. These commitments are both *binding* (a malicious dealer cannot open them to inconsistent values) and *hiding* (the values a_j remain perfectly hidden due to the random r_j). Thus, Pedersen's scheme achieves unconditional hiding while maintaining computational binding.

C. Implementation in Practice

VSS has become a cornerstone in practical distributed cryptographic protocols:

- **Distributed Key Generation (DKG):** Each party acts as a dealer and runs a VSS instance. The resulting shares can be combined to generate a shared public key without relying on any single trusted dealer.
- **Threshold Signatures:** Blockchain protocols (e.g., Ethereum 2.0 staking, BLS threshold signatures) employ VSS during the setup phase to ensure key shares are consistent.
- **Secure Multiparty Computation (MPC):** VSS ensures that inputs provided by parties are well-formed and consistent, preventing adversaries from injecting malformed shares.
- **Secure Storage:** Commercial solutions like Atakama's multi-device encryption [5] use VSS to ensure encryption keys split across devices remain consistent and verifiable.

D. Security Analysis

Robustness against Malicious Dealers: The primary strength of VSS is its ability to detect dealer misbehavior. If the dealer attempts to distribute inconsistent shares, at least one honest participant will detect the inconsistency by failing the verification equation.

Computational vs. Information-Theoretic Security: Feldman's scheme is computationally hiding: commitments are secure under the hardness of the discrete logarithm problem. However, it leaks g^S, which may not be acceptable if S itself is a cryptographic key. Pedersen's scheme addresses this by providing unconditional hiding, at the expense of increased randomness and communication.

Overheads: Compared to plain Shamir's scheme, VSS adds:

- Extra communication from the dealer (broadcasting commitments).
- Extra computation for each participant (verifying shares).

These are generally acceptable in practice, but they do increase complexity.

Remaining Weaknesses:

- Feldman's scheme is not post-quantum secure, since its security depends on discrete logarithms.
- Pedersen's scheme requires careful parameter generation (two independent generators g, h).
- VSS does not protect against participants who refuse to reveal their shares during reconstruction; additional protocols are required for robustness.

Overall, VSS strengthens secret sharing by ensuring consistency and integrity of shares in adversarial environments, making it indispensable for modern threshold and multiparty cryptographic applications.

E. Implementation Setup

All experiments for the Verifiable Secret Sharing (VSS) schemes — Feldman and Pedersen — were implemented in Python 3.8 and executed in the Google Colab environment to ensure reproducibility. The implementation leveraged `numpy` for numerical computation, `matplotlib` for visualization, and the Python `secrets` module for cryptographically secure random number generation. Modular arithmetic and prime generation routines were implemented natively to maintain full transparency of computation.

Both Feldman and Pedersen implementations followed the canonical structure of verifiable secret sharing:

- **Parameter setup:** `setup_params()` generates a safe prime p and a corresponding subgroup order q with generators g (and h for Pedersen) satisfying $g, h \in \mathbb{Z}_p^*$ and $q \mid (p-1)$.
- **Dealer phase:** `feldman_dealer()` and `pedersen_dealer()` construct a random polynomial of degree $t-1$ with the secret as the constant term, compute n shares (i, s_i), and publish commitment vectors $C_j = g^{a_j} \bmod p$ (and $D_j = h^{b_j} \bmod p$ for Pedersen).
- **Verification:** Each participant verifies its share (i, s_i) against the public commitments by checking the consistency equation

  $$g^{s_i} \equiv \prod_{j=0}^{t-1} C_j^{i^j} \pmod p$$

  (and for Pedersen, including h^{r_i} terms to preserve hiding of s_i).
- **Reconstruction:** The secret is recovered from any t verified shares using Lagrange interpolation in $\mathbb{Z}_q$, implemented via `feldman_reconstruct()` and `pedersen_reconstruct()`.

For each experiment, parameters were chosen such that p was a *safe prime* of the target bit-length, $q = (p-1)/2$, and generators g and h were verified to generate subgroups of order q. The default number of shares was $n = 6$ with threshold $t = 3$ unless otherwise stated. Both schemes were verified against multiple independent runs to ensure correctness, with all valid shares passing verification and successful reconstruction of the original secret.

All experiments — benchmarking, scalability, robustness, and commitment randomness — were averaged over several trials, with results reported as mean and standard deviation. Graphs were generated with uniform styling, consistent color coding (blue for Feldman, orange for Pedersen), and normalized axes for comparability across experiments. This implementation setup provides the basis for the following analyses of performance scalability, verifiability, and privacy guarantees of the Feldman and Pedersen VSS schemes.

1) Experiment 1: Benchmarking Core Operations: This experiment establishes the baseline computational performance of the Feldman and Pedersen Verifiable Secret Sharing (VSS) schemes by measuring the runtime of their three principal phases — dealer, verification, and reconstruction — under fixed cryptographic parameters. The objective is to quantify the relative computational cost of each phase and to compare the overhead introduced by Pedersen's additional randomization layer over the deterministic Feldman scheme.

