Novel Verifiably Distributed Multi-User Secret Sharing schemes

Likang Lu[1] and Jianzhu Lu[2]

[1] Collece of Computer science · Collece of Software, Inner Mongolia University, Hohhot, 010021, China 1213279727@qq.com
[2] Department of Computer Science, Jinan University, Guangzhou, 510630, China tljz@jnu.edu.cn

Abstract. Distributed secret sharing techniques, where a specific secret is encoded into its shares which are conveyed to the IoT device or its user via storage nodes, are considered. A verifiably distributed secret sharing (VDSS) provides a way for a legitimate user to verify the secret he reconstructs through the downloaded shares while the secrecy condition is satisfied in a weak or a perfect sense. This article examines the impact of minimizing verification information in a VDSS on the communication complexity and storage overhead, and achieves the verifiability in resource-limited IoTs by aggregating the verification information of different devices/users. Then, two secure VDSS are proposed. The first VDSS attains the lower bound on the communication complexity while providing the fault tolerance. The second VDSS simultaneously achieves the lower bounds of both communication complexity and storage overhead while providing the balanced storage load, thus showing the scheme that is optimal in terms of both parameters.

1 Introduction

Secret sharing is an important cryptographic primitive and it is used in many real-world applications,such as electronic voting [1], cloud computing [2], key management in sensor networks [3], and secure storage [4]. Secret sharing is a method by which a dealer distributes shares to participants such that only authorized subsets of participants can reconstruct the secret. It offers a real information-theoretic security with simplest usability. In a distributed secret sharing (DSS), all communication between the dealer and a user who is not directly connected to it, must pass through storage nodes in the network [5]. In the resource-limited applications, such as IoTs based distributed healthcare system [6], the design of a DSS faces the challenge of achieving efficient resource utilization while satisfying the secrecy condition in a weak or a perfect sense. In addition, the ability to verify the correctness of both secrets and their shares is desirable for users in [7] and [8].

In this article, we extend the distributed multi-user secret-sharing schemes [4] and present a notion of VDSS. The notion of a VDSS differs from a DSS protocol [4], in that the secret is recognizable and that the shares should be verifiable as

authentic. In a VDSS, the dealer has no direct communication with users and conveys the shares of a secret and its verification information (VI) to a user via storage nodes, while not only storage nodes do not communicate with each other but also the users neither communicate with each other. The benefit of a VDSS is that a user can check the correctness of a reconstructed secret. Providing this benefit for resource-limited users in a DSS is composed of two problems: the VI generation problem and its storage allocation problem. These schemes are computationally secure since the ability to detect and identify cheaters is based on computational intractability assumptions. There are some schemes which are information-theoretically secure. Pedersen [12] used a commitment scheme to remove the assumption in Feldman's scheme [9] to propose a VSS scheme which is information-theoretically secure. Based on an error-correcting code, Bhndo et al. [13] proposed their schemes in which fake shares can be detected as error codes and corrected based on coding technique. McEliece and Sarwate [14] suggested constructing a secret sharing scheme based on Reed-Solomons code. The two-type schemes need to set different verification information (VI) for different users. This makes it infeasible for a VDSS to attain their optimal storage overheads defined in Section 2.2.

In a multi-user environment, the efficiency of a VDSS is one of the most important issues in its applications. There are two general types of efficiency measures: storage overhead and communication complexity (see Section 2.2). In a communication-optimal VDSS, each user downloads exactly one symbol from the storage nodes in its access set. Note that a short VI can reduce the storage requirements at the storage nodes. In particular, by aggregating VI of different users, resulting storage overhead can be orders of magnitude smaller than approaches shortening the size of VI to reduce the storage overhead. We have investigated a Nyberg's one-way accumulator (NOWA, see Section 4.1) having the property of absorbency in addition to the one-way and quasi-communicative properties [15]. The structure of a NOWA is very suitable for the aggregation of VI since it needs only one secret key of the sealer and accumulated items are dynamic, without the shared key between parties. Then, the efficient schemes related to VDSS are proposed in this article.

1.1 Our Results

We first consider the problem of VDSS through a generalized distributed secret sharing (GDSS) scheme under a certain number of storage nodes. Then, given m users and $n-1$ storage nodes, we provide a GDSS scheme with the fault tolerance when m is a binomial coefficient of $n-1$, and prove that it is perfectly secure. By combining GDSS with a NOWA, we further construct a secure VDSS that is communication-optimal, that is, it achieves the minimum possible communication complexity. Finally, we present a secure VDSS which yields optimality of both the communication complexity and the storage overhead while providing the balanced storage load.

The rest of the article is structured as follows. The problem statement is presented in Section 2. Section 3 proposes a GDSS scheme. Section 4 describes

a communication-optimal VDSS. Section 5 presents a secure VDSS achieving optimality of both the communication complexity and the storage overhead with the balanced storage load. The article is concluded in Section 6.

2 Problem Statement

2.1 Model Constituents

Let q be a large integer, $\mathbb{Z}_q = \{0, 1, \cdots, q-1\}$, and $\mathbb{Z}_q^* = \mathbb{Z}_q \setminus \{0\}$. We denote by $[n-1]$ the set $\{1.2. \cdots, n-1\}$, and we denote by k the secret key of the dealer, where $n, k \in \mathbb{Z}_q^*$. Assume that $\lfloor x \rfloor$ is the integer part of the rational number x, and $\lceil x \rceil$ is the smallest integer not less than x. Let $|x|$ denote the length of the string x, and $|S|$ be the number of elements in the set S. We consider a VDSS system whose principal constituents are depicted in Figure 1.

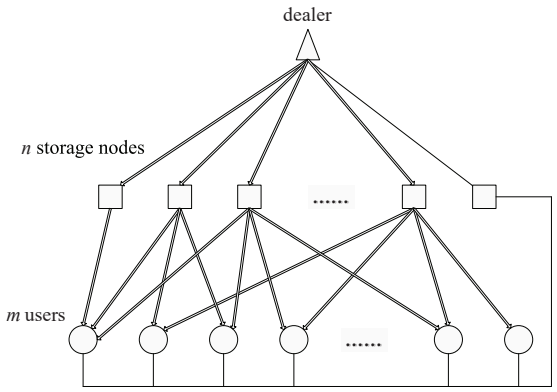

Fig. 1. The system model, where a block arrow denotes a secure channel, and the line denotes an authentic channel.

Parties. Consider a dear, a set of n storage nodes, and a set of m users, where $m > n$. The goal of this system is to enable the dealer to securely convey a specific secret to the user via storage nodes. The dealer has access to all the storage nodes but it does not have direct access to the users. Storage nodes are passive, that is, these nodes do not communicate with each other. The n-th storage node is designed for public access, which stores the VI of users to check the authenticity of both secret shares and the secrets. The remaining storage nodes, $[n-1]$, form the access structure $\mathscr{A}$ according to the access set A_j of user j. That is,

$$\mathscr{A} = \{A_j \subseteq [n-1] : j, j' \in [m], A_{t_j,j} \nsubseteq A_{j'}, \forall j \neq j'\}. \tag{1}$$

where $A_{t_j,j}$ is any t_j-subset of A_j associated with the threshold value t_j. Also, all users do not communicate with each other.

Communication Channel. In the system, there are two types of channels: an authentic channel and a secure channel. The shares of a secret are transmitted via secure channels between the storage nodes and either the dealer or the user, while the VI is sent over authentic channels between them. Note that secure channels guarantee both the authenticity and confidentiality of shares, and authentic channels only guarantee the authenticity of transmitted information but not its confidentiality. For instance, the classic telephone channel is an authentic channel, and the secure socket layer (SSL) which uses port 443 provides a secure channel.

