A Combinatorial Approach to IoT Data Security*

Anandarup Roy[1] Bimal Kumar Roy[1] Kouichi Sakurai[2]
Suprita Talnikar[3]

[1] Indian Statistical Institute, Kolkata, India

[2] Kyushu University, Kyushu, Japan

[3] Radboud University, Nijmegen, The Netherlands

Abstract

This article explores the potential of Secret Sharing-Based Internet of Things (SBIoT) as a promising cryptographic element across diverse applications, including secure data storage in commercial cloud systems (Datachest), smart home environments (encompassing sensors, cameras, smart locks, and smart assistants), and e-health applications (protecting patient data and medical records). Beyond these applications, the paper makes two key contributions: the introduction of a novel cheater identification algorithm designed to verify the correct submission of shares during secret reconstruction, and empirical validation through experimental studies to support the theoretical advancements. This multifaceted approach not only demonstrates the versatility of SBIoT but also proposes innovative mechanisms to enhance security and integrity, contributing to the development of a more robust cryptographic framework.

1 Introduction

The Internet of Things (IoT) refers to a network of interconnected devices, objects, and systems that are embedded with sensors, software, and other technologies to collect and exchange data over the internet. These devices range from smart appliances to industrial machines, enabling communication and automation across sectors like healthcare and smart cities. Verifiable secret sharing (VSS) is crucial for securing sensitive IoT data. In collaborative IoT applications, VSS facilitates secure collaboration while preserving privacy. [6] proposes a privacy-preserving VSS implementation using data splitting and encryption. [11] emphasizes VSS in IoT, particularly in healthcare for patient data protection, enhancing security and reliability in e-health systems. [5] proposes a non-interactive data aggregation scheme using additive secret sharing for mobile user privacy.

*This article expands upon the work presented in the poster "A Combinatorial Approach to IoT Data Security" at IWSEC 2023 [12].

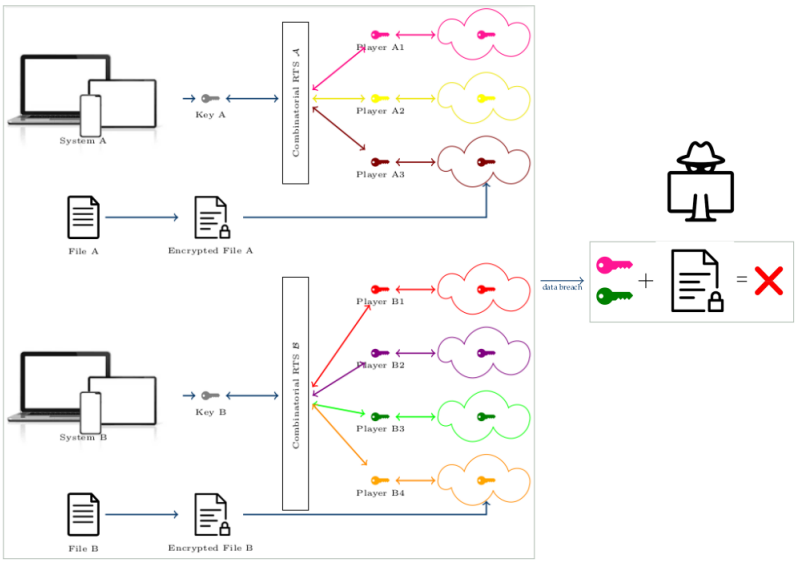
Secret sharing distributes security keys across devices, preventing single-point access. It is lightweight and computationally efficient, offering crucial protection against data modification or deletion in security-sensitive IoT applications.

For instance, [11] presents an AI heuristic decision algorithm using a best-first search (BFS) to balance energy load and reduce communication overhead in smart healthcare. Homomorphic secret sharing in IoT-based e-health offers privacy and security by securely distributing secret pairs among medical nodes, ensuring data confidentiality. This encryption prevents unauthorized data access, limiting access to authorized entities with decryption keys. Homomorphic secret sharing thus protects against unauthorized data modification. Generalization to multiple levels (e.g., combining data between hospitals, states, or countries) can be achieved through the Krönecker product of individual hospital system schemes. The underlying fields provide a basis for the homomorphism, maintained by the integer ring over which the Krönecker product is defined.

A frameproof tensor product of distribution designs is useful for lightweight IoT applications, enabling multi-level secret sharing securely and efficiently, while detecting and preventing data modification. This ensures system security even if some levels are compromised.