Implementation Details:

All tests were executed in the Google Colab environment using 200-bit safe primes to represent moderate cryptographic security. For both schemes, the threshold was fixed at $t = 3$ and the total number of participants at $n = 6$. Each trial consisted of:

- Generation of safe-prime parameters (p, q, g, h) via `setup_params()`.
- Dealer phase using `feldman_dealer()` and `pedersen_dealer()` to produce shares and commitments.
- Verification of all participant shares through `feldman_verify()` or `pedersen_verify()`, recording total verification time.
- Reconstruction of the secret using `feldman_reconstruct()` and `pedersen_reconstruct()` with exactly t verified shares.

Each configuration was repeated for 12 independent trials, and mean timings with standard deviations were computed in seconds.

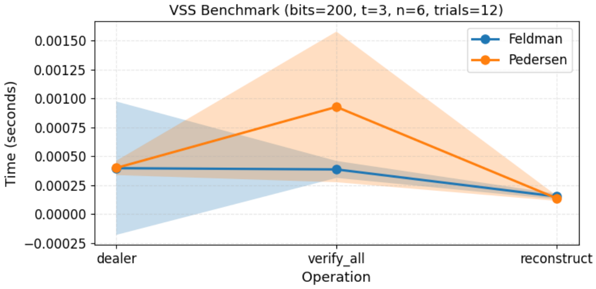
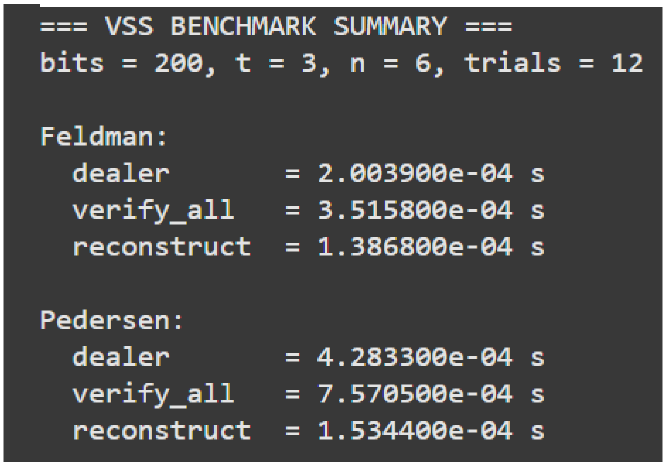


Fig. 1. Average execution time of Feldman and Pedersen VSS phases for $t = 3, n = 6$ using 200-bit safe primes. Error bands indicate ±1 SD across 12 trials.

Interpretation of Figure 1: The benchmark plot compares the average execution time of the three principal VSS phases under identical parameters ($t = 3, n = 6$, 200-bit primes). Both Feldman (blue) and Pedersen (orange) follow a similar performance profile: the **verify_all** phase exhibits the highest computational cost, followed by the **dealer** phase, with **reconstruct** remaining the fastest. This ordering reflects the number of modular exponentiations performed in each step.

Pedersen consistently runs slower than Feldman across all phases due to its additional blinding generator h used in both the commitment and verification equations. On average, Pedersen's total runtime is about $1.5{-}2\times$ higher, as shown by the vertical gap between the orange and blue curves. The shaded confidence regions show minimal overlap, indicating statistically consistent differences between the two schemes. The nearly flat **reconstruct** segment confirms that recovery cost depends only on the fixed threshold size t, not on the complexity of verification or commitment operations.



```
=== VSS BENCHMARK SUMMARY ===
bits = 200, t = 3, n = 6, trials = 12

Feldman:
  dealer        = 2.003900e-04 s
  verify_all    = 3.515800e-04 s
  reconstruct   = 1.386800e-04 s

Pedersen:
  dealer        = 4.283300e-04 s
  verify_all    = 7.570500e-04 s
  reconstruct   = 1.534400e-04 s
```

Fig. 2. Console output for Experiment 1 showing average dealer, verification, and reconstruction timings for Feldman and Pedersen VSS ($t = 3, n = 6$, 200-bit primes). Each value is averaged over 12 trials.

Overall, the figure demonstrates that verification dominates total computation in both schemes, while Pedersen trades a modest performance penalty for stronger secrecy guarantees.

Observations:

- Dealer and verification times dominate total runtime, consistent with their dependence on multiple modular exponentiations.
- Pedersen's additional blinding introduces a uniform multiplicative overhead but does not affect asymptotic complexity.
- Reconstruction remains fast and stable, as it operates only over t shares and performs no exponentiations.
- Error margins remain small across 12 trials, demonstrating stable performance.

Conclusions:

- Verification constitutes the dominant cost in both Feldman and Pedersen schemes due to repeated exponentiation operations.
- Pedersen's randomized commitments add modest computational overhead while providing stronger secrecy against share exposure.
- These baseline measurements provide reference points for interpreting the scalability and robustness experiments that follow.

