Function secret sharing (FSS) is a cryptographic primitive that is introduced by Boyle et al. (Eurocrypt 2015), motivated by application scenarios involving private access to large distributed data while minimising the overhead of communication, for example, private information retrieval. Informally, an n-party FSS scheme splits a function f into n functions f_1, ..., f_n such that f = f_1 + ··· + f_n and every strict subset of the function shares hide f. Most of the known FSS constructions only have computational hiding, namely, the hiding property holds only against a computationally bounded adversary. We consider information-theoretic hiding in this work while allowing f to be recovered from t function shares and correspondingly, any (t − 1) function shares unconditionally hide f. Call it (t, n)-threshold function secret sharing ((t, n)-TFSS for short). Using information-theoretic tools and through a series of optimizations, we show that our (t, n)-TFSS have better performance than FSS in terms of communication complexity, a criterion that measures the efficiency of such protocols. Specifically, a (t, n)-TFSS scheme with communication complexity O(l) is designed in this paper and it is better than the existing FSS schemes with lowest communication complexity O(λl), where λ is the length of pseudo-random generator’s seeds. In addition, the (t, n)-TFSS have an extra robustness property in the sense that even if up to (n − t) function shares are not available, the protocol still computes the function value at a given point correctly.

Index Terms: Function secret sharing, point function, private information retrieval, information-theoretic security.
constructed a DPF based on the pseudo-random generator (PRG) [8], [9]. First, they gave a naive solution of DPF for a function \( f_{\alpha,\beta} : [0, 1]^l \to \{0, 1\} \) that shares 2\(^l\) bits corresponding to all function values of \( f \) through a 2-party additive secret sharing. The length of each share is 2\(^l\) in this naive solution. They reduced the length of each share by using PRG to compress the length of function shares: the true random sequence is replaced by a pseudo-random sequence generated by the PRG. They then reduced the length of each share from \( O(2\lambda) \) in the initial scheme to \( O(\lambda \log^2 \lambda) \), where \( \lambda \) is the length of PRG seeds through \( l \) compressions. The security is computational due to the application of PRG.

The study of FSS then used the communication complexity as a design criterion. The communication complexity of an FSS scheme refers to the total amount of information that needs to be transmitted to complete this FSS scheme. This criterion captures the performance of FSS with respect to its applications to PIR and related scenarios such as homomorphic secret sharing [10]–[13] and fully homomorphic encryption [14], [15]. Most known constructions continued the pursuit of minimising the communication complexity within the computational security setting. In particular, DPF with the communication complexity \( 4(\lambda+1) \) was constructed in [1]. In addition, they also proposed the first n-party (\( n \geq 3 \)) FSS for point functions with the communication complexity \( O(\lambda 2^{l+n-1}/2) \). In [7], further simplified constructions of DPF through a tensor operation was proposed which reduces the communication complexity from \( 4(\lambda+1) \) to \( l(\lambda + 2) \).

An important conceptual contribution of [1] is that it defines a general FSS as a stand-alone primitive (not only as a way to build efficient computational PIR’s for example). In the general definition, a function \( f \) is split into \( n \) function shares \( f_1, \ldots, f_n \) and the recovery of \( f(x) \) from \( f_1(x), \ldots, f_n(x) \) is not restricted to \( f(x) = \sum_{i=1}^n f_i(x) \). More generally and still allows the function secret sharing to be useful for various applications. The “splitter system” [16], constructed based on FSS schemes, supports a wide variety of private search queries. The preprocessing of secure computation using FSS is presented in [17] to improve the efficiency of secure computing.

### A. OUR CONTRIBUTIONS

We study FSS with information-theoretic security in this work to complement its computational security analogues [1], [6], [7]. As observed in [1], [6] through a connection between FSS and PIR, insisting on information-theoretic security and recovery through \( f(x) = \sum_{i=1}^n f_i(x) \) will render the FSS inefficient. In fact, this is the reason that they focused on computational security. We taking a complementary approach, we insist on information-theoretic security and do not recovery \( f(x) \) through \( f(x) = \sum_{i=1}^n f_i(x) \). We consider a recovery threshold \( t \) and a privacy threshold \( (t-1) \). We note here that reconstructing from \( t \) players gives the robustness guarantee such that when up to \((n - t)\) communication channels fail and the corresponding function shares are not available, the scheme still function correctly. The security of our scheme requires that any \((t-1)\) participants do not obtain any information about the shared function.

We formally define the \((t, n)\)-Threshold function secret sharing (or \((t, n)\)-TFSS for short). We define \((t, n)\)-TFSS as an information-theoretic object using the game-based definition of security. It includes a challenger \( C \), an adversary \( A \) and a challenge index \( b \). A scheme is information-theoretically secure if for any computationally unbounded adversary \( A \) the advantage of \( A \) guessing \( b \) correctly in the indistinguishability challenge experiment about this protocol is “0”.

Subsequently, we propose three explicit constructions of \((t, n)\)-TFSS for the point function \( f_{\alpha,\beta} : [0, 1]^l \to \mathbb{F}_q \) where \( \mathbb{F}_q \) is a finite field of \( q \) elements for some constant \( q \). We start with an initial scheme, where the \( 2^l \) function values of \( f \) corresponding to all inputs in \([0, 1]^l\) are arranged into a vector over \( \mathbb{F}_q \) of length \( 2^l \). This design approach is consistent with the naive solution of [6], however we share each component of the vector using \((t, n)\)-Shamir’s scheme [18]. The length of each share in the initial scheme is \( O(2^l) \). The design of the recovery of \( f_{\alpha,\beta}(x) \) then makes use of the rich structures of polynomial based scheme to pin-point the information \( f_i(x) \), for \( i = 1, \ldots, n, x \in [0, 1]^l \) needed for reconstructing \( f_{\alpha,\beta}(x) \) (and only communicating these information) without downloading all information of the \( \mathbb{F}_q \) vector of length \( 2^l \). The communication complexity of the initial scheme is \( O(2^{l}) \).

In our first improved scheme, the vector of length \( 2^l \) is converted to the tensor products of two vectors of length \( O(\sqrt{2^l}) \). We then use \((t, n)\)-Shamir’s scheme to share each component of these two vectors over \( \mathbb{F}_q \) and the length of each share is \( O(\sqrt{2^l}) \). In this way, we reduce the communication complexity of the initial scheme from \( O(2^l) \) to \( O(\sqrt{2^l}) \). Furthermore, the vector of length \( 2^l \) is converted to the tensor products of \( l \) vectors of length \( 2 \), in our second improved scheme. We also use \((t, n)\)-Shamir’s scheme to share each component of these \( l \) vectors over \( \mathbb{F}_q \).

