박사 학위논문

Ph. D. Dissertation

분산 및 이동 네트워크에서 프라이버시 보호를 제공하는 정보 수집과 인증에 관한 연구

Privacy-Preserving Information Aggregation and Authentication in the Distributed and Mobile Networks

김 진 (金 振 Kim, Zeen)

정보통신공학과

Department of Information and Communications Engineering

KAIST

2012

분산 및 이동 네트워크에서 프라이버시 보호를 제공하는 정보 수집과 인증에 관한 연구

Privacy-Preserving Information Aggregation and Authentication in the Distributed and Mobile Networks

Privacy-Preserving Information Aggregation and Authentication in the Distributed and Mobile Networks

Advisor : Professor Kim, Kwangjo

by

Kim, Zeen

Department of Information and Communications Engineering

KAIST

A thesis submitted to the faculty of KAIST in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Information and Communications Engineering . The study was conducted in accordance with Code of Research Ethics[1].

2012. 05. 29.

Approved by

Professor Kim, Kwangjo

[Advisor]

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of KAIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

분산 및 이동 네트워크에서 프라이버시 보호를 제공하는 정보 수집과 인증에 관한 연구

김 진

위 논문은 한국과학기술원 박사학위논문으로
학위논문심사위원회에서 심사 통과하였음.

2012년 05월 29일

심사위원장	김 광 조	(인)
심사위원	강 준 혁	(인)
심사위원	이 병 천	(인)
심사위원	최 두 호	(인)
심사위원	하 정 석	(인)

DICE
20045909

김 진. Kim, Zeen. Privacy-Preserving Information Aggregation and Authentication in the Distributed and Mobile Networks. 분산 및 이동 네트워크에서 프라이버시 보호를 제공하는 정보 수집과 인증에 관한 연구. Department of Information and Communications Engineering . 2012. 63p. Advisor Prof. Kim, Kwangjo. Text in English.

ABSTRACT

The development of computer system and network enabled us to provide various applications in the field of information technology. As the exchange of information increases, the protection of the privacy becomes essential factor in the information sharing. Among the technology of the information security, this study attempted to seek a solution on the subject of secure data collection in distributed and mobile environments, and on the subject of method of effective authentication between multiple users.

The first study is on the collection and sharing of the data in distributed network while preserving the privacy preserving function. To solve this, several research has been done on the secure set operations in distributed network. We developed an polynomial operation algorithm in encrypted polynomials to improve the speed of set operation based on the polynomial representation among the existing method. Applying the method suggested by Karatsuba and divide-and-conquer method, we suggested a calculation method of polynomial multiplication, expansion, evaluation. As the result of this method, the complexity of the operation as reduced from original $\mathcal{O}(k^2)$ to $\mathcal{O}(k^{\log_2 3})$, when the cardinality of private set of the user was k.

For the problem of the distributed and multiparty communications, we suggested a method to search for the top-k elements. Earlier researches failed to provide strict attack and security goals, thus, failed to provide provable security. This results applied provably-secure concepts of PPT-k protocols, in the first time, in the method to search for top-k elements. We also defined owner privacy, which is a new security notion. Furthermore, we suggested a method to search for the top-k elements with provable security based on suggested security notion, and verified its security rigorously.

The second study is to designing method of authentication in mobile network (especially in vehicular ad-hoc network.) We studied improvement of the speed of authentication in the situation of the traffic jam, which has been a problem of the previous method that provided anonymity. To this end, we suggested a method of fast certification process, and to minimize the participation of road side units (RSUs). We achieved sufficient improvement of speed using BGN encryption, a kind of homomorphic encryption algorithm, reduced number of participation of RSUs, and suggested a method to cross-check between certified vehicles. The suggested method was able to solve the bottleneck problem of authentication when the traffic jam occurred.

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Overview and problem statements

In many applications, a collection of mutually distrustful parties must share information without compromising their privacy. The problem of *privacy-preserving distributed information sharing* is to allow parties with private data sets to compute these joint functions without use of a trusted party, and thus achieve many of the benefits obtained from combining the data sets without undesirably revealing private data. Protocols for privacy-preserving distributed information sharing must also be designed with a number of practical concerns. Firstly, many data sets are extremely large; protocols that operate on large data sets must be efficient in order to operate in the real world. In addition, we must be concerned with robustness. Recently, there proposed many techniques for privacy-preserving distributed information sharing using polynomial as a private set. We called these protocols as polynomial representation based protocols. The computational complexity of polynomial representation based protocols totally depends on the complexity of polynomial arithmetics on the encrypted polynomial. Until now, computational complexity of previous secure set operation schemes [21, 39, 59, 60, 61] are quadratic, $\mathcal{O}(k^2)$ where k is the cardinality of private sets or the degree of private polynomials. This complexity is not enough to some critical application. For example real-time monitoring system for information warfare requires fast and secure set operations for finding out the origin of threats.

Of particular interest in many applications is the problem of computing the top-k in a privacy-preserving manner. A typical scenario of an application that involves such primitive is network traffic distribution, in which n network sensors need to jointly analyze the security alert broadcasted of different sources in order to find the top-k suspect sites. In such an application, and without losing generality, each of such sensors—supposedly under control of different authorities—has a set of suspects and would like to collaboratively compute the top most frequent on each of these sets (say k of them) without revealing the set of suspects to other sensor with whom it collaborates. There has been existed several solution for PPT-k problem [5, 62, 67]. But there's no rigorous security notion on PPT-k. Therefore there is no provably-secure PPT-k protocol.