Secret Storage. Let s_j be the secret of user j, and n_j be the number, $|A_j|$, of elements in A_j. We consider a (t_j, n_j) Shamir's secret sharing scheme consisting of a pair of efficient algorithms $(\mathcal{E}(A_j, s_j, t_j), \mathcal{D}_j)$, which is used for user j. Here, $\mathcal{E}(A_j, s_j, t_j)$ is a probabilistic algorithm that is invoked by the dealer outputting the shares, $\mathbf{u}_j$, of s_j; and $\mathcal{D}_j$ is a deterministic algorithm that is invoked by the user j as $s_j \leftarrow \mathcal{D}_j(\mathbf{u}_j)$ to recover s_j. To model the secret storage of a user about how the dealer is associated with the different inputs of the storage nodes in $[n-1]$, we introduce a storing matrix $\mathbf{Z} = [z_{i,j}]_{(n-1)\times m}$ such that $z_{i,j} = 1$ if the share $u_{i,j}$ of user j is stored in the i-th storage node, otherwise $z_{i,j} = 0$. For instance, when $n = 6$, m=10, A_5=$\{a_{1,5}, a_{2,5}, a_{3,5}\}$=$\{5, 1, 2\}$, t_5=3, define $\mathcal{E}(A_5, s_5, 3) \triangleq (u_{a_{1,5},5}, u_{a_{2,5},5}, u_{a_{3,5},5}) = (u_{5,5}, u_{1,5}, u_{2,5})$. Then, $u_{5,5}$, $u_{1,5}$ and $u_{2,5}$ are stored in the 5th, 1st and 2nd storage nodes, respectively. We thus have that the fifth column of the matrix $[z_{i,j}]_{5\times 10}$ is $(1, 1, 0, 0, 1)^T$.

Download Verification. Let d be a positive integer. For the purpose of verifying the shares downloaded from storage nodes, we introduce a pair of algorithms (Prf_j, Ver_j) as follow: $Prf_j : \mathbb{Z}_q^* \times \mathbb{Z}_q^h \rightarrow \mathbb{Z}_q^d$ is invoked by the dealer to generate a proof of symbols outputted by algorithm $\mathcal{E}$, and $Ver_j : \mathbb{Z}_q \times \mathbb{Z}_q^c \rightarrow \{0, 1\}$ is invoked by the user j to validate the downloaded symbols and the reconstructed secret. For instance, for j=5, A_5=$\{5, 1, 2\}$ and t_5=3, let $\mathbf{u}$=$(u_{5,5}, u_{1,5}, u_{2,5}) \xleftarrow{\$} \mathcal{E}(A_5, s_5, 3)$, and $\mathbf{I} \leftarrow Prf(k, (s_5, \mathbf{u}))$. Here, the proof $\mathbf{I}$ satisfies that $Ver_5^{(1)}(\mathbf{u}, \mathbf{I})$=1 and $Ver_5^{(2)}(s_5', \mathbf{I})$=1, where $s_5' = \mathcal{D}_5(\mathbf{u})$.

2.2 Definition and Related Concepts

Definition of a VDSS We first defines a GDSS scheme, which can be viewed as an extension of DSSP in [4]. Then it is extended to describe a VDSS scheme.

Definition 1. *A GDSS scheme is a bundle of* $(\mathscr{A}, \mathcal{E}, \mathbf{Z}, \mathcal{D})$*, where*

i) $\mathscr{A}$ *is the access structure as defined in* (1)*. Let* $\mathbf{t} = (t_1, t_2, \cdots, t_m)^T$ *be the vector of all threshold values.*

ii) $\mathcal{E} : \mathscr{A} \times \mathbb{Z}_q^m \times [n-1]^m \rightarrow \mathbb{Z}_q^h$ *is an encoding function, for some* $h \geq m$*, which relates to the storage overhead of the system. The inputs to the encoding function* $\mathcal{E}$ *consist of* $\mathscr{A}$*,* $\mathbf{t}$*, and* $\mathbf{s} = (s_1, s_2, \cdots, s_m)^T$ *as the vector of all secrets. The output* $\mathbf{u} = \mathcal{E}(\mathscr{A}, \mathbf{s}, \mathbf{t})$ *is the vector of all shares, also called*

the encoding symbols or symbols, to be distributed and stored in the storage nodes. Here, $\mathbf{u} = (u_1, u_2, \cdots, u_h)^T$.

iii) $\mathbf{Z} = [z_{i,r}]_{(n-1)\times h}$, *where* $z_{i,r} = 1$ *if* u_r *is stored in the* i*-th storage node, otherwise* $z_{i,r} = 0$. *The* storing matrix $\mathbf{Z}$ *specifies which the symbols are stored at each storage node, is referred to as the* storage profile. *Let* $\mathbf{u}_{t_j,j}$ *denotes the vector of all data stored in nodes indexed by elements of the subset* $A_{t_j,j}$ *in* A_j.

iv) $\mathcal{D}$ *is a collection of* m *decoding algorithms* $\mathcal{D}_j : \mathbb{Z}_q^{\mathbf{u}_{t_j,j}} \to \mathbb{Z}_q$ *for* $j \in [m]$, *such that* $\mathcal{D}_j(\mathbf{u}_{t_j,j}) = s_j$. *In other words, each user is able to successfully reconstruct its own secret. This is referred to as the* correctness condition.

Again, the *weak secrecy condition* in (2) or *perfect secrecy condition* in (3) is also satisfied in an information-theoretic sense, where $\mathbf{s}_{-j} = (s_1, \cdots, s_{j-1}, s_{j+1}, \cdots, s_m)$.

$$\forall j, j' \in [m], j' \neq j : H(s_{j'}|\mathbf{u}_j) = H(s_{j'}), \tag{2}$$

$$\forall j \in [m] : H(\mathbf{s}_{-j}|\mathbf{u}_j) = H(\mathbf{s}_{-j}), \text{ or } I(\mathbf{s}_{-j}, \mathbf{u}_j) = 0 \tag{3}$$

where $H(\cdot)$ denotes Shannon entropy and $I(\cdot,\cdot)$ denotes the mutual information [3].

We now define the verifiability of all secrets and their shares. To achieve the verifiability, we define a proof function and its output by a vector in $\mathbb{Z}_m^d$ for a positive integer d.

Definition 2. *Let* d *be a positive integer. A verifiably distributed secret sharing (VDSS) scheme is a tuple* $(\mathscr{A}, \mathcal{E}, \mathbf{Z}, \mathcal{D}, n, Prf, Ver)$, *where*

i) $(\mathscr{A}, \mathcal{E}, \mathbf{Z}, \mathcal{D})$ *is a GDSS.*

ii) $Prf : \mathbb{Z}_q \times \mathbb{Z}_q^{m+h} \to \mathbb{Z}_q^d$ *is a proof generation algorithm. The input to* Prf *is* k, $\mathbf{s} = (s_1, s_2, \cdots, s_m)^T$ *and* $\mathbf{u} = \mathcal{E}(\mathscr{A}, \mathbf{s}, \mathbf{t})$. *The output* $\mathbf{I} \leftarrow Prf(k, (\mathbf{s}, \mathbf{u}))$ *is the vector in* $\mathbb{Z}_q^d$ *and stored in the* n*-th storage node.*

iii) Ver *is a collection of* $2m$ *verification algorithms* $Ver_j^{(1)} : \mathbb{Z}_q^{\mathbf{u}_j} \times \mathbb{Z}_q^d \to \{1, 0\}$ *and* $Ver_j^{(2)} : \mathbb{Z}_q \times \mathbb{Z}_q^d \to \{1, 0\}$ *for* $j \in [m]$, *such that* $Ver_j^{(1)}(\mathbf{u}_{t_j,j}, \mathbf{I}) = 1$ *and* $Ver_j^{(2)}(s_j, \mathbf{I}) = 1$. *In other words, each user is able to successfully verify its downloaded symbols and the reconstructed secret. This is referred to as the* verifiability condition.

The attack model An adversary can choose an arbitrary subset $A_{t_j,j} = \{a_{1,j}, a_{2,j}, \cdots, a_{t_j,j}\}$ of $A_j \in \mathscr{A}$ and initiate one of the following attacks.

- **Impersonation attack**: An adversary tries to create a valid message pair $(\bar{\mathbf{u}}_{t_j,j}, \bar{\mathbf{I}})$ without seeing any messages from the storage nodes.

[3] The Shanon entropy of a random variable X is defined by $H(X) = E_{x \leftarrow X}[\log \frac{1}{Pr[X=x]}]$. The mutual information between two discreet random variables X, Y jointly distributed according to $p(x,y) = Pr[(X,Y) = (x,y)]$ is given by $I(X,Y) = \sum_{x,y} p(x,y) \log(\frac{p(x,y)}{Pr[X=x]Pr[Y=y]})$.

