

## Article

# Expand-and-Randomize: An Algebraic Approach to Secure Computation

Yizhou Zhao \*  and Hua Sun \*

Department of Electrical Engineering, University of North Texas, Denton, TX 76203, USA

\* Correspondence: yizhouzhao@my.unt.edu (Y.Z.); hua.sun@unt.edu (H.S.)

**Abstract:** We consider the secure computation problem in a minimal model, where Alice and Bob each holds an input and wish to securely compute a function of their inputs at Carol without revealing any additional information about the inputs. For this minimal secure computation problem, we propose a novel coding scheme built from two steps. First, the function to be computed is *expanded* such that it can be recovered while additional information might be leaked. Second, a *randomization* step is applied to the expanded function such that the leaked information is protected. We implement this expand-and-randomize coding scheme with two algebraic structures—the finite field and the modulo ring of integers, where the *expansion* step is realized with the *addition* operation and the *randomization* step is realized with the *multiplication* operation over the respective algebraic structures.

**Keywords:** secure computation; capacity; algebraic codes



**Citation:** Zhao, Y.; Sun, H. Expand-and-Randomize: An Algebraic Approach to Secure Computation. *Entropy* **2021**, *23*, 1461. <https://doi.org/10.3390/e23111461>

Academic Editor: Cyril Branciard

Received: 25 September 2021

Accepted: 2 November 2021

Published: 4 November 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



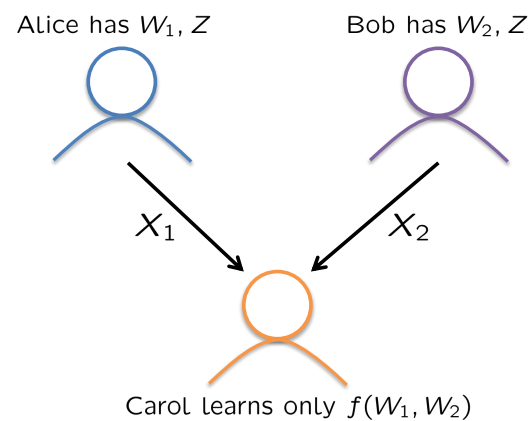
**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Cryptographic primitives are canonical and representative problems that capture the key challenges in understanding the fundamentals of security and privacy, and are essential building blocks for more sophisticated systems and protocols. There is much recent interest in using information theoretic tools to tackle classical cryptographic primitives [1–7]. Along this line, the focus of this work is on a widely studied primitive in cryptography: secure (multiparty) computation [8].

Secure computation refers to the problem where a number of users wish to securely compute a function on their inputs without revealing any unnecessary information. Interestingly, challenging as it seems, secure computation is always *feasible*, i.e., with at least three users, *any* function can be computed securely in the information theoretic sense [9,10]. We consider the most basic model with honest-but-curious users and no colluding users. This work only considers this basic model, but we note that many other variants have been considered in the literature, which may or may not be feasible depending on the specific assumptions and system parameters of the variant (see, e.g., in [11]). However, what is largely open is how to perform secure computation *optimally*, i.e., efficient secure computation solutions are not known for most cases [6].

The main motivation of this work is to make progress towards constructing efficient secure computation codes. Towards this end, we focus on a minimal model of secure computation, introduced by Feige, Kilian, and Naor in 1994 [12]. In this model (see Figure 1), there are three users: Alice, Bob, and Carol. Alice and Bob have inputs  $W_1$  and  $W_2$ , respectively, and wish to compute a function  $f(W_1, W_2)$  at Carol without revealing any additional information about their inputs beyond what is revealed by the function itself. To do so, Alice and Bob share a common random variable  $Z$  that is independent of the inputs and send codewords  $X_1$  and  $X_2$  to Carol, respectively. From  $X_1, X_2$ , Carol can recover  $f(W_1, W_2)$  and conditioned on  $f(W_1, W_2)$ , the codewords  $X_1$  and  $X_2$  are independent of  $W_1, W_2$  so that no additional information is leaked.



**Figure 1.** The minimal secure computation problem [12].

The key feature of this formulation is that the communication protocol consists of only one codeword from each party that holds the input (thus non-interactive), while for the general secure computation formulation [9,10], interactive protocols are allowed and typically used. Elemental as it seems, this minimal secure computation problem preserves most challenging features of general secure computation; in particular, feasibility results remain *strong* and optimality results remain *weak*, i.e., any function  $f$  can be computed securely while the construction of efficient codes remains open in general [12,13]. In this work, we focus exclusively on the original three-party formulation of minimal secure computation [12], but note that many interesting variants have been studied (sometimes under different names to highlight different assumptions) in the literature, e.g., more than three parties [14–16], colluding parties [17–20], other security notions [21], and unresponsive parties [22].

The main contribution of this work is a novel coding scheme that relies on algebraic structures to ensure correctness and security. To illustrate the idea of our coding scheme, let us first consider an example. Suppose Alice and Bob each holds a ternary input,  $W_1, W_2 \in \{0, 1, 2\}$ , and wish to compute if  $W_1$  is equal to  $W_2$ , i.e.,  $f(W_1, W_2) = \text{Yes}$  if  $W_1 = W_2$ , and  $f(W_1, W_2) = \text{No}$  otherwise.

As the equality function may not be easily computed in a secure manner, we first *expand* it to a linear function so that it becomes simpler to deal with. As shown in Figure 2, we use the linear function  $W_1 - W_2$  over the finite field  $\mathbb{F}_3$  (equivalent to operations modulo 3). For this expansion, we require that the original function can be fully recovered by the expanded function. This is easily verified for this example, where  $W_1 - W_2 \neq 0$  if and only if  $W_1$  is not equal to  $W_2$ . This expansion step does not solve the secure computation problem because additional information may be leaked. For example, here Carol should only know if  $W_1 - W_2 = 0$  and is not supposed to learn whether  $W_1 - W_2$  is 1 or 2. To prevent this leakage, we invoke another step of *randomization* so that the leaked information by the expanded function becomes confusable and thus protected. For this equality function example, when  $W_1 - W_2 \neq 0$ , we wish to make the result equally likely to be 1 or 2. This is realized by multiplying  $W_1 - W_2$  with  $\gamma$ , where  $\gamma$  is uniform over  $\{1, 2\}$ . The multiplication operation is also over  $\mathbb{F}_3$ . Thus,

$$\text{when } W_1 - W_2 = 1, \gamma \times (W_1 - W_2) = \gamma \text{ is equally likely to be 1 or 2;} \quad (1)$$

$$\text{when } W_1 - W_2 = 2, \gamma \times (W_1 - W_2) = \gamma \times 2 \text{ is equally likely to be 1 or 2.} \quad (2)$$

Note that  $2 \times 2 = 4 = 1$  over  $\mathbb{F}_3$ . After this randomization step, the *randomized expanded function* does not reveal any additional information beyond the original equality function. The above expand-and-randomize procedure can be easily converted to a distributed secure computation protocol. In particular, Alice and Bob share a common random variable,

$Z = (\gamma, z)$ , where  $\gamma$  and  $z$  are independent,  $\gamma$  is uniform over  $\{1, 2\}$ , and  $z$  is uniform over  $\{0, 1, 2\}$ . The codewords  $X_1, X_2$  sent by Alice and Bob to Carol are

$$X_1 = \gamma \times W_1 + z, \quad (3)$$

$$X_2 = \gamma \times W_2 + z. \quad (4)$$

To decode  $f(W_1, W_2)$  with no error, Carol subtracts  $X_2$  from  $X_1$ ,  $X_1 - X_2 = \gamma \times (W_1 - W_2)$  and claims that  $W_1$  is equal to  $W_2$  if and only if  $X_1 - X_2 = 0$ . To see why perfect security holds, note that  $(X_1, X_2)$  is invertible to  $(X_1 - X_2, X_2)$ ; both  $X_1 - X_2$  and  $X_2$  (protected by an independent uniform noise  $z$ ) do not leak any information. Specifically, the joint distributions of  $(X_1, X_2)$  remain the same for all  $(W_1, W_2)$  pairs so that  $f(W_1, W_2)$  are the same. That is, when  $W_1$  is equal to  $W_2$ , i.e.,  $(W_1, W_2) \in \{(0, 0), (1, 1), (2, 2)\}$ ,  $(X_1, X_2)$  are identically distributed ( $X_1$  is uniform over  $\{0, 1, 2\}$  and  $X_2$  is equal to  $X_1$ ) and the same observation holds for all  $(W_1, W_2)$  pairs where  $W_1$  is not equal to  $W_2$  ( $X_1 - X_2$  is uniform over  $\{1, 2\}$ ;  $X_2$  is independent of  $X_1 - X_2$  and is uniform over  $\{0, 1, 2\}$ ). Interestingly, this secure computation code is also communication optimal, i.e., the size of  $X_1$  and  $X_2$  must be no less than  $\log_2 3$  bits each (required even if there is no security constraint).

A closer inspection of the above scheme reveals that the key is to find an expanded function such that the expanded function outputs corresponding to the same original function output can be randomized to be fully confusable. The first main result of this work is to characterize the structural properties of such *confusable sets* over the finite field  $\mathbb{F}_q$ , where  $q$  is a prime power. The confusable function outputs turn out to be characterized by the property that their discrete logarithms (in exponential representation of the finite field elements) have the same remainder in modular arithmetic. Details will be presented in Section 3.1.

As it turns out, the expand-and-randomize coding scheme is not limited to the finite field. As our second main result, we implement it over the ring of integers modulo  $n$ ,  $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$ . The ring is equipped with two operations—addition and multiplication—both defined in modulo  $n$  arithmetic. Let us consider an example to illustrate how  $\mathbb{Z}_n$  is used. Consider the selected-switch function in Figure 3. Alice has a binary input,  $W_1 \in \{0, 1\}$ . Bob has a ternary input,  $W_2 \in \{0, 1, 2\}$ . When  $W_1 \geq W_2$ , the switch function  $f$  is OFF and the output is 0 (we may think that the output is not connected to the input, so it is a constant). When  $W_1 < W_2$ , the switch function  $f$  is ON and the output is equal to the input vector (all information about  $W_1, W_2$  goes through).

$f$	0	1	2		$W_1 - W_2$	0	1	2		$\gamma \times (W_1 - W_2)$	0	1	2
0	Yes	No	No	Expand $\rightarrow$	0	0	2	1	Randomize $\rightarrow$	0	0	$\{1, 2\}$	$\{1, 2\}$
1	No	Yes	No		1	1	0	2		1	$\{1, 2\}$	0	$\{1, 2\}$
2	No	No	Yes		2	2	1	0		2	$\{1, 2\}$	$\{1, 2\}$	0

**Figure 2.** The expand-and-randomize coding scheme for the *equal* function. In the function table, each row corresponds to a value of  $W_1$  and each column corresponds to a value of  $W_2$ . When the function output is a random variable, the set of possible values is shown in the table. The functions  $W_1 - W_2$  and  $\gamma \times (W_1 - W_2)$  are computed over the finite field  $\mathbb{F}_3$  and  $\gamma$  is uniform over the set  $\{1, 2\}$ .

$f$	0	1	2		$\tilde{W}_1 + \tilde{W}_2$	0	2	5		$\gamma \times (\tilde{W}_1 + \tilde{W}_2)$	0	2	5
0	0	(0, 1)	(0, 2)	Expand $\rightarrow$	4	4	0	3	Randomize $\rightarrow$	4	$\{2, 4\}$	0	3
1	0	0	(1, 2)		2	2	4	1		2	$\{2, 4\}$	$\{2, 4\}$	$\{1, 5\}$

**Figure 3.** The expand-and-randomize coding scheme for the *selected-switch* function. The expanded function  $\tilde{W}_1 + \tilde{W}_2$  and the randomized expanded function  $\gamma \times (\tilde{W}_1 + \tilde{W}_2)$  are defined over the ring of integers modulo 6,  $\mathbb{Z}_6$ .  $\gamma$  is uniform over  $\{1, 5\}$ , the set of integers that are coprime with 6.

Following the expand-and-randomize coding paradigm, we first expand the original function to the addition function over  $\mathbb{Z}_6$  such that it can be fully recovered. Note that to facilitate the construction of the expanded function, here we perform an invertible transformation on the inputs,  $W_1 \rightarrow \tilde{W}_1(0 \rightarrow 4, 1 \rightarrow 2)$ ,  $W_2 \rightarrow \tilde{W}_2(0 \rightarrow 0, 1 \rightarrow 2, 2 \rightarrow 5)$ . The expanded function reveals more information than allowed when the output is 2 or 4. To protect this information, a randomization step is realized by multiplying  $\gamma$ , which is uniform over  $\{1, 5\}$ . Now,  $2 \times \{1, 5\} = \{2, 10\} = \{2, 4\}$  modulo 6, and  $4 \times \{1, 5\} = \{4, 20\} = \{2, 4\}$  modulo 6. Therefore, the expanded function after randomization can be used to produce the following secure computation protocol. The codewords are  $X_1 = \gamma \times \tilde{W}_1 + z$ ,  $X_2 = \gamma \times \tilde{W}_2 - z$ , where  $Z = (\gamma, z)$ ,  $\gamma$  and  $z$  are independent,  $\gamma$  is uniform over  $\{1, 5\}$ , and  $z$  is uniform over  $\{0, 1, 2, 3, 4, 5\}$ . To decode, Carol will compute  $X_1 + X_2 = \gamma \times (\tilde{W}_1 + \tilde{W}_2)$ . Comparing the original function  $f$  and the randomized expanded function  $\gamma \times (\tilde{W}_1 + \tilde{W}_2)$ , it is easy to construct the decoding rule based on  $X_1 + X_2$  (see Figure 3). Following a straightforward argument as presented above, we may show that the correctness and security constraints are satisfied. Details will be presented in Theorem 1.