The applicability of these generalizations is evident in massive data management, such as in [5], which proposes a non-interactive IoT data aggregation approach using additive secret sharing, addressing privacy, security, communication overhead, and user interaction. Additive secret sharing masks original data, preventing server-side analysis. The scheme supports offline users, maintains privacy, and provides efficient result verification algorithms. However, [5] splits the secret between only two servers. A frameproof tensor product can be applied to connect numerous such systems, due to the underlying fields over which secrets are split and the generalized integer ring over which the tensor product is defined.

The figure below shows an application of tensor design in multi-system IoT. We draw the reader's attention to the applicability of our results from [14] to secret sharing applications on the Internet of Things, especially in a secure, lightweight context.



An application of the tensor product of repairable threshold schemes from [14] in multi-system IoT, where each system (say, a single hospital) may possess a separate RTS for sharing its own secret key, while multiple systems (say, a chain of hospitals) may share their individual secrets to non-colluding cloud storage providers through a tensor product of the individual schemes.

2 Vulnerabilities in Communication Networks

Some models of vulnerability and attacks, as well as malicious behaviors are studied in detail in this domain. We briefly identify them here.

- **Share Distribution Stage:** Anomalies may be introduced during data transfer from the dealer to players.
- **Framing Dynamics:** There may be risks of players framing other players.
- **Malicious Share Insertion:** There may also be threats of false share contributions during the secret reconstruction phase.

Section 3 addresses one such vulnerability: anomaly due to erroneous share distribution. One can use error-correcting codes [10, 16, 3] as well as repair techniques described in [8] to reconstruct faulty shares. [13] explore these concepts in detail.

3 Verifiability in Secret Sharing

Verifiable secret sharing schemes play a crucial role in ensuring the security, privacy, and integrity of sensitive medical data transmitted and stored by IoT devices. They are essential for maintaining the security and privacy of medical data in IoT-based healthcare systems. These schemes enable the distribution of secret pairs among peer medical sensors in a secure manner, ensuring that sensitive information is protected from unauthorized access and malicious nodes. They also help IoT devices establish trust amongst each other, as it is ensured that only authorized devices have access to the shared secrets. Moreover, verifiable secret sharing schemes help maintain the integrity of medical data transmitted and stored by IoT devices and prevent data tampering and unauthorized modifications, ensuring the reliability of the information exchanged between devices.

Existing Verifiable Secret Sharing (VSS) schemes are based on several foundational concepts in cryptography and distributed computing. The security of VSS schemes often relies on one-way functions, which are easy to compute in one direction but hard to invert. This property is used to create verification data that participants can use to check the validity of their shares. Many VSS schemes utilize homomorphic functions, which allow certain operations to be performed on the shares without needing to reconstruct the secret. For instance, Feldman's VSS scheme [4] employs a homomorphic one-way function to verify the consistency of shares. This property is crucial for ensuring that participants can validate their shares without needing to communicate with others. Some other VSS schemes incorporate zero-knowledge proofs to allow participants to prove that they possess a valid share without revealing any information about the share itself. This is particularly useful in scenarios where privacy and confidentiality are paramount. Likewise, many VSS schemes leverage public key cryptography to facilitate secure communication and verification.

Given the resource constraints of many IoT devices, existing VSS schemes are increasingly focused on efficiency in terms of computation and communication. This includes minimizing the amount of verification data required and reducing the computational overhead associated with share generation and verification. Many modern VSS schemes utilize cryptographic hash functions to ensure the integrity and authenticity of shares. Hash functions can be used to create compact representations of shares that can be easily verified.

4 Lightweight Share Verification

Cheater detection in Verifiable Secret Sharing (VSS) ensures that the secret can only be reconstructed by authorized shareholders who possess valid shares. This integrity is crucial in scenarios where the secret is sensitive or valuable, as it prevents malicious players from manipulating the reconstruction process to gain unauthorized access to the secret, or to maliciously sabotage the reconstruction of secrets without making any other gain for themselves. Since shareholders must trust that the shares they receive and use for reconstruction are legitimate, cheater detection mechanisms provide a way to identify and exclude dishonest participants from the reconstruction process. This is particularly important in environments where shareholders may not have prior relationships or trust established. In addition, identifying coalitions of dishonest shareholders attempting to reconstruct the secret using fake or manipulated shares is facilitated, allowing the VSS scheme to maintain its security. There is no doubt that the ability to detect cheaters enhances the overall robustness of the secret sharing scheme against various attacks, including those from insiders who may attempt to compromise the system. This is especially relevant in cloud computing environments. In fact, simply knowing that there are mechanisms in place for cheater detection can deter potential dishonest behavior amongst various participants. For these reasons, cheater detection is a critical component of VSS schemes, as it ensures the security, integrity, and trustworthiness of the secret sharing process, making it suitable for applications in sensitive areas such as finance, healthcare, and cloud computing.