– **Substitution attack**: An adversary substitutes $(\mathbf{u}_{t_j,j}, \mathbf{I})$ with a different message pair $(\bar{\mathbf{u}}_{t_j,j}, \mathbf{I})$.

An impersonation or a substitution attack is successful if the secret reconstructed by the legitimate user associated with $A_{t_j,j}$ differs from the secret sent by the dealer.

Under the attack model described above, it is required that in a VDSS, an impersonation or a substitution attack will successfully pass the test of Ver with negligible probability. This is referred to as the *soundness condition*. The authors in [16] suggest the use of a one-way function h to handle the problem. For an adversary whose computational power is unbounded, Ver through the use of a universal hash function $h : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ in [17] can achieve maximal security in an information-theoretic sense , i.e.,

$$Pr[\text{Successful deception}] \leq \frac{1}{|\mathcal{T}|}.$$

In the presence of a computationally bounded adversary, Ver that uses $(t, \hat{q}, \hat{\epsilon})$-pseudorandom function family $\mathcal{F} = \{h : \mathcal{K} \times \mathcal{M} \to \mathcal{T}\}$ in [17] can ensure computational security against all t-time adversaries if $\hat{q} \geq 1$, that is,

$$Pr[\text{Successful deception}] \leq \frac{1}{|\mathcal{T}|} + \hat{\epsilon}.$$

The cost of a VDSS To evaluate the efficiency of the proposed VDSSs, the notions of storage overhead as well as communication complexity are defined next. The communication complexity is the total number of $\mathbb{Z}_q$-symbols exchanged between the users and the storage nodes, which is defined as

$$C = \sum_{j=1}^{m} (d_j + c_j). \tag{4}$$

Here, d_j, c_j denote the total number of symbols that the user j downloads from the n-th storage node and the storage nodes in any subset $A_{t_j,j}$ of A_j, respectively. A VDSS with minimum communication complexity C, defined in (4), is called a communication-optimal VDSS.

The storage overhead is the ratio between the total number of symbols stored on all the storage nodes and the number of users in the system. That is, the storage overhead, called SO, of a VDSS is defined as

$$SO = \frac{k'}{m}. \tag{5}$$

where $k' = d + \sum_{i=1}^{n-1} \sum_{j=1}^{m} z_{i,j}$, $z_{i,j}$ is specified in Definition 1, and d is the total number of symbols in I.

Note that the correctness condition must be satisfied for m mutually independent and uniformly distributed secrets. Therefore, $k' \geq d + m \geq 1 + m$ and,

consequently, $SO \geq 1 + \frac{1}{m}$. we show that the lower bound $SO = 1 + \frac{1}{m}$ can be achieved under the weak secrecy condition for a certain set of parameters, thereby providing a scheme with weak secrecy and the optimal SO equal to $1 + \frac{1}{m}$.

3 GDSS With Maximum Number of Users

3.1 The Access Structure of a GDSS

In a GDSS with weak secrecy, we present a necessary condition on access sets which relates to Sperner families in combinatorics. In particular, if t_j=n_j, it is shown in [4] that this condition is necessary and sufficient for the existence of both weakly secure and perfectly secure DSSPs. The condition not only represents the relation between Sperner families and GDSS, it also provides a method for constructing GDSSs that serve maximum number of users.

Lemma 1. *For a weakly secure GDSS with access structure $\mathscr{A}$ defined in* (1)*, we have*

$$A_{t_j,j} \nsubseteq A_{t_{j'},j'}, \tag{6}$$

for all $j, j' \in [m]$ with $j \neq j'$, where $A_{t_j,j} \subseteq A_j$ and $A_{t_{j'},j'} \subseteq A_{j'}$.

Proof. Now assume to the contrary that $A_{t_j,j} \subseteq A_{t_{j'},j'}$ for some $j \neq j'$. From the model constituents in Section 2.1, we know that the entire accessible data by user j can also be accessed by user j'. This means that user j' can retrieve s_j which user j can retrieve by the correctness condition. Hence, the weak secrecy condition is violated and the scheme is not a GDSS as defined in Definition 1, which is a contradiction.

Define $\hat{\mathscr{A}}$ associated with $\mathscr{A}$ as follows.

$$\hat{\mathscr{A}} = \{A_{t_j,j} \subseteq A_j : j \in [m], A_j \in \mathscr{A}\}. \tag{7}$$

It is easy see that $\hat{\mathscr{A}}$ is a collection of subsets satisfying the condition of Lemma 1. A collection such as $\hat{\mathscr{A}}$ is called a Sperner family in [4] and [18]. More generally, for any Sperner family $\hat{\mathscr{A}}$, from [18] we have

$$|\hat{\mathscr{A}}| \leq \binom{n-1}{\lfloor (n-1)/2 \rfloor}, \tag{8}$$

where $n-1$ is the number of storage nodes stored shares. Again, the number of users in a GDSS is m. The number of t_j-subset in A_j is $\binom{n_j}{t_j}$, and $|\hat{\mathscr{A}}| = \sum_{j\in[m]} \binom{n_j}{t_j}$, where $1 \leq t_j \leq n_j \leq n-1$. Hence, $n-1$, m and (t_j, n_j) for $j \in [m]$ must satisfy the following relation.

$$\sum_{j\in[m]} \binom{n_j}{t_j} \leq \binom{n-1}{\lfloor (n-1)/2 \rfloor}. \tag{9}$$

In particular, $\binom{n-1}{\lfloor (n-1)/2 \rfloor}$ is the maximum number of users servered in a GDSS when $t_j = n_j$.

3.2 The construction of a GDSS

The bases of proposed GDSS are a (t_j, n_j) Shamir's secret sharing scheme [19]. Hence, we briefly recall the definition and known results concerning it. Then, our construction is described.

Secret sharing scheme A secret sharing scheme of $\hat{\mathscr{A}}$ as (7) consists of a pair of probabilistic algorithms $(\mathcal{E}, \mathcal{D})$. $\mathcal{E}$ gets as input a secret s_j (from a domain of secrets S), A_j and t_j, and then generates n_j shares $u_{j,1}, u_{j,2}, \cdots, u_{j,n_j}$. $\mathcal{D}$ gets as input the shares of a subset $A_{t_j,j}$ and outputs a string. The requirements are:

- For every secret s_j and every qualified set $A_{t_j,j} \subseteq A_j \in \mathscr{A}$, it holds that $Pr[\mathcal{D}(\{u_{j,i}\}_{i \in A_{t_j,j}}, A_j) = s_j] = 1$.
- For every unqualified set $A_{t_j,j} \notin \hat{\mathscr{A}}$ and every two different secrets $s_j, s_{j'} \in S$, it holds that the distributions $\{u_{j,i}\}_{i \in A_{t_j,j}}$ and $\{u_{j',i}\}_{i \in A_{t_j,j}}$ are identical.

The (t_j, n_j) secret sharing scheme is proposed by Shamir [19]. Let $P_j(x)$ be a $(t_j - 1)$-degree polynomial, where

$$P_j(x) = s_j + \sum_{l=1}^{t_j - 1} p_{j,l} x^l, \tag{10}$$

and $p_{j,l}$'s are i.i.d (independent and identically distributed) and are selected uniformly at random from $\mathbb{Z}_q$. Assume that $r_1, r_2, \cdots, r_{n_j}$ denote n_j distinct non-zero elements in $\mathbb{Z}_q$. The secret shares are then constructed by evaluating $P_j(x)$ at r_i's, i.e.,

$$u_{j,i} = P_j(r_i), \forall i \in [n_j]. \tag{11}$$

We refer to the encoder $\mathcal{E} : \mathbb{Z}_q \to \mathbb{Z}_q^{n_j}$ that takes s_j as the input and outputs $u_{j,1}, u_{j,2}, \cdots, u_{j,n_j}$ as described above as a (t_j, n_j) Shamir's secret encoder. Again, $\mathcal{D}$ is called a (t_j, n_j) Shamir's secret decoder.