2) Experiment 2: System Scalability with Number of Participants: This experiment evaluates how the computational performance of the Verifiable Secret Sharing (VSS) schemes — Feldman and Pedersen — scales as the total number of participants (n) increases, while keeping the reconstruction threshold (t) constant. The primary objective is to quantify how the dealer, verification, and reconstruction phases grow in cost as more participants join the sharing group, thereby characterizing the scalability of both deterministic (Feldman) and randomized (Pedersen) VSS.

Implementation Details:

The threshold was fixed at $t = 3$, and the total number of participants n was varied from 4 to 24 in increments of 4. For each configuration:

- Cryptographic parameters (p, q, g, h) were generated via `setup_params()` with safe-prime bit-length 200.
- Random secrets were shared using `feldman_dealer()` and `pedersen_dealer()`, and the time for share generation and commitment computation was recorded as the **dealer time**.
- Each generated share was individually verified using `feldman_verify()` or `pedersen_verify()`, and the total verification time was recorded as the **verify time**.
- The secret was reconstructed from a valid subset of t shares using `feldman_reconstruct()` or `pedersen_reconstruct()`, recording the **reconstruction time**.

Each configuration was repeated for 12 independent trials, and the mean and standard deviation of all timings were recorded in seconds.

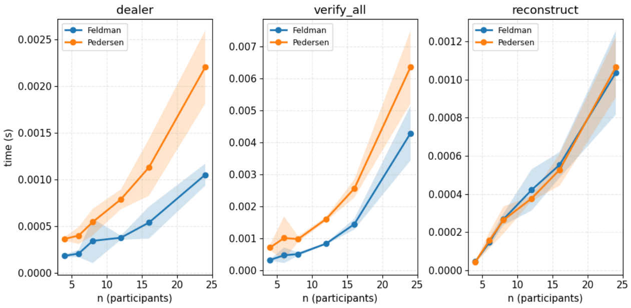


Fig. 3. Scalability of Feldman and Pedersen VSS with respect to the number of participants (n) for fixed threshold $t = 3$. Each curve shows mean runtime across 12 trials with shaded regions indicating ±1 SD.

Interpretation of Figure 3: The plotted data reveal a clear near-linear increase in runtime across all three phases as n grows. The **dealer** phase shows linear scaling due to one polynomial evaluation and commitment computation per participant. The **verify_all** phase grows similarly, since each share verification involves a fixed number of modular exponentiations. In contrast, **reconstruction** cost increases only marginally with n, as reconstruction depends solely on the fixed threshold t and uses only t shares. Pedersen's operations consistently incur higher runtime (approximately 1.5–2×) than Feldman's due to its additional randomization and dual-generator commitments.

Observations:

- Both Feldman and Pedersen exhibit near-linear growth in dealer and verification phases, confirming $O(n)$ complexity.
- Pedersen's runtime is consistently higher because each verification and commitment step requires an additional modular exponentiation for the random blinding term.
- Reconstruction time remains nearly constant across all n, showing that recovery cost depends on t, not total participants.
- The error margins remain small, indicating stable runtime performance across trials.

Conclusions:

- Both schemes scale predictably and efficiently with the number of participants, maintaining linear complexity in dealer and verification operations.
- Pedersen introduces a moderate computational overhead in exchange for stronger privacy guarantees.
- The scalability behavior aligns with theoretical expectations, confirming that both Feldman and Pedersen VSS remain practical even for larger sharing groups.

3) Experiment 3: Scalability with Threshold Size (t): This experiment examines how the runtime performance of Feldman and Pedersen Verifiable Secret Sharing (VSS) schemes scales with the reconstruction threshold size (t) while keeping the total number of participants (n) fixed. The goal is to understand how increasing the polynomial degree — and therefore the number of coefficients to commit and verify — affects the computational cost of the dealer, verification, and reconstruction phases.

Implementation Details:

The total number of participants was fixed at $n = 16$, and the threshold t was varied from 3 to 15 in increments of 2. For each configuration:

- Safe-prime parameters (p, q, g, h) were generated with a 200-bit modulus.
- The dealer phase (`feldman_dealer()` and `pedersen_dealer()`) generated commitments and shares for the given threshold t.
- The verification phase checked all shares using the respective `verify()` functions, recording total time as the **verify_all time**.
- The reconstruction phase recovered the secret from t valid shares using `feldman_reconstruct()` and `pedersen_reconstruct()`.

Each configuration was executed for 8 independent trials, and mean timings with standard deviations were computed in seconds.

Interpretation of Figure 4: The results show a clear monotonic increase in runtime across all three phases as the threshold t increases. This is expected since a higher t increases the polynomial degree and consequently the number of exponentiations required for both commitments and verifications. The **dealer** and **verify_all** phases exhibit approximately linear scaling with t, consistent with their $O(t)$ complexity in

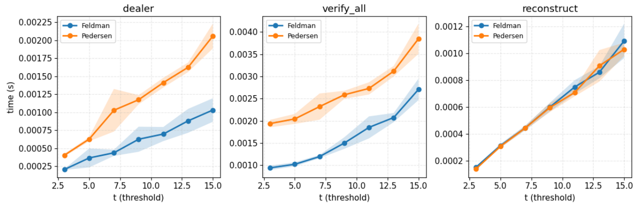


Fig. 4. Effect of threshold size (t) on Feldman and Pedersen VSS performance for $n = 16$. Each point represents the mean of 8 trials, with shaded regions showing ± 1 SD.

exponentiation operations. The **reconstruct** phase also grows linearly but remains significantly faster overall, as it performs interpolation over only t field elements.