Vehicular Ad-hoc Networks (VANETs) are one of typical applications of wireless communication technology, which provide communications among nearby vehicles and between vehicles and roadside units (RSUs) connected to the infrastructure. VANETs provide a perfect way to collect dynamic traffic information and sense various physical conditions related to traffic distribution with very low cost and high accuracy, which have a great potential to revolutionize driving environment, and will undoubtedly play an important role in the transportation system of the future. However, it is clear that security and privacy enhancing mechanisms are necessary, which are in fact a prerequisite for deployment. And a large number of agreed efforts have been undertaken to design security architectures for VANET systems. Extensive research efforts have been made by both industry and academia to make VANETs secure. To provide the privacy of the drivers in VANET, various anonymous authentication protocols [43, 45, 70] have been proposed. However, these protocols can allow the target vehicle to communicate with the nearby vehicles through the trusted authority. As the number of the vehicles in certain location area increases, these protocol may suffer the bottleneck problem in the nearby RSUs. For instance, during lunar holidays in Asian countries, as the Asian people visit their hometown using public transportation (*i.e.*, car and train), the traffic on highways is heavy. Due to the number of vehicles in the certain area, the nearby RSUs cannot support the numerous number of authentication requests. Also, the time delay to authenticate the nearby vehicle will increase.

In this thesis we try to answer the following questions about information sharing protocol and anonymous authentication scheme in VANETs.

- How to get more efficient and secure set operations under the polynomial representation.
- How to define the privacy-preserving top-k query protocol in provable security concepts and give a well-defined adversarial model for PPT-k protocol.
- How to solve the authentication bottleneck problem when traffic jam is occurred in VANETs.

As answer to the first question, we construct fast polynomial computation algorithms of encrypted polynomial. We give a fast polynomial multiplication of ciphertext, polynomial expansion, and polynomial evaluation. Our proposed algorithms reduce the previous complexity from quadratic to subquadratic.

For solving the second question, we introduce a formal structure of PPT-k protocol which consists of five probabilistic polynomial-time algorithms (Setup, Encrypt, Shuffle, Aggregate, Reveal) with formal security model including adversarial behavior and attack goals. Based on our suggestion, we propose a generic PPT-k protocol using homomorphic encryption.

Finally, we design a scalable authentication scheme using homomorphic encryption and keyword search on ciphertexts as an answer to the third question. We reduce the participation of road side units (RSUs) and fast verifying the authenticity between legitimated vehicles.

1.2 Contribution of this work

1.2.1 Speeding up polynomial arithmetics of encrypted polynomial

We claim that the efficient operations can be obtained by applying the divide-and-conquer strategy to set operation protocols. We discuss more general method that, in many (if not all) situations, allows set operation protocols to be computed at most in sub-quadratic complexity with respect to the user's input size without changing the protocols. Even though not universally applicable (our work is efficient polynomial arithmetic), it can be used for most of set operation schemes via the polynomial representation.

Our basic idea is to apply divide-and-conquer strategy to polynomial arithmetic. In particular, note that all polynomial arithmetics should be performed on ciphertexts. It is well known that one can multiply two polynomials over R of degree at most k using $\mathcal{O}(k^{\log_2 3})$ multiplications in R. In particular, we apply the well-known Karatsuba method for a *polynomial expansion* in a linear factored form using at most sub-quadratic multiplications in R.

The *polynomial evaluation* requires the heaviest operation in set operation protocol. Even using Horner's scheme, which clearly helps to eliminate large exponents, the total overhead still remains to be quadratic in the size of input multiset k. One might ask for fast Fourier transformation (FFT) which allows a fast polynomial evaluation with at most $\mathcal{O}(k \log k)$ computational complexity [58]. However, the FFT is more sophisticated and even requires to alter the set operation protocols. The requirement of FFT is the knowledge of n-th root of unity. This is main reason that FFT cannot be used for secure set operations. We design an efficient polynomial evaluation algorithm which has sub-quadratic complexity $\mathcal{O}(k^{1.28})$ using the divide-and-conquer based division algorithm and Horner's method.

1.2.2 Provably-secure and privacy-preserving top-k queries

To deal with the PPT-k algorithm from the sense of provable security at the first time. A formal construction of PPT-k query protocol for rigorous security analysis will be described based on the two privacy issues, *user privacy* and *owner privacy.* We also describe adversary's behavior for breaking *user privacy* and *owner privacy.* Based on our formal model of PPT-k algorithm, we propose provably-secure PPT-k algorithm with security proof.

1.2.3 Scalable authentication scheme for VANET

We propose a scalable privacy-preserving authentication protocol for secure vehicular communications. Through the verification of the service subscribers, the proposed authentication protocol allows the vehicle A to authenticate itself to the nearby vehicles without any participation of the nearby RSU.

If the vehicle A has authenticated with the nearby RSU, the vehicle can obtain the token, authenticating the vehicle to the nearby vehicles, from the CA. The verification of the service subscribers enable the nearby vehicles to verify whether the vehicle A has valid token or not. Therefore, the nearby vehicles can check whether the vehicle A has authenticated with the nearby RSU or not. Compared to the previous approaches [43, 70], the proposed protocol reduces computational overhead in vehicle-to-vehicle (V2V) authentication process in order to support better scalability.

1.3 Thesis outline

We begin by introducing cryptographic and mathematical preliminaries for our work in this their in Chapter 2. We discuss the related work in Chapter 3. We then introduce our techniques and algorithms for fast polynomial arithmetics on encrypted polynomial in Chapter 4. In Chapter 5, we introduce provable security concepts of PPT-k protocol and propose a generic construction of provably-secure PPT-k protocol. We propose a privacy-preserving and scalable authentication scheme for secure vehicular networks in Chapter 6. We conclude in Chapter 7.

Chapter 2. Preliminaries

2.1 Polynomial representation and arithmetics

2.1.1 Polynomial representation

Let X_i denote a multiset representation of player u_i's private datasets and $f_i \in R[x]$ denote a polynomial representation of the multiset X_i. Given a multiset X_i, the polynomial representation of this is $f_i(x) = \prod_{a_i \in X_i} (x - a_i)$. Note that some a_i's can be identical. Given N multisets $X_1, \ldots, X_N$ of the same cardinality, let us consider the problem on how to compute $I = X_1 \cap \ldots \cap X_N$. First, transform given multisets X_i into polynomial f_i's for all $i \in [1, N]$. Then choose random polynomial r_i's from $R[x]$ and compute

$$\sum_{i=1}^{N} f_i \cdot r_i$$

which is called an intersection polynomial of $\{X_i\}_{i \in [1,N]}$. The set of roots of this polynomial contains, but may not exactly the same with, $X_1 \cap \cdots \cap X_N$. However, the following **Lemma 1** shows that the intersection polynomial does not leak any information except $X_1 \cap \cdots \cap X_N$.