Proposed GDSS we use Shamir's secret sharing scheme to construct a GDSS with perfect secrecy when $\hat{\mathscr{A}}$ associated with $\mathscr{A}$ is a Sperner family. In other words, the condition in Lemma 1 is a sufficient condition for existence of a perfectly secure GDSS. The proposed GDSS is constructed as follows:

i) $\hat{\mathscr{A}}$ associated with $\mathscr{A}$ is a Sperner family consisting of subsets of $[n-1]$.
ii) $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_m)$, where $\mathcal{E}_j$ is a (t_j, n_j) Shamir's secret encoder, n_j=$|A_j|$, $j \in [m]$.
iii) Initially, let all entries of Z be zero. Then, the values of Z are updated by using $Z[a_{i,j}, z_{j-1} + i]$=1 for $j \in [m]$ if $i \in [n_j]$, where z_j=$z_{j-1} + n_j$, z_0=0, and A_j=$\{a_{1,j}, a_{2,j}, \cdots, a_{n_j,j}\}$.
iv) $\mathcal{D}_j$ is the (t_j, n_j) Shamir's secret decoder, for $j \in [m]$.

Theorem 1. *The proposed scheme is a perfectly secure GDSS satisfying all properties in Definition 1.*

Proof. The proposed scheme assigns a access set, A_j, of $[n-1]$ with a threshold value t_j to user j. By using a (t_j, n_j) Shamir's scheme, the dealer generates a random polynomial, $P_j(x)$, independently from other users, and then encodes s_j into n_j secret shares $\mathbf{u}_j$. Then, each element of $\mathbf{u}_j$ is stored on the storage node in A_j as specified by Z. It is easy to see that the user j can reconstruct its secret by running the (t_j, n_j) Shamir's secret decoder $\mathcal{D}_j$. Since $\hat{\mathscr{A}}$ associated with $\mathscr{A}$ is a Sperner family and $P_j(x)$ is a random polynomial independently from other users, we have that $I(\mathbf{s}_{-j}, \mathbf{u}_j)$=0. Therefore, the proposed scheme is a GDSS with perfect secrecy, and satisfies all properties in Definition 1.

For the design of a GDSS, we can pick a Sperner family $\hat{\mathscr{A}}$ with the maximum size $\binom{n-1}{\lfloor (n-1)/2 \rfloor}$ and then construct $\mathscr{A}$. As above, the proposed scheme is a perfectly secure GDSS satisfying all properties in Definition 1, and uses a appropriate threshold vector to serve the maximum possible number of users given a certain number of storage nodes in $[n-1]$. These is demonstrated in an example as follows.

Example 3.1: Let n=6, m=8. Compute $\binom{5}{2} = 10$. Suppose that $\hat{\mathscr{A}}$ with the maximum size 10 is a collection of all 3-subsets of [5] given as follows:

$$\hat{A}_1 = \{1,2,3\}, \hat{A}_2 = \{2,3,4\}, \hat{A}_3 = \{3,4,5\}, \hat{A}_4 = \{4,5,1\},$$
$$\hat{A}_5 = \{5,1,2\}, \hat{A}_6 = \{1,2,4\}, \hat{A}_7 = \{1,3,5\}, \hat{A}_8 = \{2,3,5\},$$
$$\hat{A}_9 = \{2,4,5\}, \hat{A}_{10} = \{3,4,1\}.$$

Then, $\mathscr{A} = \{A_j \subseteq [5] : j \in [8]\}$, where the threshold values of all users are 3, A_j=$\hat{A}_{j+1}$ for $j \geq 2$, and A_1=$\{1,2,3,4\}$ which includes three 3-subsets: $\hat{A}_1$, $\hat{A}_2$ and $\hat{A}_{10}$. For the secret reconstruction of user 1, it provides the fault tolerance.

4 Communication-Optimal VDSS

In this section, we propose a secure VDSS which is based on a GDSS in Section 3 and a Nyberg's one-way accumulator. We also study the correctness, strong secrecy, and soundness constraints. Then, we show the tight VDSS that achieves a communication-optimal VDSS under certain conditions.

4.1 Nyberg's one-way accumulator

Hence, we briefly recall the definition and known results concerning a Nyberg's one-way accumulator (NOWA).

Definition 3 (NOWA [15]). *A family of one-way accumulators is a family of one-way hash functions with quasi-commutativity. The one-way accumulator by Nyberg [15] is constructed based on the generic symmetry-based hash function*

(e.g., SHA) and simple bit-wise operations. Compared to Benaloh's scheme [20], Nyberg's scheme is more efficient without employing asymmetric cryptographic operations.

Assume that $N = 2^\theta$ is an upper bound to the number of items to be accumulated and r is an integer. Let $s_1, s_2, \cdots, s_m$ be the accumulated items with different string sizes, and $m \leq N$. Let $h(\cdot,\cdot)$ denote a NOWA from $\{0,1\}^\gamma \times \{0,1\}^*$ to $\{0,1\}^\gamma$, and $\odot$ be the bitwise operation AND. The NOWA is based on the one-way hash function $\hat{h} : \{0,1\}^* \rightarrow \{0,1\}^{\gamma\theta}$. All that is required to specify a NOWA is hashing process and AND operation. The heart of NOWA is the hashing process. The hashing process applies a hash function $\hat{h}$ to the input to produce a γ-bit output. The hashing process is composed of the following operations.

- Hashing operation: hash accumulated item s_j of the input and output a $\gamma\theta$ bits binary string v_j=$\hat{h}(s_j)$.
- Transfer α: NOWA does a transfer operation on the binary string v_j which is divided into γ blocks, $(v_{j,1}, \cdots, v_{j,\gamma})$, of length θ. The transfer of a block from a θ-bit input to a bit output is performed as follows: If $v_{j,l}$ is a string of zero bits, it is replaced by 0; otherwise, $v_{j,l}$ is replaced by 1. That is, $\alpha(v_j)$ $=(b_{j,1}, \cdots, b_{j,\gamma})$, where $b_{j,l} \in \{0,1\}$, l=$1, \cdots, \gamma$.

In this way, we can transfer an accumulated item s_j to a bit string, b_j=$\alpha(\hat{h}(s_j)) \in \{0,1\}^\gamma$, which can be considered as a value of γ independent binary random variable if $\hat{h}$ is an ideal hash function.

The NOWA on an accumulated item $s_j \in S$ with an accumulated key $k \in \{0,1\}^\gamma$ can be implemented using the AND operation described as $h(k, s_j) = k \odot \alpha(v_j) = k \odot \alpha(\hat{h}(s_j))$. And it also can be represented as $X = h(k, s_j) = k \odot \alpha(v_j) = k \odot \alpha(\hat{h}(s_j))$ $(j \in [m])$ if S is a set of accumulated items $S = \{s_1, s_2, \cdots, s_m\}$. $h(\cdot,\cdot)$ has the following properties:

- Quasi-commutativity: $h(h(k, s_1), s_2) = h(h\ (k, s_2), s_1)$.
- Absorbency: $h(h(k, s_j), s_j) = k \odot \alpha(\hat{h}(s_j)) = h\ (k, s_j)$.
- An item s_j within the accumulated value X can be verified by $h(X, s_j) = X \odot \alpha(\hat{h}(s_j)) = X$.