From this example, we find that the crux of the scheme is a *partition* of the elements of  $\mathbb{Z}_6$  into several *disjoint* confusable sets such that when any two elements of a confusable set  $\mathcal{S}$  are multiplied with  $\gamma$  which is uniform over a carefully chosen set ( $\gamma$  is referred to as the *randomizer*), they will produce identically distributed sets of values; specifically, both will produce the confusable set  $\mathcal{S}$ .

$$\begin{aligned} \mathbb{Z}_6 &= \{0\} \cup \{1, 5\} \cup \{2, 4\} \cup \{3\}, \gamma \text{ is uniform over } \{1, 5\}; \text{ over } \mathbb{Z}_6 : \\ 1 \times \{1, 5\} &= 5 \times \{1, 5\} = \{1, 5\}, 2 \times \{1, 5\} = 4 \times \{1, 5\} = \{2, 4\}, 3 \times \{1, 5\} = \{3\}. \end{aligned} \quad (5)$$

The main technical challenge is to understand which sets of elements can serve as the randomizer  $\gamma$  and how the ring  $\mathbb{Z}_n$  is partitioned into disjoint confusable sets such that security is guaranteed. For this purpose, we require a few notions from group theory and number theory. Details are presented in Section 3.2. To get a glimpse, consider the above example (see (5)), where the randomizer  $\gamma$  is from the set of integers that are coprime with 6 (1 and 5 both have no common divisor with 6), and the confusable sets are the sets of integers that have the same greatest common divisor with 6 (e.g.,  $\gcd(2, 6) = \gcd(4, 6) = 2$ ).

Our proposed coding scheme is inspired by two examples (binary logical AND function and ternary comparison function) presented in Appendix A and Appendix B of the original minimal secure computation paper [12], where modular arithmetic over a prime number  $p$  is used. Note that for a prime  $p$ , the algebraic operations (addition and multiplication) in both finite field  $\mathbb{F}_p$  and the ring of integers modulo  $p$ ,  $\mathbb{Z}_p$  are modular arithmetic. Along this line, our work can be viewed as a generalization of the examples from in [12] to a general class of achievable schemes that distill the underlying algebraic structure and work over finite fields and modulo rings of integers with general (non-prime) cardinality.

## 2. Problem Statement

Consider a pair of inputs  $(W_1, W_2) \in \{0, 1, \dots, m_1 - 1\} \times \{0, 1, \dots, m_2 - 1\}$  and a function  $f : \{0, 1, \dots, m_1 - 1\} \times \{0, 1, \dots, m_2 - 1\} \rightarrow \{0, 1, \dots, |f| - 1\}$ . We assume the function  $f$  is discrete, and use  $|f|$  to denote the cardinality of the range of  $f$ .  $W_1$  is available to Alice and  $W_2$  is available to Bob. Alice and Bob also both hold a common random variable  $Z$  whose distribution does not depend on  $W_1, W_2$ .

Alice and Bob wish to compute  $f(W_1, W_2)$  securely. To this end, Alice sends a codeword  $X_1$  and Bob sends a codeword  $X_2$  to Carol.  $X_1$  is a function of  $W_1$  and  $Z$ , and has  $L_1$  bits.  $X_2$  is a function of  $W_2$  and  $Z$ , and has  $L_2$  bits. The function  $f$  is known to Alice, Bob, and Carol. As our proposed code will have a fixed length, here we only define fixed-length codes, i.e.,  $L_1$  does not depend on the value of  $W_1$ . In general, variable-length codes might have a lower expected length (see Remark 3).

From  $X_1, X_2$ , Carol can recover  $f(W_1, W_2)$  with no error. This is referred to as the correctness constraint. To ensure Carol does not learn anything beyond  $f(W_1, W_2)$ , the following security constraint must be satisfied.

(Security) For any joint distribution of  $(W_1, W_2)$ ,  $I(X_1, X_2; W_1, W_2 | f(W_1, W_2)) = 0$ .

Equivalently, the security constraint can be stated as follows.

For any  $(W_1, W_2)$  pairs such that  $f(W_1, W_2)$  are equal,  
 $(X_1, X_2)$  are identically distributed. (6)

A rate tuple  $(L_1, L_2)$  is said to be achievable if there exists a secure computation scheme, for which the correctness and security constraints are satisfied. The closure of the set of all achievable rate tuples is called the optimal rate region.

The main result of this work is a new achievable scheme for secure computation and the new scheme works for any joint distribution of  $(W_1, W_2)$ , so we do not specify explicitly this joint distribution. Further, for simplicity, we introduce the problem statement as a scalar coding problem. Concrete distributions will be given and  $L$ -length extensions (block inputs) will be considered when they play more significant roles in the results, e.g., when we discuss  $\epsilon$ -error schemes in Section 4.2 and converse results in Section 3.3.

### 3. The Main Coding Scheme

In this section, we present a novel secure computation code that implements the expand-and-randomize scheme over the finite field  $\mathbb{F}_q$  and the ring of integers modulo  $n$ , denoted as  $\mathbb{Z}_n$ . Let us start with relevant definitions.

**Definition 1** (Confusable Sets and Randomizer). *Sets  $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \dots$  are called confusable sets if they form a partition of all elements from  $\mathbb{F}_q$  or  $\mathbb{Z}_n$  and there exists a uniform random variable  $\gamma$  over a set  $\mathcal{S}^* \subset \mathbb{F}_q$  or  $\mathbb{Z}_n$  such that  $\forall s \in \mathcal{S}_i, \gamma \times s$  is uniform over  $\mathcal{S}_i$ .  $\gamma$  is called the randomizer.*

The requirement in Definition 1 is stronger than what is needed for security. It suffices to have identical (instead of uniform) distributions over some disjoint set (instead of the confusable set). However, for our proposed scheme, it turns out that these relaxations do not lead to improved achievable rate regions such that they are not considered for simplicity. The notions introduced in Definition 1 are closely related to group actions (and orbits) studied in group theory [23]. Along this line, the main effort of this work is to identify a class of group actions that can be used in secure computation and to prove that the class found indeed forms valid group actions. Our proof is relatively elementary, relying on basic number theoretic properties. It is possible to alternatively prove the statements using group actions. We view our main findings as identifying which operations form group actions and their applicability to secure computation (while the proof of validity can be done in various ways). Another related concept that has been studied in cryptography is randomized polynomials [24,25], which also rely on extra randomization to expand the deterministic function to be computed. Our work used different randomization techniques.

**Definition 2** (Feasible Expanded Function). *For a function  $f(W_1, W_2)$ , a function  $\tilde{f}(\tilde{W}_1, \tilde{W}_2) = \tilde{W}_1 + \tilde{W}_2$  over  $\mathbb{F}_q$  or  $\mathbb{Z}_n$  is called a feasible expanded function if the mapping between  $W_1$  and  $\tilde{W}_1$ , the mapping between  $W_2$  and  $\tilde{W}_2$ , and the mapping between  $f(W_1, W_2)$  and the index of the confusable set to which  $\tilde{f}(\tilde{W}_1, \tilde{W}_2)$  belongs are all invertible.*

For an example of a feasible expanded function (for the equality function with ternary inputs) over  $\mathbb{F}_3$ , see Figure 2. Specifically,  $\mathbb{F}_3 = \{0, 1, 2\} = \mathcal{S}_0 \cup \mathcal{S}_1$ , where  $\mathcal{S}_0 = \{0\}, \mathcal{S}_1 = \{1, 2\}$ .  $\gamma$  is uniform over  $\mathcal{S}^* = \{1, 2\}$ .  $\gamma \times 1$  and  $\gamma \times 2$  are both uniformly distributed over  $\mathcal{S}_1$ .  $\tilde{W}_1 = W_1, \tilde{W}_2 = -W_2$ .  $f$  is the equality function and  $\tilde{f}$  is  $W_1 - W_2$ . The Yes output of  $f$



is mapped to  $S_0$  over  $\tilde{f}$  and the No output of  $f$  is mapped to  $S_1$  over  $\tilde{f}$ . For an example of a feasible expanded function over  $\mathbb{Z}_6$ , see Figure 3.

A feasible expanded function as defined above naturally leads to a correct and secure computation scheme, presented in the following theorem.

**Theorem 1.** For any function  $f(W_1, W_2)$ , if we have a feasible expanded function  $\tilde{f}(\tilde{W}_1, \tilde{W}_2) = \tilde{W}_1 + \tilde{W}_2$  over  $\mathbb{F}_q$  or  $\mathbb{Z}_n$ , then the following computation code is both correct and secure:

$$X_1 = \gamma \times \tilde{W}_1 + z, X_2 = \gamma \times \tilde{W}_2 - z \quad (7)$$

where  $Z = (z, \gamma)$ ,  $\gamma$  is the randomizer,  $z$  and  $\gamma$  are independent, and  $z$  is uniform over  $\mathbb{F}_q$  or  $\mathbb{Z}_n$ . Specifically, in this scheme, Alice and Bob each sends a symbol from  $\mathbb{F}_q$  or  $\mathbb{Z}_n$  to Carol.

**Proof of Theorem 1.** The proof of correctness and security follows in a straightforward manner from the definitions of the confusable sets, the randomizer, and the feasible expanded function. First, we consider the correctness constraint. To recover  $f(W_1, W_2)$  with no error, Carol may compute  $X_1 + X_2 = \gamma \times (\tilde{W}_1 + \tilde{W}_2)$ , from which Carol can uniquely identify the index of the confusable set (invertible to the original function output). Note that by the definition of the confusable sets and the randomizer, multiplying with  $\gamma$  does not change the confusable set index. Second, we consider the security constraint (6). Consider any  $(W_1, W_2)$  pairs that produce the same  $f(W_1, W_2)$  output, and we show that  $(X_1, X_2)$  are identically distributed. To see this, note that  $(X_1, X_2)$  is invertible to  $(X_1 + X_2, X_2) = (\gamma \times (\tilde{W}_1 + \tilde{W}_2), \gamma \times \tilde{W}_2 - z)$ . By the definition of the confusable sets,  $X_1 + X_2$  is uniform over the confusable set that corresponds to  $f(W_1, W_2)$ ;  $X_2$  is independent of  $X_1 + X_2$  and is uniform over  $\mathbb{F}_q$  or  $\mathbb{Z}_n$  due to the uniformity and independence of  $z$ . Therefore,  $(X_1 + X_2, X_2)$  are always uniform thus are identically distributed (so are  $(X_1, X_2)$ ). The proof is complete.  $\square$

The coding scheme in Theorem 1 relies on the structure of the confusable sets and the randomizer upon which feasible expanded functions are built. Thus, it is crucial to understand the structure of the confusable sets and the randomizer, i.e., which set of elements can be used as the randomizer and how the algebraic object is partitioned to confusable sets. This structure problem is addressed next, through algebraic characterizations. The finite field case is considered in Section 3.1 and the ring of integers modulo  $n$  case is considered in Section 3.2.

### 3.1. Finite Field

We first recall some basic facts of finite fields (refer to standard textbooks such as that in [26]). A finite field  $\mathbb{F}_q$  exists only when  $q = p^n$ , where  $p$  is a prime and  $n$  is a positive integer.  $\mathbb{F}_q$  has  $q = p^n$  elements. Any two fields with  $p^n$  elements are isomorphic, thus  $\mathbb{F}_q$  is referred to as the finite field. The  $p^n$  elements of  $\mathbb{F}_q$  are the polynomials  $a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}$ , where  $a_i \in \{0, 1, \dots, p-1\}, \forall i \in \{0, 1, \dots, n-1\}$ . The addition and multiplication operations over  $\mathbb{F}_q$  are defined modulo  $h(x)$ , where  $h(x)$  is an irreducible polynomial of degree  $n$  that always exists. The non-zero elements of  $\mathbb{F}_q$  form a multiplicative group, denoted as  $\mathbb{F}_q^\times$ .  $\mathbb{F}_q^\times$  is a cyclic group  $\{1, g, g^2, \dots, g^{q-2}\}$  that can be generated by a primitive element  $g \in \mathbb{F}_q^\times$ . Denote  $g^0 = 1$ .

**Example 1.** The finite field  $\mathbb{F}_{2^3}$  can be constructed by addition and multiplication modulo  $h(x) = x^3 + x + 1$ . The multiplicative group  $\mathbb{F}_{2^3}^\times = \{1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$  can be generated by  $g = x$ .

$$g^2 = x^2, g^3 = x^3 \bmod (x^3 + x + 1) = x + 1, g^4 = x^2 + x, \quad (8)$$

$$g^5 = x^3 + x^2 \bmod (x^3 + x + 1) = x^2 + x + 1, \quad (9)$$

$$g^6 = x^3 + x^2 + x \bmod (x^3 + x + 1) = x^2 + 1, \quad (10)$$

$$g^7 = x^3 + x \bmod (x^3 + x + 1) = 1 = g^0. \quad (11)$$

Equipped with the above results (in particular, the cyclic property of the multiplicative group  $\mathbb{F}_q^\times$ ), we are ready to state in the following theorem the algebraic characterization of the confusable sets and the randomizer over  $\mathbb{F}_q$ .