Lemma 1 *[40] Let f_1 and f_2 be polynomials in $R[x]$ such that $deg(f_1) = deg(f_2) = k$, $gcd(f_1, f_2) = 1$, and $f_1[k] \in R^*$ and $f_2[k] \in R^*$. For some $m \geq k$ let polynomials $r = \sum_{i=0}^{m} r[i]x^i$ and $s = \sum_{i=0}^{m} s[i]x^i$ where $r[i]$ and $s[i]$ are chosen uniformly at random from R for all $i \in [0, m]$.*

Let $h = f_1 \cdot r + f_2 \cdot s = \sum_{j=0}^{k+m} h[j]x^j$ for all $j \in [0, k + m]$. $h[j]$ are distributed uniformly and independently over R.

By this lemma, we see that $f_1 \cdot r + f_2 \cdot s = gcd(f_1, f_2) \cdot g$ where g is a uniformly distributed polynomial in $R[x]$ of degree $2k - |X_1 \cup X_2|$. In particular, since g is uniformly distributed, with high probability the roots of g does not represent any element in the domain of multisets.

Corollary 1 *Let $f_1, \ldots, f_N \in R[x]$ be polynomials of the degree k. Let $r_i = \sum_{j=0}^{m} r_i[j]x^j$ be a random polynomial, where $r_i[j]$ is chosen independently and uniformly from R for all $j \in [0, m]$ with $k \leq m$.Then $\sum_{i=1}^{N} f_i \cdot r_i = gcd(f_1, \ldots, f_N) \cdot g$, where g is distributed uniformly in $R^l[x]$ for $l = k + m - t$, where t is the degree of $gcd(f_1, \ldots, f_N)$.*

Using above **Lemma 1** and **Corollary 1**, one can show that given $f_1 \cdot r + f_2 \cdot s$, one cannot learn more than information on multisets X_1 and X_2 than can be deduced from $X_1 \cap X_2$.

2.1.2 Polynomial encryption

In order to perform set operations we need to support the following basic polynomial operations: addition, multiplication, and the formal derivative. These operations are given on encrypted polynomials.

By encryption $\mathcal{E}_{pk}(f)$ of polynomial $f = \sum_{i=0}^{k} f[i]x^i \in R[x]$, we denote the ordered list of encryptions of all its coefficients by an additive homomorphic encryption with some public key, *i.e.*,

$$\mathcal{E}_{pk}(f) := \langle \mathcal{E}_{pk}(f[0]), \mathcal{E}_{pk}(f[1]), \ldots, \mathcal{E}_{pk}(f[n]) \rangle$$

Utilizing the additive homomorphic encryption, one can efficiently perform a basic polynomial operations on encrypted ones without knowledge of the private key:

1. Addition. Let f_1, f_2, and g be polynomials of degree k such that $f_1(x) = \sum_{i=0}^{k} f_1[i]x^i$, $f_2(x) = \sum_{i=0}^{k} f_2[i]x^i$, and $g = \sum_{i=0}^{k} g[i]x^i$. Given two encrypted polynomials f_1 and f_2, one can efficiently compute the encryption of the polynomial $g := f_1 + f_2$ (denoted $\mathcal{E}_{pk}(f_1) \boxplus \mathcal{E}_{pk}(f_2)$) by calculating $\mathcal{E}_{pk}(g[i]) := \mathcal{E}_{pk}(f_1[i]) \boxplus \mathcal{E}_{pk}(f_2[i])$ for all $i \in [0, k]$.

2. Multiplication. Given a polynomial f_1 and an encrypted polynomial f_2, one can efficiently compute the encryption of the polynomial $g := f_1 \cdot f_2$ (denoted $f_1 \boxdot \mathcal{E}_{pk}(f_2)$) by calculating $\mathcal{E}_{pk}(g[l]) := \sum_{\substack{i+j=l \\ 0 \leq i,j \leq l}} f_1[i] \boxdot \mathcal{E}_{pk}(f_2[j])$.

3. Formal Derivative. Given an encrypted polynomial f_1, one can efficiently compute the encryption of the polynomial $g := \frac{d}{dx} f_1$ by calculating $\mathcal{E}_{pk}(g[i]) := (i+1) \boxdot \mathcal{E}_{pk}(f_1[i+1])$ for all $i \in [0, k-1]$.

In addition to the above basic operations, one can evaluate an encrypted polynomial at a point as follows:

For given an encrypted polynomial f_1 and a point a, one can efficiently compute the encryption of $b := f_1(a)$ by calculating $\mathcal{E}_{pk}(b) := \sum_{i=0}^{k} (a_i \boxdot \mathcal{E}_{pk}(f_1[i])$.

2.1.3 Polynomial multiplication for encrypted polynomial

There are different algorithms to perform the polynomial basis multiplication. Let denote the user's input size by k. The basic algorithm has an asymptotic complexity $\mathcal{O}(k^2)$. The recursive application of the Karatsuba algorithm has a running time of $\mathcal{O}(k^{\log_2 3}) \approx \mathcal{O}(k^{1.585})$ for $k = 2^m$ $(m > 0)$ [58], [38]. Knuth [37] gives a short introduction on how to multiply polynomials in an efficient way which is very similar to the Karatsuba algorithm. The l-way $(l > 2)$ generalization of the Karatsuba method has a better asymptotic complexity [47]. Bodrato [7] proposed a method to find optimal

$GF(2)[x]$ multiplication formulae. For more sophisticated algorithms, FFT and Cantor multiplications have better asymptotic complexity.

