SCIENCE CHINA
Information Sciences

Polynomial AND homomorphic cryptosystem and applications

Shundong LI[1], Sufang ZHOU[1], Jiawei DOU[2,*] and Wenli WANG[1]

Citation: SCIENCE CHINA Information Sciences **63**, 112105 (2020); doi: 10.1007/s11432-018-9789-y

View online: https://engine.scichina.com/doi/10.1007/s11432-018-9789-y

View Table of Contents: https://engine.scichina.com/publisher/scp/journal/SCIS/63/1

Published by the Science China Press

SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2020, Vol. 63 112105:1–112105:15
https://doi.org/10.1007/s11432-018-9789-y

Polynomial AND homomorphic cryptosystem and applications

Shundong LI[1], Sufang ZHOU[1], Jiawei DOU[2*] & Wenli WANG[1]

[1]*School of Computer Science, Shaanxi Normal University, Xi'an* 710062*, China;*
[2]*School of Mathematics and Information Science, Shaanxi Normal University, Xi'an* 710062*, China*

Received 17 August 2018/Revised 20 December 2018/Accepted 14 February 2019/Published online 25 December 2019

Abstract Homomorphic cryptosystems are fundamental and highly effective cryptographic primitives for addressing security problems arising in information processing, data analysis and data applications, particularly in secure cloud computing and secure multiparty computation. To privately compute functions such as $E(x_1 \wedge \cdots \wedge x_t)$, $E(x_1 \vee \cdots \vee x_t)$ and $E[(x_{11} \wedge \cdots \wedge x_{m1}) \vee \cdots \vee (x_{1n} \wedge \cdots \wedge x_{mn})]$ (m disjunctive normal form (mDNF)) on $E(x_1), \ldots, E(x_t)$ and $E(x_{11}), \ldots, E(x_{mn})$ without knowing the decryption key, Boolean homomorphic cryptosystems are necessary. Exploring new homomorphic cryptosystems to solve these problems is appealing, and is of high theoretical and practical significance. To solve problems such as these, we propose a polynomial AND homomorphic cryptosystem based on the ideal theory of abstract algebra; the scheme can be obtained from available multiplicatively homomorphic cryptosystems such as the ElGamal. We prove that the cryptosystem is semantically secure. This polynomial AND homomorphic cryptosystem is a highly effective tool for designing various cryptographic protocols. As examples, we demonstrate its applications to privately compute any DNF (i.e., $P(X_1, \ldots, X_m) = E[(x_{11} \wedge \cdots \wedge x_{m1}) \vee \cdots \vee (x_{1n} \wedge \cdots \wedge x_{mn})]$ on ciphertexts $E(x_{ij})$ of x_{ij} without knowing the decryption key) and the intersection and union of certain private sets.

Keywords polynomial AND homomorphic cryptosystem, semigroup, ideal, cloud computing security, cryptography, secure multiparty computation

Citation Li S D, Zhou S F, Dou J W, et al. Polynomial AND homomorphic cryptosystem and applications. Sci China Inf Sci, 2020, 63(1): 112105, https://doi.org/10.1007/s11432-018-9789-y

1 Introduction

Homomorphic cryptosystem was first proposed by Rivest et al. [1]. A homomorphic cryptosystem E enables us to compute a ciphertext $E(f(x_1, \ldots, x_m))$ of a function $f(x_1, \ldots, x_m)$ on ciphertexts $E(x_1), \ldots, E(x_m)$ of $x_1, \ldots, x_m$ without knowing the decryption key. Homomorphic cryptosystems are fundamental and highly effective primitives in cloud computing security [2–5] and in the construction of other secure systems such as secure voting systems, collision-resistant hash functions and private information retrieval systems. They are also highly effective building blocks for secure multiparty computation (SMC) [6–9], which is a crucial privacy-preserving technology in cooperative computation [10] and a major area of focus in the international cryptographic community.

There are two types of homomorphic cryptosystems: partially homomorphic cryposystems and fully homomorphic ones. Partially homomorphic cryptosystems include (1) multiplicatively homomorphic cryptosystems such as RSA [11], ElGamal [12] and Rabin [13], (2) additively homomorphic ones such as the Okamoto–Uchiyama [14], Paillier [15], elliptic curve [16], NTRU [17], GGH [18] (This cryptosystem

* Corresponding author (email: Jiawei@snnu.edu.cn)

had been broken by Hu et al. [19]), Benaloh [20], Naccache–Stern [21], Damgård–Jurik [22] and Ishai–Paskin cryptosystems [23], and (3) XOR-homomorphic cryptosystems such as the Goldwasser–Micali cryptosystem [24]. The BGN cryptosystem [25] permits us to homomorphically compute a quasi-2DNF formula. All the partially homomorphic cryptosystems are expressed as algebraic formulae; we can also call them algebraic homomorphic cryptosystems.

Partially homomorphic cryptosystems are highly effective primitives for addressing both secure cloud computing and SMC problems. There are no adequate partially homomorphic public key cryptosystems to be used, and the functions that can be privately computed on ciphertexts using partially homomorphic cryptosystems are limited. Exploring new homomorphic cryptosystems is appealing and is of high theoretical and practical significance.

Theoretically, fully homomorphic encryption (FHE) schemes [26–28] are the most effective building blocks for addressing cloud computing security and SMC problems [8]. They enable the computation of ciphertext $E(f(x_1,\ldots,x_t))$ of any function $f(x_1,\ldots,x_t)$ by performing a few operations on the ciphertexts $E(x_1),\ldots,E(x_t)$ of $x_1,\ldots,x_t$ without knowing the decryption key. However, unless the 'function' is expressed as a Boolean circuit $C(x_1,\ldots,x_t)$, we cannot compute it on ciphertexts. Although the concept of FHE is old, the first construction was given until 2009 by Gentry [26].

There are two types of FHE schemes: one with a packing technique (such as BGV12 [29], FV12 [30] and CKKS17 [31]) and the other one with rapid bootstrapping such as CGGI16-18 [32–34]. The first type exhibits larger parameters and consumes a longer time for bootstrapping to lower the noise such that it does not hit the maximum noise level; however, they can encrypt multiple plaintexts in a single ciphertext and therefore, have highly marginal amortized expansion rate and timing. The second type can encrypt only a bit, although it has a short latency with bootstrapping.

In the literature, bootstrapping (which has remained highly expensive [35]) is generally used to convert a somewhat homomorphic encryption into an FHE. In the last few years, researchers managed to construct significantly more efficient schemes by speeding up the bootstrapping, bringing practical applications close to reality. Chillotti et al. [35,36] made bootstrapping run in 13 ms. They also reduced the bootstrapping key size from 1 GB to 16 MB (however, it is still very large). According to their paper, achieving this will make their scheme forgo certain composability properties in the design of homomorphic circuits [35].

Gentry's FHE is based on hard problems on lattice. Lattice-based cryptography is the strongest candidate for post-quantum cryptography; moreover, in the future they are likely to replace available public key cryptosystems that are based either on the hardness of factoring or on the hardness of computing discrete logarithm problem (DLP). However, at present, the public key cryptosystems based on these two problems play important roles in cryptography. To sum up, the available FHE schemes exhibit high potential for applications in the future, although they are inefficient and impractical at present.

SMC, an area of focus in the international cryptographic community, depends substantially on partially homomorphic cryptosystems. Most SMC protocols and secure cloud computing protocols are constructed using partially homomorphic cryptosystems in conjunction with secret sharing, oblivious transfer, one-way hash function, etc. Furthermore, a number of real-world applications in areas such as medical, financial and advertising only require that the cryptosystems is partially homomorphic. To sum up, partially homomorphic cryptosystems are highly powerful albeit limited.

Numerous decisional problems can be reduced to Boolean operations; however, these problems cannot be addressed conveniently and securely owing to the limited availability of Boolean homomorphic cryptosystems. For example, the following problems cannot be addressed conveniently at present.