4.1 Existing Verification Protocols

One approach to achieving verifiability is through homomorphic commitment schemes, such as Benaloh's scheme [1]. These allow shareholders to verify that all shares are collectively consistent without revealing the secret. However, this method requires interactive proofs to ensure the dealer's integrity, which can complicate the process and make it less practical. Another method by [7] involves verifying the coherence of shares by comparing the secrets reconstructed from different subsets of players. If all subsets yield the same secret, it indicates no cheating has occurred. This method requires a coalition of players larger than the threshold to effectively detect cheaters, which can be a limitation in smaller groups.

The verification algorithms of [2] are designed to be space-efficient, meaning they do not require the storage of public data for verification, which reduces the overhead typically as-

sociated with secret sharing schemes. The proposed schemes can be used in conjunction with arbitrary secret sharing schemes and provide mechanisms for detecting cheaters among shareholders. Consequently, the design emphasizes robustness against cheaters by implementing verification routines that ensure the legality of shares independently from the secret they are generated from, unlike traditional homomorphic commitment schemes.

The Delegated Proof of Secret Sharing (DPoSS) consensus protocol proposed by [6] introduces several mathematically grounded contributions that address the challenges inherent in IoT environments. It optimizes the consensus process by leveraging secure multiparty computation (MPC) techniques [9, 17], specifically through the use of Shamir's Secret Sharing (SSS) [15]. The protocol employs a randomized selection algorithm to elect nodes for block packing, which can be mathematically represented as a uniform distribution over the set of eligible nodes. Let N be the total number of nodes, and let $S \subseteq N$ be the subset of nodes eligible for selection. The probability $P(i)$ of node i being selected is given by $P(i) = \frac{1}{|S|} \quad \forall i \in S$. This ensures that each node has an equal chance of being selected, thereby promoting fairness and reducing the risk of centralization, as no single node or small group of nodes can dominate the selection process. Further clarification of this process may be found in [6].

DPoSS incorporates verifiable secret sharing (VSS) by splitting the secret s into shares $s_1, s_2, \ldots, s_n$ using a polynomial $f(x)$ of degree $k-1$ such that $f(0) = s$. Each share s_i is computed as $s_i = f(x_i)$ for distinct x_i values. The reconstruction of the secret requires at least k shares, ensuring that any coalition of fewer than k nodes cannot derive any information about s. This framework provides a robust mechanism for protecting sensitive data in the IoT context. Furthermore, the authors propose a modular architecture that allows for the integration of various secret sharing schemes, which can be mathematically represented by defining a set of secret sharing functions $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$, where each function f_j corresponds to a different secret sharing scheme. The protocol can dynamically select f_j based on the specific requirements of the application, thus enhancing its versatility and efficiency across diverse IoT environments.

In particular, the key sharing protocol proposed in [6] incorporates VSSThe verification process involves a commitment phase in which, the dealer commits to the polynomial $f(x)$ by sending a commitment $C = \text{Commit}(f(x)) = (f(0), f(1), \ldots, f(k-1))$ to the nodes. This commitment can be done using cryptographic techniques such as hash functions or homomorphic encryption. It also involves a share verification phase in which, each node that receives a share s_i can verify it by checking if $s_i = f(x_i)$. If the share does not match the polynomial evaluation, the node can reject it and request a new share. The protocol ensures that no information about the secret is revealed unless k shares are combined, and that nodes can verify the correctness of the shares they receive

The consensus protocols of [6] based on Verifiable Random Functions (VRFs) (which are a cryptographic primitive that produce a pseudorandom output from a given input, along with a proof that the output was generated correctly) ensure that the output appears random to anyone who does not know the secret key, allows any entity to verify that the

output was generated correctly from the input and the secret key, and guarantees that for each input, the output is unique. Let K_s be the secret key and K_v be the verification key. The VRF consists of a key generation algorithm that generates a key pair (K_s, K_v) using a secure key generation algorithm, an evaluation algorithm such that for an input X, the output is $(Y, \pi) = \text{VRF\_Eval}(K_s, X)$ (Y is the pseudorandom output, and π is the proof of correctness), and a verification algorithm, which given (Y, π), checks if $\text{VRF\_Verify}(K_v, X, Y, \pi) \rightarrow$ True/False.

