A Lightweight Verifiable Secret Sharing in Internet of Things

Likang Lu[1]
College of Computer Science; College of Software
Inner Mongolia University
Hohhot, 010021, China

Jianzhu Lu[2*]
Department of Computer Science
Jinan University
Guangzhou, 510630, China

***Abstract*—Verifiable Secret Sharing (VSS) is a fundamental tool of cryptography and distributed computing in Internet of Things. Since network bandwidth is a scarce resource, minimizing the number of verification data will improve the performance of VSS. Existing VSS schemes, however, face limitations in meeting the number of verification data and energy consumptions for low-end devices, which make their adoption challenging in resource-limited IoTs. To address above limitations, we propose a VSS scheme according to Nyberg's one-way Accumulator for one-way Hash Functions (NAHFs). The proposed VSS has two distinguished features: first, the security of the scheme is based on NAHFs whose computational requirements are the basic criteria for known IoT devices and, second, upon receiving only one verification data, participants can verify the correctness of both their shares and the secret without any communication. Experimental results show that, compared to the Feldman scheme and Rajabi-Eslami scheme, the energy consumption of a participant in the proposed scheme is respectively reduced by at least 24% and 83% for a secret.**

***Keywords*—Verifiable secret sharing; one-way function; internet of things; security**

I. Introduction

The Internet of Things (IoT) is moving at such a rapid pace that there is rising demand for transforming our physical world into a complex and dynamic system of connected devices. These IoT devices will be widely used in smart homes, body/health monitoring, environmental monitoring, condition-based maintenance, among many others. IoT is not a single technology. It is a combination of sensors, devices, networks, and software that works on a collaborative basis to achieve a common goal. Secure and reliable group communication has become critical in the IoT system. Group key agreement is widely employed for secure group communications in modern collaborative and group-oriented applications. The central challenge is secure and efficient group key management [1, 2]. This is because these IoT devices have limited computing ability and the limitation of communication bandwidth. In this paper, we focus on the design of lightweight verifiable secret sharing (VSS) schemes in order to achieve the secret reconstruction among a set of IoT devices, where the reconstructed secret may be the group key of them.

A. Motivation for Lightweight VSS

To date, there are two main families of approaches that have been investigated to provide VSS to participants. The first approach provided verification data based on public key cryptography such as ASPP [3] in cyclic lattices and DLP [4]. The second approach to add verification capabilities to a scheme, was to use one-way functions to obtain fingerprints/ signatures of the involved data [5]. However, the existing schemes suffer from some major problems. Firstly, existing schemes face the challenge in very large-scale deployment of IoT devices. Since verification data grew linearly with either the number of participants [5] or the threshold value [3], their performance dropped sharply as the number of IoT devices grows. Note that network bandwidth is a scarce resource. Minimizing the number of public verification data will improve the performance of VSS. In this paper, we address this challenge and propose a VSS Scheme with only one verification data used to verify a secret and all of its shares.

In addition, for these low-cost, battery-powered IoT devices,the lightweight implementation of VSS schemes has emerged as a critical issue. Because public key cryptography uses some big integers to generate the verification data, it is much slower than symmetric key cryptography, requires more processing power, and generally increases energy consumptions of participants [6]. When the batteries are low, it may cause the IoT devices to function abnormally. Existing solutions require the public-key computation (e.g., Modular exponentiation) that is an expensive operation for IoT devices in real systems. In the VSS setting, it is a challenge to design a lightweight VSS scheme that minimizes the energy consumption of a participant. To our knowledge, this paper represents the first effort in this direction.

B. Our Contribution

In this paper, we propose a lightweight VSS scheme in IoT environments. The security of the proposed scheme is based on NAHFs which are implemented through the generic symmetry-based hash function and simple bit-wise operation. The proposed scheme dictates to generate only a NAHF value as the verification data which proves the validity of the shares for all participants. Thus, the communication cost of each participant is reduced. In addition, each participant validates a received share by running an NAHF operation. Hence, the proposed scheme is computationally efficient for each participant. Furthermore, the computation and communication costs of each participant remain unchange when the number of participants increases. That is, the proposed scheme provides the good scalability. Compared to the Feldman scheme [4] and Rajabi-Eslami scheme [3], the energy consumption of a

[*] Corresponding Author.

participant in the proposed scheme is respectively reduced by at least 24% and 83% for a secret. To the best of our knowledge, the approach of this paper is the first such technique that the number of verification data is only one value in the VSS scheme.

The rest of the article is structured as follows. Related work is presented in Section II, Section III presents a brief review of NAHF, Shamir's (t, n) secret sharing and VSS. Section IV is dedicated to the proposed VSS scheme including the security model, construction and security aspects. The performance analysis and simulation experiments for the proposed scheme are respectively discussed in Section V and Section VI. Section VII concludes the paper.

II. Related Work

The secret sharing (SS) scheme is used as a tool in IoT applications including continuous authentication [1] and key management in sensor networks [7]. Such a scheme allows one to share a secret s among a set P of participants. The participants are assigned different values called shares and only certain authorized subsets of them were able to recover the secret using these shares. A (t, n) threshold SS scheme was introduced by Shamir [8] and Blakley [9] independently in 1979. In such a scheme, the authorized subsets consisted of all subsets of P including at least t participants. The scheme was unconditionally secure which meaned that less than t participants found no information about the secret even with unlimited time and computing power. Then, many versions of SS were proposed to add some new features in the literatures [10].