Suppose t members are going to vote on a policy. To accept the policy, the vote must be passed with all in favour. For example, in the European Union (EU), certain policies that are considered to be sensitive remain subject to unanimous voting, such as taxation, social security, social protection, the accession of new states to the EU, foreign and common defence policy and operational police cooperation between the member states. If a member votes against the policy, it will not be passed. Occasionally, the voting is open; however, in most cases, if a proposal fails, the number of members who voted for or against it must be concealed. That is, although the correct result of the voting must be guaranteed, only the final

result is publicized. This problem can be abstracted as the private computation of

$$P(x_1, x_2, \ldots, x_t) = x_1 \wedge x_2 \wedge \cdots \wedge x_t. \tag{1}$$

Some other problems (for example, if we wish to know whether there is one voting for a proposal) may be reduced to the private computation of

$$Q(x_1, x_2, \ldots, x_t) = x_1 \vee x_2 \vee \cdots \vee x_t. \tag{2}$$

Oblivious transfer-based secure two-party AND computation, which needs to invoke secret sharing and oblivious transfer primitives, is rather complicated and inefficient [37, 38] and is likely to disclose some information that should not be disclosed.

Occasionally, for $X, Y \in \{0,1\}^m$, we have to privately compute the following predicate:

$$F(X_1, X_2) = (x_1 \wedge y_1) \vee \cdots \vee (x_m \wedge y_m). \tag{3}$$

This formula is called 2DNF. The problem of privately computing 2DNF has not been satisfactorily solved. Based on bilinear pairing, Boneh et al. [25] first proposed a public key cryptosystem that can privately compute a quasi-2DNF (It discloses how many $x_i y_i = 1$):

$$s = \sum_{i=1}^{m} x_i y_i = x_1 y_1 + \cdots + x_m y_m. \tag{4}$$

This pairing-based protocol needs to compute a discrete logarithm in a finite field and is of high computational complexity. Notwithstanding we use it to compute $\sum_{i=1}^{m} x_i y_i$, its applicability is highly limited due to the hardness of computing discrete logarithm. It does not compute a real 2DNF, and we do not know how to privately compute $(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_m \wedge y_m)$ with this protocol. Neither can it be used to privately compute a more complicated m-DNF:

$$F(X_1, \ldots, X_m) = (x_{11} \wedge \cdots \wedge x_{m1}) \vee \cdots \vee (x_{1n} \wedge \cdots \wedge x_{mn}). \tag{5}$$

If there is an AND homomorphic cryptosystem, by combining it with the De Morgan law, private computation problems such as (1)–(5) can be solved trivially. This is of substantial theoretical significance because based on a simple solution, we can construct a new solution that is secure in the malicious model more conveniently. Theoretically, an FHE can solve these problems; however, available FHE schemes are inefficient (in terms of computational complexity, communication complexity, ciphertext expansion rate or key size) to be applied in practice, and solutions based on FHE are not effective [39][1)].

Previous studies have attempted to design AND-homomorphic cryptosystems. For example, Sander et al. [40] changed the Goldwasser–Micali XOR-homomorphic cryptosystem [24] into an AND-homomorphic one. However, this scheme is inefficient and approximate, and the ciphertext expands several thousands times. Based on this AND-homomorphic property, Fischlin [41] constructed a protocol for the millionaires problem; the protocol is highly inefficient.

The ElGamal is one of the earliest and the most popular public key cryptosystems. It is based on the hardness of computing discrete logarithm in certain finite field and has withstood 40 years of extensive cryptanalysis. In the past few years, there have been several advancements in the number field sieve and function field sieve algorithms for computing discrete logarithms in finite fields F_{p^k}; here p is prime, and $k > 1$ is a small integer. These advancements render the DLP in small-characteristic finite fields (characteristic-two or characteristic-three field) solvable in quasi-polynomial time [42].

For computing the DLP in a finite extension field F_{q^k} with medium large prime number q, new algorithms with different computational complexities have been constructed for different q, k, since 2015. The complexities of the new algorithms are sub-exponential time; however, the asymptotic complexities

1) https://www.networkworld.com/article/3196121/security/how-to-make-fully-homomorphic-encryption-practical-and-usable.html.

are less than those of the previous algorithms. Computing the DLP in a finite field F_q with a carefully selected large prime number q is still hard [43], and the cryptosystems based on the hardness of computing DLP are still secure.

To enable private computation of functions similar to (1)–(5) on ciphertexts, we modify the ElGamal public key cryptosystem to obtain a polynomial AND homomorphic cryptosystem. The cryptosystem is of high theoretical and practical interest. Our main contributions are as follows:

• Based on the ideal theory of abstract algebra and the ElGamal public key cryptosystem, we construct a polynomial AND homomorphic cryptosystem.

• We prove that our variant is semantically secure. This cryptosystem can also be used to privately compute $E(m_1 \vee m_2)$ on $E(m_1), E(m_2)$ without knowing the private key.

• It is convenient to privately compute functions (1)–(5) using this new primitive. With regard to applications, we demonstrate its applications in privately computing the intersection set of certain private sets and function (3); this can be conveniently extended to privately compute function (5).

The remainder of this paper is organized as follows: In Section 2, we introduce a few preliminaries. Section 3 explains the concept for obtaining Boolean homomorphic cryptosystems from available public key cryptosystems and an efficient construction. In Section 4, we illustrate its applications. We enumerate our conclusion in Section 5.

2 Preliminaries

2.1 Homomorphic cryptosystem

A public key cryptosystem $\mathcal{E}$ consists of three algorithms: $\text{KeyGen}_{\mathcal{E}}$, $\text{Enc}_{\mathcal{E}}$ and $\text{Dec}_{\mathcal{E}}$.

$\text{KeyGen}_{\mathcal{E}}$ takes a security parameter k as an input and outputs a public key pk, the corresponding secret key sk as well as the definitions of the plaintext space $\mathcal{P}$ and the ciphertext space $\mathcal{C}$. $(\text{pk}, \text{sk}, \mathcal{P}, \mathcal{C}) \leftarrow \text{KeyGen}_{\mathcal{E}}(k)$.

$\text{Enc}_{\mathcal{E}}$. By taking pk and a plaintext $M \in \mathcal{P}$ as inputs, it outputs a ciphertext $C \in \mathcal{C}$. $C \leftarrow \text{Enc}_{\mathcal{E}}(\text{pk}, M)$.

$\text{Dec}_{\mathcal{E}}$. By taking a ciphertext $C \in \mathcal{C}$ and the secret key sk as inputs, it outputs the plaintext $M \in \mathcal{P}$. $M \leftarrow \text{Dec}_{\mathcal{E}}(\text{sk}, C)$.

A homomorphic public key cryptosystem $\mathcal{E}$ is a special public key cryptosystem. In addition to the three conventional algorithms described above, it also has an efficient algorithm $\text{Evaluate}_{\mathcal{E}}$; given the public key pk, a function F and a tuple of ciphertexts $C = \langle C_1, \ldots, C_t \rangle$ (where $C_i = \text{Enc}_{\mathcal{E}}(M_i)$), this algorithm can output the ciphertext of $F(M_1, \ldots, M_t)$, denoted by $\text{Enc}_{\mathcal{E}}(pk, F(M_1, \ldots, M_t))$, i.e.,

$$\text{Enc}_{\mathcal{E}}(\text{pk}, F(M_1, \ldots, M_t)) = \text{Evaluate}_{\mathcal{E}}(\text{pk}, F, (C_1, \ldots, C_t)).$$

2.2 ElGamal public key cryptosystem

The ElGamal is a popular and widely used public key cryptosystem. It is multiplicatively homomorphic. The KeyGen, Enc and Dec of the ElGamal public key cryptosystem are as follows [12]:

KeyGen. On a security parameter k, it generates a k-bit large random prime p and a generator α of the multiplicative group Z_p^*, selects a random integer $x, 1 \leqslant x \leqslant p-2$ as a private key and computes $h = \alpha^x \bmod p$ as the public key. Both the plaintext and ciphertext space are $Z_p^* = \{1, \ldots, p-1\}$.