**Theorem 2.** For  $\mathbb{F}_q$  where  $q = p^n$ ,  $p$  is a prime,  $n$  is an integer, and  $g$  is a primitive element of  $\mathbb{F}_q^\times$ , the confusable sets and the randomizer can be chosen as follows. Consider any divisor  $d$  of  $p^n - 1$ , i.e.,  $b = (p^n - 1)/d$  is an integer.

$$\gamma \text{ is uniform over } \mathcal{S}^* = \{g^0, g^d, g^{2d}, \dots, g^{(b-1)d}\}, \quad (12)$$

$$\mathbb{F}_q = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S}_2 \cup \dots \cup \mathcal{S}_d, \quad (13)$$

$$\mathcal{S}_0 = \{0\}, \mathcal{S}_i = \{g^{i-1}, g^{d+i-1}, g^{2d+i-1}, \dots, g^{(b-1)d+i-1}\}, \forall i \in \{1, 2, \dots, d\}. \quad (14)$$

**Remark 1.** In words, the elements of a confusable set are such that their discrete logarithms have the same remainder modulo a divisor of  $p^n - 1$ .

Before we prove Theorem 2, let us first understand it through an example and use it to securely compute a function.

**Example 2.** Consider  $\mathbb{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$ . A primitive element of  $\mathbb{F}_7^\times$  is 3. Setting  $d = 3$ , the following confusable sets are given by Theorem 2.

$$\begin{aligned} \mathcal{S}_0 &= \{0\}, \mathcal{S}_1 = \{3^0, 3^3\} = \{1, 27\} \bmod 7 = \{1, 6\}, \\ \mathcal{S}_2 &= \{3^1, 3^4\} = \{3, 4\}, \mathcal{S}_3 = \{3^2, 3^5\} = \{2, 5\}. \end{aligned} \quad (15)$$

Consider the function  $f(W_1, W_2)$  shown in Figure 4, for which a feasible expanded function can be built upon the confusable sets given above.

$f$	0	1	2		$\tilde{W}_1 + \tilde{W}_2$	2	3	4		$\gamma \times (\tilde{W}_1 + \tilde{W}_2)$	2	3	4
0	0	1	1	Expand over $\mathbb{F}_7$	0	2	3	4	Randomize over $\mathbb{F}_7$	0	{2, 5}	{3, 4}	{3, 4}
1	0	2	3		3	5	6	0		3	{2, 5}	{1, 6}	{0}

**Figure 4.** An expand-and-randomize secure computation code over  $\mathbb{F}_7$ , where  $\gamma$  is uniform over  $\{1, 6\}$ .

While the primitive element  $g$  of  $\mathbb{F}_q^\times$  is guaranteed to exist, there is no analytic formula for it and finding it computationally is difficult in general. Further, given the polynomial representation of  $g$ , it is generally non-trivial to determine the minimum field size  $q$  such that there exists a feasible expanded function over  $\mathbb{F}_q$  for a specific function  $f$ . A list of confusable sets for all finite fields  $\mathbb{F}_q, q < 20$  is given in Figure A1 (see the Appendix A).

**Proof of Theorem 2.** To verify that the definition of the confusable sets is satisfied, we only need to show that  $\forall s \in \mathcal{S}_i, i \in \{0, 1, \dots, d\}, \gamma \times s$  is uniform over  $\mathcal{S}_i$ . This is proved as follows. When  $i = 0$ ,  $\mathcal{S}_0 = \{0\}$  so that  $s = 0$  and  $\gamma \times s = \{0\} = \mathcal{S}_0$ . When  $i \in \{1, \dots, d\}$ , consider any element from  $\mathcal{S}_i$ , e.g.,  $s = g^{jd+i-1}, j \in \{0, 1, \dots, b-1\}$ . We have

$$\gamma \times s = \{g^0, g^d, g^{2d}, \dots, g^{(b-1)d}\} \times g^{jd+i-1} \quad (16)$$

$$= \{g^{jd+i-1}, g^{(j+1)d+i-1}, g^{(j+2)d+i-1}, \dots, g^{(j+b-1)d+i-1}\} \quad (17)$$

$$= \{g^{i-1}, g^{d+i-1}, g^{2d+i-1}, \dots, g^{(b-1)d+i-1}\} = \mathcal{S}_i \quad (18)$$

where (18) follows from the fact that  $g^{bd} = g^{p^n-1} = 1$  and the observation that any  $b$  consecutive integers form the same set under modulo  $b$ , i.e.,  $\{0, 1, \dots, b-1\} = \{j, j+1, \dots, j+b-1\} \bmod b$ . As  $\gamma$  is uniform,  $\gamma \times s$  is uniform (over  $\mathcal{S}_i$ ) as well.  $\square$

### 3.2. Ring of Integers Modulo $n$

To facilitate the presentation of the algebraic characterization of the confusable sets and the randomizer over  $\mathbb{Z}_n$ , we first introduce some definitions and preliminary results.

**Definition 3** (Set of Integers with Same gcd). Consider any proper divisor  $d$  of a given integer  $n$ , i.e.,  $d < n$  and  $n/d$  is an integer. We denote by  $\mathbb{Z}_n^{(d)}$  the set of integers in  $\mathbb{Z}_n$  so that their greatest common divisors with  $n$  are  $d$ , i.e.,  $\mathbb{Z}_n^{(d)} = \{a \in \mathbb{Z}_n \mid \gcd(a, n) = d\}$ .

For example, suppose  $n = 15 = 3 \times 5$ , which has proper divisors 1, 3, 5. Then,

$$\mathbb{Z}_{15}^{(1)} = \{1, 2, 4, 7, 8, 11, 13, 14\}, \mathbb{Z}_{15}^{(3)} = \{3, 6, 9, 12\}, \mathbb{Z}_{15}^{(5)} = \{5, 10\}. \quad (19)$$

$$\text{Further, } \mathbb{Z}_{15} = \{0, 1, \dots, 14\} = \{0\} \cup \mathbb{Z}_{15}^{(1)} \cup \mathbb{Z}_{15}^{(3)} \cup \mathbb{Z}_{15}^{(5)}. \quad (20)$$

The set  $\mathbb{Z}_n^{(1)}$  has been extensively studied in abstract algebra (see e.g., [23]) and number theory (see, e.g., in [27]), and is referred to as the multiplicative group of integers modulo  $n$  (it turns out to form a group under multiplication modulo  $n$ ), so we adopt the standard existing notation  $\mathbb{Z}_n^\times = \mathbb{Z}_n^{(1)}$ .

Note that

$$\mathbb{Z}_n^{(d)} = \{a \in \mathbb{Z}_n \mid \gcd(a, n) = d\} = d \times \{a \in \mathbb{Z}_{n/d} \mid \gcd(a, n/d) = 1\} = d \times \mathbb{Z}_{n/d}^\times. \quad (21)$$

For example,

$$\begin{aligned} \mathbb{Z}_{15}^{(3)} &= \{3, 6, 9, 12\} = 3 \times \{1, 2, 3, 4\} = 3 \times \mathbb{Z}_5^\times, \\ \mathbb{Z}_{15}^{(5)} &= \{5, 10\} = 5 \times \{1, 2\} = 5 \times \mathbb{Z}_3^\times. \end{aligned} \quad (22)$$

We present an important result on the projection of a multiplicative subgroup of  $\mathbb{Z}_n^\times$  over  $\mathbb{Z}_d^\times$  in the following lemma. To differentiate set and multiset (where an element might appear several times), we use the notation  $\bar{H}$  for a multiset  $H$ .

**Lemma 1.** Consider an arbitrary subgroup  $G_n$  of  $\mathbb{Z}_n^\times$  (under multiplication modulo  $n$ ). When we take  $G_n$  modulo  $d$  (where  $d$  is a divisor of  $n$  and  $d \neq 1$ ), we have multiple copies of a subgroup of  $\mathbb{Z}_d^\times$  (under multiplication modulo  $d$ ), i.e.,  $G_n \bmod d = \bar{\{G_d, G_d, \dots, G_d\}}$ , where  $G_d$  is a subgroup of  $\mathbb{Z}_d^\times$ .

In Lemma 1,  $G$  denotes a subgroup and the subscript specifies the original group. The proof of Lemma 1 is presented in Section 3.4. Here, for illustration, we give an example.

**Example 3.** Consider a subgroup  $G_{15} = \{1, 11\}$  of  $\mathbb{Z}_{15}^\times$ . We have  $G_{15} \bmod 3 = \{1, 2\}$  so that  $G_3 = \{1, 2\}$ , which is a subgroup of (in fact, equal to)  $\mathbb{Z}_3^\times = \{1, 2\}$ .  $G_{15} \bmod 5 = \bar{\{1, 1\}}$ , which is two copies of  $G_5 = \{1\}$ , and  $G_5$  is a (trivial) subgroup of  $\mathbb{Z}_5^\times = \{1, 2, 3, 4\}$ .

Consider another subgroup  $G_{15} = \{1, 4, 11, 14\}$  of  $\mathbb{Z}_{15}^\times$ .  $G_{15} \bmod 3 = \bar{\{1, 1, 2, 2\}}$ , which is two copies of  $G_3 = \{1, 2\} = \mathbb{Z}_3^\times$ .  $G_{15} \bmod 5 = \bar{\{1, 4, 1, 4\}}$ , which is two copies of  $G_5 = \{1, 4\}$  and  $G_5$  is a subgroup of  $\mathbb{Z}_5^\times = \{1, 2, 3, 4\}$ .

Consider  $G_{15} = \mathbb{Z}_{15}^\times = \{1, 2, 4, 7, 8, 11, 13, 14\}$ .  $G_{15} \bmod 3 = \bar{\{1, 2, 1, 1, 2, 2, 1, 2\}}$ , which is 4 copies of  $G_3 = \{1, 2\} = \mathbb{Z}_3^\times$ .  $G_{15} \bmod 5 = \bar{\{1, 2, 4, 2, 3, 1, 3, 4\}}$ , which is 2 copies of  $G_5 = \{1, 2, 3, 4\} = \mathbb{Z}_5^\times$ .

Given a subgroup  $G_d$  of the group  $\mathbb{Z}_d^\times$ , we may partition  $\mathbb{Z}_d^\times$  into cosets (see, e.g., Proposition 4 in Chapter 3 of [23] or Theorem 6.2 of [28]). Setting  $d$  as  $n/d$ , we have that  $\mathbb{Z}_{n/d}^\times$  may be partitioned into cosets with  $G_{n/d}$ . Combining with (21), i.e.,  $\mathbb{Z}_n^{(d)} = d \times \mathbb{Z}_{n/d}^\times$ , we may partition  $\mathbb{Z}_n^{(d)}$  into cosets with  $G_{n/d}$ . This partition is denoted by  $\mathbb{Z}_n^{(d)} / G_{n/d}$ .



**Example 4.** Continuing from Example 3, consider a subgroup  $G_{15} = \{1, 11\}$  of  $\mathbb{Z}_{15}^\times$ . Then,

$$\mathbb{Z}_{15}^\times / G_{15} = \{1, 11\} \cup \{2, 7\} \cup \{4, 14\} \cup \{8, 13\} \quad (23)$$

where the partition is obtained from the cosets, e.g.,  $\{2, 7\} = 2 \times \{1, 11\} = 7 \times \{1, 11\}$  is a coset of  $G_{15}$  with representative  $2 \in \mathbb{Z}_{15}^\times$  or  $7 \in \mathbb{Z}_{15}^\times$ . Similarly, when  $G_{15} = \{1, 11\}$ , from Example 3 we have  $G_3 = \{1, 2\}$ ,  $G_5 = \{1\}$  and the partitions  $\mathbb{Z}_{15}^{(5)} / G_3, \mathbb{Z}_{15}^{(3)} / G_5$  are as follows:

$$\begin{aligned} \mathbb{Z}_{15}^{(5)} / G_3 &= 5 \times \{1, 2\}, \\ \mathbb{Z}_{15}^{(3)} / G_5 &= 3 \times \{\{1\} \cup \{2\} \cup \{3\} \cup \{4\}\}. \end{aligned} \quad (24)$$

For another choice of  $G_{15}$  (again from Example 3), consider  $G_{15} = \{1, 4, 11, 14\}$  of  $\mathbb{Z}_{15}^\times$ . Then, from Example 3,  $G_3 = \{1, 2\}$ ,  $G_5 = \{1, 4\}$ . The partitions are

$$\begin{aligned} d = 1 : \mathbb{Z}_{15}^\times / G_{15} &= \{1, 4, 11, 14\} \cup \{2, 7, 8, 13\}, \\ d = 3 : \mathbb{Z}_{15}^{(3)} / G_5 &= 3 \times \{\{1, 4\} \cup \{2, 3\}\}, \\ d = 5 : \mathbb{Z}_{15}^{(5)} / G_3 &= 5 \times \{1, 2\}. \end{aligned} \quad (25)$$

For the final choice of  $G_{15} = \mathbb{Z}_{15}^\times$  from Example 3, we have  $G_3 = \mathbb{Z}_3^\times$ ,  $G_5 = \mathbb{Z}_5^\times$  and the partitions are trivial -  $\mathbb{Z}_{15}^\times / G_{15} = \mathbb{Z}_{15}^\times$ ,  $\mathbb{Z}_{15}^{(3)} / G_5 = 3 \times \mathbb{Z}_5^\times$ , and  $\mathbb{Z}_{15}^{(5)} / G_3 = 5 \times \mathbb{Z}_3^\times$ .

The collection of the cosets  $\mathbb{Z}_n^{(d)} / G_{n/d}$  for all proper divisors  $d$  is a feasible choice of the confusable sets. This result is stated in the following theorem.

**Theorem 3.** For  $\mathbb{Z}_n$ , the confusable sets and the randomizer can be chosen as follows. Consider the set of all proper divisors of  $n$ ,  $\{d_1 = 1, d_2, \dots, d_b\}$  and an arbitrary subgroup  $G_n$  of  $\mathbb{Z}_n^\times$ .

$$\gamma \text{ is uniform over } \mathcal{S}^* = G_n, \quad (26)$$

$$\mathbb{Z}_n = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S}_2 \cup \dots = \{0\} \cup \mathbb{Z}_n^\times / G_n \cup \mathbb{Z}_n^{(d_2)} / G_{n/d_2} \cup \dots \cup \mathbb{Z}_n^{(d_b)} / G_{n/d_b}. \quad (27)$$

Before presenting the proof of Theorem 3, we first give an example to illustrate its meaning.