A verifiable secret sharing (VSS) scheme is a generalization of a SS scheme [11], whose novelty is that everyone can verify whether the received share is a valid piece of the secret or not. The concept of VSS was first introduced by Chor et al [12] in 1985. Subsequently, based on "k-consistent" shares and interactive proof in [13], a VSS scheme was proposed to check the honesty of participants at the secret reconstruction phase. However, at the share generation phase, participants were unable to verify whether the shares they received from the dealer were valid. In 1987, a practical non-interactive VSS was proposed by Feldman [4, 5] through a homomorphic one-way function v for verifying consistency of each share. Indeed, let v be a $(+,\cdot)$-homomorphic one-way function (that is, $v(a+b){=}v(a)\cdot v(b)$); then, if v was evaluated over a polynomial $f(x)=\sum_{i=0}^{t-1} a_i x^i$, the equation $v(f(x))=\prod_{i=0}^{t-1} v(a_i x^i)$ held. The dealer chose two primes p, q as public values and a generator g of a subgroup of order q of $\mathbb{Z}_p^*$, where q divided $p-1$, and q was the lowest possible integer satisfying $g^q \equiv 1 \bmod p$. Then, it generated a share $s_j{=}f(x_j) \bmod q$ for each participant P_j, and published the public verification coefficients $A_i = g^{a_i} \bmod p$. Hence, the consistency of a share s_j was verified by checking the equality $g^{s_j}{=}\prod_{i=0}^{t-1} A_i^{x_j^i} (\mathrm{mod} p)$. Here, the homomorphic property of exponentiation function $v(a){=}g^a \bmod p$ was used. In the case of Feldman's scheme, the security was based on the hardness of the discrete logarithm problem (DLP). In 2019, Rajabi and Eslami [3] proposed a generic threshold VSS construction, and then presented a non-interactive VSS with security based on hardness of the approximate shortest polynomial problem (ASPP) in cyclic lattices. In the work of Tsaloli et al. [14], by combining three different primitives (i.e., homomorphic hash functions, linearly homomorphic signatures, and threshold RSA signatures) as the baseline, an approach was proposed for protecting the secret data of clients and achieving public verifiability of the computed result. Recently, Koikara et al.[15] used a bilinear map to propose a publicly verifiable secret sharing (PVSS) scheme based on 3D-cellular automata. The VSS with bilinear pairings is not suitable for IoT systems because bilinear pairings are not friendly to lightweight devices [16]. In addition, the symmetry-based VSS is more suitable for the ultra-low energy devices as compared with the public key cryptographic approaches.

A new non-trapdoor accumulator for cumulative hashing was introduced by Nyberg [17]. This kind of accumulator is called a Nyberg's one-way Accumulator for one-way Hash Function (NAHF). In practice, the NAHF is effectively implemented by using the generic symmetry-based hash function and simple bit-wise operations. Oftentimes, this results in less memory requirements than digital signature-based solutions for verification problems. In 2017, Huang et al. [18] proposed a lightweight authentication scheme with dynamic group members in IoT environments. Here, based on a public secure NAHF, the proxy computed two accumulated hash values, W and R, which were used to verify whether the node was available and unrevoked. Recently, Fan et al. [19] presented a secure region-based handover scheme with user anonymity and fast revocation, where the region secret keys of the revoked users were accumulated by NAHFs. In the proposed scheme, the dealer generates the verification data with a NHAF such that the shares of participants can be publicly and efficiently verified. This enables us to add verification capability for participants using only one verification data.

III. Preliminaries

In this section, we introduce some basic concepts of hash function, NHAF, secret sharing and VSS needed later

A. Notations

We shall use the following notations throughout the paper. A set with integers $1, 2, \cdots, n$, is written either $\{1, 2, \cdots, n\}$ or simply [n]. We denote by $|x|$ the length of the binary string corresponding to x, and $\lceil x \rceil$ the least integer that is greater than or equal to the given number x. Let $P = \{P_1, P_2, \cdots, P_n\}$ be a set of n participants and D be the dealer. The threshold is denoted by t. Let $\mathbb{Z}_p$, $\mathbb{Z}_q$ be two finite fields and $\mathbb{Z}_q^* = \mathbb{Z}_q \setminus \{0\}$, where p is a prime modulus, q is a prime divisor of $p-1$, and $q \geq n+1$. We let $H : \{0,1\}^r \times \{0,1\}^* \to \{0,1\}^r$ denote a Nyberg accumulated hash function, $h : \{0,1\}^* \to \{0, 1, \cdots, q-1\}$ and $\hat{h} : \{0,1\}^* \to \{0,1\}^{rd}$ be two one-way hash functions, where h is used to construct the required H, and $r{=}|q|$.

B. Nyberg's One-way Accumulator for One-way Hash Function

In this paper, we review the concept of Nyberg's one-way Accumulator for one-way Hash Function (NAHF).

Definition 1 (One-way hash function [17])**.** *A family of one-way hash functions is an infinite set of functions $h_l : K_l \times S_l \to V_l$ having the following properties:*

(1) *There exists a polynomial P' such that for each integer l, $h_l(k,s)$ is computable in time $P'(l,|k|,|s|)$ for all $k \in K_l$ and all $s \in S_l$.*

(2) *There is no polynomial P' such that there exists a probabilistic polynomial time algorithm which, for all sufficiently large l, when given l, a pair $(k,s) \in K_l \times S_l$, and a $s' \in S_l$, find an $k' \in K_l$ such that $h_l(k,s) = h_l(k',s')$ with probability greater than $1/P'(l)$, where (k,s) is chosen uniformly among all elements of $K_l \times S_l$ and s' is chosen uniformly form S_l.*

Definition 2 (Quasi-commutativity [17])**.** *A function $h : K \times S \rightarrow X$ is said to be quasi-commutative if for all $k \in K$ and for all $s_1, s_2 \in S$, $h(h(k,s_1),s_2) = h(h(k,s_2),s_1)$.*

Definition 3 (Nyberg's one-way accumulator [17])**.** *A family of one-way accumulators is a family of one-way hash functions with quasi-commutativity. The one-way accumulator by Nyberg [17] 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 $N = 2^d$ is an upper bound to the number of items to be accumulated and r is an integer. Let $s_1, s_2, \cdots, s_n$ be the accumulated items with different string sizes, and a set of the accumulated items $S = \{s_1, s_2, \cdots, s_n\}$, where $n \leq N$. Assume that $H(\cdot,\cdot)$ denotes an NAHF from $\{0,1\}^r \times \{0,1\}^*$ to $\{0,1\}^r$, and $\odot$ is the bitwise operation AND. The NAHF is based on the one-way hash function $h : \{0,1\}^* \rightarrow \{0,1\}^{rd}$. All that is required to specify the NAHF is hashing process and AND operation. The heart of an NAHF is the hashing process. The hashing process applies a hash function h to the input to produce a r-bit output. The hashing process is composed of the following operations.