Enc. To encrypt a message $m \in Z_p^*$, it chooses a random number $r \in Z_p^*$ and then sets

$$C = E(m) = (c_1, c_2) = (\alpha^r \bmod p, mh^r \bmod p).$$

Dec. To decrypt a ciphertext $C = (c_1, c_2) \in Z_p^*$, it computes $m = c_2 c_1^{-x} \bmod p$.

Homomorphism. If $C_1 = (g^{r_1} \bmod p, m_1 h^{r_1} \bmod p), C_2 = (g^{r_2} \bmod p, m_2 h^{r_2} \bmod p)$, then

$$C_1 C_2 = (g^{r_1+r_2} \bmod p, m_1 m_2 h^{r_1+r_2} \bmod p) = E(m_1 m_2).$$

Therefore, this cryptosystem is multiplicatively homomorphic. It can be used to encrypt ones, whereas it cannot be used to encrypt zeros. In the subsequent section, we modify this cryptosystem such that

it can be used to encrypt zeros and ones simulataneously and that when it is used to encrypt zeros and ones, it exhibits AND homomorphism.

2.3 Semantic security

An important criterion of security for a public key cryptosystem is semantic security. Semantic security is defined using the following IND-CPA mental game: (1) the adversary is given the public key generated by the challenger and the adversary can use it to encrypt whatever he wants; (2) the adversary generates two equal length messages M_0 and M_1, transmits them to the challenger and receives a ciphertext $E(M_b)$ of M_b from the challenger; here, b is randomly chosen from $\{0,1\}$; (3) the adversary guesses b' and wins the game if $b = b'$. A public cryptosystem is said to be semantically secure if no polynomial time adversary can win the game with a non-negligible advantage. Semantic security captures the intuition that given a ciphertext, the adversary learns nothing about the corresponding plaintext.

We state that a public cryptosystem $\mathcal{E}$ is semantically secure if no polynomially bounded adversary A has a non-negligible advantage against the challenger in the following game:

Setup. The challenger takes a security parameter k and executes the KeyGen algorithm. It gives the public key pk to the adversary, who can use it to encrypt whatever he wants, and keeps the secret key sk private.

Challenge. The adversary generates two equal length messages M_0, M_1 on which he wishes to be challenged and transmits them to the challenger. The challenger picks a random bit $b \in \{0,1\}$ and computes $C = E(M_b)$. He transmits C as the challenge to the adversary.

Guess. Finally, the adversary outputs a guess $b' \in \{0,1\}$ and wins the game if $b = b'$.

We refer to such an adversary A as an IND-CPA adversary. The advantage of an IND-CPA adversary A against the cryptosystem $\mathcal{E}$ is the following function of k:

$$\mathrm{Adv}_{\mathcal{E},A}(k) = \Pr[b = b'] - \frac{1}{2}.$$

The probability is over the random bits used by the challenger and adversary.

Definition 1. We state that a public key cryptosystem $\mathcal{E}$ is semantically secure if for any polynomial time IND-CPA adversary A, the function $\mathrm{Adv}_{\mathcal{E},A}(k)$ is negligible.

3 Polynomial AND homomorphic cryptosystems

If there is a multiplicatively homomorphic cryptosystem that could be used to encrypt zeros and ones simultaneously and if it is used in this manner, the cryptosystem will be AND homomorphic; this is because if $x, y \in \{0,1\}$, $x \times y = x \wedge y$ (this can be conveniently verified via truth table) and thus, $E(x) \times E(y) = E(x \times y) = E(x \wedge y)$. However, no available multiplicatively homomorphic cryptosystem can encrypt zeros and ones simultaneously. For example, RSA and Rabin cryptosystem cannot be used to encrypt either ones or zeros; ElGamal cryptosystem cannot be used to encrypt zeros.

Our goal is to modify the ElGamal cryptosystem such that it can simultaneously encrypt zeros and ones and therefore, it exhibits AND homomorphism. A polynomial AND homomorphic cryptosystem should satisfy the following criterions: (1) it can encrypt zeros and ones simultaneously; (2) it is semantically secure; (3) it is probabilistic; (4) it is AND homomorphic and (5) the homomorphism should preserve for polynomial many operations on ciphertext. Our concepts arises from the Ideal theory of abstract algebra.

3.1 Semigroup and ideal

A semigroup is an algebraic structure consisting of a set in conjunction with an associative binary operation. For example, the set of all integers mod n, denoted by Z_n, together with the multiplication modulo n forms a semigroup $(Z_n, \times)$. This semigroup is highly similar to a group except that not every element

Table 1 Multiplication table for Z_8

·	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

of it has a multiplicative inverse. If n is a prime number, $(Z_n, \times)$ forms a group. If n is an even integer, the semigroup $(Z_n, \times)$ exhibits certain interesting properties [44]:

(1) Z_n has a subset I. I together with multiplicative modulo n also forms a semigroup $(I, \times)$, which is called a subsemigroup of $(Z_n, \times)$.

(2) For $\forall x \in I, \forall y \in Z_n$, we have $x \times y \in I, y \times x \in I$ (If $\times$ is commutative, $x \times y \in I$ is sufficient).

(3) We denote $Z_n \setminus I$ by $\bar{I}$; then, for any $x, y \in \bar{I}$, $x \times y, y \times x \in \bar{I}$.

(2) and (3) can be summed up as $x \times y \in \bar{I} \Leftrightarrow x \in \bar{I} \wedge y \in \bar{I}$. Such an $(I, \times)$ is called an ideal of Z_n. For example, Table 1 presents the multiplication table for Z_8.

Here, $I = \{0, 2, 4, 6\}$ is the subset of Z_8 and $(I, \times)$ forms a subsemigroup of $(Z_8, \times)$. $(I, \times)$ is an ideal of $(Z_8, \times)$.

Formally, an ideal of a semigroup $(S, *)$ is a subsemigroup $(I, *)$ of $(S, *)$ such that if $a \in I$ and $r \in S$, $r * I \subset I$ and $I * r \subset I$.

3.2 AND homomorphic cryptosystem from ElGamal

Suppose that $\mathcal{E}$ is a multiplicatively homomorphic cryptosystem whose plaintext space is $\mathcal{P}$ and $Z_n \subset \mathcal{P}$ (where n is an integer). If n is an even integer, $(Z_n, \times)$ forms a semigroup and has an ideal.

Suppose that $(I, \times)$ is an ideal of $(Z_n, \times)$ and $Z_n \simeq \mathcal{P}$. When we need to encrypt zero, we encrypt $x \in I \setminus \{0\}$; when we need to encrypt one, we encrypt $y \in \bar{I}$. Suppose that $c = \mathrm{Enc}_{\mathcal{E}}(m)$. We stipulate that $\mathrm{Dec}_{\mathcal{E}}(c) \in I$ implies $m = 0$ and that $\mathrm{Dec}_{\mathcal{E}}(c) \in \bar{I}$ implies $m = 1$. With this modification, $\mathcal{E}$ can be used to encrypt zeros and ones simultaneously; thus, it exhibits AND homomorphic property because if $x, y \in \{0, 1\}$, $x \times y = x \wedge y$ and $\mathrm{Enc}(x) \times \mathrm{Enc}(y) = \mathrm{Enc}(xy)$ are equivalent to $\mathrm{Enc}(x) \times \mathrm{Enc}(y) = \mathrm{Enc}(x \wedge y)$. This is what AND homomorphic cryptosystem implies.