**Example 5.** Continuing from Example 4, consider a subgroup  $G_{15} = \{1, 11\}$  of  $\mathbb{Z}_{15}^\times$ . Then, from Theorem 3, the confusable sets are

$$\mathbb{Z}_{15} = \{0\} \cup \mathbb{Z}_{15}^\times / G_{15} \cup \mathbb{Z}_{15}^{(3)} / G_5 \cup \mathbb{Z}_{15}^{(5)} / G_3 \quad (28)$$

$$\stackrel{(23)(24)}{=} \{0\} \cup \{1, 11\} \cup \{2, 7\} \cup \{4, 14\} \cup \{8, 13\} \cup \{3\} \cup \{6\} \cup \{9\} \cup \{12\} \cup \{5, 10\}. \quad (29)$$

For each of the confusable set above, it is easy to verify that when an element is multiplied with  $\gamma$  (uniform over  $G_{15}$ ), the result is uniform over the confusable set.

$$\{2, 7\} = \gamma \times 2 = \gamma \times 7, \{5, 10\} = \gamma \times 5 = \gamma \times 10, \gamma \times 3 = \{1, 11\} \times 3 = \{3, 3\}. \quad (30)$$

For another example, consider  $G_{15} = \{1, 4, 11, 14\}$ . The confusable sets are

$$\mathbb{Z}_{15} = \{0\} \cup \mathbb{Z}_{15}^\times / G_{15} \cup \mathbb{Z}_{15}^{(3)} / G_5 \cup \mathbb{Z}_{15}^{(5)} / G_3 \quad (31)$$

$$\stackrel{(25)}{=} \{0\} \cup \{1, 4, 11, 14\} \cup \{2, 7, 8, 13\} \cup \{3, 12\} \cup \{6, 9\} \cup \{5, 10\}. \quad (32)$$

Let us also verify that the uniform property holds.  $\gamma$  is over  $G_{15} = \{1, 4, 11, 14\}$ . For example, consider  $7 \in \{2, 7, 8, 13\}$ , then we have  $\gamma \times 7 = \{7, 28, 77, 98\} \bmod 15 = \{7, 13, 2, 8\}$ . Consider

$12 \in \{3, 12\}$ , then we have  $\gamma \times 12 = \{12, 48, 132, 168\} \bmod 15 = \{12, 3, 12, 3\}$ , which is 2 copies of  $\{3, 12\}$ .

Finally, consider  $G_{15} = \mathbb{Z}_{15}^\times$ . The confusable sets are  $\mathbb{Z}_{15} = \{0\} \cup \mathbb{Z}_{15}^\times \cup \mathbb{Z}_{15}^{(3)} \cup \mathbb{Z}_{15}^{(5)}$ . For any element in  $\mathbb{Z}_{15}^{(3)}$ , say 6, we have  $\gamma \times 6 = \mathbb{Z}_{15}^\times \times 6 = \{6, 12, 9, 12, 3, 6, 3, 9\}$ .

**Proof of Theorem 3.** The proof relies on Lemma 1 and the property of cosets. First, the confusable sets form a partition of  $\mathbb{Z}_n$ . Second, we verify the uniform property, i.e.,  $\forall s \in \mathcal{S}_i$ ,  $\gamma \times s$  is uniform over  $\mathcal{S}_i$ . Consider any  $\mathcal{S}_i$ , e.g., a set from  $\mathbb{Z}_n^{(d_i)} / G_{n/d_i}$ ,  $i \in \{1, \dots, b\}$ . From the construction of  $\mathcal{S}_i$ , we have  $\mathcal{S}_i \times 1/d_i$  is a coset of  $G_{n/d_i}$  in  $\mathbb{Z}_{n/d_i}^\times$ . By the definition of cosets and the fact that  $s/d_i \in \mathcal{S}_i \times 1/d_i$ , we have

$$\mathcal{S}_i \times 1/d_i = G_{n/d_i} \times s/d_i. \quad (33)$$

Next, consider

$$(G_n \times s/d_i) \bmod n/d_i = ((G_n \bmod n/d_i) \times s/d_i) \bmod n/d_i \quad (34)$$

$$\stackrel{\text{Lemma 1}}{=} (\{\bar{G}_{n/d_i}, \dots, G_{n/d_i}\} \times s/d_i) \bmod n/d_i \quad (35)$$

$$\stackrel{(33)}{=} \{\bar{\mathcal{S}}_i \times 1/d_i, \dots, \mathcal{S}_i \times 1/d_i\} \bmod n/d_i \quad (36)$$

$$\Rightarrow \gamma \times s = G_n \times s = d_i \times (G_n \times s/d_i) \quad (37)$$

$$\stackrel{(36)}{=} \{\bar{\mathcal{S}}_i, \dots, \mathcal{S}_i\}. \quad (38)$$

Therefore,  $\gamma \times s$  is uniform over  $\mathcal{S}_i$ . The proof is complete.  $\square$

**Remark 2.** From Theorem 3, we see that any subgroup of  $\mathbb{Z}_n^\times$  can induce a feasible choice of the confusable sets and the randomizer. We list all possible confusable sets for  $\mathbb{Z}_n$ ,  $n < 20$  in Figure A2 (see the Appendix B). We also include in the Appendix C some discussion on the structures of the subgroups of  $\mathbb{Z}_n^\times$ , based on existing group theory and number theory results.

### 3.3. Converse

One of the challenges to understand the optimality of secure computation codes is the lack of converse results. In information theory, converse results are statements of impossibility claims and are used to prove optimality. As a starting point, we compare our achievable scheme with existing converse results with no security constraint (i.e., the pure computation problem). Interestingly, when the size of the underlying field or ring is the same as the input size, the scheme in Theorem 1 achieves the information theoretically optimal rate region. Without loss of generality, for secure computation problems, we assume there are no identical rows or columns in the function table (as Carol cannot learn anything about the exact row or column index of such identical rows and columns).

**Proposition 1.** Consider independent and uniform inputs, i.e.,  $W_1, W_2$  are independent and uniform over  $\{0, 1, \dots, m-1\}$ . For a function  $f(W_1, W_2)$ , if a feasible expanded function exists over  $\mathbb{F}_q$  or  $\mathbb{Z}_n$  where  $q = m$  or  $n = m$ , then the scheme in Theorem 1 is information theoretically optimal.

Achievability directly follows from Theorem 1 and converse ( $H(X_1), H(X_2) \geq \log_2 m$ ) follows from a simple observation that when there is no security constraint, Alice (Bob) needs to tell Carol the exact value of  $W_1$  ( $W_2$ ). The reason is that otherwise two  $W_1(W_2)$  will be mapped to the same codeword  $X_1(X_2)$  and  $f(W_1, W_2)$  has no identical rows or columns such that some value of  $f(W_1, W_2)$  cannot be decoded correctly. This (and more general) result has been proved in several different contexts in the literature, see, e.g., the classical function computation of correlated sources work by Han and Kobayashi [29] (Lemma 1) and the recent generalization [30], the computation over multiple access channel

work [31] (Lemma 1), and the network coding for computing work [32,33]. Note that the converse holds for block inputs as well, where the rate is defined as the number of bits in the codeword per input symbol. As eliminating the security constraint cannot help, the same converse holds for the secure computation problem as well.

Note that Proposition 1 characterizes the optimal rate region for a class of secure computation problems (which contain infinite instances). One could start from the confusable sets of  $\mathbb{F}_q$  or  $\mathbb{Z}_n$  and invert them into a function  $f(W_1, W_2)$  with input size  $m = q$  or  $n$ . Functions constructed from this method satisfy Proposition 1 and thus we obtain the optimal rate region.

To the best of our knowledge, the only existing information theoretic converse results for the secure computation problem are the ones obtained in [6], whose expression involves common information terms and an optimization over a class of distributions so that the exact bound needs to be evaluated for each individual instance and is generally not trivial to compute. Interestingly, for some small instances, we find that our achievable scheme is information theoretically optimal (see Remark 3 of Example 6 and Remark 4 of Example 7). For most cases, however, there is a gap in the rate region between the achievable scheme in Theorem 1 and the converse results from in [6], while it is not clear if and by how much the scheme and the converse can be improved. The model considered in [6] is the general secure computation problem that allows interactive multi-round protocols. Therefore, the converse results therein might be generally too strong for the minimal secure computation problem. We note that there are instances where we know better schemes than that in Theorem 1 (see Examples 9 and 10 in the discussion section).

### 3.4. Proof of Lemma 1

The proof of Lemma 1 consists of two parts.

First, we show that the set of elements of  $G_n \bmod d$ ,  $G_d$ , forms a subgroup of  $\mathbb{Z}_d^\times$ . This is proved by two claims: (1)  $G_d \subset \mathbb{Z}_d^\times$  and (2)  $G_d$  is closed under multiplication modulo  $d$ . Note that for finite groups, the verification of subgroups only requires the check of the closure property (i.e., associativity and the existence of identity and inverse elements are automatically guaranteed. Refer to Proposition 1 in Chapter 2 of [23]).

For (1), note that any element  $g$  of  $G_n$  belongs to  $\mathbb{Z}_n^\times$ , so  $\gcd(g, n) = 1$ . As  $d$  is a divisor of  $n$ , we have  $\gcd(g, d) = 1$  and  $\gcd(g \bmod d, d) = 1$ . Thus,  $g \bmod d$  of  $G_d$  belongs to  $\mathbb{Z}_d^\times$  and  $G_d \subset \mathbb{Z}_d^\times$ .

For (2), consider any two elements of  $G_d$ , e.g.,  $g_1, g_2 \in G_n$  and  $g_1 \bmod d, g_2 \bmod d \in G_d$ . As  $G_n$  forms a group, we have for some  $g_3 \in G_n$ ,  $(g_1 \times g_2) \bmod n = g_3$ , i.e.,  $g_1 \times g_2 = k \times n + g_3$  for some integer  $k$ . Then,

$$\left( (g_1 \bmod d) \times (g_2 \bmod d) \right) \bmod d = (g_1 \times g_2) \bmod d \quad (39)$$

$$= (k \times n + g_3) \bmod d \quad (40)$$

$$= g_3 \bmod d \quad (d \text{ is a divisor of } n) \quad (41)$$

$$\in G_d \quad (42)$$

Therefore  $G_d$  is closed under multiplication.

Second, we show that in the multiset  $G_n \bmod d$ , each element of  $G_d$  appears for the same number of times. Denote  $G_n = \{g_1, g_2, \dots, g_T\}$ . As  $G_n$  is a subgroup of  $\mathbb{Z}_n^\times$ , we have

$$\forall i \in \{1, \dots, T\}, (G_n \times g_i) \bmod n = \{g_1 \times g_i, g_2 \times g_i, \dots, g_T \times g_i\} \bmod n = G_n. \quad (43)$$

Denote the multiset  $\bar{G}_d = G_n \bmod d = \{\bar{h}_1, \dots, \bar{h}_1, \bar{h}_2, \dots, \bar{h}_Q\}$ , where  $h_q, q \in \{1, \dots, Q\}$  appears  $|h_q|$  times and  $\forall q_1 \neq q_2, h_{q_1} \neq h_{q_2}$ . Assume without loss of generality that  $|h_1| \geq |h_2| \geq \dots \geq |h_Q|$ . We need to show that  $|h_1| = |h_Q|$ . This proof is presented next.

From the first part of the proof, we know that  $G_d = \{h_1, h_2, \dots, h_Q\}$  is a subgroup of  $\mathbb{Z}_d^\times$ . Applying (43) to  $G_d$  and  $\mathbb{Z}_d^\times$ , we have

$$\forall j \in \{1, \dots, Q\}, (G_d \times h_j) \bmod d = \{h_1 \times h_j, h_2 \times h_j, \dots, h_Q \times h_j\} \bmod d = G_d. \quad (44)$$

Further, setting  $j = 1$  in (44), we have

$$(G_d \times h_1) \bmod d = \{h_1 \times h_1, h_2 \times h_1, \dots, h_Q \times h_1\} \bmod d = G_d = \{h_1, \dots, h_Q\}. \quad (45)$$

Note that multiplication mod  $d$  is commutative. Then, there exists  $j^* \in \{1, \dots, Q\}$  such that

$$(h_{j^*} \times h_1) \bmod d = (h_1 \times h_{j^*}) \bmod d = h_Q. \quad (46)$$

As  $h_{j^*} \in G_d$ , there exists  $i^* \in \{1, \dots, T\}$  such that

$$g_{i^*} \bmod d = h_{j^*}. \quad (47)$$

On the one hand,

$$\begin{aligned} (\bar{G}_d \times h_{j^*}) \bmod d &= \underbrace{\{h_1 \times h_{j^*}, \dots, h_1 \times h_{j^*}, h_2 \times h_{j^*}, \dots, h_Q \times h_{j^*}\}}_{|h_1| \text{ times}} \bmod d \\ &\stackrel{(46)(44)}{=} \underbrace{\{h_Q, \dots, h_Q, h_1, \dots, h_{Q-1}\}}_{|h_1| \text{ times}} \end{aligned} \quad (48)$$

On the other hand,

$$(\bar{G}_d \times h_{j^*}) \bmod d \stackrel{(47)}{=} (\bar{G}_d \times g_{i^*}) \bmod d = ((G_n \bmod d) \times g_{i^*}) \bmod d \quad (49)$$

$$= (G_n \times g_{i^*}) \bmod d \quad (50)$$

$$\stackrel{(43)}{=} G_n \bmod d \quad (51)$$

$$= \bar{G}_d = \{h_1, \dots, h_{Q-1}, \underbrace{h_Q, \dots, h_Q}_{|h_Q| \text{ times}}\} \quad (52)$$

Comparing (48) and (52) (i.e., the number of times that  $h_Q$  appears), we have proved that  $|h_1| = |h_Q|$ . The proof of the second part, and thus the proof of the lemma, are now complete.