- Hashing operation: Hash the accumulated item s_i of the input and output a rd bits binary string v_i=$h(s_i)$.
- Transfer α: the NAHF does a transfer operation on the binary string v_i which is divided into r blocks, $(v_{i,1}, \cdots, v_{i,r})$, of length d. The transfer of a block from a d-bit input to a bit output is performed as follows: If $v_{i,j}$ is a string of zero bits, it is replaced by 0; otherwise, $v_{i,j}$ is replaced by 1. That is, $\alpha(v_i)$ =$(b_{i,1}, \cdots, b_{i,r})$, where $b_{i,j} \in \{0,1\}$, j=$1, \cdots, r$.

In this way, we can transfer the accumulated item s_i to a bit string, b_i=$\alpha(h(s_i)) \in \{0,1\}^r$, which can be considered as the values of r independent binary random variables if h is an ideal hash function.

The NAHF on an accumulated item $s_i \in S$ with an accumulated key $k \in \{0,1\}^r$ can be implemented using the AND operation described as $H(k,s_i) = k \odot \alpha(v_i) = k \odot \alpha(h(s_i))$. And it also can be represented as $Z = H(k,s_i)$ = $k \odot \alpha(v_i) = k \odot \alpha(h(s_i))$ for an accumulated item $s_i \in S$ ($i \in [n]$). The proposed VSS relies on the following properties of the NAHF $H(\cdot,\cdot)$:

- Quasi-commutativity: $H(H(k,s_1),s_2) = H(H(k,s_2),s_1)$.
- Absorbency: $H(H(k,s_i),s_i) = k \odot \alpha(h(s_i)) = H(k,s_i)$.
- An item s_i within the accumulated value Z can be verified by $H(Z,s_i) = Z \odot \alpha(h(s_i)) = Z$.

C. Shamir's Threshold Secret Sharing

There are n participants, $P = \{P_1, P_2, \cdots, P_n\}$ and a dealer D. In Shamir's secret sharing scheme [8], it consisted of two phases: the share distribution phase and the secret reconstruction phase. During share distribution, the secret was $s = f(0)$, where $f(x)$ was a polynomial of degree $t-1$ with random coefficients (except for the constant term), computed over a finite field. The participant $P_j \in P$ holding shares knew $s_j = f(x_j)$, where x_j was P_j's unique nonzero identifier, $j \in [n]$. In secret reconstruction, any t out of n participants, $P_{j_1}, \cdots, P_{j_t}$, were able to recover the secret s by using the Lagrange interpolation formula (1) or solving the following linear equations (2), where

$$s = f(0) = \sum_{i=1}^{t} s_{j_i} \Big(\prod_{r=1, r\neq i}^{t} \frac{-x_{j_r}}{x_{j_i} - x_{j_r}} \Big), \tag{1}$$

and

$$\begin{aligned} s_{j_1} &= s + a_1 \times x_{j_1} + \cdots + a_{t-1} \times x_{j_1}^{t-1}, \\ s_{j_2} &= s + a_1 \times x_{j_2} + \cdots + a_{t-1} \times x_{j_2}^{t-1}, \\ &\vdots \\ s_{j_t} &= s + a_1 \times x_{j_t} + \cdots + a_{t-1} \times x_{j_t}^{t-1}. \end{aligned} \tag{2}$$

Note that the above coefficient matrix is a square Vandermonde matrix, which is invertible, since the x_js are distinct.

D. VSS

In a SS scheme, participants must trust that shares they receive are correct. In a VSS scheme, additional verification data are given that allow each participant to check whether its share is correct. Each message that must be checked contains additional verification data. The verification data are sent in the clear, and can be used by the recipient to determine whether the share in the message is correct. That is, recipients use them to check that a point, (x_j, s_j), sent to it is on the polynomial $f(x)$ and that the polynomial, $f(x)$, used as the basis for the sent shares equals the secret at x =0. The VSS is able to resist the following two kinds of active attacks: (1)some shares are tampered before being sent to the participants in the secret distribution phase; (2)participants submit error shares to others in the secret reconstruction phase.

IV. A Lightweight (t,n) VSS Scheme

In the section, a lightweight (t,n) VSS scheme is proposed. We discuss techniques involving the security model, construction and the security aspects of the proposed scheme.

A. The Security Model of Proposed Scheme

In this section, we give the definition of a noninteractive (t,n) VSS scheme. There are n participants, $P = \{P_1, P_2, \cdots, P_n\}$, and a dealer D. In the definition, there are four algorithms: share generation(SG), share verification(SHV), secret reconstruction(SR) and secret verification

(SEV). The proposed scheme consists of the share distribution phase and the secret reconstruction phase. We define a noninteractive (t, n) VSS scheme as follows:

A noninteractive (t, n) VSS scheme is a pair (share generation, secret reconstruction) of phases as follows.

- Share distribution: In this phase, on input a secret s and P_j's identity x_j, D first runs SG algorithm to output a share for each participant and some verification data, where the shares are sent to the corresponding participants through a secure channel. Then, on input verification data and his share, each participant runs SHV algorithm to output accept or reject the share.
- Secret reconstruction: The input of this phase are the shares corresponding to a subset of participants. At first, the validity of each share is verified by other cooperating participants running SHV algorithm. Then, if the number of participants with valid shares is at least t, the secret can be computed by applying SR algorithm on the provided shares, and the recovered secret is verified by running SEV algorithm.

A non-interactive (t, n) VSS Scheme is called secure if it satisfies the following properties:

- Threshold. Every secret can only be recovered by any t or more participants who have received the shares, and any subset of participants with less than t participants cannot obtain any information about the secret.
- Verifiability/reconstructability: Every participant can verify his share in the share generation phase. During the secret reconstruction phase, the participants can validate the received shares and check if a reconstructed secret is correct.
- Security. The VSS scheme must be able to resist up to $t-1$ colluded inside adversaries. In addition, any outside adversary cannot impersonate to be a member by forging a valid value after knowing at most $t-1$ values from other members. The VSS scheme is secure, if the adversary cannot obtain the shares in polynomial time.

In addition, the following properties for a VSS are very much tailored to IoT devices as participants:

- Efficiency. The proposed scheme should have low calculation requirements and low communication costs at the participants to reduce their energy consumptions. This makes VSS for implementation on battery-powered IoT devices that have limited computing power.
- Scalability. Even if the number of participants in large-scale deployments is big, the communication cost of the scheme should be kept small to reduce the cost of the supporting network infrastructure.