4.2 Proposed VDSS with Perfect Secrecy

Here, by using a GDSS and a NOWA, one can construct a VDSS with perfect secrecy when $\hat{\mathscr{A}}$ associated with $\mathscr{A}$ is a Sperner family consisting of subsets of $[n-1]$. The proposed VDSS with perfect secrecy is constructed as follows:

i) $\hat{\mathscr{A}}$ associated with $\mathscr{A}$ in (7) is a Sperner family with the threshold vector $\mathbf{t}$.
ii) $\mathbf{u} = \mathcal{E}(\mathscr{A}, \mathbf{s}, \mathbf{t})$ for $\mathbf{s} = (s_1, s_2, \cdots, s_m)^T$, where $\mathcal{E}$ is the encoder of the GDSS.
iii) Let $\mathbf{u}$=$(u_1, u_2, \cdots, u_h)$. Initially, let all entries of Z be zero. Then, the values of Z are updated by $z_{i,r} = 1$ if u_r of $\mathbf{u}$ is stored in i-th storage node,

iv) $Prf = (Prf_1, Prf_2, \cdots, Prf_m)$, where Prf_j defines as

$$Prf_j(k, (s_j, \mathbf{u}_j)) = h(\cdots h(h(k, s_j), u_{1,j}), \cdots, u_{n_j,j}). \tag{12}$$

Here, $j \in [m]$, k is a secret key of the dealer, h is a NOWA generated from a one-way hash function $\hat{h} : \{0,1\}^* \to \{0,1\}^{\gamma\theta}$, and $(m+h) \leq 2^\theta$. By abuse of notation, we will often refer to a proof algorithm by $Prf_j(s_j, \mathbf{u}_j)$, and omit k when clear from the context.

v) For $j \in [m]$, the output $I_j = Prf_j(s_j, \mathbf{u}_j)$ is stored in the n-th storage node.

vi) Given $j \in [m]$, $Ver_j^{(1)}(\mathbf{u}_{t_j,j}, I_j) = 1$ if and only if $h(I_j, u_{i,j}) = I_j$ for all $i \in A_{t_j,j} \subseteq A_j$, where $\mathbf{u}_{t_j,j} = (u_{1,j}, \cdots, u_{t_j,j})$ and I_j are downloaded by the user j.

vii) $\mathcal{D}_j(\mathbf{u}_{t_j,j})$ is the decoder of the GDSS.

viii) Given $j \in [m]$, $Ver_j^{(2)}(s'_j, I_j) = 1$ if and only if $h(I_j, s'_j) = I_j$, where $s'_j = \mathcal{D}_j(\mathbf{u}_{t_j,j})$.

Lemma 2. *Assume that h is a secure Nyberg's one-way accumulator. The proposed VDSS satisfies the soundness condition under a computationally bounded adversary.*

Proof. Let $\mathcal{B}$ be an impersonation adversary. Assume that p_{ia} denotes the following probability

$$\begin{aligned} Pr[k \xleftarrow{\$} \mathbb{Z}_q, ((\bar{s}_j, \bar{\mathbf{u}}_{t_j,j}) \leftarrow \mathcal{B}((s_j, \mathbf{u}_{t_j,j}), Prf_j(s_j, \mathbf{u}_{t_j,j})) : \\ Prf_j(\bar{s}_j, \bar{\mathbf{u}}_{t_j,j}) = \bar{I}_j \wedge ((\bar{s}_j, \bar{\mathbf{u}}_{t_j,j}), \bar{I}_j) \neq ((s_j, \mathbf{u}_{t_j,j}), I_j)] \end{aligned}$$

where $Prf_j(s_j, \mathbf{u}_{t_j,j})$ is computed as (12). Thus, we need to bound the value p_{ia}. Note that h is constructed based on the generic symmetry-based hash function $\hat{h}$ and simple bit-wise operations, that is, $h(k, s_j) = k \odot \alpha(h(s_j))$. When k is chosen uniformly from $\mathbb{Z}_q$, the corresponding hash values $h(k, (s_j, \mathbf{u}_{t_j,j})) = I_j$ and $h(k, (\bar{s}_j, \bar{\mathbf{u}}_{t_j,j})) = \bar{I}_j$ are independent and have a uniform distribution over $\mathbb{Z}_q$. Therefore, p_{ia} is negligible from [15] and [21] when h is a secure Nyberg's one-way accumulator. For a substitution adversary, we assume that p_{sa} denotes the following probability

$$\begin{aligned} Pr[k \xleftarrow{\$} \mathbb{Z}_q, ((\bar{s}_j, \bar{\mathbf{u}}_{t_j,j}) \leftarrow \mathcal{B}((s_j, \mathbf{u}_{t_j,j}), Prf_j(s_j, \mathbf{u}_{t_j,j})) : \\ Prf_j(\bar{s}_j, \bar{\mathbf{u}}_{t_j,j}) = Prf_j(s_j, \mathbf{u}_{t_j,j}) \wedge (\bar{s}_j, \bar{\mathbf{u}}_{t_j,j}) \neq (s_j, \mathbf{u}_{t_j,j})]. \end{aligned}$$

Due to the collision resistance of h, we have that p_{sa} is negligible. This completes the proof.

Theorem 2. *The proposed scheme in this section is a secure VDSS satisfying all properties in Definition 2.*

Proof. In the proposed scheme, each user j has access to all data stored in a subset $A_{t_j,j}$ of A_j Hence, the correctness and perfect secrecy conditions are satisfied by invoking the secret encoder and decoder of a GDSS. Also, note that from Lemma 2 it satisfies the soundness condition under a computationally

bounded adversary since h is a secure Nyberg's one-way accumulator. What remains to show is that the verifiability condition is also satisfied.

The proof of s_j and $\mathbf{u}_j$ is provided by computing I_j through (12). From the quasi-commutativity and absorbency properties in Section 4.1, we have that $h(I_j, s_j)$=I_j and $h(I_j, u_{i,j})$=I_j, where $u_{i,j}$ is the element of $\mathbf{u}_j$. This together with the definitions of $Ver_j^{(1)}$ implies that $Ver_j^{(1)}(\mathbf{u}_{t_j,j}, I_j) = 1$ if and only if $h(I_j, u_{i,j}) = I_j$ for all $i \in A_{t_j,j} \subseteq A_j$. Furthermore. the reconstructed secret as s'_j=$\mathcal{D}_j(\mathbf{u}_{t_j,j})$, can be checked through $h(I_j, s'_j) = I_j$. This is because $Ver_j^{(2)}(s'_j, I_j) = 1$ if and only if $h(I_j, s'_j) = I_j$. Therefore, the proposed scheme also satisfies the verifiability condition. This completes the proof.

Theorem 3. *The proposed scheme is a communication-optimal VDSS.*

Proof. For each user j, $A_{t_j,j}$ is a subset of A_j with the minimum size such that user j can reconstruct its secret s_j. Note that at least one symbol in $\mathbb{Z}_q$ has to be download by user j from each node in $A_{t_j,j}$ and the n-th storage node. From the definition of communication complexity in (4), we have that $\sum_{j\in[m]}(1+t_j) \leq C$. Define

$$C_0 = \sum_{j\in[m]} (1 + t_j). \tag{13}$$

That is, C_0 is a lower bound on the communication complexity of VDSSs.

In the proposed scheme of $\hat{\mathscr{A}}$ as (7), its communication complexity is $\sum_{j\in[m]}(|I_j|+|A_{t_j,j}|)$. Again, each user downloads exactly one data symbol from the nodes in its access subset $A_{t_j,j}$, and $|I_j|$=1. This shows that the proposed VDSS attains the lower bound in (13). This completes the proof.

The proposed scheme together with the storage nodes is demonstrated in an example, discussed next.

Example 4.1: Let $n = 6$, $m = 8$. The access structure $\mathscr{A}$ is defined as *Example 3.1.* Also, let q=31, and non-zero and distinct evaluation values r_1=1, r_2=2, r_3=3 and r_4=4 are chosen from $\mathbb{Z}_q$. Then, the encoded data of s_j about $A_j \in \mathscr{A}$ are generated by using (11), where $n_1 = 4$, n_j=3 for $2 \leq j \leq 8$. Note that $\lceil|q|\rceil$=5, and $\sum_{j=1}^{8}(n_j + 1) = 33$. Let θ =32 satisfying $\sum_{j=1}^{m}(n_j + 1) < 2^\theta$. SHA-1 is chosen as $\hat{h}$, and then a NOWA $h : \mathbb{Z}_q \times \{0, 1\}^* \to \mathbb{Z}_q$ is constructed. Thus, I_j is computed for the access set A_j as

$$\begin{aligned} I_1 &= h(h(h(h(k, s_1), u_{a_{1,1}}), u_{a_{2,1}}), u_{a_{3,1}}), u_{a_{4,1}}), \\ I_j &= h(h(h(h(k, s_j), u_{a_{1,j}}), u_{a_{2,j}}), u_{a_{3,j}}), j = 2, \cdots, 8. \end{aligned}$$

The storage profile is shown in Table 2. The communication complexity of the proposed communication-optimal VDSS is 32. Note that d=8, which is equal to m. In the next section, we discuss approaches to generate the proof with only one data symbol in a more structured way such that $d = 1$.

Table 1. Storage Profile in Example 4.1.