Finally, we reduce the communication complexity of the initial scheme from \( O(2^l) \) to \( O(l) \) and it is optimal. Our constructions are based on the polynomial over the finite

---

**TABLE 1. Comparisons to the existing FSS schemes.**

| Schemes          | \((t, n)\)            | \(\text{CP}\) | Security                  |
|------------------|------------------------|----------------|--------------------------|
| Gilboa et al.    | \((2, 2)\)             | \(O(\lambda \log^3 \lambda)\) | computational security   |
| Boyle et al.     | \((2, 2)\)             | \(O(\lambda l)\)     | computational security   |
| Boyle et al.     | \((2, 2)\)             | \(O(\lambda l)\)     | computational security   |
| Boyle et al.     | \((n, n)\)             | \(O(\lambda 2^{n/2})\) | computational security   |
| Yuan et al.      | \((t, n)\) and \(t \geq l + 1\) | \(O(l)\)           | no security proof        |
| Our initial scheme | \((t, n)\) and \(1 < t < n\) | \(O(2^l)\)         | information-theoretic security |
| Our improved scheme I | \((t, n)\) and \(1 < t < n\) | \(O(\sqrt{2^l})\) | information-theoretic security |
| Our improved scheme II | \((t, n)\) and \(1 < t < n\) | \(O(l)\)           | information-theoretic security |
field and without any additional assumption (e.g., the exist of PRG). We compare the existing FSS schemes with our three constructions and list them in TABLE 1. We note that Yuan et al. [19] proposed a construction of FSS that can be verified to give \((t, n)\)-TFSS for \(t \geq l + 1\). Its communication complexity is \(O(l)\), same as our second improved scheme which supports any threshold \(t\). However, there was no formal proof in their paper.

### B. PAPER ORGANIZATION

The rest of the paper is organized as follows. In Section II, we give some related concepts and definitions. In Section III, we define the information-theoretic function secret sharing and give the security definition under the game-based security model. In Section IV, we design an initial scheme based on Shamir’s scheme. In Section V, we use tensor products to reduce the communication complexity of the initial scheme and design two improved schemes. Finally, we conclude the paper in Section VI.

### II. PRELIMINARIES

In this section, we introduce the definitions of point function, tensor products and Shamir scheme. We list some of the symbols used in this paper in TABLE 2.

**TABLE 2. The descriptions of symbols.**

| Symbols | Descriptions |
|---------|--------------|
| \(\mathbb{F}_q\) | The finite field of order \(q\) |
| || | The string concatenation |
| \(\otimes\) | The tensor products |
| \([N]\) | The set \(\{1, \ldots, N\}\) |
| \(k[i]\) | The \(i\)-th entry of vector \(k\) |
| \(e_i\) | The unit vector where the \(i\)-th entry is “1” |
| \(|k|\) | The length of the vector \(k\) |
| \(k_i\) | The \(i\)-th component of vector \(k\) |
| \(\{0, 1\}^*\) | The bit string with arbitrary length |
| \(\{0, 1\}^l\) | The bit string with length \(l\) |
| \(D_f\) | The domain of the function \(f\) |
| \(R_f\) | The range of the function \(f\) |

**Definition 1 (Point Functions):** Let \(\{0, 1\}^l\) be the domain and \(\mathbb{F}_q\) is the range, for any \(\alpha \in \{0, 1\}^l\), \(\beta \in \mathbb{F}_q\), the point function \(f_{\alpha, \beta} : \{0, 1\}^l \rightarrow \mathbb{F}_q\) is defined as follows: for any \(x \in \{0, 1\}^l\)

\[
f_{\alpha, \beta}(x) = \begin{cases} 
\beta, & x = \alpha; \\
0, & x \neq \alpha. 
\end{cases}
\]  

**Definition 2 (Tensor Products):** Let \(\widetilde{V}^1 = (v_1^1, \ldots, v_l^1)\), \(\widetilde{V}^2 = (v_1^2, \ldots, v_l^2)\) and \(|\widetilde{V}| = U, |\widetilde{V}^2| = V\). Let \(V\) be the tensor products of \(\widetilde{V}^1, \widetilde{V}^2\) and \(V = \widetilde{V}^1 \otimes \widetilde{V}^2 = (v_1^{12}, \ldots, v_{1l}^{12}, v_1^{21}, \ldots, v_{1l}^{21}, v_1^{22}, \ldots, v_{1l}^{22}) = (v_1, \ldots, v_N)\) and \(|V| = N = UV\). Any \(j\) can be uniquely denoted by \(j = (u', v')\) where \(j \in [N] u' \in [U], v' \in [V]\).

**Definition 3 ((t,n)-Shamir’s Scheme):** Let \(\mathbb{F}_q\) be the finite field with \(q\) elements, where \(q\) is a large prime and \(q > n\), that contains the domain of possible secrets. Let \(\lambda_1, \ldots, \lambda_n\) be distinct nonzero elements in \(\mathbb{F}_q\) that are fixed and publicly known. There is a trusted dealer and \(n\) participants \(\{P_i\}_{i \in [n]}\). The following scheme is a so-called \((t, n)\)-threshold secret sharing scheme, which means any \(t\) shares reconstruct the secret while any \(\leq t - 1\) shares contain no information about the secret.

- **Share generation:** Given a secret \(s \in \mathbb{F}_q\), the dealer chooses uniformly \(a_1, \ldots, a_{t-1} \in \mathbb{F}_q\) and defines the polynomial \(P(X) = s + \sum_{i=1}^{t-1} a_iX^i\). This is a random polynomial of degree at most \(t - 1\) with constant term fixed to \(s\). The share of the participant \(P_i\) for \(i = 1, \ldots, n\), is

\[
s_i := P(\lambda_i) \in \mathbb{F}_q.
\]

Each share \(s_i\) is distributed to the corresponding participant \(P_i\) independently and privately through a secure channel.

- **Secret reconstruction:** Any \(t\) participants \(P_{i_1}, \ldots, P_{i_t}\), from the participants set \(\{P_i\}_{i \in [n]}\), work together to recover the secret \(s\) by the polynomial interpolation as follows.

\[
s = \sum_{r=1}^{t} b_i \cdot s_{i_r},
\]

where \(b_i = \prod_{r'=1, r' \neq r}^{t} \frac{\lambda_{i_r}}{\lambda_{i_r} - \lambda_{i}}\), for \(r = 1, \ldots, t\), are the interpolation coefficients.