For the ElGamal probabilistic cryptosystem, the plaintext space is $Z_p^* = \{1, 2, \ldots, p-1\}$; here p is a large prime number. We know that $(Z_{p-1}, \times)$ forms a semigroup with $Z_{p-1} \simeq Z_p^* (Z_{p-1} = Z_p^* \cup \{0\} \setminus \{p-1\})$; moreover, it is highly convenient to verify that semigroup $(Z_{p-1}, \times)$ has an ideal $(I, \times)$. Here $I = \{0, 2, 4, \ldots, p-3\}$ and $\bar{I} = \{1, 3, \ldots, p-2\}$. Because the ElGamal cryptosystem cannot encrypt zero and $0 \notin Z_p^*$, when we need to encrypt zero, we choose $x \in I \setminus \{0\}$ and encrypt it; when we need to encrypt one, we choose $y \in \overline{I}$ and encrypt it. We also stipulate that if $c = \mathrm{Enc}(m)$ and $\mathrm{Dec}(c) \in I$, $m = 0$; otherwise if $\mathrm{Dec}(c) \in \bar{I}$, then $m = 1$. Because the ElGamal is multiplicatively homomorphic, we immediately obtain a polynomial AND homomorphic cryptosystem. The KeyGen, Enc, Dec and Evaluate algorithms of the AND homomorphic cryptosystem are as follows:

KeyGen. On a security parameter k, it generates a k-bit random prime p and a generator α of the multiplicative group $(Z_p^*, \times)$, selects a random integer $x, 1 < x \leqslant p-2$ as the private key and computes $h = \alpha^x \bmod p$ as the public key. The semigroup is Z_{p-1} which has an ideal $(I, \times)$; here, $I = \{0, 2, \ldots, p-3\}, \bar{I} = \{1, 3, \ldots, p-1\}$.

Enc. To encrypt a message m, choose two random numbers $r \in Z_p^*$ and s and compute

$$C = \mathrm{Enc}(m) = (c_1, c_2) = (\alpha^r \bmod p, sh^r \bmod p).$$

Here, if $m = 0$, $s \in I \setminus \{0\}$; otherwise, $s \in \bar{I}$.

Dec. To decrypt a ciphertext $C = (c_1, c_2)$, where $c_1, c_2 \in Z_p^*$, compute $s' = c_2 c_1^{-x} \bmod p \bmod p-1$. If $s' \in \bar{I}$, $\mathrm{Dec}(C) = 1$; otherwise, $\mathrm{Dec}(C) = 0$.

Remark 1. Because for Z_{p-1}, $I \subset Z_{p-1}$ contains element zero, but the ElGamal cryptosystem cannot encrypt zero, s should not be zero; i.e., one must choose $s \in Z_{p-1} \backslash \{0\}$.

Evaluate. Suppose $C_1 = \mathrm{Enc}(m_1), C_2 = \mathrm{Enc}(m_2)$ $(m_1, m_2 \in \{0, 1\})$. This implies that there are two random numbers $s_1, s_2 \in Z_{p-1} \backslash \{0\}$ such that

$$C_1 = (\alpha^{r_1} \bmod p, s_1 h^{r_1} \bmod p), \quad C_2 = (\alpha^{r_2} \bmod p, s_2 h^{r_2} \bmod p).$$

Then,

$$C_1 \cdot C_2 = \mathrm{Enc}(m_1) \cdot \mathrm{Enc}(m_2) = (\alpha^{r_1+r_2}, s_1 s_2 h^{r_1+r_2}) = (c_1, c_2).$$

To decrypt $C_1 C_2$, compute $s' = (c_2 c_1^{-x} \bmod p) \bmod p - 1 = s_1 s_2 \bmod p \bmod p - 1$.

If $s_1 s_2 < p-1$, $c_2 c_1^{-x} \bmod p \bmod p-1 = s_1 s_2$. By the properties of the ideal, $s_1 s_2 \in \bar{I} \Leftrightarrow s_1 \in \bar{I} \wedge s_2 \in \bar{I}$; i.e.,

$$\mathrm{Enc}(m_1) \cdot \mathrm{Enc}(m_2) = \mathrm{Enc}(1) \Leftrightarrow \mathrm{Enc}(m_1) = \mathrm{Enc}(1) \wedge \mathrm{Enc}(m_2) = \mathrm{Enc}(1).$$

This implies that $\mathrm{Enc}(m_1) \cdot \mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 \wedge m_2)$.

Parameters. If $\mathrm{Enc}(x_i) = (g^{r_i} \bmod p, s_i h^{r_i} \bmod p)$, and if we denote

$$\prod_{i=1}^{l} \mathrm{Enc}(x_i) = \left(g^{\sum_{i=1}^{l} r_i} \bmod p, h^{\sum_{i=1}^{l} r_i} \prod_{i=1}^{l} s_i \bmod p \right) = (c_1, c_2),$$

then, $c_2 c_1^{-x} \bmod p = \prod_{i=1}^{l} s_i \bmod p$. $\prod_{i=1}^{l} s_i \bmod p = \prod_{i=1}^{l} s_i \Leftrightarrow \prod_{i=1}^{l} s_i < p$. The number of times that we can perform homomorphic operations on ciphertext depends substantially on s_i and p. More or less, the number is equal to $\log_2 p / \log_2 \max\{s_i\}$. The higher the number of operations that you wish to perform, the smaller the s_i that you should choose. Typically, $\log_2 p = 1024$. If we choose $\log_2 \max\{s_i\} = 16$, we can perform 64 homomorphic operations on ciphertexts rather than infinite homomorphic operations. Therefore, we call this a polynomial AND homomorphic cryptosystem. For most applications, a polynomial AND homomorphic cryptosystem is adequate.

3.3 Performance

Security. The security of this AND homomorphic cryptosystem is identical to that of the original ElGamal cryptosystem because it does not change the ElGamal cryptosystem. It uses $s \in I \setminus \{0\} \subset Z_p^*$ to respresent zero, and $s \in \bar{I} \subset Z_p^*$ to represent one and encrypts s. Thus, we have the following theorem.

Theorem 1. If the original ElGamal cryptosystem is semantically secure, the variant is also semantically secure.

Here 'secure' implies that no polynomial time algorithm can distinguish $\mathrm{Enc}(m_1) = (g^{r_1} \bmod p, s_1 h^{r_1} \bmod p)$ $(s_1 \in I \backslash \{0\})$ from $\mathrm{Enc}(m_2) = (g^{r_2} \bmod p, s_2 h^{r_2} \bmod p)$ $(s_2 \in \bar{I})$ with non-negligible advantage.

Proof sketch. Suppose there was a polynomial time algorithm A that can distinguish $\mathrm{Enc}(m_1)(m_1 = 0)$ from $\mathrm{Enc}(m_2)(m_2 = 1)$ with advantage ε (implying that given $(g^r \bmod p, \mathrm{sh}^r \bmod p)$, A can determine whether $s \in \bar{I}$ with advantage ε); we could use this algorithm to construct a new algorithm B that would win the IND-CPA game that defines the semantic security of the original ElGamal cryptosystem with the same advantage ε as follows.

In the challenge phase, algorithm B, simulating the adversary, generates two messages $M_0, M_1 \in Z_p^*$ of identical length (although $M_0 \in I, M_1 \in \bar{I}$) and transmits M_0, M_1 to the challenger. The challenger randomly chooses a bit $b \in \{0, 1\}$, encrypts M_b to obtain $\mathrm{Enc}(M_b)$ and transmits it back to B; B invokes A with $\mathrm{Enc}(M_b)$. Because A can determine whether $M_b \in \bar{I}$ with advantage ε, if A outputs that $M_b \in \bar{I}$, B outputs that $b = 1$; otherwise, B outputs that $b = 0$. Because A's advantage is ε and B uses A's output to make its guess, B will win the game with an identical advantage ε. This implies that the original ElGamal cryptosystem is not semantically secure.

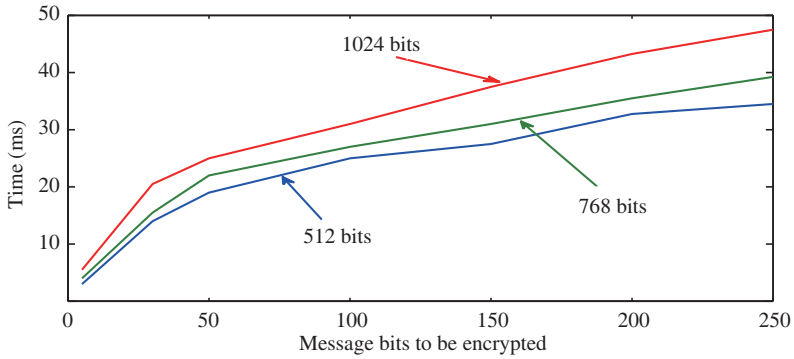

Figure 1 (Color online) The execution time increases almost linearly as the bits to be encrypted increase and does not significantly increase as the length of the module increases.