#### 4. Generalization

In this section, we consider several generalizations of the coding scheme presented in the previous section, to illustrate how the insights generalize beyond the basic setting.

##### 4.1. Optimized Additive Randomness

In the coding scheme presented in Theorem 1, the additive common randomness  $z$  appeared in the codewords  $X_1, X_2$  is uniform over  $\mathbb{F}_q$  or  $\mathbb{Z}_n$  (refer to (7)), which is not necessary but a universal and convenient choice that works for all cases and admits a simple proof. We show, through the following example, that an optimized  $z$  (which does not have full-support over  $\mathbb{Z}_n$ ) might help to further reduce the communication cost.

**Example 6.** Consider the function  $f(W_1, W_2)$  shown in Figure 5, where a feasible expanded function over  $\mathbb{Z}_4$  is also depicted. The confusable sets are obtained from Theorem 3 using  $G_4 = \mathbb{Z}_4^\times = \{1, 3\}$ .

$$\mathbb{Z}_4^\times = \{0\} \cup \{1, 3\} \cup \{2\}, \gamma \text{ is uniform over } G_4 = \{1, 3\}. \quad (53)$$

From Theorem 1, Alice will send  $X_1 = \gamma \times \tilde{W}_1 + z$  and Bob will send  $X_2 = \gamma \times \tilde{W}_2 - z$  to Carol, where  $\gamma$  is uniform over  $\{1, 3\}$ ,  $z$  is uniform over  $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ , and  $\gamma, z$  are independent. That is, Alice and Bob each sends a symbols from  $\mathbb{Z}_4$  (i.e., 2 bits) to Carol.

Interestingly, if we choose  $z$  to be uniform over  $\{0, 2\}$  (instead of uniform over  $\{0, 1, 2, 3\}$ ), the scheme will also work. Correctness remains the same and for security, we only need  $(X_1, X_2)$  (or equivalently  $(X_1, X_1 + X_2)$ ) to be identically distributed when  $(W_1, W_2) \in \{(0, 0), (0, 1)\}$ . Note that

$$\begin{aligned} (W_1, W_2) = (0, 0) &\rightarrow (\tilde{W}_1, \tilde{W}_2) = (1, 0) \rightarrow (X_1, X_1 + X_2) = (\gamma + z, \gamma), \\ (W_1, W_2) = (0, 1) &\rightarrow (\tilde{W}_1, \tilde{W}_2) = (1, 2) \rightarrow (X_1, X_1 + X_2) = (\gamma + z, \gamma \times 3). \end{aligned} \quad (54)$$

Both  $(\gamma + z, \gamma)$  and  $(\gamma + z, \gamma \times 3)$  are uniform over  $\{(1, 1), (3, 1), (3, 3), (1, 3)\}$ . Therefore, the scheme satisfies the security constraint. Importantly, now  $X_2 = \gamma \times \tilde{W}_2 - z$  can only take value 0 or 2. Therefore, Bob only needs to send 1 bit (instead of 2 bits) to Carol.

$f$	0	1	Expand $\xrightarrow{\gamma}$ over $\mathbb{Z}_4$	$\tilde{W}_1 + \tilde{W}_2$	0	2	Randomize $\xrightarrow{\gamma}$ over $\mathbb{Z}_4$	$\gamma \times (\tilde{W}_1 + \tilde{W}_2)$	0	2
0	2	2		1	1	3		1	{1, 3}	{1, 3}
1	0	1		0	0	2		0	0	2

**Figure 5.** An expand-and-randomize secure computation code over  $\mathbb{Z}_4$ , where  $\gamma$  is uniform over  $\{1, 3\}$ .

**Remark 3.** If variable-length codes are allowed, then the above code can be further improved. Specifically, Alice does not need to distinguish whether  $X_1$  is 1 or 3, e.g., Alice may simply send 1 when  $X_1$  is 1 or 3 (this happens when  $\tilde{W}_1 = 1$ ). Interestingly, the rate region of this code coincides with an existing converse result from Theorem 9 in [6] for any joint distribution of  $(W_1, W_2)$  with full support. Thus, this improved code with optimized additive randomness and variable-length codewords turns out to be information theoretically optimal (i.e., even if block codes are allowed). We also note that an alternative optimal code construction based on a different idea is presented in [6] (see Algorithm 3).

For a general given function  $f(W_1, W_2)$ , to find the optimal choice of  $z$  of minimum randomness, we may list all identically distributed conditions in the security constraint (such as (54)) and solve for the  $z$  variable that satisfies all the constraints and has minimum entropy (a uniform full-support  $z$  will always work but has maximum entropy).

#### 4.2. $\epsilon$ -Error Schemes with Block Codes

Hitherto, we have focused exclusively on scalar codes and zero-error schemes that work for any joint distribution of  $(W_1, W_2)$ . In this subsection, we show how to use classical source coding techniques (specifically, structured linear codes, or Korner–Marton coding [34]) that exploit the specific distribution of  $(W_1, W_2)$  to improve the communication rate when long block codes and vanishing-error are allowed. This is explained through the following binary AND function example.

**Example 7.** Consider the binary AND function  $f(W_1, W_2) = W_1 \text{ AND } W_2$ , for which a feasible expanded function over  $\mathbb{F}_3$  is shown in Figure 6. The confusable sets are obtained from Theorem 2 using the primitive element  $g = 2$  of  $\mathbb{F}_3^\times$  and the divisor  $d = 1$ .

$$\mathbb{F}_3 = \{0\} \cup \{2^0, 2^1\} = \{0\} \cup \{1, 2\}, \quad \gamma \text{ is uniform over } \{1, 2\}. \quad (55)$$

Then, from Theorem 1, we set  $X_1 = \gamma \times \tilde{W}_1 + z$ ,  $X_2 = \gamma \times \tilde{W}_2 - z$  so that it suffices to send a symbol from  $\mathbb{F}_3$  (i.e.,  $\log_2 3$  bits) each from Alice and Bob to Carol. In other words, the rate tuple  $(\log_2 3, \log_2 3)$  is achievable. As mentioned in the introduction, this zero-error scalar code first appeared in Appendix B of [12].

We note that for correct decoding, Carol will compute  $X_1 + X_2 = \gamma \times (\tilde{W}_1 + \tilde{W}_2)$ , denoted by  $U$ . As our goal is only to recover  $U$  (securely of course), the amount of information required is simply the entropy of  $U$  (which is smaller than  $\log_2 3$  bits as long as it is not uniform). The only caveat is that encoding is done in a distributed manner at Alice and Bob respectively, so we just need to compress  $U$  with a linear code such that it is compatible with the decoding procedure of  $X_1 + X_2$ . Fortunately, this distributed source compression for sum computation problem has been studied in network information theory. In particular, structured linear codes apply and we will use (the secure version of) Korner–Marton coding [34].

The improvement of the communication rate comes from the observation that in our proposed code, we consider the worst case, i.e.,  $U$  is not compressed and a symbol from  $\mathbb{F}_3$  is sent to represent  $X_1$  regardless of the distribution of  $U$ , the variable we wish to recover. When  $U$  is not uniform, further compression over long blocks is possible. As a simple example, suppose  $W_1$  and  $W_2$  are two independent uniform binary variables. As a result,  $U = X_1 + X_2 = \gamma \times (\tilde{W}_1 + \tilde{W}_2)$  is 0 with probability  $1/4$ , is 1 with probability  $3/8$ , and is 2 with probability  $3/8$  (see Figure 6) and the entropy of  $U$  is

$$H(U) = H\left(\frac{1}{4}, \frac{3}{8}, \frac{3}{8}\right) = \frac{1}{4}(11 - 3\log_2 3) < \log_2 3.$$

Next, we outline how to use structured linear source codes to achieve the rate tuple  $(R_1, R_2) = (H(U) \log_2 3, H(U) \log_2 3)$  bits per input symbol over long block-length with vanishing probability of error. Consider  $L$ -length extension of the two inputs  $W_1, W_2$ , denoted by  $\tilde{W}_1, \tilde{W}_2$ , i.e.,  $\tilde{W}_1, \tilde{W}_2$  are two sequences of i.i.d. uniform bits of length  $L$ . A similar vector notation is used for  $L$ -length extensions of other variables, e.g.,  $\vec{z}$  represents a length  $L$  sequence of i.i.d. uniform symbols over  $\mathbb{F}_3$ . We apply our proposed scheme to each bit of the input sequence and then multiply (over  $\mathbb{F}_3$ ) the vector codeword with a matrix  $A$  of size  $(H(U) + \epsilon)L \times L$ .

$$\begin{aligned} \vec{X}_1 &= A_{(H(U)+\epsilon)L \times L} \cdot (\underbrace{\tilde{\gamma}_{L \times 1} \times \tilde{W}_1}_{L \times 1} + \vec{z}_{L \times 1}), \quad \vec{X}_2 = A_{(H(U)+\epsilon)L \times L} \cdot (\tilde{\gamma} \times \tilde{W}_2 - \vec{z}) \\ \Rightarrow \vec{X}_1 + \vec{X}_2 &= A_{(H(U)+\epsilon)L \times L} \cdot \underbrace{(\tilde{\gamma} \times (\tilde{W}_1 + \tilde{W}_2))}_{\triangleq \vec{U}} = A_{(H(U)+\epsilon)L \times L} \cdot \vec{U} \end{aligned}$$

where the ‘+’ and ‘×’ operators are symbol-wise, and the ‘·’ operator is the matrix multiplication operator. Further optimizations of the common randomness consumption are possible, i.e., the same randomizer can be used for each input bit and it suffices to use an additive common randomness variable with entropy  $LH(U) \log_2 3$  bits (instead of  $L \log_2 3$  bits). Note that the same matrix  $A$  must be used by both Alice and Bob. We need to ensure that from  $A_{(H(U)+\epsilon)L \times L} \cdot \vec{U}$ , we can recover  $\vec{U}$ . In other words, we now have the well-known point-to-point source coding problem with a linear compressor. Thus, there exists a deterministic matrix  $A$  of size  $(H(U) + \epsilon)L \times L$  such that we can recover  $\vec{U}$  from  $A \cdot \vec{U}$  with  $\epsilon$  probability of error and  $\epsilon \rightarrow 0$  when  $L \rightarrow \infty$ . Specifically, a random generation of  $A$  (i.e., choosing each element of  $A$  independently and uniformly over  $\mathbb{F}_3$ ) will work with high probability. The structured linear coding technique has appeared in the literature many times, e.g., it was introduced by Elias in the context of channel coding over a binary symmetric channel [35], was used by Wyner in the context of distributed source coding of binary sources (the Slepian–Wolf problem, see Section VI. C of [36]), was used by Korner and Marton in the context of encoding module-two sum of binary sources [34], and generalizations to finite fields are immediate (see, e.g., in [37] and Remark 10.2 of [38]).

The security constraint is easily verified. The scalar code is secure by Theorem 1. Then independent application of the scalar code to  $L$ -length extensions is also secure. Multiplying with a deterministic matrix  $A$  will not leak any information.

Therefore, the optimized block code has vanishing probability of error and is secure. The achieved rate tuple is  $(H(U) \log_2 3, H(U) \log_2 3)$  bits for the codewords per input bit.

Finally, we note that the idea of using Korner–Marton coding for secure computation is not new, e.g., it has been applied to secure sum computations [6,39]. While our objective is not sum



computation, Korner–Marton coding still applies because the decoding procedure  $X_1 + X_2$  relies on a linear operation.

$$\begin{array}{c|cc} f & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array} \xrightarrow[\text{over } \mathbb{F}_3]{\text{Expand}} \begin{array}{c|cc} \tilde{W}_1 + \tilde{W}_2 & 1 & 2 \\ \hline 0 & 1 & 2 \\ 1 & 2 & 0 \end{array} \xrightarrow[\text{over } \mathbb{F}_3]{\text{Randomize}} \begin{array}{c|cc} \gamma \times (\tilde{W}_1 + \tilde{W}_2) & 1 & 2 \\ \hline 0 & \{1, 2\} & \{1, 2\} \\ 1 & \{1, 2\} & 0 \end{array}$$

**Figure 6.** An expand-and-randomize secure computation code (for the binary AND function) over  $\mathbb{F}_3$ , where  $\gamma$  is uniform over  $\{1, 2\}$ .

**Remark 4.** Interestingly, the zero-error code presented above for AND computation is information theoretically optimal in terms of the communication rate (refer to Theorem 11 of [6]). That is, communicating  $\log_2 3$  bits per input bit from Alice and Bob each to Carol is the minimum possible. Note that as  $\epsilon$ -error codes achieve an improved rate performance than the best of that of zero-error codes, we know that for secure computation problems,  $\epsilon$ -error capacity may be different from zero-error capacity (this fact has been established in prior work [6,39]). We also note that when  $\epsilon$ -error is allowed, the optimal rate region for AND computation remains open.