### III. INFORMATION-THEORETIC \((t, n)\)-TFSS

In this section, we formally define the information-theoretic \((t, n)\)-TFSS we consider in this work and point out the difference between our definition and the definition of the computational FSS considered in [1], [6], [7]. Then we give the definition of the communication complexity of the TFSS scheme.

**Definition 4 \((t, n)\)-Threshold Function Secret Sharing \((t, n)\)-TFSS:** Let \(\mathbb{F}_q(q > n)\) be a finite field. Let \(n\) be the number of participants. Let \(t\) \((t < n)\) be the reconstruction threshold and the privacy threshold is \(t - 1\). Let \(f : D_f \rightarrow R_f\) be a function that is efficiently computable and can be succinctly described. A \((t, n)\)-function secret sharing is a triple of probabilistic polynomial time algorithms \((\text{Gen}, \text{Eval}, \text{Dec})\) with the following syntax.

- **Gen(\(1^k, f)\):** This is the share generation algorithm run by the dealer. It takes the security parameter \(1^k\) and function description \(f\) as inputs, and outputs \(n\) shares \(k_1, \ldots, k_n\).
- **Eval(\(k_i, x)\):** This is the evaluation algorithm run by each participants \(P_i\), for \(i = 1, \ldots, n\). It takes the share \(k_i\), an evaluation point \(x \in D_f\) as inputs and outputs a value \(y_i\), corresponding to the participant’s share of \(f(x)\).
- **Dec((\(f_1, y_1), \ldots, (f_t, y_t))\):** This is the output decoder. When it receives any \(t\) values \(y_1, \ldots, y_t\), it outputs a value \(f(x)\).
The triple (Gen, Eval, Dec) should satisfy the following requirements.

- Correctness. For each function \( f : D_f \rightarrow R_f \), any \( x \in D_f \) and any subset \( \{i_1, \ldots, i_t\} \subseteq [n] \), we have

\[
\Pr[(k_1, \ldots, k_n) \leftarrow \text{Gen}(1^k); \text{Dec}(i_1, \text{Eval}(i_1, k_{i_1}, x)), \ldots, (i_t, \text{Eval}(i_t, k_{i_t}, x))) = f(x)] = 1.
\] (4)

It means that the function value \( f(x) \) can be computed correctly if the above three algorithms run normally.

- \((t-1)\)-Security. We consider the following indistinguishability experiment \( \Pi \) for a set \( T \subseteq \{P_1, \ldots, P_n\} \) of corrupted participants, where \(|T| = t-1\):

- Any computationally unbounded adversary \( A \) inputs security parameter \( 1^k \) outputting \((f_0, f_1) \leftarrow \text{Gen}(1^k)\) with \( D_{f_0} = D_{f_1}\).
- The challenger \( C \) chooses a challenge index \( b \in \{0, 1\} \) randomly to generate \((k_1, \ldots, k_n) \leftarrow \text{Gen}(1^k, f_b)\).
- The adversary \( A \) exploits \( (k_i)_{i \in T} \), offered by corrupted participants in \( T \), outputting a guess \( b' \leftarrow \mathcal{A}(k_i)_{i \in T} \).

Let \( \text{Adv}(1^k, A, T) := \Pr[b = b'] - \frac{1}{2} \) be the advantage of \( A \) guessing \( b \) correctly in the above indistinguishability experiment. We say a \((t, n)\)-TFSS scheme is information-theoretically \( t-1 \)-secure if for any corrupted participants set \( T \) of cardinality \( t-1 \) and any computationally unbounded adversary \( A \), \( \text{Adv}(1^k, A, T) = 0 \).

 Remark 1 (Information-Theoretic FSS): This indistinguishability challenge experiment is consistent with [1], [20], where only focus on the security of information-theoretic, that is, without any computational assumptions. The previous FSS schemes in [1], [6], [7] is secure under the assumption of the existence of PRG and they are computationally secure. In definition 3 the security requires it can resist any computationally unbounded adversaries even for non-uniform probability polynomial time adversaries and we restrict the value \( \text{Adv}(1^k, A, T) = 0 \).

Remark 2 (Robust Decoding): Obviously, definition 3 allows the algorithm \( \text{Dec} \) successful decoding the function value at an evaluation point even if at most \( n-t \) inputs cannot given in time. This is more flexible than the previous FSS in which \( \text{Dec} \) requires all \( n \) inputs are given to decode the function value.

Definition 5 (Communication Complexity): The communication complexity of a \((t, n)\)-TFSS scheme will be denoted as \( \Theta = \Phi + \Psi \), where \( \Phi = \sum_{i=1}^{n} |k_i| \) and \( \Psi = \sum_{b=0}^{t} |y_{b'}| \).

IV. AN INITIAL SCHEME TFSS\(_1\) (GEN, EVAL, DEC)

From this section, we start construct a concrete \((t, n)\)-TFSS schemes for point functions based on \((t, n)\)-Shamir’s scheme over the finite field \( \mathbb{F}_q \). We start with an initial scheme TFSS\(_1\)(Gen, Eval, Dec). In a nutshell, we arranged the truth table (containing \( N = 2^t \) field elements of \( \mathbb{F}_q \)) of a point function \( f_{\alpha, \beta} : \{0, 1\}^t \rightarrow \mathbb{F}_q \) in to a vector of length \( N \). This approach is consistent with the naive solution designed in [6], however we use \( N (t, n)\)-Shamir’s schemes to share the vector. The function share for the participant \( P_i \) is obtained by packing the \( i \)-th shares of the \((t, n)\)-Shamir’s schemes. After determining a particular evaluation point \( \alpha' \in \{0, 1\}^t \), each participant outputs the component of his/her function share that is corresponding to \( \alpha' \), which is a share of the \((t, n)\)-Shamir’s scheme that shares the secret \( f_{\alpha, \beta}(\alpha') \). The correctness and \( t-1 \)-security then follow naturally from the \((t, n)\)-Shamir’s scheme.