B. The Proposed (t, n) VSS Scheme

Figure 1 shows the proposed (t, n) VSS scheme, where the combiner may be each participant in P. In the proposed scheme, the algorithms SG, SHV, SR and SEV are the mathematical processes in the Step 1, 2, 3 and 4, respectively. The security of the scheme is based on an NAHF, which is quasi-commutative and has the absorbency property.

Share generation

Step 1: Dealer D chooses a $(t-1)$ degree polynomial $f(x) = a_o + a_1 x + \cdots + a_{t-1}x^{t-1}$, where $a_0 = s$ is the secret in $\mathbb{Z}_q^*$, and for $i = 1, 2, \cdots, t-1$, a_i are picked uniformly from $\mathbb{Z}_q^*$. The dealer distributes the share $s_j = f(x_j)$ to the corresponding participant P_j secretly, where x_j is P_j's unique nonzero identifier, $j \in [n]$. Then, based on an NAHF H, the dealer selects k as its long term secret key to compute the verification data V, and publishes H and V, where $V = H(\cdots H(H(k, s), s_1), \cdots, s_n)$.
Step 2: After receiving s_j, each participant P_j checks if $H(V, s_j) = V$ holds, $j \in [n]$. If true, P_j confirms that the received share s_j is correct; otherwise, it repeats the step 1.

Secret reconstruction

Step 3: The participant P_j releases its share s_j, and the combiner confirms the correctness of s_j via $H(V, s_j) = V$.
Step 4: Assume that the combiner receives t correct shares $s_{j_1}, s_{j_2}, \cdots, s_{j_t}$. The secret s is recovered by the formula (1) or solving equation (2). Then, the combiner validates the recovered secret s as $H(V, s) = V$.

Fig. 1. The Proposed (t, n) VSS Scheme.

The correctness of the proposed (t, n) VSS scheme is guaranteed by the following theorem 1 and 2.

Theorem 1. *In the share generation, the correctness of each share s_j can be validated by the receiver through $H(V, s_j) = V$, $j \in [n]$.*

Proof 1. *If the dealer D follows the scheme accurately, we have that $V = H(\cdots H(H(k, s), s_1), \cdots, s_n)$. Based on the absorbency property of H, it is known that the share s_n satisfies $H(V, s_n)$ =V. In fact, $H(V, s_n) = H(H(\cdots H(H(k, s), s_1), \cdots, s_n), s_n) = H(\cdots H(H(k, s), s_1), \cdots, s_n) = V$, where the second equality holds for the absorbency property of H.*

Generally, in accordance with the quasi-commutativity of H, we have

$$\begin{aligned} V &= H(\cdots H(H(\cdots H(H(k,s),s_1),\cdots,s_j),s_{j+1}),\cdots,s_n) \\ &= H(\cdots H(H(\cdots H(H(k,s),s_1),\cdots,s_{j+1}),s_j),\cdots,s_n) \\ &\;\;\vdots \\ &= H(H(\cdots H(\cdots H(H(k,s),s_1),\cdots,s_{j+1}),\cdots,s_n),s_j). \end{aligned} \tag{3}$$

where $j = 1, 2, \cdots, n-1$. Combining the absorbency property of H and equation (3), we obtain that $H(V, s_j) = H(H(H(\cdots H(\cdots H(H(k, s), s_1), \cdots, s_{j+1}), \cdots, s_n), s_j), s_j) = H(H(\cdots H(\cdots H(H(k, s), s_1), \cdots, s_{j+1}), \cdots, s_n), s_j) = V$, where the second equality holds for the absorbency property of H, and the third equality holds due to equation (3). This completes the proof.

Theorem 2. *In the secret reconstruction, the received shares*

s_{j_θ} and the recovered secret s can be publicly and efficiently verified via $H(V, s_{j_\theta}) = V$ and $H(V, s) = V$, respectively, $\theta \in [t]$.

Proof 2. *In the secret reconstruction, the share s_{j_θ} can be publicly and efficiently verified via $H(V, s_{j_\theta})$ =V, for $\theta \in [t]$. This proof is the same as that of Theorem 1. In addition, similar to the derivation of equation (3), the secret s satisfies the following equation:*

$$\begin{aligned} V &= H(\cdots H(H(k,s), s_1), \cdots, s_n) \\ &= H(\cdots H(H(k,s_1), s), \cdots, s_n) \\ &\;\;\vdots \\ &= H(H(\cdots H(H(k,s_1), s_2), \cdots, s_n), s). \end{aligned} \tag{4}$$

By using the absorbency property of H and equation (4), for the secret s we see that $H(V,s) = V$. This is because $H(V,s) = H(H(H(\cdots H(H(k,s_1),s_2),\cdots,s_n),s),s)$ $= H(H(\cdots H(H(k,s_1),s_2),\cdots,s_n),s) = V$, where the second equality holds due to the absorbency property of H, and the third equality holds by equation (4). This completes the proof.

Remark 1. *The correctness of algorithms $H(V, s_j)$ =V and $H(V, s)$ =V depends on the assumption that the output length, rd, of h satisfies $(n+1) \leq 2^d$, where an NAHF $H : \{0,1\}^r \times \{0,1\}^* \rightarrow \{0,1\}^r$ is constructed through $h : \{0,1\}^* \rightarrow \{0,1\}^{rd}$. When $(n+1) > 2^d$, it is feasible to replace V with $(V_0, V_1, \cdots, V_{u-1})$, where u =$\lceil \frac{n+1}{2^d} \rceil$. For $\varsigma = 0, 1. \cdots, u-1$, $V^{(\varsigma)}$ is generated as follows: (1) different hash functions, $h^{(\varsigma)} : \{0,1\}^* \rightarrow \{0,1\}^{rd}$, are chosen. (2) the NAHF $H^{(\varsigma)} : \{0,1\}^r \times \{0,1\}^* \rightarrow \{0,1\}^r$ is generated by the hash function $h^{(\varsigma)}$. (3) Let $s_{n+1} = s$, the ς-th value is computed as $V^{(\varsigma)} = H^{(\varsigma)}(\cdots H^{(\varsigma)}(k, s_{\varsigma+1}), \cdots, s_{\varsigma+2^d})$. To verify the correctness of $s_{\varsigma+j}$, we can check if $H^{(\varsigma)}(V^{(\varsigma)}, s_{\varsigma+j}) = V^{(\varsigma)}$, where $\varsigma = 0, 1. \cdots, u-1$, and $j \in [2^d]$.*

The following theorems ensure the security of the proposed (t, n) VSS scheme.