The above linear compression technique applies to all secure computation codes over  $\mathbb{F}_q$  (refer to Theorem 2), i.e., instead of sending a symbol from  $\mathbb{F}_q$ , we may compress it to  $H(U) \log_2 q$  bits. However, we note that the same result does not hold for codes over  $\mathbb{Z}_n$  when  $n$  is not a prime. While the same linear compression technique can be applied, the rate performance is not known. That is, it is not known how large the matrix  $A$  needs to be, if we wish to recover  $\tilde{U}$  from  $A \cdot \tilde{U}$ . In particular,  $H(U)$  may not suffice (we need to understand more on the open problem of source coding with restricted encoding structures, e.g., modular arithmetic. For related results on source coding with group codes, see, e.g., in [40,41] and references therein).

#### 4.3. Equality Function with Non-Prime-Power Inputs

In this subsection, we continue the discussion on the equality function in the introduction. We consider the equation function with arbitrary input size, i.e.,  $W_1, W_2 \in \{0, 1, \dots, m-1\}$  for an arbitrary integer  $m$  and wish to securely compute if  $W_1$  is equal to  $W_2$ . The approach taken in the introduction works when  $m$  is a prime power  $p^n$  such that there exist an invertible mapping between  $\{0, 1, \dots, p^n-1\}$  and the elements from the finite field  $\mathbb{F}_{p^n}$  (say  $W_1(W_2)$  is mapped to  $\tilde{W}_1(\tilde{W}_2)$ ). Then,  $\tilde{W}_1 - \tilde{W}_2$  is a feasible expanded function where the zero element and the non-zero elements are two confusable sets. Now, what if  $m$  is not a prime power? We may increase  $m$  to a prime power and then use the previous approach. This approach will require that Alice and Bob each sends a symbol of size larger than  $\log_2 m$  bits. Interestingly, we show that  $\log_2 m$  bits are always sufficient for any  $m$  (no matter whether  $m$  is a prime power or not). To this end, we need to use a variant of the expand-and-randomize scheme from Theorem 1. To illustrate the idea, in the following we consider the simplest example where  $m$  is not a prime power, i.e.,  $m = 6$ .

**Example 8.** Consider  $W_1, W_2 \in \{0, 1, \dots, 5\}$ .  $f(W_1, W_2) = \text{Yes}$  if  $W_1$  is equal to  $W_2$  and otherwise  $f(W_1, W_2) = \text{No}$ . While 6 is not a prime power, we may decompose it into products of prime powers, i.e.,  $6 = 2 \times 3$ . The following scheme works by using a product of our expand-and-randomize schemes over decomposed domains with uniformly permuted inputs, i.e., using two schemes of equality function with prime inputs in parallel.

Alice and Bob share a common random variable  $Z = (\pi, \gamma_1, \gamma_2, z_1, z_2)$ , where all the random variables are independent and uniform,  $\pi$  is from the set of all possible permutations with 6

elements,  $\gamma_1$  is from  $\mathbb{F}_2^\times = \{1\}$ ,  $z_1$  is from  $\mathbb{F}_2 = \{0, 1\}$ ,  $\gamma_2$  is from  $\mathbb{F}_3^\times = \{1, 2\}$ , and  $z_2$  is from  $\mathbb{F}_3 = \{0, 1, 2\}$ . The codewords are

$$\begin{aligned} X_1 &= (a_1, a_2), \text{ where } a_1 \stackrel{\mathbb{F}_2}{=} \gamma_1 \times (\pi(W_1) \bmod 2) + z_1, a_2 \stackrel{\mathbb{F}_3}{=} \gamma_2 \times (\pi(W_1) \bmod 3) + z_2; \\ X_2 &= (b_1, b_2), \text{ where } b_1 \stackrel{\mathbb{F}_2}{=} \gamma_1 \times (\pi(W_2) \bmod 2) + z_1, b_2 \stackrel{\mathbb{F}_3}{=} \gamma_2 \times (\pi(W_2) \bmod 3) + z_2. \end{aligned}$$

The '+' and '×' operations in computing  $a_1, b_1(a_2, b_2)$  are over  $\mathbb{F}_2(\mathbb{F}_3)$ . Note that the same permutation  $\pi$  is applied to  $W_1$  and  $W_2$ . The decoding rule of Carol is as follows.

$$\text{Carol claims } W_1 \text{ is equal to } W_2 \text{ if and only if } (a_1, a_2) = (b_1, b_2). \quad (56)$$

We have zero error because  $\pi(W_1) = \pi(W_2)$  if and only if  $(\pi(W_1) \bmod 2, \pi(W_1) \bmod 3) = (\pi(W_2) \bmod 2, \pi(W_2) \bmod 3)$  (this result is typically referred to as the Chinese Remainder Theorem). Next, we verify that the security constraint is satisfied, i.e., when  $W_1$  is not equal to  $W_2$ ,  $X_1, X_2$  are identically distributed. Consider any  $(W_1, W_2)$  such that  $W_1 \neq W_2$  and  $\pi(W_1) \neq \pi(W_2)$ . Note that  $(X_1, X_2)$  is invertible to  $(a_1, a_2, b_1 - a_1, b_2 - a_2)$ , and the 3 variables  $a_1, a_2, (b_1 - a_1, b_2 - a_2)$  are independent. Further,  $a_1$  is uniform over  $\{0, 1\}$ ,  $a_2$  is uniform over  $\{0, 1, 2\}$ , and  $(\pi(W_1), \pi(W_2))$  is uniform over  $(i, j), i \neq j, i, j \in \{0, 1, 2, 3, 4, 5\}$  so that

$$\Pr(\pi(W_1) \bmod 2 = \pi(W_2) \bmod 2, \pi(W_1) \bmod 3 \neq \pi(W_2) \bmod 3) = 2/5, \quad (57)$$

$$\Pr(\pi(W_1) \bmod 2 \neq \pi(W_2) \bmod 2, \pi(W_1) \bmod 3 = \pi(W_2) \bmod 3) = 1/5, \quad (58)$$

$$\Pr(\pi(W_1) \bmod 2 \neq \pi(W_2) \bmod 2, \pi(W_1) \bmod 3 \neq \pi(W_2) \bmod 3) = 2/5 \quad (59)$$

$$\Rightarrow (b_1 - a_1, b_2 - a_2) \text{ is uniform over } \{(0, 1), (0, 2), (1, 0), (1, 1), (1, 2)\}. \quad (60)$$

Therefore, security is guaranteed and sending  $1 + \log_2 3 = \log_2 6$  bits from Alice and Bob to Carol each is sufficient.

**Remark 5.** The above scheme generalizes in a straightforward manner to any integer  $m$  using a prime-power decomposition of  $m$  (this result is typically referred to as the fundamental theorem of arithmetic). As the achieved communication rate  $\log_2 m$  bits for Alice and Bob each matches the optimal rate for independent and uniform inputs with no security constraint (see Proposition 1), the above secure computation scheme achieves the information theoretically optimal rate region.

## 5. Discussion

We introduced the expand-and-randomize scheme for the secure computation problem and implement it over the finite field and the ring of integers modulo  $n$ . We characterized the algebraic structures of the feasible expanded functions through the notion of confusable sets. We find it interesting that while we consider only information theoretic security, the tools invoked from algebra and number theory arise frequently and lie in the core of cryptography under computational security (see textbooks, e.g., in [42,43]). The proposed scheme is very efficient and sometimes optimal when the original function is (close to) an isomorphism of such confusable sets. In particular, the information theoretically optimal communication cost is characterized for a new class of functions that include an infinite number of instances (refer to Proposition 1). However, we are also aware of functions where there exist better schemes such that our scheme is strictly sub-optimal. In the following, we present two such examples to expose more diverse insights for the challenging open problem of minimal secure computation.

### Sub-Optimal Examples

**Example 9.** Consider the function shown in Figure 7. A feasible expanded function over  $\mathbb{F}_7$  is also depicted. The confusable sets are obtained from Theorem 2 using the primitive element  $g = 3$  of  $\mathbb{F}_7^\times$  and the divisor  $d = 2$ .

$$\mathbb{F}_7 = \{0\} \cup \{3^0, 3^2, 3^4\} \cup \{3^1, 3^3, 3^5\} = \{0\} \cup \{1, 2, 4\} \cup \{3, 5, 6\}, \gamma \text{ is uniform over } \{1, 2, 4\}.$$

Therefore, according to Theorem 1, it suffices to send a symbol from  $\mathbb{F}_7$  (i.e.,  $\log_2(7)$  bits) each from Alice and Bob to Carol. In other words, the rate tuple  $(\log_2(7), \log_2(7))$  is achievable.

However, an improved rate tuple  $(2, 2)$  is achievable using a different coding scheme. Specifically, we use the coding scheme from Section 2 of [12]. The scheme (when applied to this example) is described as follows. The common random variable shared is  $Z = (\pi, z_1, z_2)$ , where  $\pi, z_1, z_2$  are independent and uniform binary random variables. The codewords are

$$X_1 = \begin{cases} (1, z_1) & \text{when } \pi = 0, W_1 = 0 \\ (2, z_2) & \text{when } \pi = 0, W_1 = 1 \\ (2, z_1) & \text{when } \pi = 1, W_1 = 0 \\ (1, z_2) & \text{when } \pi = 1, W_1 = 1 \end{cases}, X_2 = \begin{cases} (f(0, W_2) + z_1, f(1, W_2) + z_2) & \text{when } \pi = 0 \\ (f(1, W_2) + z_2, f(0, W_2) + z_1) & \text{when } \pi = 1 \end{cases}$$

where  $X_1$  and  $X_2$  each contains 2 bits. The coding idea is that the first (second) row of the function table is protected by the uniform noise  $z_1$  ( $z_2$ ). Bob does not know the value of  $W_1$ , so he will send both  $f(0, W_2)$  and  $f(1, W_2)$  (after masked by  $z_i$ ) in a random order. Alice will use the first element of  $X_1$  to indicate if the first or the second element of  $X_2$  contains the desired function and use the second element to carry the noise that masks the desired function output. Therefore, the decoding rule is as follows. Denote  $X_1 = (a_1, a_2)$  and  $X_2 = (b_1, b_2)$ . Carol can recover  $f(W_1, W_2)$  with no error from  $b_{a_1} - a_2$ . Security is guaranteed by the observation that Carol only knows the noise that masks the desired function while obtains nothing else. Therefore in this scheme, sending  $2 < \log_2 7$  bits each by Alice and Bob is sufficient.

$f$	0	1	2	$\xrightarrow[\text{over } \mathbb{F}_7]{\text{Expand}}$	$\tilde{W}_1 + \tilde{W}_2$	1	2	3	$\xrightarrow[\text{over } \mathbb{F}_7]{\text{Randomize}}$	$\gamma \times (\tilde{W}_1 + \tilde{W}_2)$	1	2	3
	0	0	1		0	1	2	3		0	$\{1, 2, 4\}$	$\{1, 2, 4\}$	$\{3, 5, 6\}$
	1	0	1		3	4	5	6		3	$\{1, 2, 4\}$	$\{3, 5, 6\}$	$\{3, 5, 6\}$

**Figure 7.** An expand-and-randomize secure computation code over  $\mathbb{F}_7$ , where  $\gamma$  is uniform over  $\{1, 2, 4\}$ .

**Example 10.** Consider the function shown in Figure 8. A feasible expanded function over  $\mathbb{Z}_8$  is also depicted. The confusable sets are obtained from Theorem 3 using the subgroup  $G_8 = \{1, 3\}$  of  $\mathbb{Z}_8^\times = \{1, 3, 5, 7\}$ .

$$\mathbb{Z}_8 = \{0\} \cup \{1, 3\} \cup \{5, 7\} \cup \{2, 6\} \cup \{4\}, \gamma \text{ is uniform over } G_8 = \{1, 3\}. \quad (61)$$

Therefore, according to Theorem 1, it suffices to send a symbol from  $\mathbb{Z}_8$  (i.e., 3 bits) each from Alice and Bob to Carol. In other words, the rate tuple  $(3, 3)$  is achievable.

However, an improved rate tuple is achievable using a different coding scheme. To see this, we assume that  $W_1, W_2$  are independent and each of them is uniform over its support (note that the scheme above does not depend on the joint distribution of  $(W_1, W_2)$ ). We now describe a scheme that achieves the rate tuple  $(2, \log_2 3)$ , which is strictly better than  $(3, 3)$ . This scheme is inspired by Algorithm 3 of [6], which is for the function in Example 6 and we generalize it to the function in Figure 8. Alice and Bob share the common random variable  $Z = (z, z')$ , where  $z, z'$  are independent uniform binary random variables. The codewords sent are

$$X_1 = \begin{cases} W_1, z' & \text{when } W_1 = 0 \\ W_1, z & \text{when } W_1 = 1 \end{cases}, X_2 = \begin{cases} (W_2 + z)_{\mathbb{F}_2} & \text{when } W_2 \in \{0, 1\} \\ 2 & \text{when } W_2 = 2 \end{cases} \quad (62)$$

where  $X_1$  contains 2 bits and  $X_2$  contains  $\log_2 3$  bits. Obviously  $z'$  is useless and it appears here to produce a fixed-length code. An important feature of this function is that from  $f(W_1, W_2)$ , we can always recover  $W_1$ . Therefore,  $W_1$  is always sent from Alice. Further,  $W_2$  should be protected when  $W_2 \in \{0, 1\}$  and  $W_1 = 0$ . Thus,  $W_2$  is protected by a uniform noise  $z$  and when it should be revealed (i.e., when  $W_1 = 1$ ), Alice will send the noise  $z$ ; otherwise, when it should be protected, Alice will send an independent (thus useless) noise  $z'$ . After the coding idea is explained, the decoding rule is now obvious. Carol will first check the value of  $W_1$  (sent from Alice). If  $W_1 = 0$ , Carol will claim  $f = 0$  if  $X_2 \in \{0, 1\}$  and  $f = 1$  if  $X_2 = 2$ . If  $W_1 = 1$ , Carol will claim  $f = 4$  if  $X_2 = 2$  and otherwise if  $X_2 \in \{0, 1\}$ , Carol will use  $z$  to recover  $W_2$  and based on  $(W_1, W_2)$ ,  $f$  is decoded with no error. Security is guaranteed because when  $(W_1, W_2) \in \{(0, 0), (0, 1)\}$ , for both cases we have that  $X_1, X_2$  are independent, in  $X_1 = (W_1, z')$ ,  $W_1$  is fixed to 0,  $z'$  is uniform and  $X_2 = W_2 + z$  is also uniform. Therefore, using this improved scheme, it suffices to send  $2 < 3$  bits by Alice and  $\log_2 3 < 3$  bits by Bob, respectively.