More concretely, in the algorithm of \( \text{Gen} \), the dealer \( D \) chooses \( N = 2^t \) random polynomials over \( \mathbb{F}_q \) with degree at most \( t-1 \), labeled by elements in \( \{0, 1\}^t \). Let the constant term of the \( \alpha \)-th polynomial be \( f_{\alpha, \beta}(\alpha') \), for all \( \alpha' \in \{0, 1\}^l \). The dealer \( D \) generates \( n \) function shares \( f_{\alpha, \beta} \) through the \( N \) polynomials evaluating at \( n \) publicly known distinct non-zero elements \( \lambda_1, \ldots, \lambda_n \in \mathbb{F}_q \). In the algorithm of \( \text{Eval} \), the participant \( P_i \), upon receiving \( \alpha' \in \{0, 1\}^t \), outputs the evaluation of the \( \alpha' \)-th polynomial at the point \( \lambda_i \in \mathbb{F}_q \). After the algorithm \( \text{Dec} \) receives any \( t \) evaluations of the \( \alpha' \)-th polynomial at distinct points, it recovers the constant of the \( \alpha' \)-th polynomial using polynomial interpolation, which is \( f_{\alpha, \beta}(\alpha') \) by construction. The initial scheme TFSS\(_1\)(Gen, Eval, Dec) is detailed in FIGURE 1.

We now show that the initial scheme TFSS\(_1\)(Gen, Eval, Dec) satisfies the following requirements: correctness, \((t-1)\)-security as defined in Section III.

THEOREM 1: Let \( \mathbb{F}_q \) be a finite field and \( 1^k \) be the security parameter. The initial scheme TFSS\(_1\)(Gen, Eval, Dec) is a \((t, n)\)-TFSS for point functions \( f_{\alpha, \beta} : \{0, 1\}^t \rightarrow \mathbb{F}_q \). The communication complexity of TFSS\(_1\) is \( O(2^t) \).

Proof 1: Correctness. It suffices to show that for any point function \( f_{\alpha, \beta} : \{0, 1\}^t \rightarrow \mathbb{F}_q \), any \( \alpha' \in \{0, 1\}^t \) and any subset \( \{i_1, \ldots, i_t\} \subseteq [n] \),

\[
\text{Pr}(k_1, \ldots, k_n) \leftarrow \text{Gen}(1^k; \alpha, \beta); \text{Dec}(i_1, \text{Eval}(i_1, k_{i_1}, \alpha')), \ldots, (i_t, \text{Eval}(i_t, k_{i_t}, \alpha')) \rightleftharpoons f_{\alpha, \beta}(\alpha') = 1.
\]

FIGURE 1. The scheme TFSS\(_1\). The function being shared is \( f_{\alpha, \beta} : \{0, 1\}^t \rightarrow \mathbb{F}_q \). The number of participants is \( n \). The reconstruction threshold is \( t \) and the privacy threshold is \( t-1 \). The \( l_1, \ldots, l_n \) are \( n \) non-zero distinct public values chosen from \( \mathbb{F}_q \).
we only need to verify the following equation.

\[
\text{Dec}(i_1, y_{i_1}, \ldots, (i_t, y_{i_t})) = \sum_{r=1}^{t} b_{i_r} \cdot y_{i_r} = \sum_{r=1}^{t} b_{i_r} \cdot k_{i_r}[\alpha'] = \sum_{r=1}^{t} b_{i_r} \cdot s_{a',i_r} = f_{a'}(0) = f_{a,\beta}(\alpha').
\]

This equation holds for every possible value of \(i_1, \ldots, i_t\) and \(y_{i_1}, \ldots, y_{i_t}\). Therefore, the decoder is correct.

\(t-1\)-Security. We consider the following indistinguishability challenge experiment \(\Pi\) for corrupted participants set \(T \subseteq \{P_1, \ldots, P_n\}\) of size \(|T| \leq t - 1\). Without loss of generality, we assume that \(T = \{P_1, \ldots, P_{t-1}\}\).

- The adversary \(A\) takes the security parameter \(1^k\) as the input and outputs two point functions \((f_{a_0,b_0}, f_{a_1,b_1}) \leftarrow A(1^k)\), where \(a_0, a_1 \in \{0,1\}^t\), \(b_0, b_1 \in \mathbb{F}_q\).
- The challenger \(C\) chooses the challenge index \(b\) from \(\{0,1\}\) randomly and generates \((k_1, \ldots, k_n) \leftarrow \text{Gen}(1^k, f_{a,b})\) as follows:
  - Choose \(N\) random polynomials \(f_j(x) = a_{j,0} + a_{j,1} x + \cdots + a_{j,t-1} x^{t-1}\) over \(\mathbb{F}_q\), where \(a_{j,0} = b_0, a_{j,0} = 0, j \neq b\), for \(j = 1, \ldots, N\).
  - Compute \(k_i = (s_{i,1}, s_{i,2}, \ldots, s_{i,N})\) and output \((k_1, \ldots, k_n)\), where \(s_{i,j} = f_j(k_i)\), for \(i = 1, \ldots, n\).
- The adversary \(A\) exploits \((k_i)_{i \in T}\), offered by corrupted participants in the set \(T\), outputting a guess \(b' \leftarrow A((k_i)_{i \in T})\).

The \((t-1)\) shares sent to \(A\) satisfy the following equation system:

\[
\begin{pmatrix}
  k_1 \\
  \vdots \\
  k_{t-1}
\end{pmatrix}
= \begin{pmatrix}
  a_{1,0} & \cdots & a_{N,0} \\
  \vdots & \ddots & \vdots \\
  a_{1,0} & \cdots & a_{N,0}
\end{pmatrix}
\begin{pmatrix}
  \lambda_1 & \cdots & \lambda_{t-1}^{l-1} \\
  \vdots & \ddots & \vdots \\
  \lambda_{t-1} & \cdots & \lambda_{t-1}^{l-1}
\end{pmatrix}
A
\begin{pmatrix}
  \lambda_1 & \cdots & \lambda_{t-1}^{l-1} \\
  \vdots & \ddots & \vdots \\
  \lambda_{t-1} & \cdots & \lambda_{t-1}^{l-1}
\end{pmatrix}
\times
\begin{pmatrix}
  a_{1,1} & \cdots & a_{N,1} \\
  \vdots & \ddots & \vdots \\
  a_{1,t-1} & \cdots & a_{N,t-1}
\end{pmatrix}
\end{pmatrix}
\]

We note that \(A\) is a random matrix of size \((t-1) \times N\) and \(\Lambda\) is a nonsingular matrix of order \((t-1)\). As a result, \(K = AA_0 \in \mathbb{F}_q^N\) is a random vector. It contains no information about \(A_0\) and hence no information about \(b\). Therefore, \(Pr[b' = b] = 1/2\).