Theorem 3. *Assume that q is a large prime number. The share s_j obtained by the polynomial $f(x)$, has a uniform distribution on $\mathbb{Z}_q$, $j \in [n]$.*

Proof 3. *Let A and X be two independent random variables defined on $\mathbb{Z}_q$. A basic result from the theory of random variables is that if A has a uniform distribution on $\mathbb{Z}_q$ and X has an arbitrary distribution on $\mathbb{Z}_q$, then $B_1 = A + X \pmod q$ and $B_2 = A \cdot X \pmod q$ have a uniform distribution on $\mathbb{Z}_q$, where X is chosen from $\mathbb{Z}_q^*$ in the latter case. If b_1 is chosen uniformly from all possible values of B_1, the probability of $B_1 = b_1$ is given as:*

$$\begin{aligned} Pr[B_1 = b_1] &= Pr[A + X = b_1] \\ &= \sum_{x_j} Pr[A = b_1 - x_j] Pr[X = x_j] \\ &= 1/q \cdot \sum_{x_j} Pr[X = x_j] = 1/q. \end{aligned}$$

Similarly, when b_2 is chosen uniformly from all possible values of B_2, we have

$$\begin{aligned} Pr[B_2 = b_2] &= Pr[A \cdot X = b_2] \\ &= \sum_{x_j} Pr[A = b_2 \cdot (x_j)^{-1}] Pr[X = x_j] \\ &= 1/q \cdot \sum_{x_j} Pr[X = x_j] = 1/q. \end{aligned}$$

It can be easily shown that the above argument can be extended to the random polynomial function $f(x)$. Since $a_0, a_1, \cdots, a_{t-1}$ are uniformly distributed on $\mathbb{Z}_q$ and x_j is P_j's unique nonzero identifier, hence $a_0, a_1 x_j, \cdots, a_{t-1} x_j^{t-1}$ are uniformly distributed on $\mathbb{Z}_q$. Then, $f(x_j) = a_0 + a_1 x_j + \cdots + a_{t-1} x_j^{t-1}$ is uniformly distributed on $\mathbb{Z}_q$. Therefore, $s_j = f(x_j)$ is uniformly distributed on $\mathbb{Z}_q$, that is, s_j has a uniform distribution on $\mathbb{Z}_q$.

Theorem 4. *Under the assumption that H is a secure NAHF, the secret s and some shares s_j cannot be obtained by an attacker from V, $j \in [n]$.*

Proof 4. *Recall from Definition 3 that an NAHF H is a one-way hash function with quasi-commutativity. Suppose the accumulated item s_j is computed in the j-th iteration of V, thus, V $= H(\cdots H(H(\cdots H(H(k,s),s_1),\cdots,s_j),s_{j+1}),\cdots,s_n)$. Note that $V = H(H(\cdots H(\cdots H(H(k,s),s_1),\cdots,s_{j+1}),\cdots,s_n),s_j)$ $= H(Q, s_j)$, where the first equality holds due to equation (3), and $Q = H(\cdots H(\cdots H(H(k,s),s_1),\cdots,s_{j+1}),\cdots,s_n)$. Furthermore, we have that $H(V, s_j)$=V. We now need to prove that it is hard for the attacker presented with V to find (Q', s_j) such that $V = H(Q', s_j)$. At this point, One-way property of H in Definition 1 ensures that this is computationally infeasible, that is, there is no polynomial P' such that there exists a probabilistic polynomial time algorithm which finds an $s_j \in \mathbb{Z}_q$ such that $V = H(Q', s_j)$ with probability greater than $1/P'(l)$, where Q' is chosen uniformly form $\{0,1\}^r$. Hence, it is computationally infeasible to find an s_j such that $H(V, s_j) = V$, $j \in [n]$. Similarly,it is computationally infeasible to derive the share s from V.*

Theorem 5. *In the proposed VSS scheme, any subset of participants of size less than t cannot obtain any information about the secret s.*

Proof 5. *Here, we consider the worst case, where $t-1$ participants take part in recovering the secret s. Any $t-1$ participants with different identities $x_{j_1}, \cdots, x_{j_{t-1}}$ cannot compute the secret s since they cannot solve the linear system of $(t-1)$ equations and t unknowns: $s_{j_l} = s + a_1 \times x_{j_l} + \cdots + a_{t-1} \times x_{j_l}^{t-1}$, $l \in [t-1]$, which has a degree of freedom, where $a_0 = s$. We can consider the coefficient, a_{t-1}, of the last term in $f(x)$ as a free variable from $\mathbb{Z}_q$. In this case, the secret s has a unique representation as a linear combination of a_{t-1} and the shares $\{s_{j_1}, \cdots, s_{j_{t-1}}\}$, where a_{t-1} is uniformly distributed over $\mathbb{Z}_q$. From the proof of Theorem 3, it follows that s has a uniform distribution over $\mathbb{Z}_q$. Hence, no information about the secret s can be extracted from these $t-1$ shares.*

Combining Theorem 3, 4 and 5, we have the following theorem:

Theorem 6. *The proposed (t, n) VSS scheme is secure under the assumption that H is a secure NAHF.*

TABLE I. THE COMMUNICATION COSTS OF D AND P_j IN THE VSS SCHEMES

	share $\|f(x_j)\|$	verification data	D	P_j
Rajabi-Eslami[3]	m$n_0\|p_0\|$	t $\|F(a[i])\|$ = t$n_0\|p_0\|$	(t+nm)$n_0\|p_0\|$	(m(t+1)+t)$n_0\|p_0\|$
Feldman [4]	$\|q\|$	t $\|A_i\|$=t$\|p\|$	n$\|q\|$+$t\|p\|$	t$\|p\|$+ (t+1)$\|q\|$
proposed scheme	$\|q\|$	$\|V\|$=r	n$\|q\|$+r	(t+1)$\|q\|$ +r