Another implementations of polynomial multiplication algorithms utilize the hybrid approach [26]. Such implementations first perform several Karatsuba iterations to reduce the whole complexity, and then a basic algorithm on small input. Especially there are various techniques to efficiently perform the later multiplications. For example, the quadratic comb methods [42] are used in [26]. The table-lookup technique is often applied to perform the polynomial multiplication efficiently.

2.2 Homomorphic encryption

A public-key encryption algorithm $E()$ is *homomorphic* if given $E(x)$ and $E(y)$, one can compute $E(x \circ y)$ without decrypting x and y for some algebraic operation $\circ$. Formally, for some homomorphic evaluation algorithm $H()$, given encryption algorithm $E()$ satisfies that $H(D(m)) = D(H(m))$ where $D()$ is a corresponding decryption algorithm with $E()$ [9].

If the homomorphic property holds for addition and multiplication operations, we call this algorithm is to be *fully homomorphic* [16, 11].

In order to perform group operations on ciphertexts, we need semantically secure public-key encryption schemes satisfying additively homomorphic property. Let $\mathcal{E}_{pk}$ be such a public-key encryption algorithm with a public key pk. Then this public-key encryption scheme allows the following operations on ciphertexts without knowing the corresponding private key sk:

- Given the encryptions of a and b, $\mathcal{E}_{pk}(a)$ and $\mathcal{E}_{pk}(b)$, one can efficiently compute the encryption of $a+b$, $\mathcal{E}_{pk}(a+b)$. Here $\mathcal{E}_{pk}(a+b) := \mathcal{E}_{pk}(a) \boxplus \mathcal{E}_{pk}(b)$ where $\boxplus$ denotes an addition on ciphertexts.
- Given a constant c and an encryption of a, $\mathcal{E}_{pk}(a)$, one can efficiently compute the encryption of $c \cdot a$, denoted $\mathcal{E}_{pk}(c \cdot a) := c \boxdot \mathcal{E}_{pk}(a)$. Here $\boxdot$ is a scalar multiplication which is equivalent to successive addition given constant times on ciphertexts.

We review three additive homomorphic encryption schemes used for the set operations.

2.2.1 The Paillier cryptosystem

Paillier [50] proposed an additive homomorphic cryptosystem. The cryptosystem takes plaintext from $\mathbb{Z}_n^*$ as an input and outputs ciphertext in $\mathbb{Z}_n^*$. Operations on encrypted values require arithmetic under $\mod n^2$. This cryptosystem has been proven to be semantically secure under the the decisional composite residuosity assumption. We describe the Paillier cryptosystem in brief.

Key Generation. Let $N = pq$, where p and q are primes. Choose $g \in \mathbb{Z}_{n^2}^*$ such that the order of g is divisible by N. Any such g is of the form $g \equiv (1+N)^a b^N \mod N^2$ for a pair (a, b), where $a \in \mathbb{Z}_N$ and

$b \in \mathbb{Z}_N^*$. Note that $(1+N)^a \equiv 1+aN \mod N^2$, so $g \equiv (1+aN)b^N \mod N^2$. Let $\lambda = lcm(p-1, q-1)$. The public key is (g, N) and the private key is λ.

Encryption: For message m and blinding factor $r \in \mathbb{Z}_N$, Paillier encryption is defined as $E_P(m, r, g, N) = g^m r^N \mod N^2$.

Decryption: In the Paillier cryptosystem, decryption is more complicated than encryption. First note that for any $x \in \mathbb{Z}_{N^2}$, $x^\lambda \equiv 1 \pmod{N}$ and $x^{N\lambda} \equiv 1 \pmod{N^2}$. Given $c = E_P(m, r, g, N) = g^m r^N \mod N^2$, we can see that $c^\lambda \equiv g^{m\lambda} r^{N\lambda} \equiv 1 + am\lambda N \pmod{N^2}$. Note also that $g^\lambda \equiv 1 + a\lambda N \pmod{N^2}$. Then we decrypt by computing

$$m = D_P(c, g, \lambda, N) = \frac{f_N(c^\lambda)}{f_N(g^\lambda)} \mod N$$

where $f_N(x) = \frac{(x \mod N^2)-1}{N}$.

2.2.2 A modified version of ElGamal encryption

Camenisch *et al.* [18] and Dachman-Soled *et al.* [20] made use of a standard variant of ElGamal encryption. Their initial setup requires to choose a cyclic group G of prime order q such that the discrete logarithm problem is difficult in G. Let g be a generator of G and x be a secret key chosen uniformly at random in $\mathbb{Z}_q^*$. The public key is $(g, h = g^x)$. To encrypt a message $\alpha \in \mathbb{Z}_q^*$, choose r at random in $\mathbb{Z}_q^*$, and compute $(g^r, g^\alpha h^r)$. Then one may easily verify that the cryptosystem satisfies the additive homomorphic property. However it is not possible to efficiently perform decryption; but the decryption will not be necessary.

As with Paillier, the scheme is also semantically secure and allows efficient proofs of knowledge. It is worth to note that arithmetic operations in G is much more efficient than in $\mathbb{Z}_{n^2}^*$ for the same level of security and the ciphertexts are smaller.

2.2.3 BGN encryption

As mentioned in the introduction, we need to get rid of privacy concern regarding the abuse of subscription information in authentication server. In other words, the authentication server should verify whether an end-user is one of subscribers without information leakage of the subscription information. To address this issue, we employ the previous approaches [21, 39], converting the searching of the sets to an evaluation of polynomial representations of a given set.

In 2005, Boneh *et al.* proposed new encryption scheme [6] which supports additively homomorphic operation and one multiplicative operation on encrypted data. Before describing the scheme, we explain the notation used in the scheme and how to construct the bilinear groups.

The homomorphic encryption scheme proposed by Boneh *et al.* uses the following notation:

1. $\mathbb{G}$ and $\mathbb{G}_{\mathbb{1}}$ are two (multiplicative) cyclic groups of finite order n.