A. COMMUNICATION COMPLEXITY

In TFSS, \(k_i = (s_{i,1}, s_{i,2}, \ldots, s_{i,N})\), for \(i = 1, \ldots, n\), \(b_i = k_i[\alpha'] = s_{a',i_i} \) for \(\alpha' \in \{0,1\}^t\). For \(j = 1, \ldots, N, i = 1, \ldots, n\) has \(s_{j,i} = f_j(k_i) \in \mathbb{F}_q\) thus \(|s_{j,i}| = \log q\). Therefore, \(\Phi = n|k_i| \log q = n2^t \log q\), \(\Psi = t \log q\), hence \(\Theta = \Phi + \Psi = n2^t \log q + t \log q = O(2^t)\). Therefore, the communication complexity of TFSS is \(O(2^t)\).

V. IMPROVED SCHEMES

In this section, we first reduced the communication complexity of TFSS from \(O(2^t)\) to \(O(\sqrt{2^t})\) in the improved scheme I TFSS1 (Gen, Eval, Dec) from \(O(2^t)\) to \(O(\sqrt{2^t})\) in the improved scheme I TFSS1 (Gen, Eval, Dec). In a nutshell, we split the truth table of the point function \(f_{a,b} : \{0,1\}^t \rightarrow \mathbb{F}_q\), which can be seen as a vector of length \(2^t\), into a tensor products of two vectors \(\bar{U}, \bar{V}\) and let \(|\bar{U}| = U, |\bar{V}| = V\). Therefore, \(N = U \times \bar{V}\) and \(U = V = O(\sqrt{2^t})\). We share \(\bar{U}, \bar{V}\), using \(U + V\) (i.e., Shamir’s schemes). The function share for the participant \(P_i\) is obtained by packing the \(i\)-th shares of the \(U + V\) (i.e., Shamir’s schemes). After determining a particular evaluation point \(\alpha' = (\alpha'_1, \alpha'_2, \ldots, \alpha'_t)\), where \(\alpha' \in \{0,1\}^t\), each participant outputs two components of his/her function share that is corresponding to \(\alpha'_1, \alpha'_2\) and using them to calculate the secret \(f_{a,\beta}(\alpha')\). The correctness and
The constant terms of $\alpha$ polynomials are 0. In the second group random polynomials is $V$, the number of the second group random polynomials is $U$. The number of first group random polynomials is $t$, two group random polynomials with degree $\lfloor tf/2\rfloor$ is $T$.

In this section, we construct the improved scheme $I$ TFSS$_2$ (Gen, Eval, Dec) for the point function $f_{a,\beta} : [0,1]^l \rightarrow \mathbb{F}_q$. In Gen, let $\alpha = (\alpha_1, \alpha_2)$, for $\alpha_1 \in [U]$, $\alpha_2 \in [V]$ and $\beta = \beta_1 \cdot \beta_2$, for $\beta_1, \beta_2 \in \mathbb{F}_q$. The dealer $D$ chooses two group random polynomials with degree $(t-1)$ over $\mathbb{F}_q$. The number of first group random polynomials is $U$ and the number of the second group random polynomials is $V$.

In the first group random polynomials, the constant terms of $\alpha_1$-th polynomial is $\beta_1$ and the constant term of the rest polynomials are 0. In the second group random polynomials, the constant terms of $\alpha_2$-th polynomial is $\beta_2$ and the constant term of the rest polynomials are 0. Then $D$ shares the vectors $\bar{U}, \bar{V}$ using these two group random polynomials and generates $n$ shares of $f_{a,\beta}$ through the two group random polynomials evaluating at $n$ publicly known distinct no-zero element $\lambda_1, \ldots, \lambda_n \in \mathbb{F}_q$. In the algorithm $Evaluat$, the participant $P_i$, upon determining $\alpha' = (\alpha'_1, \alpha'_2)$, for $\alpha'_1 \in [N]$ output the evaluation of $\alpha'_1$-th in the first group and $\alpha'_2$-th random polynomial in the second group at point $\lambda_i$. After the algorithm $Dec$ receives any $t$ evaluation of the $\alpha'_1$-th and $\alpha'_2$-th random polynomial at distinct point. It recovers the constant terms of $\alpha'_1$-th and $\alpha'_2$-th random polynomial respectively utilizing polynomial interpolation, and then computing their product. The improved scheme $I$ TFSS$_2$ (Gen, Eval, Dec) is detailed in FIGURE 3.

THEOREM 2: Let $\mathbb{F}_q$ be a finite field and $l^t$ be the security parameter. The initial scheme TFSS$_2$ (Gen, Eval, Dec) is a $(t,n)$-TFSS for point functions $f_{a,\beta} : [0,1]^l \rightarrow \mathbb{F}_q$. The communication complexity of TFSS$_1$ is $O(\sqrt{2^n})$.

The idea of the proof of Theorem 2 is very similar to the proof of Theorem 1. We present them in Appendix 1.

A. IMPROVED SCHEME I: TFSS$_2$ (GEN, EVAL, DEC)

In this section, we construct the improved scheme $I$ TFSS$_2$ (Gen, Eval, Dec) for the point function $f_{a,\beta} : [0,1]^l \rightarrow \mathbb{F}_q$. In Gen, let $\alpha = (\alpha_1, \alpha_2)$, for $\alpha_1 \in [U]$, $\alpha_2 \in [V]$ and $\beta = \beta_1 \cdot \beta_2$, for $\beta_1, \beta_2 \in \mathbb{F}_q$. The dealer $D$ chooses two group random polynomials with degree $(t-1)$ over $\mathbb{F}_q$. The number of first group random polynomials is $U$ and the number of the second group random polynomials is $V$.