TABLE II. THE COMPUTATION COSTS OF D AND P_j IN THE VSS SCHEMES, WHERE T$_o$ IS THE COMPUTATION TIME FOR THE OPERATION $o \in \{F, H, M$(MULTIPLICATION), e(EXPONENTIATION), E(EXPONENTIATION ON R_{p_0}), f (COMPUTING $f(x_j)$ON R_{p_0})$\}$, pm(POLYNOMIAL MULTIPLICATION ON R_{p_0})

	share $\|f(x_j)\|$	verification data	verify $f(x_j)$	get s	D	P_j
Rajabi-Eslami[3]	$T_{f(x_j)}$=T_f	t $T_{F(a[i])}$=tT_F	(t-1)T$_E$+T$_F$	mtT$_M$	nT$_f$ + tT$_F$	t(t-1)T$_E$ + mt(T$_{pm}$ +T$_M$)
Feldman[4]	(t-1)T_M	t T_{A_i} = tT_e	tT_e	tT_M	n(t-1)T_M + tT_e	t^2T_e + tT_M
proposed scheme	(t-1)T_M	T_V = (n+1)T_H	T_H	tT_M	n(t-1)T$_M$+(n+1)T$_H$	(t+1)T_H +tT_M

V. PERFORMANCE OF PROPOSED VSS SCHEME

In this section we present and discuss the efficiency and scalability for the proposed scheme in Section IV-B. We mainly consider the costs of an extension of the SS scheme to achieve verifiability. By decreasing the number of verification data, we improves on the previous VSS schemes [3, 4]. We estimate the efficiency by counting the number of basic cryptographic operations required in the extension, and also calculate its communication cost. To evaluate the scalability of the proposed scheme, it suffices to show that the costs of each participant remain unchange in the increase in the size of the IOT network (i.e. the number of participants).

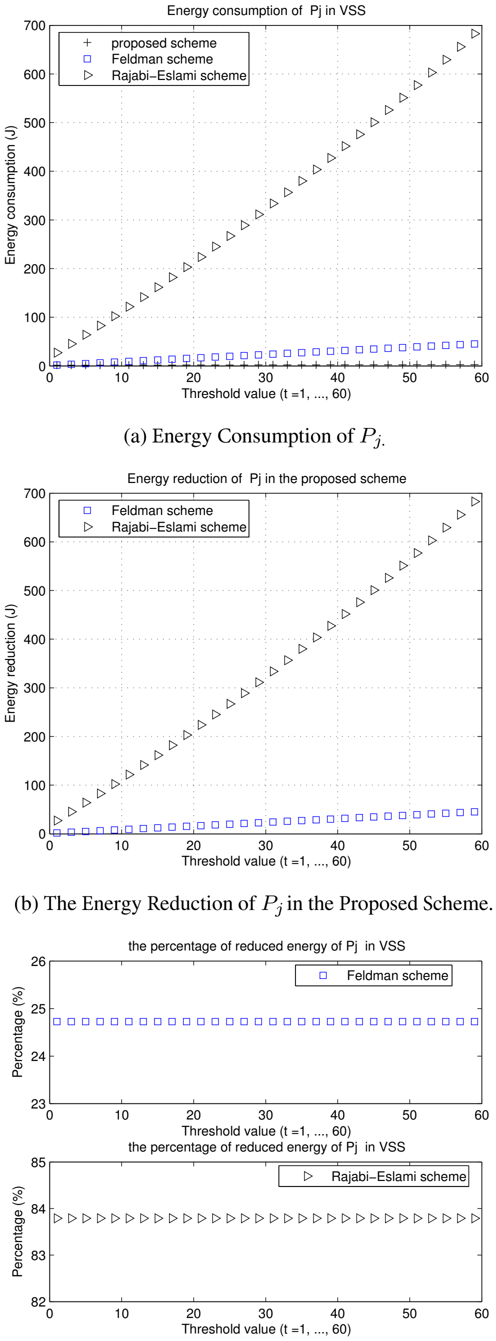
Bandwidth is a scarce resource. In a VSS, the communication cost is dominated by the sizes of both verification data and a share. From Table I, we see that in the proposed scheme, the communication costs of the D and P_j are significantly lower than Feldman scheme and Rajabi-Eslami scheme since $|q|$ is much less than $|p|$ and $mn_0|p_0|$ (see Section VI). In the proposed scheme, the verification data V is only a value in $\mathbb{Z}_q$, so is any share. Specifically, at the share generation phase, the dealer D broadcasts V to participants in P and transmits s_j=$f(x_j)$ to each participant P_j, $j \in [n]$, where $|V| + \sum_{j=1}^{n} |s_j| = r + n|q|$ bits. Upon receiving V and s_j from D at the share generation phase, each P_j obtains at least $(t-1)$ different shares s_{j_θ} from the others in P while sending s_j to them at the secret reconstruction phase. Here, $|V| + |s_j| + \sum_{\theta=1}^{t-1} |s_{j_\theta}| + |s_j| = r + (t+1)|q|$ bits. In the Feldman scheme[4], the verification data included t elements $A_0, \cdots, A_{t-1}$ (see Section II) in $\mathbb{Z}_p$, and the size of each share was the same as the proposed scheme. Therefore, the communication costs at D and P_j were $n|q| + t|p|$ and $t|p| + (t+1)|q|$ bits, respectively. In the Rajabi-Eslami scheme [3], the verification data was composed of (n_0-1)-degree polynomials $F(a[0]), \cdots, F(a[t-1])$ in $\mathbb{Z}_{p_0}$ and each share contained m polynomials in R_{p_0}. Here, the polynomial ring R_{p_0} = $\mathbb{Z}_{p_0}[\alpha]/(\alpha^{n_0}-1)$, and D_{p_0} was an appropriate subset of "small" elements of R_{p_0} [1], where the dimension $m > 1$, the integer module $p_0 \geq 2$ and an error distribution δ. Note that each share $f(x_j)$ and the secret s were respectively composed of m polynomials in R_{p_0} and D_{p_0}, and $F(a[i])$ was a polynomial in R_{p_0}. Thus, the communication costs at D and P_j were $(t+nm)n_0|p_0|$ and $(m(t+1)+t)n_0|p_0|$ bits, respectively.