User	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
1	$u_{1,1}$	$u_{2,1}$	$u_{3,1}$	$u_{4,1}$		I_1
2			$u_{3,2}$	$u_{4,2}$	$u_{5,2}$	I_2
3	$u_{5,3}$			$u_{4,3}$	$u_{5,3}$	I_3
4	$u_{1,4}$	$u_{2,4}$			$u_{5,4}$	I_4
5	$u_{1,5}$	$u_{2,5}$		$u_{4,5}$		I_5
6	$u_{1,6}$	$u_{2,6}$			$u_{5,6}$	I_6
7		$u_{2,7}$	$u_{3,7}$		$u_{5,7}$	I_7
8		$u_{2,8}$		$u_{4,8}$	$u_{5,8}$	I_8

5 VDSS With Optimal Storage Overhead

In this section, a VDSS is proposed to attain the optimal storage overhead equal to $1 + \frac{1}{m}$. The basis of proposed VDSS is a distributed secret sharing protocol (DSSP) in [4]. Hence, we briefly recall the definition and known results concerning this DSSP. We then describe our scheme.

5.1 The Definition and Known Results for DSSP

The definition of DSSP in [4] is a special case of GDSS in Definition 1. In the case when $t_j = |A_j|$ in the access structure $\mathscr{A}$ defined in (1), a GDSS is a DSSP. Assume that there are m users and $\bar{n}$ storage nodes storing the shares. For convenience, the access sets for the first $\bar{n}$ users are specified as follows:

$$A_j = \{j, j+1, \cdots, j+\bar{k}-1\}, j = 1, 2, \cdots, \bar{n}. \tag{14}$$

where $\bar{k} \in [\bar{n}]$ is a parameter. The remaining access sets, A_j $(j = \bar{n}+1, \cdots, m)$, can be arbitrary as long as $\mathscr{A}$ form a Sperner family. The indices of storage nodes are considered modulo $\bar{n}$, i.e., $\bar{n} + l = l$.

A DSSP is said to be a tight DSSP if every user downloads exactly one $\mathbb{Z}_q$-symbol from each node in its access set. Then, To find a communication-optimal DSSP, the parameter $\bar{k}$ in (14) is determined by Theorem 5 in [4] as follows.

Theorem 4. *For a given number of users m and storage nodes $\bar{n}$, a communication-optimal DSSP is constructed by any tight DSSP with the following access structure $\mathscr{A}$: $\mathscr{A}$ is a Sperner family that contains α_i^* of i-subsets of $\bar{n}$ and α_{i+1}^* of $(i+1)$-subsets of $\bar{n}$, where i is the maximum integer that satisfies $\binom{\bar{n}}{i} \leq m$, and α_i^* and α_{i+1}^* are calculated as*

$$\alpha_i^* = \frac{(\binom{\bar{n}}{i+1} - m) \cdot \binom{\bar{n}}{i}}{\binom{\bar{n}}{i+1} - \binom{\bar{n}}{i}}, \alpha_{i+1}^* = \frac{(m - \binom{\bar{n}}{i}) \cdot \binom{\bar{n}}{i+1}}{\binom{\bar{n}}{i+1} - \binom{\bar{n}}{i}}.$$

To encode the secrets, s_j's, of the users $j \in [\bar{n}]$, their $(\bar{k}-1)$ degree polynomials $P_j(x)$'s in (10) are constructed by the following system of linear equations:

$$\begin{array}{cccc} P_1(r_1)=u_1, & P_2(r_1)=u_2, & \cdots, & P_{\bar{n}}(r_1)=u_{n-1}, \\ P_1(r_2)=u_2, & P_2(r_2)=u_3, & \cdots, & P_{\bar{n}}(r_2)=u_1, \\ \vdots, & \vdots, & \cdots, & \vdots, \\ P_1(r_{\bar{k}})=u_{\bar{k}}, & P_2(r_{\bar{k}})=u_{\bar{k}+1}, & \cdots, & P_{\bar{n}}(r_{\bar{k}})=u_{\bar{k}-1}. \end{array} \tag{15}$$

under certain conditions, the system has a unique solution for $p_{j,l}$'s and u_j's, $j \in [\bar{n}]$ and $l \in [\bar{k}-1]$. Note that u_j is the encoding symbol of the secret s_j. To simplify such conditions and also for ease of calculation, let r be a primitive element of $\mathbb{Z}_q$, and $r_i = r^i$ for $i \in [\bar{k}]$. The following corollary means that there is a non-singular matrix E such that

$$(u_1, u_2, \cdots, u_{\bar{n}})^T = E(s_1, s_2, \cdots, s_{\bar{n}})^T. \tag{16}$$

Corollary 1. *If* $(p-1) \nmid i\bar{n}$ *for* $i \in [\bar{k}]$, *then* (16) *defines a one-to-one mapping between* $(s_1, s_2, \cdots, s_{\bar{n}})$ *and* $(u_1, u_2, \cdots, u_{\bar{n}})$.

Consider user j for $j{=}\bar{n}+1, \cdots, m$. Without loss of generality suppose that $A_j = (a_{1,j}, a_{2,j}, \cdots, a_{\bar{k},j})$. Note that $A_j \subseteq [\bar{n}]$. To encode the secret s_j, a $(\bar{k}-1)$ degree polynomial $P_j(x)$'s in (10) is constructed by the dealer considering a system of linear equations for the case(a) in (17).

$$(a)\begin{cases} P_j(r_1)=u_{a_{1,j}}, \\ P_j(r_2)=u_{a_{2,j}}, \\ \vdots, \\ P_j(r_{\bar{k}-1})=u_{a_{\bar{k}-1,j}}. \end{cases} \quad (b)\begin{cases} P_j(r_2)=u_{a_{2,j}}, \\ P_j(r_3)=u_{a_{3,j}}, \\ \vdots, \\ P_j(r_{\bar{k}})=u_{a_{\bar{k},j}}. \end{cases} \tag{17}$$

Since the coefficient matrices of them are a Vandermonde matrix,$(p_{j,1}, p_{j,2}, \cdots, p_{j,\bar{k}-1})$ has a unique solution in the case. That is, $P_j(x)$ in (10) is determined since s_j is known. Then, s_j is encoded as $P_j(r_{\bar{k}})$ which is stored in the remaining node, $a_{\bar{k},j}$, of A_j. Similarly, a polynomial $P_j(x)$ for the case (b) in (17) can be obtained, and then $P_j(r_1)$ is computed as the encoding data of s_j and stored in the remaining node, $a_{1,j}$, of A_j.

5.2 Proposed VDSS with weak secrecy

In the proposed VDSS with weak secrecy, the optimal storage overhead equal to $1+\frac{1}{m}$ is attained without the need to the external randomness. The idea is to modify the proposed DSSP with weak secrecy in [4]. In particular, the proof with only one symbol in $\mathbb{Z}_q$ is achieved by using a NOWA in a structured way. In addition, an efficient strategy is found which makes the storage profile balanced when considering individual storage loads across the storage nodes in $[n-1]$.

For generating the parameters and a Sperner family $\mathscr{A}$, an algorithm $Init(n,m)$ is proposed. Then, an algorithm $Prf(k,(\mathbf{s},\mathbf{u}))$ is constructed to output a proof with only one symbol in $\mathbb{Z}_q$. They are presented as follows.

$(\mathbb{Z}_q, r, p, h, \mathscr{A}) \leftarrow Init(n,m)$: It first checks if $m \le \binom{n-1}{\lfloor (n-1)/2 \rfloor}$. If true, it then runs all the following operations.

– The parameter $\bar{k}$ is determined by using Theorem 4, where $\bar{n}$=$n-1$.
– Choose an integer $p > \bar{k}m$ such that a multiplicative subgroup of $\mathbb{Z}_p$ forms a cyclic group, $\mathbb{U}(q)$, of order p, where $(p-1) \nmid i(n-1)$ for $i \in [\bar{k}]$. Let r be the primitive element of the cyclic group.
– Select a one way hash function $\hat{h}$ with the output length $\gamma\theta$, and generate a NOWA $h : \mathbb{Z}_q \times \{0,1\}^* \to \mathbb{Z}_q$. Here, θ is an integer which satisfies $(\bar{k}+1)m < 2^\theta$, and γ=$|q|$.
– $\mathscr{A}$ is a Sperner family with at least m subsets of size $\bar{k}$ of $[n-1]$. First, for $j \in [n-1]$, the access set A_j is defined as (14). Second, the remaining access sets, A_j for $j = n, n+1, \cdots, m$, can be partitioned into balanced collections of size at most $n-1$. In a balanced collection of size $n-1$, the union of the first elements of all sets with even subscripts and the last elements of all sets with odd subscripts is $[n-1]$. This is referred to as the *balanced storage strategy*.