The simple example of scheme TFSS$_2$, the function being shared is $f_{a,\beta} : [0,1]^l \rightarrow \mathbb{F}_q$. The number of participants is $n$. The privacy threshold is $t-1$ and the reconstruction threshold is $t$. The $\lambda_1, \ldots, \lambda_n$ are $n$ non-zero distinct public values chosen from $\mathbb{F}_q$.
- \( \text{Gen}(1^\lambda, f_{\alpha, \beta}) \rightarrow (k_1, \ldots, k_n) \).
- Let \( \alpha \rightarrow (\alpha_1, \ldots, \alpha_l) \) and \( \beta = \beta_1 \cdot \beta_2 \cdot \ldots \cdot \beta_l \), where \( \alpha_w \in \{0, 1\}, \beta_w \in \mathbb{F}_q \), for \( w = 1, \ldots, l \).
- Choose \( l \) group random polynomials
  \[
  \begin{align*}
  f_{\beta}^w &= a_{\beta, w}^0 + a_{\beta, w}^1 x + \cdots + a_{\beta, w}^{l-1} x^{l-1}, \\
  f_{\alpha}^w &= a_{\alpha, w}^0 + a_{\alpha, w}^1 x + \cdots + a_{\alpha, w}^{l-1} x^{l-1},
  \end{align*}
  \]
  where \( a_{\alpha, w}, a_{\beta, w} = 0 \) if \( \alpha_w, \beta_w = 0 \).
- Compute \( k_i = (s_{0, i}^w, s_{1, i}^w, s_{2, i}^w, \ldots, s_{l-1, i}^w, \alpha_{l, i}) \), where \( s_{0, i}^w = f_{\beta}^w(\lambda_i), s_{1, i}^w = f_{\alpha}^w(\lambda_i), i = 1, \ldots, n \).
- Output \((k_1, \ldots, k_n)\).
- \( \text{Eval}(i, k_i, \alpha') \rightarrow y_i \).
  - For any \( \alpha' \in \{0, 1\}^l \), let \( \alpha' = (\alpha'_1, \ldots, \alpha'_l) \), where \( \alpha'_w \in \{0, 1\} \).
  - Output \( y_i = (k_i[1 + \alpha'_1], \ldots, k_i[2(l - 1) + 1 + \alpha'_l]) = (y_{i1}, \ldots, y_l) \).
- \( \text{Dec}((i_1, y_{i_1}), \ldots, (i_t, y_{i_t})) \rightarrow S \).
  - Output \( S = \left(\sum_{r=1}^{i_t} b_{r_1} \cdot y_{i_1}^1, \ldots, \sum_{r=1}^{i_t} b_{r_l} \cdot y_{i_1}^l\right) \), where \( b_r = \prod_{r=1}^{t} l_{i_r} = \lambda_{i_r}^2 - \lambda_{i_r} \).

![FIGURE 5. The scheme TFSS3. The function being shared is \( f_{\alpha, \beta} : \{0, 1\}^l \rightarrow \mathbb{F}_q \). The number of participants is \( n \). The privacy threshold is \( t - 1 \) and the reconstruction threshold is \( t \). The \( \lambda_1, \ldots, \lambda_n \) are \( n \) non-zero distinct public values chosen from \( \mathbb{F}_q \).](image)

2) EXAMPLE
We give a simple example of the scheme TFSS2 as a validation in FIGURE 4.

**B. IMPROVED SCHEME II: TFSS3(GEN, EVAL, DEC)**

In this section we construct the improved scheme II TFSS3 (Gen, Eval, Dec) for the point function \( f_{\alpha, \beta} : \{0, 1\}^l \rightarrow \mathbb{F}_q \).

In Gen, let \( \alpha = (\alpha_1, \ldots, \alpha_l) \) with \( \alpha_w \in \{0, 1\} \), for \( w = 1, \ldots, l \), and \( \beta = \beta_1 \cdot \beta_2 \cdot \ldots \cdot \beta_l \), for \( \beta_1, \ldots, \beta_l \in \mathbb{F}_q \). The dealer \( D \) chooses \( l \) group random polynomials with degree \( t - 1 \) and the number of each set of polynomials is 2.

In the \( w \)-th set of polynomials, the constant term of \( (1 + \alpha_w) \)-th polynomial is \( \beta_w \), and \( 0 \) is the constant term of another polynomial. Similarly, \( D \) uses these \( l \) group random polynomials to share the vectors \( \mathbf{V}_1, \ldots, \mathbf{V}_l \), and then \( D \) generates \( n \) shares of \( f_{\alpha, \beta} \) through evaluating at \( n \) publicly known distinct non-zero element \( \lambda_1, \ldots, \lambda_n \) of the \( l \) group random polynomials. In the algorithm Eval, after determining \( \alpha' = (\alpha'_1, \ldots, \alpha'_l) \), for \( \alpha' \in \{0, 1\}^l \), \( \alpha'_w \in \{0, 1\} \), \( w = 1, \ldots, l \) the participant \( P_i \) outputs the evaluation of \((1 + \alpha'_w)\)-th polynomial in \( w \)-th group at point \( \lambda_i \). After the algorithm \( \text{Dec} \) receives any \( t \) evaluation of the \((1 + \alpha'_w)\)-th random polynomial at distinct point. It recovers the constant terms of \((1 + \alpha'_w)\)-th polynomial respectively by applying the polynomial interpolation, and then computing their product. The improved scheme II TFSS3 (Gen, Eval, Dec) is detailed in FIGURE 5.

THEOREM 3: Let \( \mathbb{F}_q \) be a finite field and \( k^l \) be the security parameter. The improved scheme TFSS3 (Gen, Eval, Dec) is a \((t, n)\)-TFSS for the point function \( f_{\alpha, \beta} : \{0, 1\}^l \rightarrow \mathbb{F}_q \). The communication complexity of TFSS3 is \( O(l^t) \).

The idea of the proof of Theorem 3 is also similar to the proof of Theorem 1. We present them in Appendix 2.

1) COMMUNICATION COMPLEXITY
In TFSS3, \( k_i = (s_{0, i}^1, s_{1, i}^1, s_{2, i}^1, \ldots, s_{l-1, i}^1, \alpha_{l, i}) \), \( y_i = (y_{i1}^1, \ldots, y_{il}^1) = (y_{i1}^1, \ldots, y_{il}^1) \), where \( s_{0, i}^w, s_{l, i}^w \in \mathbb{F}_q \), \( i = 1, \ldots, n \).