It is generally assumed that in an IoT system, the dealer or server has powerful computing resources and the computing power of IoT devices is limited [22]. Another advantage of proposed scheme is that the computation costs of participants are low since computational requirements are the basic criteria for known IoT devices. For each participant P_j in the proposed scheme, its computation cost is $(t+1)T_H + tT_M$, where $H(V, s_j)$, $H(V, s_{j_\theta})$ and $H(V, s)$ are respectively computed for verifying s_j, s_{j_θ} $(\theta \in [t-1])$ and the recovered s, and t multiplication operation in the Lagrange interpolation formula (1) are performed to recover s. Note that in Rajabi-Eslami scheme [3], $T_f = m(t-1)T_m$, and $T_F = mT_{pm}$. This was because m polynomials with degree $(t-1)$ needed to be computed for each $f(x_j)$ in R_{p_0} and $F(X) = \sum_{i=1}^{m} X_i b_i$. From the experimental results in Section VI, we know that $T_M < T_{pe} < T_H < T_e < T_E$. Table II shows that the computation cost of P_j is the lowest in the proposed scheme. In contrast, the computation cost of D in the proposed scheme, where the time to compute V increases with n, increases due to the use of NAHF H. To compute $s_j = f(x_j)$ for each participant P_j and verification data V, D needs to execute $t-1$ multiplication operations for $f(x_j)$ and $n+1$ NAHF operations for V, $j \in [n]$. It means that the computation cost of D is $n(t-1)T_M + (n+1)T_H$. Furthermore, the proposed scheme provides the good scalability since the computation and communication costs of P_j remain unchange when the number of participants increases.

VI. SIMULATION EXPERIMENTS

We further evaluate the performance of proposed scheme using simulation experiments. The experiments are conducted on an Intel(R) Core(TM) i7-6700 CPU@3.40 GHz machine with 8.00 GB memory and Windows7 using JDK1.8. We choose to focus on SHA-512 for hashing h in NAHF H with a 128 bit output, where $N = 2^4$ is an upper bound to the number of accumulated items. When $N > 2^4$, we do this by selecting $u = \lceil N/(2^4) \rceil$ different SHA-512 as Remark 1. For Feldman scheme, the parameters p, q were chosen as suggested (see page 21 in [23]), i.e.,$|p|$=1024bits, and $|q|$=160bits. As for Rajabi-Eslami scheme, according to the LWE parameters for hardware tests (see Table 4 in [24]), the corresponding parameters $(n_0, |p_0|) = (128, 12)$. In addition, let $m = 2$. To give a detailed quantitative analysis, we assume

[1] $\mathbb{Z}_{p_0}$ was the set of integers from 0 to $p_0 - 1$, $\mathbb{Z}_{p_0}[\alpha]$ denoted the set of polynomials with coefficients in $\mathbb{Z}_{p_0}$. R_{p_0} containd all polynomials of degree less than n_0 with coefficients in $\mathbb{Z}_{p_0}$, as well as two ring operations, which were polynomial addition and multiplication modulo $\alpha^{n_0} - 1$. Each polynomial in R_{p_0} had n_0 coefficients in $\mathbb{Z}_{p_0}$, so there was a bijection between R_{p_0} and $\mathbb{Z}_{p_0}^{n_0}$. The compact knapsack problem over R_{p_0} was defined in [21] as follows: given $m = \mathcal{O}(log_2 n_0)$ elements $b_1, \cdots, b_m \in R_{p_0}$ and a target value $c \in R_{p_0}$, found coefficients $X_1, \cdots, X_m \in D_{p_0}$ such that $\sum_{i=1}^{m_1} X_i b_i$=c.



(a) Energy Consumption of P_j.

(b) The Energy Reduction of P_j in the Proposed Scheme.

(c) The Percentage of Reduced Energy of P_j.

Fig. 2. The Energy Consumption, Energy Reduction and the Percentage of Reduced Energy for P_j in the Proposed Scheme.

that participants are MICA2 motes, which work at 8 MHz with a 8-bit processor ATmega128L, and which adopt IEEE 802.15.4 standard. As described in Cao et al. [25], the power level of a MICA2 mote is U = 3.0 V, the current draw in active mode is I = 8.0 mA, the receiving current draw is I_r = 10 mA, the transmitting current draw is I_t=27 mA, and the data rate is r_d = 12.4 kbps. The cost of receiving (or transmitting) one byte is $E_r = UI_r(8/r_d)$ = $19.35\mu J$ (or $E_t = UI_t(8/r_d)$ = $52.26\mu J$). The parameters are fixed in all experiments.

Experiment 1 examines the average time required to run an operation in Table II. With the above parameter settings, we consider the average value of over 160 trials for an operation. The results are as follows: T_M= 0.0022milliseconds (ms), T_H=0.0858(ms), T_e=1.3445(ms), T_{pm}=0.0169(ms), T_E = 1.6071(ms). Especially, the average time performing the addition operation is 0.0007ms, which is negligible compared with the others.

Experiment 2 examines the energy consumption of a participant. To compute the electrical energy consumed by a participant during t_p seconds, we apply Joule's law as $E = UIt_p$. From Table II and Table I, we have that $t_p = (t+1)T_H + tT_M = (t+1) \times 0.0858 + t \times 0.0022$ (ms), and $(t+1)|q| + r$ is equal to $4 + 40$ bytes, where transmitting bytes are 20 and receiving bytes are $t + 20$. For P_j, the energy cost of communication is $20 \times E_t + (t+20) \times E_r$ $=19.36t + 1432.2(\mu J)$, and the energy cost of computation is $3 \times 8 \times t_p = 2.112t + 2.0592(\mu J)$. Thus, the energy consumption of P_j is $21.472t + 1434.2592(\mu J) \approx 0.0215t + 1.4343(J)$. We find that the energy cost of computation is cheap compared to data communication. Again, we compare the energy consumption of P_j in the proposed scheme with that of Feldman scheme and Rajabi-Eslami scheme. From Figure 2 (a), it is evident that the energy consumption of P_j increases with the threshold value, but it is relatively stable in the proposed scheme. In particular, the proposed scheme makes P_j have the smallest energy consumption. Note that given a threshold value t, the energy reduction is equal to the difference of the corresponding ordinate values of two schemes in Figure 2(a). For each participant P_j, the results of energy reduction are shown in Figure 2(b). Compared with the Feldman scheme, the energy reduction of the proposed scheme is larger in the Rajabi-Eslami scheme, and the difference increases with t. Furthermore, Figure 2 (c) shows that, compared to the Feldman scheme and Rajabi-Eslami scheme, the energy consumption of P_j in the proposed scheme is respectively reduced by at least 24% and 83% for a secret.