Table 2 Ciphertext expansion and noise comparison

Cryptosystem	G-M	O-U	Paillier	FHE-derived [a)]	Ours
Ciphertext expansion	$2k$	$3k$	$4k$	16000	$2k$

a) FHE refers to the FHE from [35].

Computational complexity. Goldwasser-Micali (G-M), Okamoto-Uchiyama (O-U), Paillier and FHE-derived cryptosystems as well as our new cryptosystems can be used to encrypt zeros and ones. To encrypt a bit, the G-M cryptosystem requires only a modular multiplication, the O-U cryptosystem requires a modular exponentiation, the Paillier cryptosystem requires a modular exponentiation and our cryptosystem requires two modular exponentiations. The computational complexity of the G-M cryptosystem is the lowest. Because it is challenging to theoretically analyze the complexity of FHE cryptosystem, we conduct an experimental comparison.

We do not know the implementation detail of [35]; we accept its result. There is no data to reveal the time required to encrypt a bit using the fastest FHE. Ref. [36] demonstrates that the bootstrapping of a gate requires 13 ms (on a 2.9 GHz computer with 64-bit single core (i7-4910MQ)).

We implement our cryptosystem on a benchmark computer with an Intel(R) Core(T) i5-6600 3.30 GHz, 3.31 GHz, 8.00 GHz RAM, myeclipse 10 and without optimization. We choose three prime numbers with 512, 768 and 1024 bits to test. The results are shown in Figure 1.

Ciphertext expansion. Because the G-M, the O-U, the Paillier and the FHE-derived cryptosystems as well as our new cryptosystem can be used to encrypt zeros and ones, we compare their ciphertext expansion when the plaintext space is $\{0,1\}$. Suppose that the bit number of p used in our cryptosystem is k and that the p,q used in the G-M, the O-U and the Paillier cryptosystems are also k-bit prime numbers. Their ciphertext expansions are presented in Table 2. Typically, k =768.

Remark 2. By the De Morgan law, $\overline{x \vee y} = \overline{x} \wedge \overline{y}$, this variant can be used to privately compute $x \vee y$ as follows: (1) Alice and Bob compute $E(\overline{x}), E(\overline{y})$; (2) compute $c = E(\overline{x}) \cdot E(\overline{y})$; (3) decrypt c to obtain $\mathrm{Dec}(c) = \overline{x} \wedge \overline{y} = \overline{x \vee y}$; (4) flip the value of $\mathrm{Dec}(c)$ to obtain $x \vee y$.

4 Applications

Homomorphic cryptosystems are highly effective primitives for addressing SMC problems. The AND-homomorphic cryptosystem can be directly used by multiple parties to privately address numerous SMC problems in a trivial manner, such as the problems described in the introduction, i.e., privately computing $P(x_1, \ldots, x_t) = x_1 \wedge x_2 \wedge \cdots \wedge x_t$, and $Q(x_1, \ldots, x_t) = x_1 \vee x_2 \vee \cdots \vee x_t$.

In this section, we provide a few examples to illustrate the applications of this new homomorphic cryptosystem to privately address the set intersection problem and 2-DNF problem in the semi-honest model. Rather than demonstrate that the solution to a specific problem based on the new homomorphic cryptosystem will certainly outperform the available solutions, we demonstrate that our new homomorphic

cryptosystem can provide a new tool to solve these problems and that in certain cases, it is likely to be the most effective tool, e.g., to privately compute the intersection of over two private sets.

4.1 Security of SMC

In SMC, we generally do not consider the external attackers and rather consider three (models) types of internal attackers: (1) fully honest model; (2) semi-honest model, or passive attacker model, or honest-but-curious attacker, and (3) malicious model or active attacker model.

A fully honest party would properly follow the prescribed protocol and would delete the record of all its intermediate computations at the end of the protocol. A protocol that is secure against fully honest attackers is the weakest secure protocol, and it is not reasonable to ask a party to necessarily delete its record. Therefore, a protocol that is secure against fully honest parties is not considered in SMC.

Loosely speaking, a semi-honest party is one who follows the protocol properly with the exception that it keeps a record of all its intermediate computations. He may use the record to attempt to derive other parties' private inputs. This constrained model may be justified in certain settings and certainly provides an effective methodological locus. In addition to being of independent interest, the semi-honest model will play a major role in the construction of protocols for the malicious model [37].

A malicious party is likely to arbitrarily deviate from the prescribed functionality. However, we do not consider three malicious behaviours [37]: (1) parties refusing to participate in the protocol; (2) parties substituting their local input (and entering the protocol with an input other than the one provided to them), and (3) parties aborting the protocol prematurely (e.g., before sending their last message). A protocol that is secure in the malicious model should prevent all other malicious behaviours.

Goldreich [37] designed a compiler which can automatically produce a protocol that is secure in the malicious model, given a protocol that is secure in the semi-honest model. Furthermore, whereas general malicious behaviour may be infeasible for numerous users, semi-honest behaviour is likely to be feasible for them (moreover, one cannot assume that they behave only in a fully honest way). Consequently, in a number of settings, one may assume that although the users are likely to desire to cheat, they can behave in a semi-honest way. Therefore, designing protocols that are secure in the semi-honest model is of high theoretical and practical significance.

Definition 2 (Security in semi-honest model [37]). Let $f : (\{0,1\})^m \to (\{0,1\})^m$ be m-ary functionality, where $f_i(x_1, \ldots, x_m)$ denotes the i-th element of $f(x_1, \ldots, x_m)$. For $I = \{i_1, \ldots, i_s\} \subseteq [m] = \{1, \ldots, m\}$, we let $f_I(x_1, \ldots, x_m)$ denote the subsequence $f_{i_1}(x_1, \ldots, x_m), \ldots, f_{i_s}(x_1, \ldots, x_m)$. Let Π be m-party protocol for computing f. The view of the i-th party during an execution of Π on $\overline{x} = (x_1, \ldots, x_m)$, denoted as $\mathrm{view}_i^{\Pi}(\overline{x})$, is defined as $(x_i, r_i, m_i^1, \ldots, m_i^\tau)$; here, r_i represents the outcome of P_i's internal coin tosses and m_i^j represents the j-th message P_i has received. Moreover, for $I = \{i_1, \ldots, i_s\}$, we let $\mathrm{view}_I^{\Pi}(\overline{x}) = (I, \mathrm{view}_{i_1}^{\Pi}(\overline{x}), \ldots, \mathrm{view}_{i_s}^{\Pi}(\overline{x}))$.

In case f is a deterministic m-ary functionality, we consider that Π privately computes f if there exists a probabilistic polynomial-time algorithm, denoted by S, such that for each $I \subseteq [m]$, it holds that

$$\{S(I, (x_{i_1}, \ldots, x_{i_s}), f_I(\overline{x}))\}_{\overline{x} \in (\{0,1\}^*)^m} \stackrel{c}{\equiv} \{\mathrm{view}_I^{\Pi}(\overline{x})\}_{\overline{x} \in (\{0,1\}^*)^m},$$

where $\stackrel{c}{\equiv}$ denotes computational indistinguishability.

4.2 Set intersection

There are t parties $P_1, \ldots, P_t$, and each party has a private set $X_i \subseteq U$. They wish to compute the intersection of these sets $X = X_1 \cap X_2 \cap \cdots \cap X_t$ without disclosing X_i.

This problem has numerous practical applications and has been extensively studied. Meadows [45] introduced the first private set intersection protocol with high communication complexity. Freedman et al. [46] proposed a protocol based on oblivious polynomial evaluation. Kissner et al. [47] extended and improved Freedman's approach. The most efficient protocol was proposed by Pinkas et al. [48] and was improved in [49, 50]. Chen et al. [51] proposed the first FHE based set intersection protocol, and we

will compare the computational efficiency of our protocol with that of this protocol subsequently. We are not going to review these studies and compare our protocol with all efficient solutions; however, our solution may not outperform the solution proposed in [46,47,51]. We wish to demonstrate that in certain scenarios, where the available solutions are not effective, our solution is effective.