$I \leftarrow Prf(k, (\mathbf{s}, \mathbf{u}))$: By inputting k, $\mathbf{s}$ and $\mathbf{u}$, algorithm transfer them into an element $I \in \mathbb{Z}_q$, where

$$\begin{aligned} I = & \quad h(\cdots h(h(\cdots h(k, s_1), \cdots, s_m), u_1), \cdots, u_h), & (18) \\ = & \quad h(\cdots h(Prf(k, \mathbf{s}), u_1), \cdots, u_h) \\ = & \quad Prf(Prf(k, \mathbf{s}), \mathbf{u}). & (19) \end{aligned}$$

Here, $Prf(k, \mathbf{s}) = h(\cdots h(h(k, s_1), s_2), \cdots, s_m)$, and $\mathbf{u} = (u_1, u_2, \cdots, u_h)$ is the output of $\mathcal{E}$. Note that $h(I, s_j)$=I and $h(I, u_i)$=I from the quasi-commutativity and absorbency properties in Section 4.1.

Let $A_j = \{j_1, j_2, \cdots, u_{j_{\bar{k}}}\}$, $j \in [m]$. Based on the results in Section 5.1, where $\bar{n} = n-1$, the proposed VDSS with weak secrecy is constructed as follows:

i) The dealer runs algorithm $Init(n, m)$ to output $(\mathbb{Z}_q, r, p, h, \mathscr{A})$.
ii) $(u_1, u_2, \cdots, u_{n-1})^T \leftarrow \mathcal{E}(s_1, s_2, \cdots, s_{n-1})^T$, where $\mathcal{E}$ is the encoder of the DSSP defined as (15) and (16). Then, u_j is stored in the storage node j for $j \in [n-1]$.
iii) For A_j $(j \geq n)$ in a balanced collection, s_j is encoded as u_j=$\mathcal{E}(s_j)$ via the balanced storage strategy in $Init(n, m)$. Specifically, if $j \bmod 2 = 1$, u_j=$P_j(r_{\bar{k}})$ defined as (a) in (17) and it is stored in the last node, $u_{j_{\bar{k}}}$, of A_j; otherwise, u_j=$P_j(r_1)$ and is stored in the first node, j_1, of A_j, where $P_j(r_1)$ is computed as (b) in (17).
iv) $I \leftarrow Prf(k, \mathbf{s}, \mathbf{u})$, where Prf_j defines as (18) and (19), and $\mathbf{u} = (u_1, u_2, \cdots, u_m)$. Then, the output, I, is stored in the n-th storage node.
v) $Ver_j^{(1)} = (Ver_{j_1}^{(1)}, Ver_{j_2}^{(1)}, \cdots, Ver_{j_{\bar{k}}}^{(1)})$ for $j \in [m]$, where $Ver_{j_l}^{(1)}(u_{j_l}, I) = 1$ if and only if $h(I, u_{j_l}) = I$, $l \in [\bar{k}]$. Here, $\mathbf{u}_j = (u_{j_1}, u_{j_2} \cdots, u_{j_{\bar{k}}})$ and I are downloaded by the user j.
vi) $\mathcal{D}_j(\mathbf{u}_j)$ is the decoder of the DSSP.
vii) $Ver_j^{(2)}(s'_j, I) = h(I, s'_j)$, and $Ver_j^{(2)}(s'_j, I) = 1$ if and only if $h(I, s'_j) = I$ for $j \in [m]$, where s'_j=$\mathcal{D}_j(\mathbf{u}_j)$.

Theorem 5. *The proposed scheme in this section is a weakly secure VDSS satisfying all conditions in Definition 2 and has the storage overhead, defined in (5), equal to* $1 + \frac{1}{m}$.

Proof. It is easy to prove that the proposed scheme satisfies the verifiability condition. Similar to the proof of Theorem 6, we have that the proposed scheme satisfies the soundness condition. Note that the number of symbols generated in this scheme is m and each symbol is stored exactly once. Again, the proof has only one symbol in $\mathbb{Z}_q$. The storage overhead is $\frac{m+1}{m}$, which is the optimal value. Then, the rest of the proof is similar to the proof of Theorem 11 in [4].

For certain parameters m and n, the proposed scheme in this section satisfies the soundness condition, and is also a communication-optimal VDSS. This is summarized in the following theorem.

Theorem 6. *Let* $m=\binom{n-1}{\bar{k}}$ *for some* $\bar{k} \leq \lceil \frac{n-1}{2} \rceil$, *and the access structure* $\mathscr{A}$ *be picked as the set of all* $\bar{k}$*-subsets of* $[n-1]$. *The proposed scheme simultaneously attains the optimal value for both the communication complexity and the storage overhead under the weak secrecy and soundness conditions*

Proof. Let $\mathcal{B}$ be an impersonation adversary. Assume that p_{ia} denotes the following probability

$$\begin{aligned} Pr[k \xleftarrow{\$} \mathbb{Z}_q, (\bar{\mathbf{s}}, \bar{\mathbf{u}}) \leftarrow \mathcal{B}((\mathbf{s}, \mathbf{u}), Prf(k, (\mathbf{s}, \mathbf{u}))) : \\ Prf(k, (\bar{\mathbf{s}}, \bar{\mathbf{u}})) = \bar{I} \wedge ((\bar{\mathbf{s}}, \bar{\mathbf{u}}), \bar{I}) \neq ((\mathbf{s}, \mathbf{u}), I)] \end{aligned}$$

where $Prf(k, (\mathbf{s}, \mathbf{u}))$ is defined as (18) and (19). Again, for a substitution adversary, we assume that p_{sa} denotes the following probability

$$\begin{aligned} Pr[k \xleftarrow{\$} \mathbb{Z}_q, ((\bar{s}_j, \bar{\mathbf{u}}_{t_j,j}) \leftarrow \mathcal{B}((\mathbf{s}, \mathbf{u}), Prf(k, (\mathbf{s}, \mathbf{u}))) : \\ Prf(k, (\bar{\mathbf{s}}, \bar{\mathbf{u}})) = Prf(k, (\mathbf{s}, \mathbf{u}))) \wedge (\bar{\mathbf{s}}, \bar{\mathbf{u}}) \neq (\mathbf{s}, \mathbf{u})]. \end{aligned}$$

Similar to the proof of Theorem 2, we have that p_{ia} and p_{sa} are negligible when h is a secure Nyberg's one-way accumulator. Then, the rest of the proof is similar to the proof of Theorem 12 in [4].

Example 5.1: Let $n = 6$, $m = 10$. From Theorem 4, $i = 3$ is the maximum integer that satisfies $\binom{5}{i} \leq 10$, α_3^*=10 and α_4^*=0. This means that $\mathscr{A}$ is a Sperner family that contains 10 of 3-subsets of $[n-1]$ as follows.

$$\begin{aligned} A_1 = \{1,2,3\}, A_2 = \{2,3,4\}, A_3 = \{3,4,5\}, A_4 = \{4,5,1\}, \\ A_5 = \{5,1,2\}, A_6 = \{1,3,5\}, A_7 = \{1,2,4\}, A_8 = \{2,3,5\}, \\ A_9 = \{2,4,5\}, A_{10} = \{3,4,1\}. \end{aligned}$$