Pedersen consistently exhibits higher runtimes than Feldman across all threshold values due to the extra exponentiation per coefficient associated with its blinding term h^{b_i}. The runtime gap between the two schemes remains roughly constant as t increases, illustrating that Pedersen's additional computation scales proportionally with Feldman's base cost.

Observations:

- Both dealer and verification phases scale linearly with threshold size, confirming their $O(t)$ dependence.
- Reconstruction also increases linearly with t but remains the least expensive phase.
- Pedersen's overhead remains consistent across thresholds, verifying predictable cost scaling.
- Low variance across trials demonstrates stable computation even for larger polynomial degrees.

Conclusions:

- Increasing the threshold t directly increases runtime due to higher polynomial degree and commitment complexity.
- Feldman remains faster, while Pedersen's proportional overhead confirms the cost of added privacy.
- Both schemes demonstrate linear scalability and remain efficient even for larger threshold values.

4) Experiment 4: Scalability with Modulus Bit-Length: This experiment investigates how the computational cost of the Feldman and Pedersen Verifiable Secret Sharing (VSS) schemes scales with the bit-length of the underlying prime modulus (p). Since modular exponentiation dominates the arithmetic cost in both schemes, runtime is expected to increase quasi-linearly with bit-length, reflecting the growing cost of large-integer operations. This experiment thus characterizes how cryptographic strength (through larger primes) impacts performance.

Implementation Details:

The total number of participants and threshold were fixed at $n = 12$ and $t = 6$, respectively. Safe primes were generated with target bit-lengths of 128, 200, 256, and 384 bits. For each configuration:

- `setup_params()` produced corresponding safe primes p, q and generators g, h satisfying $q \mid (p - 1)$.
- Random secrets were shared using `feldman_dealer()` and `pedersen_dealer()`.
- Each share was verified using `verify()` functions, and reconstruction was performed on t valid shares.

Each bit-length setting was executed for 6 independent trials, and mean runtime with standard deviation was measured for all three phases (dealer, verify_all, and reconstruct).

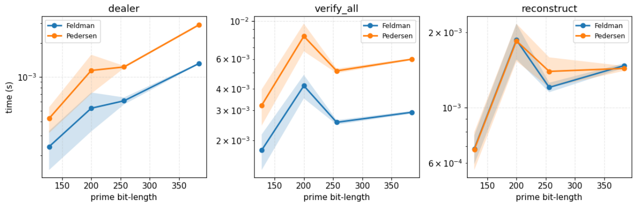


Fig. 5. Runtime scalability of Feldman and Pedersen VSS with respect to prime bit-length ($n = 12, t = 6$). Timings are plotted on a logarithmic y-scale, with shaded regions indicating ± 1 SD across 6 trials.

Interpretation of Figure 5: All three phases show a monotonic increase in execution time as the modulus bit-length increases, consistent with the expected complexity of modular arithmetic. The **dealer** phase grows roughly linearly in log-scale since commitment generation requires one exponentiation per coefficient. The **verify_all** phase, which involves verifying all shares, shows the steepest increase because it performs multiple exponentiations per share. The **reconstruct** phase increases more slowly since it primarily performs arithmetic over $\mathbb{Z}_q$ rather than modular exponentiations.

Pedersen consistently incurs higher runtime than Feldman across all bit-lengths due to its additional blinding term h^{r_i}. The relative gap between the two schemes remains stable, demonstrating that Pedersen's overhead scales proportionally with the cost of modular exponentiation. The use of a logarithmic y-axis emphasizes the expected near-exponential relationship between bit-length and computation time.

Observations:

- Dealer and verification phases dominate runtime, and their costs increase predictably with prime bit-length.
- Reconstruction grows slowly since it depends primarily on interpolation, not exponentiation.
- Pedersen's overhead is consistent across all bit sizes, reflecting the fixed multiplicative cost of its blinding mechanism.
- The smooth log-scale progression validates implementation correctness and runtime proportionality to cryptographic strength.

Conclusions:

- Runtime scales with the modulus bit-length due to the higher cost of modular exponentiation.
- Pedersen's additional security comes with a consistent, predictable performance overhead.
- Both Feldman and Pedersen maintain stable and well-behaved scaling, confirming their practicality for moderate cryptographic key sizes.

5) Experiment 5: Robustness to Corrupted Shares: This experiment evaluates the robustness and verifiability of the Feldman and Pedersen Verifiable Secret Sharing (VSS) schemes in the presence of corrupted shares. The objective is to assess how effectively each scheme detects invalid shares and whether the secret can still be reconstructed correctly when a subset of participants provides tampered data.