The above-mentioned protocols for this problem are applicable only to two parties, i.e., $t = 2$; moreover, it is highly challenging to extend these protocols to multiparty (over two parties) cases. Although we can first compute $X_1 \cap X_2$ and then compute $X_1 \cap X_2 \cap X_3$, etc., computing in this manner will disclose certain information that should not be disclosed. We develop a protocol applicable to both two-party and multiparty cases. In a multiparty case, collusion attack must be considered. To resist collusion attack, a threshold decryption cryptosystem is necessary. We use the ElGamal cryptosystem to construct a threshold AND homomorphic cryptosystem.

In (m, t) threshold decryption cryptosystem, the public key is open, whereas the private key is shared among t parties. Over m parties can cooperate to decrypt a ciphertext; however, parties numbering less than m cannot obtain anything about the plaintext. In SMC, we wish to resist the collusion attack launched by the maximum possible number of parties; therefore, we require a simple (t, t) threshold decryption cryptosystem. A (t, t) threshold decryption ElGamal cryptosystem can be constructed as follows.

KeyGen. To use the threshold decryption ElGamal cryptosystem with prime number p and generator α of Z_p^*, each party P_i chooses a random number x_i as its secret key, computes $h_i = \alpha^{x_i} \bmod p$ and publishes h_i. All the parties cooperate to compute the public key:

$$h = \prod_{i=1}^{t} h_i = \prod_{i=1}^{t} \alpha^{x_i} = \alpha^{x_1+\cdots+x_t} (\text{mod } p).$$

Enc. To encrypt $M \in Z_p^*$, choose a random number $r \in Z_p^*$, and compute

$$E(M) = (c_1, c_2) = (\alpha^r \bmod p, Mh^r \bmod \ p).$$

Dec. To decrypt ciphertext (c_1, c_2), each party computes $y_i = c_1^{x_i} \bmod \ p$. Finally, they compute

$$M = c_2 \left(\prod_{i=1}^{t} y_i \bmod p \right)^{-1} \bmod \ p.$$

Correctness.

$$c_2 = Mh^r \bmod p = M\alpha^{(x_1+\cdots+x_t)r} \bmod \ p,$$

$$\prod_{i=1}^{t} y_i = \prod_{i=1}^{t} c_1^{x_i} = c_1^{x_1+\cdots+x_t} = \alpha^{r(x_1+\cdots+x_t)} (\text{mod } p)$$

$$\Rightarrow c_2 \left(\prod_{i=1}^{t} y_i \right)^{-1} = M\alpha^{(x_1+\cdots+x_t)r} \alpha^{-r(x_1+\cdots+x_t)} = M (\text{mod } p).$$

With a specific encoding transformation, the intersection problem can be solved using the AND homomorphic cryptosystem. Suppose $U = \{u_1, \ldots, u_m\}$ and $u_1 < u_2 < \cdots < u_m$. The parties can address the problem as Protocol 1.

Correctness. In this threshold decryption cryptosystem, $c_j = E(\prod_{i=1}^{t} s_{ij})$. $s_j \in \bar{I}(v_j = 1)$ implies that all $s_{ij} \in \bar{I}(i = 1, \ldots, t)$ and all $v_{ij} = 1$. By the encoding scheme, this implies that $u_j \in X_i$ $(i = 1, \ldots, t)$. Therefore, $u_j \in X_1 \cap \cdots \cap X_t$.

Security. The cryptosystem is probabilistic; therefore, by using the simulation paradigm [37], we can establish the following theorem.

Theorem 2. Protocol 1 privately computes the intersection set of private sets $X_1, \ldots, X_t$ in the semi-honest model.

Protocol 1 Privately compute $X = X_1 \cap \cdots \cap X_t$ in the semi-honest model

Inputs: Private sets $X_1, \ldots, X_t$.
Output: $X = X_1 \cap \cdots \cap X_t$.
Setup: All the parties cooperate to generate a public key $h = \prod_{i=1}^{t} h_i \bmod p$. P_i keeps x_i private. The secret key is shared among all the t parties. They also agree that s_{ij} should be chosen from a subset of I and should be adequately small. For example, $s_{ij} \in \{2, 4, 6, 8, 12, 16, 18, 20\}$.
(1) Each party P_i $(i = 1, \ldots, t)$ constructs a vector $V_i = (v_{i1}, \ldots, v_{im})$ as follows: if $u_j \in X_i$, $v_{ij} = 1$; otherwise, $v_{ij} = 0$. P_i encrypts V_i with the public key h to obtain

$$C_i = (c_{i1}, \ldots, c_{im}) = (E(v_{i1}), \ldots, E(v_{im})),$$

where $c_{ij} = (\alpha^{r_{ij}} \bmod p, s_{ij} h^{r_{ij}} \bmod p)$. If $v_{ij} = 0$, $s_{ij} \in I \backslash \{0\}$; otherwise, $s_{ij} \in \bar{I}$. P_i publishes C_i.
(2) Each party can compute

$$C = \prod_{i=1}^{t} C_i = \left(\prod_{i=1}^{t} c_{i1}, \ldots, \prod_{i=1}^{t} c_{im} \right) = (c_1, \ldots, c_m),$$

where $c_j = (\alpha^{r_{1j} + \cdots + r_{tj}} \bmod p, (\prod_{i=1}^{t} s_{ij}) h^{r_{1j} + \cdots + r_{tj}} \bmod p) = (\alpha^{r_j} \bmod p, s_j h^{r_j} \bmod p)$.
(3) All the parties cooperate to decrypt C to obtain $S = (s_1, \ldots, s_m)$. Then, set $X = \phi$. If $s_j \in \bar{I}$ $(j = 1, \ldots, m)$ and $u_j \in X$, set $X \leftarrow X \cup \{u_j\}$; otherwise, $u_j \notin X$.

Proof. In our protocol, all the parties have an identical status. By Definition 2, it suffices to prove that for the largest collusion structure set $T = \{P_1, \ldots, P_{t-1}\}$ (Even if all the members of T cooperate, they cannot obtain information beyond what X reveals. Neither can a subset of T obtain further information), there exists a polynomial time algorithm S such that

$$\{S(T, (x_1, \ldots, x_{t-1}), f_T(\overline{x}))\}_{\overline{x} \in (\{0,1\}^*)^m} \stackrel{c}{\equiv} \{\mathrm{view}_T^{\Pi}(\overline{x})\}_{\overline{x} \in (\{0,1\}^*)^m}.$$

During the execution, $f_T(\overline{x}) = X_1 \cap \cdots \cap X_t$,

$$\mathrm{view}_T^{\Pi}(\overline{x}) = ((x_1, \ldots, x_{t-1}), (r_1, \ldots, r_{t-1}), \mathrm{view}_1^{\Pi}(\overline{x}), \ldots, \mathrm{view}_{t-1}^{\Pi}(\overline{x})).$$

As a group, the messages of the view that are not generated by the group are $E(v_{t1}), \ldots, E(v_{tm})$. Even if all the members of T conspire, they cannot decrypt $E(v_{t1}), \ldots, E(v_{tm})$ without P_t's cooperation. $\mathrm{view}_T^{\Pi}(\overline{x})$ can be expressed as follows:

$$\mathrm{view}_T^{\Pi}(\overline{x}) = ((x_1, \ldots, x_{t-1}), (r_1, \ldots, r_{t-1}), E(v_{t1}), \ldots, E(v_{tm})).$$

Algorithm S operates as follows: on inputs $(T, (X_1, \ldots, X_{n-1}), f_T(\overline{x}))$, S chooses X_t' such that $f_T(X_1, \cdots, X_{t-1}, X_t') = f_T(X_1, \ldots, X_{t-1}, X_t)$, and then simulates the execution of the protocol with $X_1, \ldots, X_{t-1}, X_t'$ to obtain $E(v_{t1}'), \ldots, E(v_{tm}')$. Because the cryptosystem used in the protocol is probabilistic,

$$E(v_{t1}), \ldots, E(v_{tm}) \stackrel{c}{\equiv} E(v_{t1}'), \ldots, E(v_{tm}').$$

Let $S(T, (x_1, \ldots, x_{t-1}), f_T(\overline{x})) = ((x_1, \ldots, x_{t-1}), r, E(v_{t1}'), \ldots, E(v_{tm}'))$. We have

$$\{S(T, (x_1, \ldots, x_{t-1}), f_T(\overline{x}))\}_{\overline{x} \in (\{0,1\}^*)^m} \stackrel{c}{\equiv} \{\mathrm{view}_T^{\Pi}(\overline{x})\}_{\overline{x} \in (\{0,1\}^*)^m}.$$

This completes the proof.