2. g is a generator of $\mathbb{G}$.

3. e is a bilinear map e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_{\mathbb{1}}$. Namely, for all u, $v \in \mathbb{G}$ and a, $b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$.

4. $e(g, g)$ is a generator of $\mathbb{G}_{\mathbb{1}}$.

If $\mathbb{G}$ is a bilinear group, we can say that a group $\mathbb{G}_{\mathbb{1}}$ and a bilinear map as above also exist.

Let $n > 3$ be a given square-free integer that is not divisible by 3. Then, a bilinear group $\mathbb{G}$ of order n can be constructed as follows:

1. Find the smallest positive integer $l \in \mathbb{Z}$ such that $p = ln - 1$ is prime and $p = 2mod3$.

2. Consider the group of points on the (super-singular) elliptic curve $y^2 = x^3 + 1$ defined over $\mathbb{F}$. Since $p = 2mod3$, the curve has $p + 1 = ln$ points in $\mathbb{F}$. Therefore the group of points on the curve has a subgroup of order n which denote by $\mathbb{G}$.

3. Let $\mathbb{G}_{\mathbb{1}}$ be the subgroup of $\mathbb{F}^*$ of order n. The modified Weil pairing on the curve [?] gives a bilinear map e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_{\mathbb{1}}$ with the required properties.

The homomorphic encryption scheme consists of the three algorithms as follows:

1. KeyGen(τ): Let $n = q_1 q_2$. Pick two random generators g and u from the group $\mathbb{G}$ and set $h = u^{q_2}$. Then h is a random generator of the subgroup of $\mathbb{G}$ of order q_1. The public key PK_{BGN} is $(n, \mathbb{G}\ , \mathbb{G}_{\mathbb{1}}\ , e, g, h)$. The private key SK_{BGN} is q_1.

2. Encrypt(PK_{BGN}, m): When the message space consists of integers in the set $\{0, 1, \ldots, T\}$ with $T < q_2$ and r is a random number from 0 to $n-1$, the encryption is computed as $C = g^m h^r \in \mathbb{G}$. C is the encrypted message of m.

3. Decrypt(SK_{BGN}, C): The decryption is computed as $C^{q_1} = (g^m, h^r)^{q_1} = (g^{q_1})^m$.

Then, the encryption of $m_1 + m_2$ and the encryption of $m_1 m_2$ can be computed as $g^{m_1} h^{r_1} \cdot g^{m_2} h^{r_2}$ and $e(g^{m_1} h^{r_1}, g^{m_2} h^{r_2})$ where e is a bilinear mapping from $\mathbb{G} \times \mathbb{G}$ to $\mathbb{G}_{\mathbb{1}}$ and the encryption of m_i is $g^{m_i} h^{r_i}$ in BGN scheme. Also, the expected decryption time using the lambda method proposed by Pollard is $\tilde{O}(\sqrt{|m|})$ [6] although the authentication server has the private key $vk = q_1$.

Since the encryption of m_i is $g^{m_i} h^{r_i}$, the encryption of $m_1 + m_2$ is $g^{m_1+m_2} h^{r'} = g^{m_1} g^{m_2} h^{r'}$. By assigning $r' = r_1 \cdot r_2$, the encryption result of $m_1 + m_2$ can be expressed as $g^{m_1} h^{r_1} \cdot g^{m_2} h^{r_2}$. To allow one multiplication of two encrypted messages, we use the bilinear map. When $g_1 = e(g, g)$ and

$h_1 = e(g, h)$, the encryption of $m_1 m_2$ is $g_1^{m_1 m_2} h_1^{r''}$.

$$e(g^{m_1} h^{r_1}, g^{m_2} h^{r_2}) h_1^{r'} = \begin{cases} e(g^{m_1 + q_2 r_1},\ g^{m_2 + q_2 r_2}) h_1^{r'} \\ e(g, g)^{(m_1 + q_2 r_1) \cdot (m_2 + q_2 r_2)} h_1^{r'} \\ g_1^{(m_1 m_2 + \alpha q_2 r_2 m_1 + \alpha q_2 r_1 m_2 + \alpha^2 q_2^2 r_2 r_1)} h_1^{r'} \\ g_1^{m_1 m_2} h_1^{r' + r_2 m_1 + r_1 m_2 + \alpha q_2 r_2 r_1} \\ g_1^{m_1 m_2} h_1^{r''} \end{cases}$$

Note that $h_1 = e(g, h) = e(g, g^{\alpha q_2}) = e(g, g)^{\alpha q_2} = g_1^{\alpha q_2}$. Also, we can assign $r'' = r' + r_2 m_1 + r_1 m_2 + \alpha q_2 r_2 r_1$ since r'' is a random number in $\mathbb{Z}_n$.

2.2.4 Keywords search on the encrypted data

A keyword search on encrypted data is introduced to share audit log and email on a public server while minimizing information leakage. Previous protocols [4, 24, 8] have three common entities in their system models: a data provider, public server, and data retriever. The data provider generates shared information and stores it on a public server in an encrypted form. Only an entity having a proper trapdoor (*i.e.*, access permission) can retrieve the stored information. This approach can remove privacy concern regarding the abuse of subscription information stored in authentication server. However, no access control is provided [71] and the server can link two different sessions to the same group using the relationship between the stored data and the submitted trapdoor.

To provide access control only, [71] proposed an idea to convert the searching of the sets to an evaluation of polynomial representations of a given set [21, 39] using BGN encryption [6]. Interestingly, this approach can address the second problem due to non-deterministic property of BGN encryption which will be explained in the following section. However, the proposed approach is not efficient in view of computational overhead. Denote S_1 and S_2 by a set of access keys and a set of keywords, respectively. Then, the data retriever should compute $|S_1| + |S_2| + 1$ exponent multiplications and BGN encryptions [6] per each query. Also, the server should compute $|S_1| + |S_2| + 1$ pairing operations and $2 \cdot |S_1| + |S_2| + 1$ exponent multiplications per each query. Figure 2.1 illustrates the evaluation result proposed by Yau *et al.*.