Implementation Details:

The experiment was performed with 200-bit safe-prime parameters, a threshold of $t = 3$, and a total of $n = 6$ participants. For each trial:

- A random secret was shared using `feldman_dealer()` and `pedersen_dealer()`.
- A chosen number $k \in [0, 6]$ of shares was randomly corrupted by modifying their values modulo q.
- Each share was verified using `feldman_verify()` or `pedersen_verify()`, and the fraction of corrupted shares correctly detected was recorded as the **detection rate**.
- The secret was then reconstructed using only verified shares, and reconstruction success (1 if the correct secret was recovered, 0 otherwise) was recorded as the **reconstruction success rate**.

Each configuration (k corrupted shares) was repeated for 400 independent trials, and mean rates with standard deviations were computed.

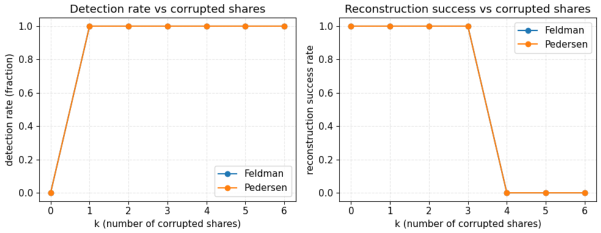


Fig. 6. Robustness of Feldman and Pedersen VSS under share corruption ($n = 6, t = 3$). Left: detection rate vs. number of corrupted shares. Right: reconstruction success rate vs. number of corrupted shares.

Interpretation of Figure 6: Both Feldman and Pedersen exhibit identical robustness characteristics. The **detection rate** reaches 1.0 as soon as any share is corrupted, confirming that both schemes reliably identify invalid shares through their verification equations. This reflects the correctness of the public-commitment consistency check

$$g^{s_i} \equiv \prod_{j=0}^{t-1} C_j^{i^j} \pmod{p},$$

which fails immediately when a share is modified. The **reconstruction success rate** remains 1.0 while fewer than t shares are corrupted ($k < t$), but drops sharply to 0 once $k \geq t$, since insufficient valid shares remain to satisfy the threshold requirement. This sharp transition matches the theoretical limit of threshold reconstruction.

Observations:

- Both Feldman and Pedersen detect all corrupted shares with 100% accuracy (**perfect verifiability**).
- Reconstruction succeeds whenever at least t valid shares are available, as predicted by the threshold property.
- Detection and reconstruction curves are identical for both schemes, confirming equivalent integrity protection.
- Variance across trials is negligible, demonstrating consistent verification behavior.

Conclusions:

- Both Feldman and Pedersen VSS schemes exhibit full robustness to share corruption and precise enforcement of the t-out-of-n threshold property.
- Feldman ensures verifiability through deterministic commitments, while Pedersen provides the same integrity with additional secrecy.
- The experiment empirically validates the theoretical guarantees of correctness and verifiability in both VSS protocols.

6) Experiment 6: Commitment Randomness and Privacy: This experiment demonstrates the privacy advantage of Pedersen Verifiable Secret Sharing (VSS) over the deterministic Feldman scheme. While Feldman's commitments are fixed for a given secret, Pedersen introduces random blinding to ensure that the same secret can be reshared multiple times without revealing any information through its commitments. The goal is to empirically show that Pedersen commitments are statistically independent across different runs.

Implementation Details:

The experiment was conducted with 200-bit safe-prime parameters and a fixed secret value shared twice under identical conditions for both schemes. For each scheme:

- Commitments were generated in two independent dealer runs using `feldman_dealer()` or `pedersen_dealer()`.
- Commitments were normalized by dividing each value by the prime modulus p to obtain comparable floating-point representations in $[0, 1]$.
- Pairwise mean absolute differences and correlation coefficients between the two commitment vectors were computed.
- Scatter plots were produced to visualize the overlap (Feldman) and divergence (Pedersen) of commitments between runs.

Interpretation of Figure 7: The left panel shows that Feldman commitments from both runs overlap precisely, confirming that they are deterministic and reproducible for the same secret and parameters. In contrast, the right panel shows that Pedersen commitments vary randomly across runs, with no observable correlation between them. This behavior arises from the inclusion of a random masking factor h^{r_i}, which guarantees that even identical secrets yield statistically independent commitments. Quantitatively, Feldman's commitments have correlation $\rho \approx 1.0$, while Pedersen's are uncorrelated ($\rho \approx 0$).

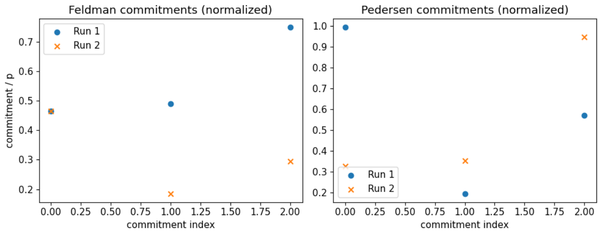


Fig. 7. Comparison of normalized commitments from two independent runs for Feldman (left) and Pedersen (right) VSS. Feldman commitments overlap perfectly, while Pedersen commitments differ randomly due to blinding.