$f$	0	1	2	Expand $\xrightarrow{\text{over } \mathbb{Z}_8}$	$\tilde{W}_1 + \tilde{W}_2$	0	2	6	Randomize $\xrightarrow{\text{over } \mathbb{Z}_8}$	$\gamma \times (\tilde{W}_1 + \tilde{W}_2)$	0	2	6
0	0	0	1		1	1	3	7		1	{1, 3}	{1, 3}	{5, 7}
1	2	3	4		2	2	4	0		2	{2, 6}	4	0

**Figure 8.** An expand-and-randomize secure computation code over  $\mathbb{Z}_8$ , where  $\gamma$  is uniform over  $\{1, 3\}$ .

Going forward, while the characterization of the algebraic structures of the confusable sets over  $\mathbb{F}_q$  and  $\mathbb{Z}_n$  is given, i.e., the confusable sets in Theorems 2 and 3 are complete (a simple consequence of the confusable sets definition is the closure property under multiplication), the algorithmic aspect of the expand-and-randomize scheme over  $\mathbb{F}_q$  and  $\mathbb{Z}_n$  is wide open, i.e., we do not have efficient algorithms that can help us quickly identify (minimal) feasible expanded functions when they exist. The solutions to current examples are mainly found through the lists of confusable sets in Appendix A and Appendix B. Efficient algorithms that can classify functions and realize associations with isomorphisms of confusable sets are of immediate interest, but appear challenging at the same time. Going beyond the finite field and the ring of integers modulo  $n$ , it is interesting to explore other widely studied algebraic objects in abstract algebra [23], e.g., the matrix ring and the polynomial ring. Generally speaking, the expand-and-randomize scheme captures the idea of *embedding* the function to compute in another function that guarantees security. The potential of this general embedding theme remains to be fully explored. Finally, we note that while we focus on the basic model of minimal (non-interactive three-user) secure computation, the proposed scheme generalizes immediately to interactive protocols (by first interactively generating the common randomness) and to more users (the notions of expanded functions generalize in a natural manner). Exploration of the proposed scheme to various models of secure computation [11] is an interesting research avenue.

**Author Contributions:** Conceptualization, Y.Z. and H.S.; methodology, Y.Z. and H.S.; formal analysis, Y.Z. and H.S.; writing—original draft preparation, Y.Z. and H.S.; writing—review and editing, Y.Z. and H.S.; funding acquisition, H.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by funding from NSF grants CCF-2007108 and CCF-2045656.

**Conflicts of Interest:** The authors declare no conflict of interest.

# Appendix A. Confusable Sets of $\mathbb{F}_q, q < 20$

Field	$\gamma \in \mathcal{S}^*$	Confusable Sets $\mathcal{S}_0, \mathcal{S}_1, \dots$	$g$	$h(x)$
$\mathbb{F}_5$	$\{1, 4\}$	$\{0\}, \{1, 4\}, \{2, 3\}$	2	—
$\mathbb{F}_7$	$\{1, 6\}$	$\{0\}, \{1, 6\}, \{2, 5\}, \{3, 4\}$	3	—
	$\{1, 2, 4\}$	$\{0\}, \{1, 2, 4\}, \{3, 5, 6\}$		
$\mathbb{F}_{3^2}$	$\{1, 2\}$	$\{0\}, \{1, 2\}, \{x, 2x\}, \{x+2, 2x+1\}, \{x+1, 2x+2\}$	$x$	$x^2 + x + 2$
	$\{1, 2, x+2, 2x+1\}$	$\{0\}, \{1, 2, x+2, 2x+1\}, \{x, 2x, x+1, 2x+2\}$		
$\mathbb{F}_{11}$	$\{1, 10\}$	$\{0\}, \{1, 10\}, \{2, 9\}, \{3, 8\}, \{4, 7\}, \{5, 6\}$	2	—
	$\{1, 3, 4, 5, 9\}$	$\{0\}, \{1, 3, 4, 5, 9\}, \{2, 6, 7, 8, 10\}$		
$\mathbb{F}_{13}$	$\{1, 12\}$	$\{0\}, \{1, 12\}, \{2, 11\}, \{3, 10\}, \{4, 9\}, \{5, 8\}, \{6, 7\}$	2	—
	$\{1, 3, 9\}$	$\{0\}, \{1, 3, 9\}, \{2, 5, 6\}, \{4, 10, 12\}, \{7, 8, 11\}$		
	$\{1, 5, 8, 12\}$	$\{0\}, \{1, 5, 8, 12\}, \{2, 3, 10, 11\}, \{4, 6, 7, 9\}$		
	$\{1, 3, 4, 9, 10, 12\}$	$\{0\}, \{1, 3, 4, 9, 10, 12\}, \{2, 5, 6, 7, 8, 11\}$		
$\mathbb{F}_{2^4}$	$\{1, x^2 + x, x^2 + x + 1\}$	$\{0\}, \{1, x^2 + x, x^2 + x + 1\}, \{x, x^3 + x^2, x^3 + x^2 + x\}, \{x+1, x^3 + 1, x^3 + x\}, \{x^2, x^3 + x + 1, x^3 + x^2 + x + 1\}, \{x^2 + 1, x^3, x^3 + x^2 + 1\}$	$x$	$x^4 + x + 1$
	$\{1, x^3, x^3 + x, x^3 + x^2, x^3 + x^2 + x + 1\}$	$\{0\}, \{1, x^3, x^3 + x, x^3 + x^2, x^3 + x^2 + x + 1\}, \{x, x+1, x^2 + x + 1, x^3 + x + 1, x^3 + x^2 + 1\}, \{x^2, x^2 + 1, x^2 + x, x^3 + 1, x^3 + x^2 + x\}$		
$\mathbb{F}_{17}$	$\{1, 16\}$	$\{0\}, \{1, 16\}, \{2, 15\}, \{3, 14\}, \{4, 13\}, \{5, 12\}, \{6, 11\}, \{7, 10\}, \{8, 9\}$	3	—
	$\{1, 4, 13, 16\}$	$\{0\}, \{1, 4, 13, 16\}, \{2, 8, 9, 15\}, \{3, 5, 12, 14\}, \{6, 7, 10, 11\}$		
	$\{1, 2, 4, 8, 9, 13, 15, 16\}$	$\{0\}, \{1, 2, 4, 8, 9, 13, 15, 16\}, \{3, 5, 6, 7, 10, 11, 12, 14\}$		
$\mathbb{F}_{19}$	$\{1, 18\}$	$\{0\}, \{1, 18\}, \{2, 17\}, \{3, 16\}, \{4, 15\}, \{5, 14\}, \{6, 13\}, \{7, 12\}, \{8, 11\}, \{9, 10\}$	2	—
	$\{1, 7, 11\}$	$\{0\}, \{1, 7, 11\}, \{2, 3, 14\}, \{4, 6, 9\}, \{5, 16, 17\}, \{8, 12, 18\}, \{10, 13, 15\}$		
	$\{1, 7, 8, 11, 12, 18\}$	$\{0\}, \{1, 7, 8, 11, 12, 18\}, \{2, 3, 5, 14, 16, 17\}, \{4, 6, 9, 10, 13, 15\}$		
	$\{1, 4, 5, 6, 7, 9, 11, 16, 17\}$	$\{0\}, \{1, 4, 5, 6, 7, 9, 11, 16, 17\}, \{2, 3, 8, 10, 12, 13, 14, 15, 18\}$		

For any  $q$ ,  $\mathcal{S}_0 = \{0\}$ ,  $\mathcal{S}_1 = \mathcal{S}^* = \mathbb{F}_q^\times$  are always feasible and omitted (e.g.,  $q = 3, 2^2, 2^3$  only have such confusable sets).

**Figure A1.** A list of confusable sets for  $\mathbb{F}_q, q < 20$ .  $\mathcal{S}^*$  is the set of elements over which the randomizer  $\gamma$  is uniformly distributed.  $g$  is a primitive element of  $\mathbb{F}_q^\times$  and  $h(x)$  is an irreducible polynomial for  $\mathbb{F}_{p^n}, n > 1$ .

## Appendix B. Confusable Sets of $\mathbb{Z}_n, n < 20$

Ring	$\gamma \in G_n$	Confusable Sets $\mathcal{S}_0, \mathcal{S}_1, \dots$
$\mathbb{Z}_4$	$\{1, 3\}$	$\{0\}, \{1, 3\}, \{2\}$
$\mathbb{Z}_6$	$\{1, 5\}$	$\{0\}, \{1, 5\}, \{2, 4\}, \{3\}$
$\mathbb{Z}_8$	$\{1, 3\}$	$\{0\}, \{1, 3\}, \{2, 6\}, \{4\}, \{5, 7\}$
	$\{1, 5\}$	$\{0\}, \{1, 5\}, \{2\}, \{3, 7\}, \{4\}, \{6\}$
	$\{1, 7\}$	$\{0\}, \{1, 7\}, \{2, 6\}, \{3, 5\}, \{4\}$
	$\{1, 3, 5, 7\}$	$\{0\}, \{1, 3, 5, 7\}, \{2, 6\}, \{4\}$
$\mathbb{Z}_9$	$\{1, 8\}$	$\{0\}, \{1, 8\}, \{2, 7\}, \{3, 6\}, \{4, 5\}$
	$\{1, 4, 7\}$	$\{0\}, \{1, 4, 7\}, \{2, 5, 8\}, \{3\}, \{6\}$
	$\{1, 2, 4, 5, 7, 8\}$	$\{0\}, \{1, 2, 4, 5, 7, 8\}, \{3, 6\}$
$\mathbb{Z}_{10}$	$\{1, 9\}$	$\{0\}, \{1, 9\}, \{2, 8\}, \{3, 7\}, \{4, 6\}, \{5\}$
	$\{1, 3, 7, 9\}$	$\{0\}, \{1, 3, 7, 9\}, \{2, 4, 6, 8\}, \{5\}$
$\mathbb{Z}_{12}$	$\{1, 5\}$	$\{0\}, \{1, 5\}, \{2, 10\}, \{3\}, \{4, 8\}, \{6\}, \{7, 11\}, \{9\}$
	$\{1, 7\}$	$\{0\}, \{1, 7\}, \{2\}, \{3, 9\}, \{4\}, \{5, 11\}, \{6\}, \{8\}, \{10\}$
	$\{1, 11\}$	$\{0\}, \{1, 11\}, \{2, 10\}, \{3, 9\}, \{4, 8\}, \{5, 7\}, \{6\}$
	$\{1, 5, 7, 11\}$	$\{0\}, \{1, 5, 7, 11\}, \{2, 10\}, \{3, 9\}, \{4, 8\}, \{6\}$
$\mathbb{Z}_{14}$	$\{1, 13\}$	$\{0\}, \{1, 13\}, \{2, 12\}, \{3, 11\}, \{4, 10\}, \{5, 9\}, \{6, 8\}, \{7\}$
	$\{1, 9, 11\}$	$\{0\}, \{1, 9, 11\}, \{2, 4, 8\}, \{3, 5, 13\}, \{6, 10, 12\}, \{7\}$
	$\{1, 3, 5, 9, 11, 13\}$	$\{0\}, \{1, 3, 5, 9, 11, 13\}, \{2, 4, 6, 8, 10, 12\}, \{7\}$
$\mathbb{Z}_{15}$	$\{1, 4\}$	$\{0\}, \{1, 4\}, \{2, 8\}, \{3, 12\}, \{5\}, \{6, 9\}, \{7, 13\}, \{10\}, \{11, 14\}$
	$\{1, 11\}$	$\{0\}, \{1, 11\}, \{2, 7\}, \{3\}, \{4, 14\}, \{5, 10\}, \{6\}, \{8, 13\}, \{9\}, \{12\}$
	$\{1, 14\}$	$\{0\}, \{1, 14\}, \{2, 13\}, \{3, 12\}, \{4, 11\}, \{5, 10\}, \{6, 9\}, \{7, 8\}$
	$\{1, 2, 4, 8\}$	$\{0\}, \{1, 2, 4, 8\}, \{3, 6, 9, 12\}, \{5, 10\}, \{7, 11, 13, 14\}$
	$\{1, 4, 7, 13\}$	$\{0\}, \{1, 4, 7, 13\}, \{2, 8, 11, 14\}, \{3, 6, 9, 12\}, \{5\}, \{10\}$
	$\{1, 4, 11, 14\}$	$\{0\}, \{1, 4, 11, 14\}, \{2, 7, 8, 13\}, \{3, 12\}, \{5, 10\}, \{6, 9\}$
	$\{1, 2, 4, 7, 8, 11, 13, 14\}$	$\{0\}, \{1, 2, 4, 7, 8, 11, 13, 14\}, \{3, 6, 9, 12\}, \{5, 10\}$
$\mathbb{Z}_{16}$	$\{1, 7\}$	$\{0\}, \{1, 7\}, \{2, 14\}, \{3, 5\}, \{4, 12\}, \{6, 10\}, \{8\}, \{9, 15\}, \{11, 13\}$
	$\{1, 9\}$	$\{0\}, \{1, 9\}, \{2\}, \{3, 11\}, \{4\}, \{5, 13\}, \{6\}, \{7, 15\}, \{8\}, \{10\}, \{12\}, \{14\}$
	$\{1, 15\}$	$\{0\}, \{1, 15\}, \{2, 14\}, \{3, 13\}, \{4, 12\}, \{5, 11\}, \{6, 10\}, \{7, 9\}, \{8\}$
	$\{1, 3, 9, 11\}$	$\{0\}, \{1, 3, 9, 11\}, \{2, 6\}, \{4, 12\}, \{5, 7, 13, 15\}, \{8\}, \{10, 14\}$
	$\{1, 5, 9, 13\}$	$\{0\}, \{1, 5, 9, 13\}, \{2, 10\}, \{3, 7, 11, 15\}, \{4\}, \{6, 14\}, \{8\}, \{12\}$
	$\{1, 7, 9, 15\}$	$\{0\}, \{1, 7, 9, 15\}, \{2, 14\}, \{3, 5, 11, 13\}, \{4, 12\}, \{6, 10\}, \{8\}$
	$\{1, 3, 5, 7, 9, 11, 13, 15\}$	$\{0\}, \{1, 3, 5, 7, 9, 11, 13, 15\}, \{2, 6, 10, 14\}, \{4, 12\}, \{8\}$
$\mathbb{Z}_{18}$	$\{1, 17\}$	$\{0\}, \{1, 17\}, \{2, 16\}, \{3, 15\}, \{4, 14\}, \{5, 13\}, \{6, 12\}, \{7, 11\}, \{8, 10\}, \{9\}$
	$\{1, 7, 13\}$	$\{0\}, \{1, 7, 13\}, \{2, 8, 14\}, \{3\}, \{4, 10, 16\}, \{5, 11, 17\}, \{6\}, \{9\}, \{12\}, \{15\}$
	$\{1, 5, 7, 11, 13, 17\}$	$\{0\}, \{1, 5, 7, 11, 13, 17\}, \{2, 4, 8, 10, 14, 16\}, \{3, 15\}, \{6, 12\}, \{9\}$