VII. Conclusion

In this paper, we give a lightweight (t, n) VSS scheme based on an NAHF [17]. Different from previous VSS schemes, the proposed scheme generates only a NAHF value as the verification data which proves the validity of the shares for all participants. This means that the scheme has less communication cost than previous approaches to achieve the share verification. Another important property is that the computation and communication costs of each participant remain unchange when the number of its participants increases. It is convenient for building a secure scalable IoT network. At the same time, because the correctness of each share can be efficiently checked, the new participant can verify whether his share is right or not. Using the proposed scheme, we can make a IoT system more secure and efficient. The presented scheme can be applied to multi-party computation, electronic voting, secure databases and many circumstances.

Acknowledgment

This work is supported in part by National Key Research and Development Plan of China under Grants 2018YFB1003701, in part by the Science and Technology Planning Project of Guangdong under Grants 2021B0101420003, 2020B0909030005, 2020B1212030003, 2020ZDZX3013, and in part by the 22nd batch of Teaching Reform Research Project of Jinan University under Grant 55611518.

References

[1] O. O. Bamasag, K. Youcef-Toumi, Towards continuous authentication in internet of things based on secret sharing scheme, in: Proc. 10st Workshop on Embedded Systems Security, 2015, pp.1-8.

[2] B. Schoenmakers, A simple publicly verifiable secret sharing scheme and its application to electronic voting, in: Proc. 19th Annu. Int. Cryptol. Conf. Adv. Cryptology, 1999, pp. 148–164.

[3] B. Rajabi, Z. Eslami, A verifiable threshold secret sharing scheme based on lattices, Information Sciences 501(2019) 655-661.

[4] P. Feldman, A practical scheme for non-interactive verifiable secret sharing, in: Proc. 28th Annual Symposium on Foundations of Computer Science, 1987, pp. 427–438.

[5] M. Cafaro, P. Pellè, Space-efficient verifiable secret sharing using polynomial interpolation, IEEE Trans. Cloud Comput. 6(2)(2018) 453-463.

[6] A. Wander, N. Gura, H. Eberle, V. Gupta, S. C. Shantz, Energy analysis of public-key cryptography for wireless sensor networks, in: PerCom 2005, 2005, pp. 324-328.

[7] C. Wu, S. Li, Y. Zhang, Key management scheme based on secret sharing for wireless sensor network, in: Proc. 4th Int. Conf. Emerging Intell. Data Web Technol., 2013, pp. 574–578.

[8] A. Shamir, How to share a secret, Commun. ACM 22(11)(1979) 612–613.

[9] G. R. Blakley, Safeguarding cryptographic keys. in: Proc. Nat. Comput. Conf., 48, 1979, pp.313-317.

[10] H. Pilaram, T. Eghlidos: An efficient lattice based multistage secret sharing scheme. IEEE Trans. Dependable Secur. Comput. 14(1)(2017) 2-8.

[11] L. Harn, C. Lin, Strong (n, t, n) verifiable secret sharing scheme, Information Sciences 180(2010) 3059–3064.

[12] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract), in: Proc. 26th Annual Symposium on Foundations of Computer Science, 1985, pp. 383–395.

[13] J.C. Benaloh, Secret sharing homomorphisms: keeping shares of a secret secret, in: Proc. on Advances in cryptology–CRYPTO '86, 1987, pp. 251–260 .

[14] G. Tsaloli, G. Banegas, A. Mitrokotsa, Practical and provably secure distributed aggregation: verifiable additive homomorphic secret sharing. Cryptogr. 4(3): 25 (2020)

[15] R. Koikara, E.-J. Yoon, A. Paul, Publicly verifiable threshold secret sharing based on three-dimensional-cellular automata. Concurr. Comput. Pract. Exp. 33(22) (2021)

[16] S. Zhang, Q. Wen, W. Li, H. Zhang, Z.Jin, A multi-user public key encryption with multi-keyword search out of bilinear pairings. Sensors 20(23): 6962 (2020)

[17] K. Nyberg, Fast accumulated hashing, in: Proc. 3rd Int. Workshop Fast Softw. Encryption, 1996, pp. 83–87.

[18] J.-J. Huang, W.-S. Juang, C.-I Fan, Y.-F. Tseng, H. Kikuchi, Lightweight authentication scheme with dynamic group members in IoT environments. in: Adjunct Proc. MobiQuitous 2016, 2016, pp. 88-93.

[19] C.-I Fan, J.-J. Huang, M.-Z. Zhong, R.-H. Hsu, W.-T. Chen, J. Lee, ReHand: Secure region-based fast handover with user anonymity for small cell networks in mobile communications, IEEE Trans. Inf. Forensics Secur. 15(2020) 927-942.

[20] J. Benaloh, M. de Mare, One-way accumulators: A decentralized alternative to digital signatures, in: Proc. Workshop Theory Appl. Cryptograph. Techn. Adv. Cryptol., 1993, pp. 274–285.

[21] V. Lyubashevsky, D. Micciancio, Generalized compact knapsacks are collision resistant, in: Proc. 33rd ICALP 2006, 2006, pp. 144-155.

[22] L. Tawalbeh, F. Muheidat, M. Tawalbeh, M. Quwaider, IoT Privacy and Security: Challenges and Solutions, Appl. Sci. 10(2020) 4102.

[23] FIPS PUB 186-2, Digital signature standard (DSS). 2000, January 27. http://csrc.nist.gov/publications/PubsFIPS .html♯fips186-3.

[24] N. Göttert, T. Feller, M. Schneider, J. Buchmann, S. A. Huss, On the design of hardware building blocks for modern lattice-based encryption schemes, in: Proc. Int. Workshop CHES, 2012, pp. 512-529.

[25] X. Cao, W. Kou, L. Dang, B. Zhao, IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks, Comput. Commun. 31(4)(2008) 659-667.