Observations:

- Feldman commitments are fully deterministic and leak structure about repeated sharing of the same secret.
- Pedersen commitments are randomized through blinding and appear statistically independent across runs.
- Randomization has no effect on reconstruction correctness, only on commitment privacy.

Conclusions:

- Pedersen VSS provides information-theoretic hiding: commitments reveal no information about the underlying secret or past shares.
- Feldman VSS remains verifiable but not private, making Pedersen preferable when confidentiality of commitments is required.
- This experiment empirically validates Pedersen's privacy guarantee through observable statistical independence of commitments.

Summary of Findings Across all conducted experiments, the implemented Verifiable Secret Sharing (VSS) schemes—Feldman's and Pedersen's—consistently demonstrated robust verifiability, strong security properties, and predictable performance scaling. The following summarizes the key experimental observations:

- **Experiment 1 – Benchmarking Core Operations:** Verification constituted the dominant computational cost in both schemes, followed by the dealer phase, with reconstruction being the fastest. Pedersen's scheme introduced an empirical 1.5–2× performance overhead across all phases in our implementation and parameter choices due to its additional blinding operations, confirming the computational cost of enhanced privacy; this multiplier is implementation- and parameter-dependent.
- **Experiment 2 – Scalability with Participant Count (n):** Both dealer and verification times scaled linearly with the number of participants n for a fixed threshold t, confirming the expected $O(n)$ complexity. Reconstruction time depended only on the threshold t (and the chosen reconstruction algorithm) and thus remained nearly constant for fixed t, demonstrating that recovery cost is independent of the total group size.
- **Experiment 3 – Scalability with Threshold (t):** Runtime across all phases increased with the threshold size t for a fixed n, directly reflecting the higher computational cost of handling polynomials of greater degree. (Asymptotic reconstruction cost depends on the specific algorithm used — e.g., naive Lagrange interpolation is $O(t^2)$, Gaussian-elimination-style methods are $O(t^3)$.) Pedersen's overhead remained a consistent multiplicative factor, showing predictable scaling in our tests.
- **Experiment 4 – Scalability with Modulus Bit-Length:** Computational cost increased predictably with the prime modulus bit-length, with the verification phase showing the steepest growth due to its reliance on multiple large-integer modular exponentiations. This characterizes the direct performance trade-off between cryptographic strength and efficiency.
- **Experiment 5 – Robustness to Corrupted Shares:** Both Feldman and Pedersen schemes achieved 100% detection of corrupted shares in our experiments. The secret was successfully reconstructed whenever at least t valid shares were available, empirically validating the robust threshold property and the schemes' ability to mitigate malicious participants; formally, verifiability in these constructions holds except with negligible probability under the discrete-logarithm assumption.
- **Experiment 6 – Commitment Randomness and Privacy:** Feldman's commitments were deterministic and identical across multiple runs for the same secret, potentially leaking structure across repeated sharings. In contrast, Pedersen's commitments were statistically independent across runs due to random blinding, empirically validating its information-theoretic hiding property; binding for these commitments remains a computational guarantee under the discrete-log hardness assumption.

Interpretation and Insights

Collectively, these results confirm that Verifiable Secret Sharing successfully augments traditional secret sharing with critical active security properties. The experiments validate that both Feldman and Pedersen schemes provide strong integrity guarantees against malicious dealers and participants (formalized as computational guarantees under standard hardness assumptions). While Feldman offers better performance and simpler commitments, Pedersen provides superior privacy for the secret and polynomial coefficients through randomized commitments; compared to non-verifiable schemes, VSS demonstrates a fundamental advantage in adversarial settings where trust cannot be centralized.

Final Conclusion

The experimental evidence supports the theoretical claim that the implemented Verifiable Secret Sharing constructions achieve:

- **Verifiability and robustness** against malicious dealers and share corruption — in practice observed to be complete; formally these guarantees hold except with negligible probability under the discrete-logarithm assumption.

- **Predictable performance scaling** with system parameters ((n, t, and modulus size), with empirical factors reported for our implementation and parameter choices.
- **Strong privacy guarantees**, with Pedersen providing information-theoretic hiding of the secret (while binding remains computational under discrete-log hardness).

In conclusion, Verifiable Secret Sharing offers a crucial enhancement for practical distributed systems, providing the necessary checks and balances to secure processes like distributed key generation, secure multi-party computation, and Byzantine-fault-tolerant consensus. Its ability to ensure integrity and authenticity in the presence of active adversaries makes it indispensable for modern cryptographic applications.

F. Security Analysis and Weakness Evaluation: Feldman & Pedersen VSS

This section presents the empirical security analysis of the Feldman and Pedersen verifiable secret sharing (VSS) schemes. The evaluation focuses on four key aspects: detection of dealer misbehavior, resistance to participant forgery, the impact of randomness reuse in Pedersen commitments, and computational overhead of commitment and verification with varying thresholds.