While the DPoSS protocol presents several advantages, it also has some drawbacks that can impact its implementation and performance in practical scenarios. DPoSS relies on the distribution of secret shares among nodes, which can lead to increased communication overhead, especially in large networks. The effectiveness of the secret sharing scheme is contingent upon the threshold k. If the number of malicious nodes exceeds $n-k$, the protocol's security can be compromised. The process of collecting shares, performing polynomial interpolation, and reconstructing the secret can introduce latency in reaching consensus. The time taken for nodes to communicate and verify shares can delay the block packing process, which may not be suitable for applications requiring real-time or near-real-time processing. Moreover, DPoSS assumes that a certain proportion (majority) of players are honest to function correctly. Additionally, the protocol's fault tolerance is inherently linked to the robustness of the underlying secret sharing scheme, which may not be sufficient in all scenarios.

Evidently, many existing VSS schemes face challenges such as requiring multiple rounds of communication (which can be inefficient and impractical in real-world applications), needing large numbers of polynomials or additional verification data, inadequate robustness against collusion among dishonest participants, etc. It is clear that despite significant progress in the field of verifiable secret sharing, challenges remain in terms of efficiency, robustness, and practicality. There is hence, a need for continued research to develop more effective and secure VSS schemes that can be applied in real-world scenarios.

4.2 An Improved Cheater Detection Algorithm

Recall how [14] uses two Shamir schemes on points $x_1, \ldots, x_{v_1}$ of a BIBD $\mathcal{A}$, and $y_1, \ldots, y_{v_2}$ of a BIBD $\mathcal{B}$ to construct a tensor design $\mathcal{F}(\mathcal{A}, \mathcal{B})$, which is frameproof.

We shall now describe a verification protocol not based on hash functions. This protocol has a better computation complexity than standard hash-based verifiers and is moreover based on simple algebraic functions.

- The dealer chooses a (not very large) prime p such that none of $x_1, \ldots, x_{v_1}$, $y_1, \ldots, y_{v_2}$ are divisible by p, and declares p beforehand.
- He also produces a chart of inverse pairs (a, a^{-1}) $\forall a \in \mathbb{Z}_p$.
- The dealer then runs the share generation algorithm as described in [14].

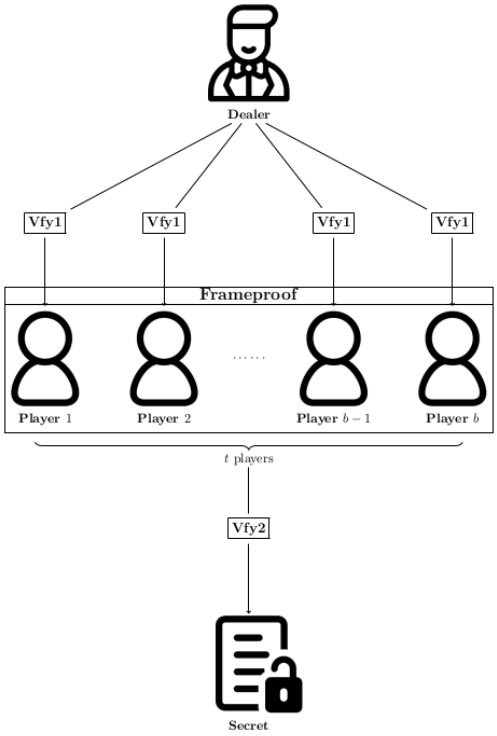
- He then computes $\sum_{i,j} x_i^{-1} y_j^{-1} \pmod{p}$ for each player P, where the share of P consists of elements x_i from $\mathcal{A}$ and y_j from $\mathcal{B}$.
- Finally, the dealer attaches this value to each share and distributes the shares to the participants.
- The secret reconstruction is done by an authorised collection of participants as in [14] and the verification can commence in this phase.

Complexity: In short, our computation complexity comes out to be $\mathcal{O}(n)$ – which is better than the previous $\mathcal{O}(n \log n)$.

Residue computation: at most $\mathcal{O}(\log^2 n)$.

Summation: $\mathcal{O}(n)$.

The storage space required is at most $p - 1$. However, the communication size increases.



5 Conclusion

In this article, we have demonstrated the broad applicability of the proposed scheme from [14] and the novel verification protocol in a variety of IoT contexts. This work underscores the critical need for ongoing research to develop more robust and secure VSS schemes suitable for

real-world deployment. A promising avenue for future exploration involves in-depth analysis of specific use cases.

Cheater detection is a cornerstone of VSS, ensuring that only authorized shareholders with valid shares can reconstruct the secret. This integrity is paramount in scenarios where the secret carries significant value or sensitivity, as it safeguards against malicious attempts to manipulate the reconstruction process. We have discussed several existing VSS verification protocols that have explored various approaches. Homomorphic commitment schemes, such as Benaloh's, provide one method. Harn and Lin's technique focuses on verifying share coherence. Cafaro and Pellè's algorithms prioritize space efficiency by eliminating the need for public data storage during verification, thereby reducing overhead. The DPoSS consensus protocol offers mathematically sound solutions to the unique challenges posed by IoT environments.