Computational complexity. The computational complexity of Protocol 1 depends on the number of parties and the cardinality of set U. It requires t modular exponentiations to generate the public key. Because each party needs to encrypt m bits, they need to totally encrypt mt bits. Encrypting a bit requires two modular exponentiations, and totally, $2mt$ modular exponentiations are required. They need to decrypt m bits and each cooperative decryption requires $t + 2$ modular exponentiations. The protocol totally requires $2mt + m(t + 2) + t$ modular exponentiations.

Intersection cardinality. There are t parties $P_1, \ldots, P_t$, and each party has a private set $X_i \subseteq U$. They wish to compute $|X_1 \cap X_2 \cap \cdots \cap X_t|$ without disclosing either X_i or X. Suppose $U = \{u_1, \ldots, u_m\}$ and $u_1 < u_2 < \cdots < u_m$. A minor modification of Protocol 1 (P_t and P_1 each re-randomize and randomly permutes $(c_{t1}, c_{t2}, \ldots, c_{tm})$ one time; performing this will keep the elements of the intersection set private) can straightforwardly solve this problem.

Subset problem. Alice has a private set $X = \{x_1, \ldots, x_a\}$, and Bob has a private set $Y = \{y_1, \ldots, y_b\}$, with $X, Y \subseteq U$ and $a \leqslant b$. They wish to determine whether X is a subset of Y without disclosing X, Y. Suppose $U = \{u_1, \ldots, u_m\}$ and $u_1 < u_2 < \cdots < u_m$. As $X \subseteq Y \Leftrightarrow |X \cap Y| = |X|$, we can also straightforwardly solve this problem by using the protocol for the intersection cardinality.

Set union problem. By the De Morgan law, we can transform OR operations to AND operations. Therefore, we can also use the AND homomorphic cryptosystem to privately compute the union of private sets $X_1, \ldots, X_t$.

4.3 Implementation and performance

We implement our private set intersection protocol described above and compare the results with that of the available protocol based on FHE. We perform a test for a two-party case. We do not consider collusion attack of the parties and therefore, the threshold decryption cryptosystem is not necessary. Thus, the key generation requires only one modular exponentiation, and decryption only requires $2m$ modular exponentiations. The computational complexity is $6m+1$ modular exponentiations. We test 20 times and compute the average time required without preprocessing. The benchmark computer has an Intel(R) Core(T) i5-6600 3.30 GHz, 3.31 GHz 8.00 GHz RAM with myeclipse 10. Suppose $|U| = 10000$, and that a 512-bit prime number p is chosen. For any two private sets $X, Y \subseteq U$, the execution time of the protocol is approximately 2.9 s. Because our protocol involves only the encryption of one and zero and all the encryptions can either be outsourced to a cloud or be preprocessed, its performance can be significantly improved. This is highly critical for a mobile equipment such as smart phone.

The most computationally efficient PSI protocols have been constructed using tools such as hash functions and oblivious transfer; however, a potential limitation with these approaches is the communication complexity which increases linearly with the size of the larger set. The fastest FHE-derived protocol proposed in [51] uses batching and hashing to mainly reduce the communication complexity and uses windowing and partitioning to reduce the circuit depth. Even with these optimizations, its execution time on a benchmark computer with two 18-core Intel Xeon CPU E5-2699 v3 2.3 GHz and 256 GB of RAM is approximately 120 s. The advantage of this protocol is its low communication complexity. It functions particularly well when one of the two sets is significantly smaller than the other. For example, if the private set sizes are N_Y, N_X and $N_Y \ll N_X$, the communication overhead will be $O(N_Y \log N_X)$. This is important when a user has a constrained device (smart phone) and a service provider has a large set and they wish to privately compute the intersection, such as in the private contact discovery application.

4.4 DNF problem

DNF problem. Alice has a private vector $X = (x_1, \ldots, x_m) \in \{0,1\}^m$, and Bob has another private vector $X = (x_1, \ldots, x_m) \in \{0,1\}^m$. They wish to cooperatively compute

$$P(X, Y) = (x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_m \wedge y_m)$$

without disclosing X, Y. Using the threshold decryption AND homomorphic cryptosystem, we can privately and more efficiently compute any DNF (real DNF), i.e., mDNF, for variables in $\{0,1\}$. For example, the 2DNF problem can be addressed as Protocol 2.

This protocol can be conveniently extended to privately compute mDNF

$$P(X_1, \ldots, X_m) = (x_{11} \wedge \cdots \wedge x_{m1}) \vee \cdots \vee (x_{1n} \wedge \cdots \wedge x_{mn}). \tag{6}$$

Security. For the security of this protocol, we have the following theorem.

Theorem 3. Protocol 2 privately computes a 2DNF.

We can use the simulation paradigm to prove the security of this theorem similarly as the proof of Theorem 3. Owing to the space limitation, we omit the proof.

Remark 3. With certain transformation, the AND homomorphic cryptosystem can be used to privately compute the Hamming distance between two private binary strings.

Protocol 2 Privately compute 2DNF

Inputs: $X = (x_1, \ldots, x_m),\ Y = (y_1, \ldots, y_m) \in \{0,1\}^m$.
Output: $(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_m \wedge y_m)$.
Setup: Alice and Bob cooperate to generate a public key h. The private key sk is shared by Alice and Bob.
(1) Alice encrypts $X = (x_1, \ldots, x_m)$ using the threshold decryption AND-homomorphic cryptosystem with the public key pk to obtain $C_1 = (E(x_1), \ldots, E(x_m))$ and transmits it to Bob.
(2) Bob performs the following:

- Encrypts $Y = (y_1, \ldots, y_m)$ using the threshold decryption AND-homomorphic cryptosystem with the public key pk to obtain $C_2 = (E(y_1), \ldots, E(y_m))$.
- Computes $C = C_1C_2 = (E(x_1)E(y_1), \ldots, E(x_m)E(y_m)) = (c_1, c_2, \ldots, c_m)$.
- Generates a random permutation of C, denoted by $\pi_1(C) = (c_{\pi_1(1)}, \ldots, c_{\pi_1(m)})$ and transmits $\pi_1(C)$ to Alice.

(3) Alice re-randomizes $\pi_1(C)$ by multiplying each element of $\pi_1(C)$ with $E(1)$, permutes it and publishes

$$\pi(C) = (c_{\pi(1)}, \ldots, c_{\pi(m)}) = (c_{\pi_2\pi_1(1)}, \ldots, c_{\pi_2\pi_1(m)}).$$

(4) For $i = 1$ to m, Alice and Bob cooperate to decrypt $c_{\pi(i)}$. If there exists i such that $D(c_{\pi(i)}) = 1$, terminate the protocol and output $(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_m \wedge y_m)$=1; otherwise, output $(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_m \wedge y_m)$=0.
Because $(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_m \wedge y_m)$=0 iff $\forall i\ (x_i \wedge y_i) = 0$. If there exists i such that $x_i \wedge y_i = 1$, then $(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_m \wedge y_m)$=1. This is the reason why step (4) works.