When the encrypted data is the the subscription information regarding a service, the size of S_1 and S_2 are the number of the subscribers in the service. If the data is the keywords specifying the service, the size of S_1 (or S_2) is the number of the subscribers in the service (or the number of the keywords specifying the service). Usually, the keywords specifying the service is less than 15. As a result, we need more lightweight approach for keyword search on the encrypted data than the existing approach [71].

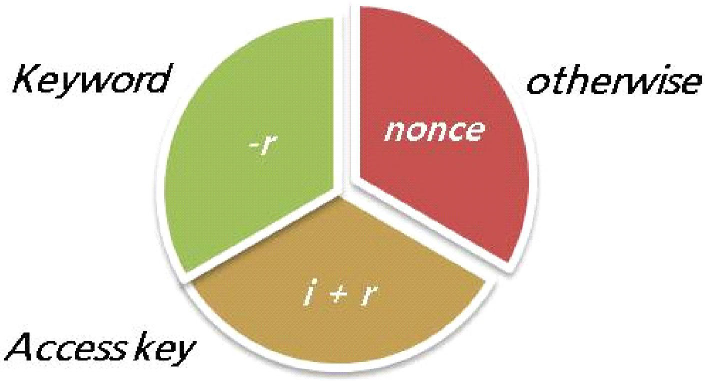

Figure 2.1: Evaluation result of the polynomial f(x) in [71]

Kim *et al.* in [36] proposed an enhanced idea which reduces the degree of the polynomial representations of a given set. As the BGN encryption can support non-deterministic property, the public server cannot find any relationship between the different forms of the same access key i. Hence, the polynomial representation of a given set does not require the degree presenting a set of access keys. By dividing the set presenting the number of the service subscribers into several subsets, the processing time verifying the polynomial representation of a given set can be adjusted according to the desire performance. Note that the set presents the number of the service subscribers. Using this idea, Kim *et al.* proposed VSS (Verification of the Service Subscribers). While the encrypted list of the service subscribers is given to the public server, the public server can verify whether the data retriever is one of the legitimate subscribers or not. In order to prevent the public server from identifying the legitimate service subscribers, the data provider encrypts the list of the service subscribers using the BGN encryption [6]. The data retriever only requires ($|S_2|$ - 1) modular exponentiations and S_2 BGN encryption [6] per each query.

2.3 Shamir's secret sharing

Secret sharing is a technique for protecting sensitive data, such as cryptographic keys. It is used to distribute a secret value to a number of parts–shares–that have to be combined together to access the original value. These shares can then be given to separate parties that protect them using standard means, e.g., memorize, store in a computer or in a safe. Secret sharing is is used in modern cryptography to lower the risks associated with compromised data. Sharing a secret spreads the risk of compromising the value across several parties. Standard security assumptions of secret sharing schemes state that if an adversary gains access to any number of shares lower than some defined threshold, it gains no information of the secret value. The first secret sharing schemes were proposed by Shamir [56].

The idea of Shamir's threshold scheme [56] is that k points are sufficient to define a polynomial of degree $k-1$. Suppose we want to use (k,n) threshold scheme to share our secret s, without loss of generality assuming that an element exists in a finite field $\mathbb{F}$. Choose at random $k-1$ coefficients $a_1, a_2, \ldots, a_{k-1}$ in $\mathbb{F}$, and let $a_0 = s$. Build the polynomial $f(x) = \sum_{n=0}^{k-1} a_n x^n$. Let us construct any n distinct points it, for instance set $i = 1, 2, \ldots, n$ to retrieve $(i, f(i))$. Every participant is given a point (a pair of input to the polynomial and output). Given any k subset of these pairs, we can find the coefficients of the polynomial using interpolation and the secret is the constant term a_0.

Chapter 3. Related work

3.1 Set operations using polynomial representation

Throughout the thesis $R[x]$ denotes a polynomial ring over a ring R. We restrict our attention to protocols running over polynomial arithmetic.

3.1.1 Two-party protocols

Freedman, Nissim, and Pinkas (FNP) [21] considered the *Set Intersection problem* and presented the first corresponding secure protocols against active adversaries. We describe their idea in detail, since it underlies most of the subsequent work on this topic. They use a semantically-secure public-key cryptosystem holding a group homomorphic property.

The main idea is as follows: A client defines a polynomial f whose roots are the private datasets

$$f = (x - a_1)(x - a_2) \cdots (x - a_k) = \sum_{i=0}^{k} \alpha_i x^i.$$

The client sends all the encrypted coefficients of f to a server by homomorphic encryptions. The server evaluates and re-randomizes the encrypted polynomial at the datasets, and then returns to the client the result of the computation. The client decrypts each encrypted coefficient β_j and checks it in the datasets, where j means the cardinality of the server's datasets.

In order to reduce the computational complexity, they suggested some naive methods that enable to evaluate the polynomial at multipoint efficiently. One is to use Horner's rule [28] and the other is to reduce the degree of polynomials. Applying Horner's rule is clearly an efficient way to evaluate a k-th degree polynomial at a single point. But regarding n points, this does not help to enhance the total complexity $\mathcal{O}(k^2)$. The second way leads us to change the protocol.

Recently, Camenisch and Zaverucha [18] investigated the set intersection problem in a different direction. They ask for a trusted third party to sign the input datasets of the two participants. This implies that the computational complexity still remains unchanged.

Instead Dachman-Soled *et al.* [20] concentrated on improving security in set intersection by suggesting that one participant prepare as many polynomials as the size of his input. Each polynomial is used to send all datasets using Shamir's secret sharing. Therefore, this scheme requires additional polynomial interpolation.