![FIGURE 6. The simple example of scheme TFSS3. The function being shared is \( f_{\alpha, \beta} : \{0, 1\}^3 \rightarrow \mathbb{F}_q \). The number of participants is \( 3 \). The privacy threshold is \( t = 1 \). The reconstruction threshold is \( 2 \) and \( \lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 3 \).](image)

2) EXAMPLE
We give a simple example of the scheme TFSS3 as a validation in FIGURE 6.
VI. CONCLUSION

In this work we consider a FSS scheme with threshold allowing the secret function \( f \) to be recovered from \( t \) function shares and correspondingly, any \((t - 1) \) function shares unconditionally hide \( f \). Call it \((t, n)\)-threshold function secret sharing \((t, n)\)-TFSS. After given the definition of \((t, n)\)-TFSS, we then design three schemes and proved they are \((t, n)\)-TFSS schemes with information-theoretic security according to the definition of \((t, n)\)-TFSS. We construct an initial scheme based on \((t, n)\)-Shamir’s scheme with communication complexity \(O(2^t)\). Subsequently, we use the tensor operation to improve the communication complexity of the initial scheme and construct two improved schemes. After analysis, we note the communication complexity of our second improved scheme TFSS\(_3\) is \(O(l)\). It is slight better than the optimal two-party computational FSS designed in \([7]\) with communication complexity \(O(\lambda l)\), where \(\lambda\) is the length of the pseudo-random generator’s seeds. Furthermore, due to the nature of the threshold, our schemes have additional robustness property, that is, the function value at the given point can still be reconstructed when \((n - t)\) participants are offline. Based on the work of this article, we will consider the following work in the future. 1) Attempt to construct a verifiable TFSS schemes based on the TFSS schemes constructed in this paper. 2) Attempt to transplant the method of constructing TFSS in this paper to homomorphic secret sharing to construct a threshold homomorphic secret sharing scheme with information-theoretic security.

APPENDICES

APPENDIX 1

PROOF OF THEOREM

THEOREM 2. Let \(\mathbb{F}_q\) be a finite field and \(1^k\) be the security parameter. The improved scheme TFSS\(_2\) (Gen, Eval, Dec) is a \((t, n)\)-TFSS for the point functions \(f_{a, \beta} : [0, 1]^t \rightarrow \mathbb{F}_q\). The communication complexity of TFSS\(_2\) is \(O(\sqrt{2^t})\).

Proof 2: Correctness. It suffices to show that for any point functions \(f_{a, \beta} : [0, 1]^t \rightarrow \mathbb{F}_q\), any \(\alpha' \in [0, 1]^t\) and any subset \((i_1, \ldots, i_t) \subseteq [n]\),

\[
\Pr[(k_1, \ldots, k_n)] \leftarrow \text{Gen}(1^k, \alpha, \beta) : \text{Dec}((i_1, \text{Eval}(i_1, k_{i_1}, \alpha'))), \ldots, (i_t, \text{Eval}(i_t, k_{i_t}, \alpha'))) = f_{a, \beta}(\alpha') = 1.
\]

we only need to verify the following equation.

\[
\sum_{r=1}^t y_{i_r} \cdot \sum_{r=1}^t y_{i_r} = \sum_{r=1}^t b_{r} \cdot k_{i_r} [\alpha'_{r}] \cdot \sum_{r=1}^t b_{r} \cdot k_{i_r} [U + \alpha'_{r}] = (b_1 s_{1, i_1} + \cdots + b_n s_{n, i_t}) \cdot (b_1 s_{1, i_1} + \cdots + b_n s_{n, i_t}) = f_{\alpha'}(0) \cdot f_{\alpha'}(0) = f_{a, \beta}(\alpha').
\]

(7) (t - 1)-Security: We consider the following indistinguishability challenge experiment \(\Pi\) for corrupted participants set \(T \subseteq \{P_1, \ldots, P_n\}\) of size \(|T| \leq (t - 1)\). Without loss of generality, we assume that \(T = \{P_1, \ldots, P_{t-1}\}\).

- The adversary \(A\) takes the security parameter \(1^k\) as the input and outputs two point functions \((f_{a, \beta}, f_{a, \beta}) \leftarrow A(1^k)\), where \(a, \alpha' \in [0, 1]^t\), \(\beta, \beta' \in \mathbb{F}_q\).
- The challenger \(C\) chooses the challenge index \(b \in \{0, 1\}\) and generates \((k_1, \ldots, k_n)\) for corrupted participants as follows:

\[
\begin{align*}
&\Pr[(k_1, \ldots, k_n) \leftarrow \text{Gen}(1^k, f_{a, \beta}, \beta) : \text{Dec}((i_1, \text{Eval}(i_1, k_{i_1}, \alpha'))), \ldots, (i_t, \text{Eval}(i_t, k_{i_t}, \alpha'))) = f_{a, \beta}(\alpha') = 1.\end{align*}
\]

we only need to verify the following equation.

\[
\sum_{r=1}^t y_{i_r} \cdot \sum_{r=1}^t y_{i_r} = \sum_{r=1}^t b_{r} \cdot k_{i_r} [U + \alpha'_{r}] = f_{a, \beta}(\alpha').
\]
We consider the following indistinguishability challenge experiment Π for corrupted participants set T ⊆ \{P_1, \ldots, P_n\} of size |T| ≤ t − 1. Without loss of generality, we assume that T = \{P_1, \ldots, P_{t−1}\}.

- The adversary A takes the security parameter 1^λ as the input and outputs two functions (f_{\alpha^0, \beta^0}, f_{\alpha^1, \beta^1}) \leftarrow \mathcal{A}(1^\lambda), where \alpha^0, \alpha^1 \in \{0, 1\}^l, \beta^0, \beta^1 \in \mathbb{F}_q.

- The challenger C chooses the challenge index b from \{0, 1\} randomly and generates (k_1, \ldots, k_n) \leftarrow \text{Gen}(1^\lambda, f_{\alpha^0, \beta^0}) as follows:

  - Let \alpha^b \rightarrow (\alpha^b_0, \ldots, \alpha^b_n) and \beta^b = \beta^b_1 \cdot \beta^b_2 \cdots \beta^b_l;

  - Choose l sets random polynomials

    \begin{align*}
    f^w_0 &= a^w_{0,0} + a^w_{1,0}x + \cdots + a^w_{0,l−1}x^{l−1}, \\
    f^w_1 &= a^w_{0,0} + a^w_{1,0}x + \cdots + a^w_{1,l−1}x^{l−1},
    \end{align*}

    where \(a^w_{0,0} = b^b_0, a^w_{0,0}=1.0 = 0, w = 1, \ldots, l.\)

  - Compute \(k_i = (s^1_{0,i}, s^1_{1,i}, s^2_{0,i}, s^2_{1,i}, \ldots, s^l_{0,i}, s^l_{1,i}) = (k^1_i, k^2_i, \ldots, k^l_i)\) with \(s^w_{0,i} = f^w_0(\lambda_i), s^w_{1,i} = f^w_1(\lambda_i), for i = 1, \ldots, n.\)

  - Output \((k_1, \ldots, k_n).\)

- The adversary A exploits \((k_i)_{i \in T}\), offered by corrupted participants in the set T, outputting a guess \(b^* \leftarrow \mathcal{A}(k_i)_{i \in T}\).