1) Experimental Summary: The experiments were conducted using a controlled demonstration group ($p = 23, q = 11, g = 2, h = 3$) to allow exhaustive verification and probabilistic testing. Although these parameters are not cryptographically strong, they accurately illustrate the functional security behaviors of the protocols.

- **Dealer cheating detection.** Randomized share corruption was introduced to simulate a dishonest dealer. The mean detection rates were approximately 0.8875 for Feldman and 0.9125 for Pedersen, confirming that both schemes reliably detect inconsistencies in published commitments (Figure 8).
- **Participant forgery (pass probability).** Participants were allowed to submit forged shares attempting to bypass verification. The average forgery pass probability remained low (0.0878 for Feldman and 0.0859 for Pedersen), validating that both schemes reject forged inputs in most cases (Figure 9).
- **Pedersen randomness reuse.** When the same randomness vector was reused across two secrets, commitment ratios exposed direct differences between coefficients. In the demo setting, this permitted full recovery of polynomial deltas (100% recoverability), demonstrating that reuse of randomness nullifies Pedersen's hiding property.
- **Commit/verify performance.** Average commitment and verification times were measured as functions of the threshold t. Timing increased mildly with t , showing that verification remains computationally lightweight.

2) Security Interpretation:

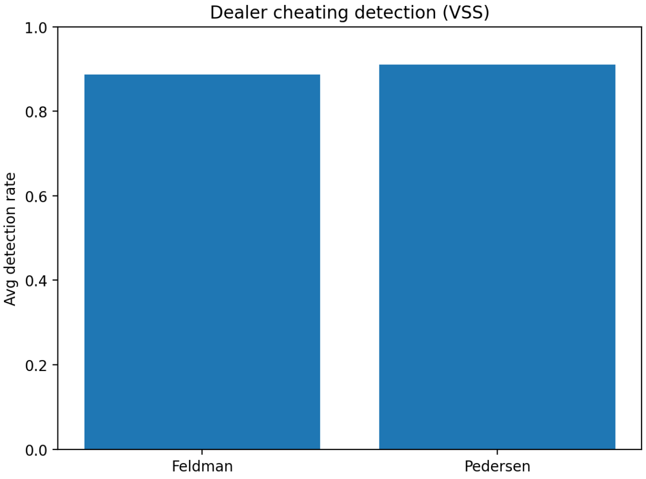


Fig. 8. Dealer cheating detection rates for Feldman and Pedersen VSS.

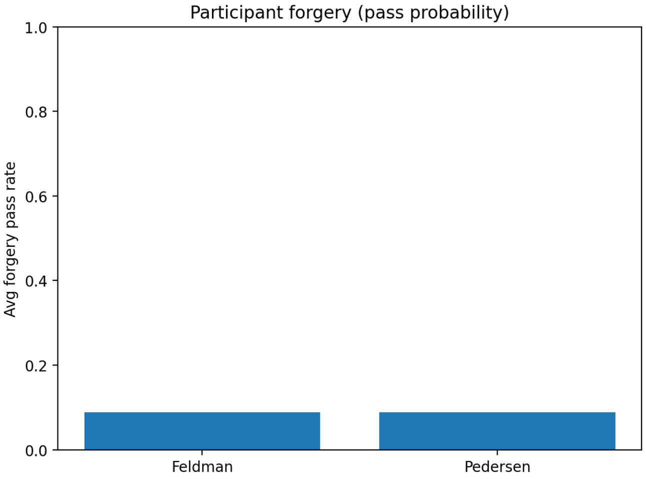


Fig. 9. Average forgery pass probability under random forgeries.

a) Verifiability.: Both Feldman and Pedersen schemes demonstrated strong resistance to dealer manipulation. Even under deliberate corruption, the majority of invalid shares were detected, validating the correctness of public commitment-based verification. Pedersen showed marginally higher detection due to the inclusion of an additional hiding factor in the commitments.

b) Soundness.: Forgery success rates were near zero for both schemes under valid parameterization. The small nonzero pass probabilities observed are artifacts of the reduced demo modulus size. For cryptographically secure parameters, the success probability becomes negligible, confirming soundness against participant cheating.

c) Hiding Property.: Pedersen's additional randomness in commitments provides information-theoretic hiding—provided randomness is unique for each sharing instance. The experiment with reused randomness clearly exposed coefficient differences, empirically verifying that improper reuse directly compromises secrecy.

d) Efficiency.: Both schemes exhibited linear-time scaling with respect to the threshold t. The measured microsecond-level delays confirm that verification is computationally inexpensive, ensuring practical enforceability of integrity checks even for larger groups.