Obviously, for $j \in [5]$, the elements of A_j satisfy the relation in (14). Again, a balanced collection of size 5 is composed of access sets A_j, $6 \leq j \leq 10$. Note that the union of the first elements of all sets with even subscripts and

the last elements of all sets with odd subscripts is [5]. Therefore, a Sperner family outputted by $Init(n, m)$ is described as $\mathscr{A} = \{A_j : j \in [10]\}$, where $\bar{k}$ = 3. Let also q=31, r=15, and p=10, which satisfy the condition of Corollary 1 as $9 \nmid 5i$ for $i \in [3]$. Here, $\mathbb{U}(31)$=$\{15, 8, 27, 2, 30, 16, 23, 4, 29, 1\}\subseteq \mathbb{Z}_{31}^*$ and $15^{10} \bmod 31 = 1$. In the encoding secret phase, it involves encoding $s_1, \cdots, s_5$ as the random seed. Let r_1=15, r_2=8, and r_3=27. Then, a one-to-one mapping between $(s_1, s_2, \cdots, s_{n-1})$ and $(u_1, u_2, \cdots, u_{n-1})$ is defined as follows.

$$\begin{cases} 15u_1 + 13u_2 + 16u_3 = 13s_1, \\ 15u_2 + 13u_3 + 16u_4 = 13s_2, \\ 15u_3 + 13u_4 + 16u_5 = 13s_3, \\ 16u_1 + 15u_4 + 13u_5 = 13s_4, \\ 13u_1 + 16u_2 + 15u_5 = 13s_5. \end{cases}$$

Hence, u_j is computed from $s_1, s_2, \cdots, s_5$ as

$$\begin{cases} u_1 = 18s_1 + 16s_2 + 16s_4 + 13s_5, \\ u_2 = 13s_1 + 18s_2 + 16s_3 + 16s_5, \\ u_3 = 16s_1 + 13s_2 + 18s_3 + 16s_4, \\ u_4 = 16s_2 + 13s_3 + 18s_4 + 16s_5, \\ u_5 = 16s_1 + 16s_3 + 13s_4 + 18s_5. \end{cases}$$

For the remaining secrets s_j $(j \geq 6)$, their encoded data symbols are generated by (17). Furthermore, I is computed through (18). The resulting encoded secrets and their verification information together with the storage profile are shown in Table 2. It is clear that from (5), SO=1.1, which is the optimal SO, where d=1,k'=$d + 10$=11, and m=10. Observe that it has a balanced storage profile, and attains the lower bound, 40, of the communication complexity in (13).

Table 2. Storage Profile in Example 5.1.

Storage node	Random symbol	Generated data symbol
Node 1	u_1	u_6=$5s_6 + 26u_2 + u_5$
Node 2	u_2	u_8=$5s_8 + 26u_3 + u_5$
Node 3	u_3	u_{10}=$5s_{10} + u_1 + 26u_4$
Node 4	u_4	u_7=$26s_7 + 5u_1 + u_2$
Node 5	u_5	u_9=$26s_9 + 5u_2 + u_4$
Node 6		I

5.3 The complexity of two VDSS shcemes

The encoding computation complexity is determined by how many times the dealer has to generate the encoding symbols of secrets and their VI for m user. In the VDSS with optimal SO, the computation complexity of encoding the first $n - 1$ secrets is $\mathcal{O}((n-1)^2)$, due to the multiplication of the $(n-1) \times (n-1)$

seed encoding matrix as specified in (16) by the vector $(s_1, s_2, \cdots, s_{n-1})^T$. Let $\tilde{k}$ be the average size of access sets of the remaining $m-(n-1)$ users. Notes that the computation of a vector inner product, as specified in [4], is needed for each of the $m-(n-1)$ remaining users. Again, the number of NOWA operations in (18) is $(\tilde{k}+1)m$. Hence, the encoding complexity of VDSS with optimal SO is $\mathcal{O}((n-1)^2+\tilde{k}m)$. Similarly, the computation complexity of the encoder in the VDSS with nearly optimal SO is $\mathcal{O}(\tilde{k}m)$, where $\tilde{k}$ is the average size of access sets of all m users.

Regarding the decoding computation complexity, each user j utilizes (t_j, n_j) Shamir's decoder after receiving t_j correct shares, where a polynomial is interpolated to reconstruct s_j. Note that for the decoding of s_j, the user j needs to perform t_j NOWA operations and t_j multiplication operations. Hence, the decoding computation complexity of both schemes is $\mathcal{O}(\tilde{k}m)$, where $\tilde{k}$ is the average size of threshold values of all m users.

6 Conclusion

We have considered a VDSS, where the dealer conveys a specific secret to the user via storage nodes. Our design examines the impact of minimizing VI in a VDSS on the communication complexity and storage overhead. Our main result is the design of two secure VDSS schemes, where the first scheme attains the lower bound on the communication complexity while providing the fault tolerance, and the second scheme simultaneously achieves the lower bounds of both communication complexity and storage overhead while providing the balanced storage load.

References

1. B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," in *Advances in Cryptology — CRYPTO'99*, M. Wiener Ed. Santa Barbara, California, USA, 1999, pp. 148—164.
2. V. Attasena, N. Harbi, and J. Darmont, "Sharing-based privacy and availability of cloud data warehouses," In *Proc EDA 2013*, Blois, France, 2013, pp. 17-–32.
3. C. Wu, S. Li, and Y. Zhang, "Key management scheme based on secret sharing for wireless sensor network," in *Proc. EIDWT2013*, Xi'an, Shaanxi, China, 2013, pp. 574—578.
4. M. Soleymani and H. Mahdavifar, "Distributed Multi-User Secret Sharing," *IEEE Trans. Inf. Theory*, vol. 67, no. 1, pp. 164–178, Jan. 2021.
5. N. B. Shah, K. V. Rashmi, and K. Ramchandran, "Secure network coding for distributed secret sharing with low communication cost," in *Proc. ISIT 2013*, Istanbul, Turkey, 2013, pp. 2404–2408.
6. K. Matousek, "Security and reliability considerations for distributed healthcare systems," in *Proc. 42nd Annual IEEE International Carnahan Conference on Security Technology*, Prague, Czech Republic, 2008, pp.346–348.
7. R. A. Chou and A. Yener, "Strongly secure multiuser communication and authentication with anonymity constraints," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 572–586, Jan. 2020.

8. M. Yoshida and S. Obana, "Verifiably Multiplicative Secret Sharing," *IEEE Trans. Inf. Theory*vol. 65, no. 5, pp. 3233–3245, May 2019.
9. P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proc. FOCS 1987*, Los Angeles, California, USA, 1987, pp. 427—437.
10. C. Tartary and H. Wang, "Dynamic threshold and cheater resistance for shamir secret sharing scheme," in *Proc. Inscrypt 2006*, Beijing, China, 2006, pp. 103—117.
11. H.-Y. Lin and L. Harn, "A generalized secret sharing scheme with cheater detection," in *Advances in Cryptology-ASIACRYPT '91*, Fujiyoshida, Japan, 1991, pp. 149—158.
12. T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology-CRYPTO'91*, Santa Barbara, California, USA, 1991, pp. 129—140.
13. C. Blundo, A. De Santis, L. Gargano, and U. Vaccaro, "Secret sharing schemes with veto capabilities," in *Proc. Algebraic Coding 1993*, Paris, France, 1993, pp. 82—89.
14. R. J. McEliece and D. V. Sarwate, "On sharing secrets and Reed-Solomon codes," *Commun. ACM* , vol. 24, no. 9, pp. 583–584, Jan. 1981.
15. K. Nyberg, "Fast accumulated hashing," in *Proc. FSE 1996*, Cambridge, UK, 1996, pp. 83—87.
16. E. D. Karnin, J. W. Greene,and M. E. Hellman, "On secret sharing systems," *IEEE Trans. Inf. Theory*, vol. 29, no. 1, pp. 35–41, Jan. 1983.
17. S. Laur, Message authenitcation [Online]. Available: https://courses.cs.ut.ee/MTAT.07.003/2017_fall/uploads/Main/lecture-vi.pdf
18. E. Sperner, "Ein satz über untermengen einer endlichen menge," *Mathematische Zeitschrift*, vol. 27, no. 1, pp. 544—548, 1928. [Online]. Available: https://doi.org/10.1007/BF01171114
19. A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612-–613, Nov. 1979.
20. J. Benaloh, M. de Mare, "One-way accumulators: A decentralized alternative to digital signatures," in *Advances in Cryptology-EUROCRYPT '93*, Lofthus, Norway, 1993, pp. 274—285.
21. X. Yao, X. Han, X. Du, and X.i Zhou, "A lightweight multicast authentication mechanism for small scale IoT applications," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3693–3701, Oct. 2013.