5 Conclusion

To solve the problems listed in the introduction, we analysed the property of the available homomorphic cryptosystems and used the properties of ideal of abstract algebra and multiplicatively homomorphic cryptosystems to design AND homomorphic one. Our main concept is to modify the original homomorphic cryptosystems so they can be used to encrypt zeros and ones such that the ciphertexts of zeros and those of ones are indistinguishable while maintaining the original homomorphic property. This new homomorphic cryptosystem can be used to solve the problems listed in the introduction as well as others. We also demonstrated the applications of the scheme to solve other SMC problems in the semi-honest model. Compared to TFHE derived AND homomorphic cryptosystems, our AND homomorphic cryptosystem is more time-space efficient, albeit less functional. Compared to the schemes of [29, 30], our scheme is more effective in terms of latency, although not in terms of amortized timing or expansion rate. It can be used to privately compute certain Boolean formulae straightforwardly, and the protocols based on it are conveniently constructed and understood. In the future, we will explore the feasibility of modifying this scheme so it is AND homomorphic, as well as method for reducing the ciphertext expansion, and the feasibility of applying them to solve certain SMC problems in the malicious model.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 62172435). We are deeply grateful to all reviewers.

References

1 Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms. Found Secure Comput, 1978, 4: 169–180
2 López-Alt A, Tromer E, Vaikuntanathan V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Proceedings of the 44th ACM Symposium on Theory of Computing, 2012. 1219–1234
3 Atayero A A, Feyisetan O. Security issues in cloud computing: the potentials of homomorphic encryption. J Emerg Trends Comput Inf Sci, 2011, 2: 546–552
4 Zhang R, Ma H, Lu Y, et al. Provably secure cloud storage for mobile networks with less computation and smaller overhead. Sci China Inf Sci, 2017, 60: 122104
5 Kuang L W, Yang L T, Feng J, et al. Secure tensor decomposition using fully homomorphic encryption scheme. IEEE Trans Cloud Comput, 2018, 6: 868–878
6 Anunay K, Akshay R, Matthew D, et al. Cryptographically secure multiparty computation and distributed auctions using homomorphic encryption. Cryptography, 2017, 1: 25
7 Lin H Y, Tzeng W G. An efficient solution to the millionaires' problem based on homomorphic encryption. In: Proceedings of the 3rd International Conference Applied Cryptography and Network Security, 2005. 456–466
8 Jiang B B, Zhang Y. Securely min and k-th min computations with fully homomorphic encryption. Sci China Inf Sci, 2018, 61: 058103
9 Wang W, Hu Y, Chen L, et al. Exploring the feasibility of fully homomorphic encryption. IEEE Trans Comput, 2015, 64: 698–706
10 Cramer R, Damgård I B, Nielsen J B. Secure Multiparty Computation. Cambridge: Cambridge University Press, 2015

11 Rivest R L, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Commun ACM, 1978, 21: 120–126
12 ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. In: Proceedings of Annual International Cryptology Conference, Santa Barbara, 1984. 10–18
13 Rabin M O. Digitalized Signatures and Public-key Functions as Intractable as Factorization. Massachusetts INST of Tech Cambridge Lab For Computer Science, Technical Report, No. ADA078415. 1979
14 Okamoto T, Uchiyama S. A new public-key cryptosystem as secure as factoring. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, Espoo, 1998. 308–318
15 Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, Prague, 1999. 223–238
16 Miller V. Use of elliptic curves in cryptography. In: Proceedings of Annual International Cryptology Conference, Santa Barbara, 1985. 417–426
17 Hoffstein J, Pipher J, Silverman J H. NTRU: a ring-based public key cryptosystem. In: Proceedings of the 3rd International Symposium on Algorithmic Number Theory, Portland, 1998. 267–288
18 Goldreich O, Goldwasser S, Halevi S. Public-key cryptosystems from lattice reduction problems. In: Proceedings of Annual International Cryptology Conference, Santa Barbara, 1997. 112–131
19 Hu Y P, Jia H W. Cryptanalysis of GGH map. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, 2016. 537–565
20 Benaloh J. Dense probabilistic encryption. In: Proceedings of the Workshop on Selected Areas of Cryptography, Kingston, 1994. 120–128
21 Naccache D, Stern J. A new public key cryptosystem based on higher residues. In: Proceedings of the 5th ACM conference on Computer and Communications Security, San Francisco, 1998. 59–66
22 Damgård I, Jurik M. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In: Proceedings of Public Key Cryptosystem, 2001. 119–136
23 Ishai Y, Paskin A. Evaluating branching programs on encrypted data. In: Proceedings of International Theory of Cryptography Conference, Amsterdam, 2007. 575–594
24 Goldwasser S, Micali S. Probabilistic encryption. J Comput Syst Sci, 1984, 28: 270–299
25 Boneh D, Goh E J, Nissim K. Evaluating 2-DNF formulas on ciphertexts. In: Proceedings of International Theory of Cryptography Conference, Cambridge, 2005. 325–341
26 Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, Bethesda, 2009. 169–178
27 Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping. ACM Trans Comput Theor, 2014, 6: 1–36
28 Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE. SIAM J Comput, 2014, 43: 831–871
29 Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping. In: Proceedings of Innovations in Theoretical Computer Science, Cambridge, 2012. 309–325
30 Fan J F, Vercauteren F. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144. http://eprint.iacr.org/
31 Cheon J H, Kim A, Kim M, et al. Homomorphic encryption for arithmetic of approximate numbers. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, 2017. 409–437
32 Chillotti I, Gama N, Georgieva M, et al. Faster fully homomorphic encryption: bootstrapping in less than 0.1 seconds. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, 2016. 3–33
33 Chillotti I, Gama N, Georgieva M, et al. Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, 2017. 377–408
34 Chillotti I, Gama N, Georgieva M, et al. TFHE: fast fully homomorphic encryption over the torus. IACR Cryptol ePrint Arch, 2018, 2018: 421
35 Chillotti I, Gama N, Georgieva M, et al. Faster fully homomorphic encryption: bootstrapping in less than 0.1 seconds. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, 2016. 3–33
36 Chillotti I, Gama N, Georgieva M, et al. TFHE: fast fully homomorphic encryption over the torus. https://eprint.iacr.org/2018/421
37 Goldreich O. Foundations of Cryptography: Basic Applications. Cambridge: Cambridge University Press, 2004
38 Nielsen J B, Nordholt P S, Orlandi C, et al. A new approach to practical active-secure two-party computation. In: Proceedings of Annual International Cryptology Conference, Santa Barbara, 2012. 681–700
39 Naehrig M, Lauter K, Vaikuntanathan V. Can homomorphic encryption be practical? In: Proceedings of the 3rd ACM Cloud Computing Security Workshop, Chicago, 2011. 113–124
40 Sander T, Young A, Yung M. Non-interactive crypto-computing for NC^1. In: Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science, 1999. 554–566
41 Fischlin M. A cost-effective pay-per-multiplication comparison method for millionaires. In: Proceedings of the Cryptographer's Track at the RSA Conference, San Jose, 2001. 457–471
42 Barbulescu R, Gaudry P, Joux A, et al. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of

small characteristic: improvements over FFS in small to medium characteristic. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, 2014. 1–16
43 Menezes A, Sarkar P, Singh S. Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography. In: Proceedings of International Conference on Cryptology in Malaysia, Kuala Lumpur, 2016. 83–108
44 Thomas W J, Stephen F. Abstract Algebra Theory and Applications. Nacogdoches: Austin State University Press, 2014
45 Meadows C. A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party. In: Proceedings of 1986 IEEE Symposium on Security and Privacy, Oakland, 1986. 134
46 Freedman M J, Hazay C, Nissim K, et al. Efficient set intersection with simulation-based security. J Cryptol, 2016, 29: 115–155
47 Kissner L, Song D. Privacy-preserving set operations. In: Proceedings of Annual International Cryptology Conference, Santa Barbara, 2005. 241–257
48 Pinkas B, Schneider T, Segev G, et al. Phasing: private set intersection using permutation-based hashing. In: Proceedings of the 24th USENIX Security Symposium, Washington, 2015. 515–530
49 Pinkas B, Schneider T, Zohner M. Scalable private set intersection based on OT extension. ACM Trans Priv Secur, 2018, 21: 1–35
50 Orrú M, Orsini E, Scholl P. Actively secure 1-out-of-n OT extension with application to private set intersection. In: Proceedings of Cryptographers' Track at the RSA Conference, San Francisco, 2017. 381–396
51 Chen H, Laine K, Rindal P. Fast private set intersection from homomorphic encryption. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Dallas, 2017. 1243–1255