Building upon these foundations, we have introduced an improved cheater detection algorithm that departs from traditional hash-based verification methods. This algorithm exhibits superior computational efficiency while relying on simple algebraic operations. Although it reduces storage requirements, it comes at the cost of increased communication overhead. This article lays the groundwork for further advancements in VSS, with a particular focus on enhancing security, efficiency, and practical implementation for IoT applications.

References

[1] Josh Cohen Benaloh. Secret Sharing Homomorphisms: Keeping Shares of A Secret Sharing. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 1986.

[2] Massimo Cafaro and Piergiuseppe Pellè. Space-Efficient Verifiable Secret Sharing Using Polynomial Interpolation. *IEEE Trans. Cloud Comput.*, 6(2):453–463, 2018.

[3] Dingding Dong, Nitya Mani, and Yufei Zhao. On the Number of Error Correcting Codes. *Comb. Probab. Comput.*, 32(5):819–832, 2023.

[4] Paul Feldman. A Practical Scheme for Non-interactive Verifiable Secret Sharing. In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 427–437. IEEE Computer Society, 1987.

[5] Yanxia Fu, Yanli Ren, Guorui Feng, Xinpeng Zhang, and Chuan Qin. Non-Interactive and Secure Data Aggregation Scheme for Internet of Things. *Electronics*, 10(20), 2021.

[6] Tieming Geng, Laurent Njilla, and Chin-Tser Huang. Delegated Proof of Secret Sharing: A Privacy-Preserving Consensus Protocol Based on Secure Multiparty Computation for IoT Environment. *Network*, 2(1):66–80, 2022.

[7] Lein Harn and Changlu Lin. Detection and Identification of Cheaters in (t , n) Secret Sharing Scheme. *Des. Codes Cryptogr.*, 52(1):15–24, 2009.

[8] Bailey Kacsmar and Douglas R. Stinson. A Network Reliability Approach to the Analysis of Combinatorial Repairable Threshold Schemes. *Adv. Math. Commun.*, 13(4):601–612, 2019.

[9] Yinghui Luo, Xiaoshi Deng, Yilin Wu, and Junhuan Wang. Mpc-dpos: An Efficient Consensus Algorithm Based on Secure Multi-Party Computation. In *Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*, pages 105–112, 2019.

[10] Alain Poli. Some Algebraic Tools for Error-Correcting Codes. In Jacques Calmet, editor, *Algebraic Algorithms and Error-Correcting Codes, 3rd International Conference, AAECC-3, Grenoble, France, July 15-19, 1985, Proceedings*, volume 229 of *Lecture Notes in Computer Science*, pages 43–60. Springer, 1985.

[11] Amjad Rehman, Tanzila Saba, Khalid Haseeb, Souad Larabi Marie-Sainte, and Jaime Lloret. Energy-Efficient IoT e-Health Using Artificial Intelligence Model with Homomorphic Secret Sharing. *Energies*, 14(19), 2021.

[12] Anandarup Roy, Bimal Kumar Roy, Kouichi Sakurai, and Suprita Talnikar. A Combinatorial Approach to IoT Data Security, 2023.

[13] Anandarup Roy, Bimal Kumar Roy, Kouichi Sakurai, and Suprita Talnikar. Access Structure Hiding Verifiable Tensor Designs. *Journal of Statistics and Applications, Special Issue in Memory of Prof. C R Rao*, 22(3):535 – 554, 2024 (New Series).

[14] Bimal Kumar Roy and Anandarup Roy. IoT-Applicable Generalized Frameproof Combinatorial Designs. *IoT*, 4(3):466–485, 2023.

[15] Adi Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, 1979.

[16] Yu Dong Yao and Shixin Cheng. Generalization of Hadamard Matrices and a Class of Two-Dimensional Error-Correcting Codes. In *IEEE International Conference on Communications: Integrating the World Through Communications, ICC 1986, Toronto, Canada, June 22-25, 1986, Proceedings*, pages 997–1001. IEEE, 1986.

[17] Hanrui Zhong, Yingpeng Sang, Yongchun Zhang, and Zhicheng Xi. Secure Multi-Party Computation on Blockchain: An Overview. In Hong Shen and Yingpeng Sang, editors, *Parallel Architectures, Algorithms and Programming - 10th International Symposium, PAAP 2019, Guangzhou, China, December 12-14, 2019, Revised Selected Papers*, volume 1163 of *Communications in Computer and Information Science*, pages 452–460. Springer, 2019.