3.1.2 Multi-party protocols

Kissner and Song (KS) [39], [40] proposed improved protocols fodr more general set operations, as well as protocols for set operations in the multi-party setting. In order to ensure to be secure against an active adversary, KS employes generic zero-knowledge proofs [17], [12]. Their protocols cover set intersection, over-threshold Set Union, and Subset Relation problem.

Sang *et al.* [59], [60], [61] provide specific protocols for set intersection. Their protocols are also based on FNP, but improve the previous result by the fact of $\mathcal{O}(N)$ in the computation where N is the number of participants in the protocol. Also their scheme still has $\mathcal{O}(k^2)$ computational complexity where k is the size of private multiset.

Li and Wu [46] presented secure set intersection protocol in the information theoretical model, which is improved by Patra *et al.* [51]. This is different from our interest since we are interested in protocols in cryptographic model.

3.2 Privacy-preserving top-k query protocol

There has been proposed many protocols in the open literature to solve the above problems of privacy-preserving data aggregation. We can classify these researches under the type of the configuration of a set: fully centralized, fully decentralized, and semi-centralized. While the centralized schemes assume the existence of a trusted third party (TTP), which makes such schemes of less useful from the practical point of views under the strong assumption that TTP is not immune to compromise and malicious behavior, the fully decentralized schemes utilize cryptographic primitives and protocols to replace the centralized TTP. On the other hand, semi-centralized schemes try to bridge the functional and security gaps between both of the previous approaches. As they are of particular relevance to the work in hand, in the following we elaborate on the two latter ideas: the *decentralized* and *semi-decentralized* privacy preserving data aggregation schemes. As mentioned earlier, decentralized solutions to the problem try to replace the centralized TTP using cryptographic constructions, which come in different forms leading to several research approaches. One approach is based on secure multi-party computation like in [5, 62, 67]. Vaidya and Clifton's (VC) scheme [62] uses a generalized muliparty computation (MPC) protocol. A drawback of this protocol is inefficiency. VC scheme required overwhelmingly huge computational resources since the scheme uses Yao's garbled circuits [69]. Furthermore, as the datasets become disjoint, the efficiency of such construction decreases sharply. In [67], Xiong *et al.* have devised a unique method to aggregate the largest k values over numeric data using randomization techniques. In [5], Burkhart and Dimitropoulos has devised a feasible construction in which the round complexity is linear to the number of bits in the data elements. However, due to using *sketches* (*i.e.*, hash tables) to aimprove the efficiency of MPC subprotocols (used for `equality` and `lessthan` operations) the aggregate results are probabilistic. The authors claimed the proposed protocol is efficient in terms of its computational complexity, but the round complexity is huge expensive. In [3], Applebaum *et al.* proposed a solution for the top-k problem using semi-centralized constructions. The authors aim to enhance the efficiency of fully-decentralized instantiations by adding new entities: proxy and database (DB).

3.3 Privacy-preserving authentication protocols

3.3.1 Membership verification through keyword search

In order to enforce proper access control on keyword search, the subscription information should be encrypted. Typical approach presenting the subscription information on the specific service was presented in the polynomial form so that the searching of the set can be converted to the evaluation of the given polynomial [21, 39]. The existing approach [71] in the open literature, being able to employ access control to keyword search on the encrypted data, requires homomorphic operations in addition and multiplication. Among the homomorphic encryptions [25], the encryption scheme proposed by Boneh *et al.* in 2005 [6] can support the unlimited additive homomorphic operations and one multiplicative homomorphic operation. That's why we employ the encryption scheme proposed by Boneh *et al.* to hide the subscription information during keyword search.

3.3.2 Anonymous authentication

There are many approaches to solve user privacy and security challenges in ubiquitous computing environmen[2, 1, 10, 13, 41, 49, 63, 72, 53, 54, 33, 29]. However, most of these results fall in the scope of establishing general security framework and identifying general security requirements, without providing concrete security protocols. Some work [2, 1, 10, 41, 49, 33] focused on designing specific security infrastructures to protect user context privacy like location information from service providers. Creese *et al.* [13] and Wu *et al.* [63] revised authentication and privacy requirements and Zugenmaier *et al.* [72] showed that the use of a combination of devices using incompatible anonymous mechanisms can compromise the anonymity, which is achieved when each device is used separately.

Jendricke *et al.* [33] introduced an identity management system for ubiquitous computing environment. A user can issue multiple identities and use them depending on the applications. Using these virtual identities, the scheme can protect user privacy while providing access control and user authentication. However, there is no concrete protocol. He *et al.* [29] presented a simple anonymous ID scheme for ubiquitous computing environment. However, this scheme cannot prevent the double spending problem, a kind of replay attack, since there is no verification about the actual holder anonymous ID based on Chaum's blind signature technique [15]. In 2005, Ren *et al.* [53, 54] proposed new scheme supporting a part of the requirements for ubiquitous computing environment. While the scheme prevents double spending problem by combining two cryptographic primitives, blind signature and hash chain, the scheme reduces the number of signature verifications on the authentication server side. Additionally, the scheme provides non-linkability and differentiated service access control and does not rely on underlying system infrastructure such as the "lighthouse" or "mist routers" [1]. However, a mobile user should store all hash chains of his/her anchor to avoid repetitions of the same hash computations and perform one public key operation whenever the user sends a service access request message. Moreover, the service providers may have privacy concern regarding the abuse of their subscription information which is stored in the authentication server to enforce proper access control.

Gruteser and Grunwald [22] offered a method for hiding user's MAC address with anonymous IDs so that the user cannot be tracked in a wireless LAN environment.

3.3.3 Privacy-preserving authentication protocols for VANETs

Lin *et al.* [45] proposed a secure and privacy-preserving protocol for vehicular communications, called GSIS which is based on group signature [14] and identity-based signature [57]. While guaranteeing anonymity, confidentiality, and other security primitives, the GSIS can provide traceability of each vehicle. Only if any dispute happens, the identity of the message sender will reveal. In order to provide V2V communication between the vehicle A and nearby vehicles, the GSIS employs the group signature. The identity-based signature scheme is used to sign each message sent by each RSU for ensuring its authenticity. However, when the adversary compromises many of the RSUs, the adversary can track any movement of the target vehicle.