The \((t−1)\) shares sent to A satisfy the following equation (12), as shown at the top of this page:

We note that A_1, \ldots, A_l are random matrix of size \((t−1) \times 2\) and \(\Lambda\) is a nonsingular matrix of order \((t−1)\). As a result, \(K_1, \ldots, K_l \in \mathbb{F}_q^{t−1}\) is a random vector. It contains no information about \(A^0_1, \ldots, A^0_l\) and hence no information about b. Therefore, \(\text{Pr}[b^* = b] = 1/2.\)

REFERENCES

[1] E. Boyle, N. Gilboa, and Y. Ishai, “Function secret sharing,” in Advances in Cryptology—EUROCRYPT. Sofia, Bulgaria: Springer, Apr. 2015, pp. 337–367.

[2] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, “Private information retrieval,” in Proc. 36th Ann. Symp. Found. Comput. Sci., 1995, pp. 965–982.

[3] B. Chor and N. Gilboa, “Computationally private information retrieval (extended abstract),” in Proc. 29th Ann. ACM Symp. Theory Comput., El Paso, TX, USA, May 1997, pp. 304–313.
[4] A. Beimel, Y. Ishai, and E. Kushilevitz, “General constructions for information-theoretic private information retrieval,” J. Comput. Syst. Sci., vol. 71, no. 2, pp. 213–247, Aug. 2005.

[5] R. Ostrovsky and V. Shoup, “Private information storage (extended abstract),” in Proc. 29th Annu. ACM Symp. Theory Comput., El Paso, TX, USA, May 1997, pp. 294–303.

[6] N. Gilboa and Y. Ishai, “Distributed point functions and their applications,” in Advances in Cryptology—EUROCRYPT, Copenhagen, Denmark: Springer, May 2014, pp. 640–658.

[7] E. Boyle, N. Gilboa, and Y. Ishai, “Function secret sharing: Improvements and extensions,” in Proc. ACM Conf. Comput. Commun. Secur., Vienna, Austria, Oct. 2016, pp. 1292–1303.

[8] R. Impagliazzo, L. A. Levin, and M. Luby, “Pseudo-random generation from one-way functions (extended abstracts),” in Proc. Annu. Symp. Theory Comput., Seattle, WA, USA, May 1989, pp. 12–24.

[9] [Author's name], [Author's name], [Author's name], and [Author's name], “A pseudorandom generator from any one-way function,” SIAM J. Comput., vol. 28, no. 4, pp. 1364–1396, Jan. 1999.

[10] E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, and M. Orrù, “Homomorphic secret sharing: Optimizations and applications,” in Proc. ACM Conf. Comput. Commun. Secur., Dallas, TX, USA, Oct./Nov. 2017, pp. 2105–2122.

[11] SIAM J. Comput., vol. 28, no. 4, p. 585, Aug. 2016.

[12] E. Boyle, L. Kohl, and P. Scholl, “Homomorphic secret sharing from lattices without FHE,” in Proc. 36th Annu. Int. Conf. Theory Appl. Cryptograph. Techn., Adv. Cryptol.-EUROCRYPT, Darmstadt, Germany, May 2019, pp. 3–33.

[13] R. W. F. Lai, G. Malavolta, and D. Schröder, “Homomorphic secret sharing for low degree polynomials,” in Proc. Adv. Cryptol.-ASIACRYPT, 24th Int. Conf. Theory Appl. Cryptol. Inf. Secur., Brisbane, QLD, Australia, Dec. 2018, pp. 279–309.

[14] T. Plantard, W. Susilo, and Z. Zhang, “Fully homomorphic encryption using hidden ideal lattice,” IEEE Trans. Inf. Forensics Security, vol. 8, no. 12, pp. 2127–2137, Dec. 2013.

[15] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in Proc. 41st Annu. ACM Symp. Theory Comput., Bethesda, MD, USA, May/June 2009, pp. 169–178.

[16] F. Wang, C. Yun, S. Goldwasser, V. Vaikuntanathan, and M. Zabaria, “Splinter: Practical private queries on public data,” in Proc. 14th USENIX Symp. Networked Syst. Design Implement. (NSDI), Boston, MA, USA, May 2017, pp. 299–313.

[17] E. Boyle, N. Gilboa, and Y. Ishai, “Secure computation with preprocessing via function secret sharing,” IACR Cryptol. ePrint Arch., vol. 2019, p. 1095, Dec. 2019.

[18] A. Shamir, “How to share a secret,” Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.

[19] D. Yuan, M. He, S. Zeng, X. Li, and L. Lu, “(t, p)-threshold point function secret sharing scheme based on polynomial interpolation and its application,” in Proc. 9th Int. Conf. Utility Cloud Comput., Shanghai, China, Dec. 2016, pp. 269–275.

[20] E. Boyle, N. Gilboa, Y. Ishai, H. Lin, and S. Tessaro, “Foundations of homomorphic secret sharing,” in Proc. 9th Innov. Theor. Comput. Sci. Conf., Cambridge, MA, USA, Jan. 2018, Art. no. 21.

JINGLONG LUO received the B.S. degree in mathematics and computing science from Shanxi Datong University, Fujian, China, in 2017. He is currently pursuing the M.S. degree with the College of Mathematics and Informatics, Fujian Normal University, Fujian. His research interests include cryptography and information security.

JINWEI ZHANG received the B.S. degree in information and computer science from Xiangtan University, Xiangtan, China, in 2016. He is currently pursuing the Ph.D. degree in information security at the State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, Beijing, China. His research interests include cryptography and information security.

LIANG FENG ZHANG received the Ph.D. degree in cryptography and information security from Nanyang Technological University, Singapore. He is currently an Assistant Professor with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. His main research interests focus on the design of protocols for secure multiparty computation problems, such as the private information retrieval and the verifiable computation.

FUCHUN LIN received the Ph.D. degree in mathematics from the Division of Mathematical Sciences, Nanyang Technological University, Singapore, in 2014. From 2014 to 2017, he was a Postdoctoral Scholar with the University of Calgary, Canada. He is currently a Research Fellow with Nanyang Technological University. His research interests focus on the information-theoretic security aspects in cryptography, such as secret sharing, wiretap codes, and non-malleable codes.

CHANGLU LIN received the B.S. and M.S. degrees in mathematics from Fujian Normal University, China, in 2002 and 2005, respectively, and the Ph.D. degree in information security from the State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, China, in 2010. He currently works with the College of Mathematics and Informatics, and the Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University. He is interested in cryptography and network security. He has conducted research in diverse areas, including secret sharing, multiparty computation, public key cryptography and their applications.