Figure A2. A list of confusable sets for  $\mathbb{Z}_n, n < 20, n$  is not a prime (prime  $n$  has been covered in Figure A1).

## Appendix C. Useful Properties of Subgroups of $\mathbb{Z}_n^\times$

- It is established by Gauss that  $\mathbb{Z}_n^\times$  is cyclic if and only if  $n \in \{2, 4, p^b, 2p^b\}$ , where  $p$  is an odd prime and  $b$  is a positive integer (see, e.g., Theorem 42 in [27]).  $\mathbb{Z}_n^\times$  can be generated by a single element  $g$ , typically referred to as the primitive root modulo  $n$ . After  $g$  is found, we can enumerate all subgroups of  $\mathbb{Z}_n^\times$  (similar to Theorem 2). There is no analytic formula or fast algorithm to find the primitive root modulo  $n$  in general (see Section 1.4 of [44]).
- For other values of  $n$  not covered above, we do not have a full understanding of all the subgroups of  $\mathbb{Z}_n^\times$  in general. A useful approach is prime-power decomposition, based on the Chinese Remainder Theorem (see, e.g., Section 7.6 in [23]).  $\mathbb{Z}_n^\times$  is a direct product of the groups corresponding to each of its prime power factors, i.e.,  $\mathbb{Z}_n^\times = \mathbb{Z}_{p_1^{k_1}}^\times \times \mathbb{Z}_{p_2^{k_2}}^\times \cdots$ , where  $n = p_1^{k_1} p_2^{k_2} \cdots$  for distinct primes  $p_i$  and integers  $k_i$ . Any product of subgroups is a subgroup of the product group. However, the reverse



argument is not true, i.e., some subgroup of  $\mathbb{Z}_n^\times$  cannot be written as a product of subgroups of  $\mathbb{Z}_{p_i^{k_i}}$ . To go beyond subgroup products, we may resort to the fundamental theorem of finite Abelian group (see, e.g., Section 5.2 in [23]), which help decompose  $\mathbb{Z}_n^\times$  to direct groups of  $\mathbb{Z}_b$  (see Theorem 1.4.1 in [44]) and then the subgroup enumeration problem becomes that of counting the subgroups of a finite abelian group, where the case of product of two groups  $\mathbb{Z}_{b_1} \times \mathbb{Z}_{b_2}$  is fully solved and otherwise open [45–47] (for analytic solutions of some simple cases, see in [48]). Finally, the total number of subgroups of  $\mathbb{Z}_n$  has order  $O(\frac{\log n}{\log \log n})$  [49].

## References

- Beimel, A.; Orlov, I. Secret sharing and non-shannon information inequalities. *IEEE Trans. Inf. Theory* **2011**, *57*, 5634–5649. [CrossRef]
- Martín, S.; Padró, C.; Yang, A. Secret sharing, rank inequalities, and information inequalities. *IEEE Trans. Inf. Theory* **2016**, *62*, 599–609. [CrossRef]
- Sun, H.; Jafar, S.A. The Capacity of Private Information Retrieval. *IEEE Trans. Inf. Theory* **2017**, *63*, 4075–4088. [CrossRef]
- Banawan, K.; Ulukus, S. The Capacity of Private Information Retrieval from Coded Databases. *IEEE Trans. Inf. Theory* **2018**, *64*, 1945–1956. [CrossRef]
- Lee, E.J.; Abbe, E. Two shannon-type problems on secure multi-party computations. In Proceedings of the 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 30 September–3 October 2014; pp. 1287–1293.
- Data, D.; Prabhakaran, V.M.; Prabhakaran, M.M. Communication and randomness lower bounds for secure computation. *IEEE Trans. Inf. Theory* **2016**, *62*, 3901–3929. [CrossRef]
- Zhou, Y.; Sun, H.; Fu, S. On the Randomness Cost of Linear Secure Computation. In Proceedings of the 2019 53rd Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 20–22 March 2019; pp. 1–6.
- Yao, A.C. Protocols for secure computations. In Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982), Chicago, IL, USA, 3–5 November 1982; pp. 160–164.
- Ben-Or, M.; Goldwasser, S.; Wigderson, A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, 2–4 May 1988; pp. 1–10.
- Chaum, D.; Crépeau, C.; Damgård, I. Multiparty unconditionally secure protocols. In Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, 2–4 May 1988; pp. 11–19.
- Cramer, R.; Damgård, I.B.; Nielsen, J.B. *Secure Multiparty Computation and Secret Sharing*; Cambridge University Press: Cambridge, UK, 2015.
- Feige, U.; Killian, J.; Naor, M. A minimal model for secure computation. In Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, Montreal, QC, Canada, 23–25 May 1994; pp. 554–563.
- Applebaum, B.; Holenstein, T.; Mishra, M.; Shayevitz, O. The communication complexity of private simultaneous messages, revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin, Germany, 2018; pp. 261–286.
- Ishai, Y.; Kushilevitz, E. Private simultaneous messages protocols with applications. In Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems, Ramat Gan, Israel, 17–19 June 1997; pp. 174–183.
- Beimel, A.; Kushilevitz, E.; Nissim, P. The complexity of multiparty PSM protocols and related models. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin, Germany, 2018; pp. 287–318.
- Assouline, L.; Liu, T. Multi-Party PSM, Revisited. *Cryptology ePrint Archive*, Report 2019/657. 2019. Available online: <https://eprint.iacr.org/2019/657> (accessed on 3 October 2021).
- Beimel, A.; Gabizon, A.; Ishai, Y.; Kushilevitz, E.; Meldgaard, S.; Paskin-Cherniavsky, A. Non-interactive secure multiparty computation. In *Annual Cryptology Conference*; Springer: Berlin, Germany, 2014; pp. 387–404.
- Benhamouda, F.; Krawczyk, H.; Rabin, T. Robust non-interactive multiparty computation against constant-size collusion. In *Annual International Cryptology Conference*; Springer: Berlin, Germany, 2017; pp. 391–419.
- Yoshida, M.; Obana, S. On the (in) efficiency of non-interactive secure multiparty computation. *Des. Codes Cryptogr.* **2018**, *86*, 1793–1805. [CrossRef]
- Agarwal, N.; Anand, S.; Prabhakaran, M. Uncovering Algebraic Structures in the MPC Landscape. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin, Germany, 2019; pp. 381–406.
- Halevi, S.; Ishai, Y.; Kushilevitz, E.; Rabin, T. Best possible information-theoretic MPC. In *Theory of Cryptography Conference*; Springer: Berlin, Germany, 2018; pp. 255–281.
- Beimel, A.; Ishai, Y.; Kushilevitz, E. Ad hoc PSM protocols: Secure computation without coordination. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin, Germany, 2017; pp. 580–608.
- Dummit, D.S.; Foote, R.M. *Abstract Algebra*; John Wiley & Sons: Hoboken, NJ, USA, 2004.

24. Ishai, Y.; Kushilevitz, E. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In Proceedings of the 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, CA, USA, 12–14 November 2000; pp. 294–304.
25. Yuval, I.; Kushilevitz, E. Perfect constant-round secure computation via perfect randomizing polynomials. In *International Colloquium on Automata, Languages, and Programming*; Springer: Berlin, Germany, 2002; pp. 244–256.
26. Lidl, R.; Niederreiter, H. *Finite Fields*; Cambridge University Press: Cambridge, UK, 1997; Volume 20.
27. Shanks, D. *Solved and Unsolved Problems in Number Theory*; Chelsea Publishing Company: New York, NY, USA, 1978.
28. Judson, T. *Abstract Algebra: Theory and Applications*; Stephen F. Austin State University: Nacogdoches, TX, USA, 2014.
29. Han, T.S.; Kobayashi, K. A Dichotomy of Functions  $F(x, y)$  of Correlated Sources  $(X, Y)$  from the Viewpoint of the Achievable Rate Region. *IEEE Trans. Inf. Theory* **1987**, *33*, 69–76. [\[CrossRef\]](#)
30. Kuzuoka, S.; Watanabe, S. On distributed computing for functions with certain structures. *IEEE Trans. Inf. Theory* **2017**, *63*, 7003–7017. [\[CrossRef\]](#)
31. Nazer, B.; Gastpar, M. Computation over multiple-access channels. *IEEE Trans. Inf. Theory* **2007**, *53*, 3498–3516. [\[CrossRef\]](#)
32. Appuswamy, R.; Franceschetti, M.; Karamchandani, N.; Zeger, K. Network coding for computing: Cut-set bounds. *IEEE Trans. Inf. Theory* **2011**, *57*, 1015–1030. [\[CrossRef\]](#)
33. Huang, C.; Tan, Z.; Yang, S.; Guang, X. Comments on cut-set bounds on network function computation. *IEEE Trans. Inf. Theory* **2018**, *64*, 6454–6459. [\[CrossRef\]](#)
34. Korner, J.; Marton, K. How to encode the modulo-two sum of binary sources. *IEEE Trans. Inf. Theory* **1979**, *25*, 219–221. [\[CrossRef\]](#)
35. Elias, P. Coding for noisy channels. *IRE Conv. Rec.* **1955**, *3*, 37–46.
36. Wyner, A. Recent results in the shannon theory. *IEEE Trans. Inf. Theory* **1974**, *20*, 2–10. [\[CrossRef\]](#)
37. Csiszar, I. Linear codes for sources and source networks: Error exponents, universal coding. *IEEE Trans. Inf. Theory* **1982**, *28*, 585–592. [\[CrossRef\]](#)
38. Gamal, A.E.; Kim, Y.-H. *Network Information Theory*; Cambridge University Press: Cambridge, UK, 2011.
39. Data, D.; Dey, B.K.; Mishra, M.; Prabhakaran, V.M. How to securely compute the modulo-two sum of binary sources. In Proceedings of the 2014 IEEE Information Theory Workshop (ITW 2014), Hobart, Australia, 2–5 November 2014; pp. 496–500.
40. Sahebi, A.G.; Pradhan, S.S. Abelian group codes for channel coding and source coding. *IEEE Trans. Inf. Theory* **2015**, *61*, 2399–2414. [\[CrossRef\]](#)
41. Heidari, M.; Pradhan, S.S. How to compute modulo prime-power sums. In Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, Spain, 10–15 July 2016; pp. 1824–1828.
42. Katz, J.; Lindell, Y. *Introduction to Modern Cryptography*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2014.
43. Shoup, V. *A Computational Introduction to Number Theory and Algebra*; Cambridge University Press: Cambridge, UK, 2009.
44. Cohen, H. *A Course in Computational Algebraic Number Theory*; Springer Science & Business Media: New York, NY, USA, 2013; Volume 138.
45. Tărnăxău, M. An arithmetic method of counting the subgroups of a finite abelian group. *Bull. Math. Soc. Sci. Math. Roum.* **2010**, *53*, 373–386.
46. Tóth, L. Subgroups of finite abelian groups having rank two via Goursat’s lemma. *Tatra Mt. Math. Publ.* **2014**, *59*, 93–103. [\[CrossRef\]](#)
47. Bauer, K.; Sen, D.; Zvengrowski, P. A generalized goursat lemma. *arXiv* **2011**, arXiv:1109.0024.
48. Petrillo, J. Counting subgroups in a direct product of finite cyclic groups. *Coll. Math. J.* **2011**, *42*, 215–222. [\[CrossRef\]](#)
49. Martin, G.; Troupe, L. The distribution of the number of subgroups of the multiplicative group. *J. Aust. Math. Soc.* **2017**, *108*, 46–97. [\[CrossRef\]](#)