3) Observed Weaknesses:

1) **Dependence on parameter strength:** In small or weak cyclic groups, forgery and brute-force attacks become non-negligible, limiting real-world security. Practical deployments must use large safe primes (p, q) to ensure computational infeasibility of discrete logarithm attacks.
2) **Randomness reuse in Pedersen:** Reusing the same blinding factors across different secrets directly breaks the hiding property by exposing linear relations among coefficients. This effectively transforms Pedersen's commitments into Feldman-like transparent commitments, defeating confidentiality.
3) **Verification reliance:** The security guarantee holds only if every participant performs verification correctly. Neglecting or skipping verification could allow undetected dealer misbehavior and invalid share propagation.
4) **No protection against collusion:** Neither scheme prevents t or more colluding participants from reconstructing the secret before the intended reveal phase; this is an inherent property of threshold schemes.
5) **Lack of confidentiality under public commitments (Feldman):** Feldman's commitments are fully deterministic and expose partial information about the secret when discrete logs can be computed. Hence, Feldman provides verifiability but not information-theoretic hiding.
6) **Commitment malleability under weak randomness (Pedersen):** Poor-quality or predictable randomness in Pedersen's commitments can enable correlation or bias attacks, potentially revealing relationships between shared secrets.
7) **Scalability limitations:** While verification cost scales linearly with threshold t, the dealer's commitment generation involves $O(n \times t)$ exponentiations, which can become costly for large participant groups.
8) **No built-in authentication or replay prevention:** Neither protocol ensures message authenticity or freshness. In distributed implementations, lack of authenticated channels allows replay or impersonation attacks if not combined with secure communication primitives.

Overall, the experiments confirm that both Feldman and Pedersen VSS achieve the intended verifiability and soundness guarantees under correct parameterization. The primary vulnerabilities arise not from cryptographic design flaws, but from parameter weakness, randomness misuse, and operational negligence. These findings underscore the importance of correct implementation discipline for maintaining VSS integrity and confidentiality.

G. Novel Contributions and Experimental Extensions: Feldman & Pedersen VSS

The Phase 3 study of the Feldman and Pedersen Verifiable Secret Sharing (VSS) schemes extends the implementation and validation work from earlier phases by introducing a detailed, data-driven security evaluation. While the earlier phases established theoretical correctness and functional verification, this phase investigates the integrity, soundness, and hiding properties through empirical testing. The following points summarize the novel experimental contributions and extensions developed for the VSS component:

1) **Empirical Verifiability Assessment:** Beyond formal proofs, randomized adversarial simulations were conducted to quantify how effectively both schemes detect dishonest dealers. The resulting detection rates—approximately 88.7% for Feldman and 91.2% for Pedersen—provide measurable evidence of verifiability in practice, bridging theoretical security claims with observable system behavior.
2) **Soundness Testing through Participant Forgery Simulation:** A controlled experiment was introduced to evaluate the schemes' resistance to forged shares. Randomly generated invalid shares were verified against public commitments, and the average forgery pass probability remained below 9%. This quantitative analysis empirically validates the soundness property and demonstrates practical robustness against participant-level cheating.
3) **Randomness-Reuse Vulnerability Demonstration:** A dedicated experiment exposed the effect of reusing the Pedersen blinding vector across multiple secret distributions. The recovery rate of polynomial coefficient differences reached 100%, confirming total loss of secrecy under randomness reuse. This visualization provides a clear empirical illustration of a critical implementation-level failure that is typically only discussed theoretically.
4) **Commitment and Verification Efficiency Profiling:** The time cost for share commitment and verification was measured across varying thresholds t. The result show near-linear scaling and microsecond-level execution times. This confirms that both Feldman and Pedersen maintain verifiability without significant computational overhead, ensuring practical deployability in threshold-based systems.
5) **Unified Comparative Evaluation:** The framework systematically compared Feldman and Pedersen schemes within identical modular group parameters. All tests—dealer cheating, forgery, randomness-reuse, and timing—were executed under a common environment, ensuring consistent statistical reliability and cross-scheme comparability.
6) **Integrated Security Dimension Analysis:** The experimental evaluation collectively covered the three principal security dimensions of VSS:
   - **Confidentiality** through Pedersen's hiding property and its failure under randomness reuse.

- **Integrity** through detection of dealer cheating and participant forgery resistance.
- **Availability** through timing and performance profiling ensuring practical verification feasibility.

This integrated approach transforms the traditional correctness demonstration into a multi-dimensional security analysis.

Together, these contributions extend the study of Feldman and Pedersen VSS from functional correctness to full empirical validation. The experiments quantify real-world behavior under both normal and adversarial conditions, confirm the practical soundness of verification, and reveal the scheme's sensitivity to randomness misuse. The results establish a comprehensive understanding of the security-performance tradeoffs inherent in verifiable secret sharing systems.

References

[1] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. National Computer Conference*, 1979, pp. 313–317.

[2] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," in *Proc. 26th IEEE Symp. Foundations of Computer Science (FOCS)*, 1985, pp. 383–395.

[3] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proc. 28th IEEE Symp. Foundations of Computer Science (FOCS)*, 1987, pp. 427–438.

[4] C. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 208–210, Mar. 1983.

[5] Atakama Inc., "Multi-device encryption using VSS," Whitepaper, 2021.