Lu *et al.* [43] proposed an efficient conditional privacy preservation protocol for secure vehicular communications, called ECPP, which issues on-the-fly short-time anonymous certificate to vehicles by using a group signature scheme. Since RSUs can check the validity of the requesting vehicle during the short-time anonymous certificate generation phase, such revocation check by vehicle itself of GSIS is not required. Therefore message verification is more efficient that GSIS. The ECPP provides authentication, anonymity, unlinkability and traceability under the strong assumption that most RSUs will not disclose any internal information without the authorization of the trusted authority. However, due to a large number of RSUs, cost considerations prevent the RSUs from having sufficient protection facilities against malicious attacks. Therefore, it is possible for an attacker to access RSUs and disclose the information in the RSUs. When multiple RSUs are compromised, an attacker can trace the movement of a vehicle by using the information stored in the compromised RSUs, because each RSU stores unchanged pseudonyms for OBUs in ECPP. As a result, ECPP does not provide unlinkability when some RSUs are compromised.

In 2009, Yim *et al.* [70] proposed a anonymous authentication scheme in VANETs. The proposed scheme guaranteed authentication, anonymity, unlinkability, and traceability simultaneously. The unlinkability which enables privacy preservation and the traceability which enables conditional tracking are contradictory. Yim *et al.*'s protocol utilize the traceable ring signature scheme with the k-times anonymous authentication scheme to address the contradictory requirements. In addition, the proposed scheme has three advantages compared with other previous works. First, the scheme does not have revocation list update process in authentication process. Second, the scheme always provides unlinkability although multiple RSUs are compromised. Finally, the scheme requires only one authentication process for mutual authentication when the vehicle communicate with the same RSU, because proposed scheme has key agreement functionality that makes secure channel to communicate. These advantages make Yim *et al.*'s scheme efficient in large-scale and busy networks like VANETs.

Chapter 4. Generic polynomial arithmetics for secure set operations with sub-quadratic complexity

Performing set operations without leaking the private dataset have played a crucial role in various applications which used over the distributed networks. Much recent work to compute secure set operations are designed using homomorphic public-key encryption and the technique of representing sets as polynomials in the cryptographic model. These polynomial representation based solutions require intensive polynomial arithmetic including polynomial expansion, multiplication, and evaluation. Previously proposed solutions require a quadratic complexity in computation.

This chapter deals with the speeding up the legacy set operation protocols based on polynomial representation. We present techniques to improve the computational complexity from quadratic to sub-quadratic. In particular, we show that without modification of given solutions, one can obtain the product of two polynomials of degree k with $\mathcal{O}(k^{\log_2 3})$ multiplications using the Karatsuba method in case of polynomial multiplication which is performed on ciphertexts and $\mathcal{O}(k^{1.28})$ for polynomial evaluation using the divide-and-conquer based division algorithm.

Using our new polynomial arithmetics, the total computational complexity of polynomial representation based set operations reduce from $\mathcal{O}(k^2)$ to $\mathcal{O}(k^{\log_2 3})$.

4.1 Introduction

For N ($N \geq 2$) different datasets over the distributed networks, the set operations including intersection and union play one of the fundamental tasks for a practical business application. One typical example is that the commercial companies want to share intersection of their customer lists while their own list except the common customers is protected. When one set operation does not reveal any other elements except for the elements in the resulting set, it is said to be privacy-preserving set operation. In other words, schemes in the literature attempt to guarantee privacy of the inputs of honest players even in the presence of adversary. Adversaries may be allowed to insert or delete some messages in transit, to inject false messages, and to gain access to users' information.

There have been intensive studies to solve this problem in different settings. Freedman, Nissim, and Pinkas [21] presented the first efficient privacy-preserving set intersection operation using polynomial representation working merely between two users. Kissner and Song [39] subsequently showed that polynomial representation allows more various set operations even in multi-party setting. This result was followed by a series of papers including [59], [60], [61].

The common property of such these schemes is that the cost of the polynomial arithmetic operations fully dominates their performance. Throughout the their, R denotes a ring. We assume that a polynomial $f = \sum_{i=0}^{k} a_i x^i \in R[x]$ of degree k. A list of polynomials arithmetic operations includes:

- **Polynomial expansion.**

  Given a polynomial,

  $$f = \prod_{i=1}^{k}(x - a_i) = (x - a_1)(x - a_2)\ldots(x - a_k)$$

  in a linear factored form, compute the expansion coefficients $f[i] \in R$ for each term of f

  $$f = \sum_{i=0}^{k} f[i]x^i = f[k]x^k + \cdots + f[1]x + f[0]$$

- **Polynomial multiplication.**

  Given polynomials $g = \sum_{i=0}^{k} g[i]x^i$ and $h = \sum_{i=0}^{m} h[i]x^i$, compute

  $$f = g \cdot h = \sum_{l=0}^{k+m} c_l x^l$$

  where $c_l = \sum_{l=x+y} (g[x] \cdot h[y])$.

- **Polynomial multipoint evaluation.**

  Given a polynomial f and k points $\langle a_1, a_2, \ldots, a_k \rangle$, compute

  $$\langle f(a_1), f(a_2), \ldots, f(a_k) \rangle .$$

Previouly proposed privacy preserving set operations based on polynomial representation require quadratic computational complexity in total.

In this research, we claim that the efficient operations can be obtained by applying the divide-and-conquer strategy to set operation protocols. We discuss more general method that, in many (if not all) situations, allows set operation protocols to be computed at most in sub-quadratic complexity with respect to the user's input size without changing the protocols. Even though not universally applicable (our work is efficient polynomial arithmetic), it can be used for most of set operation schemes via the polynomial representation.

Our basic idea is to apply divide-and-conquer strategy to polynomial arithmetic. In particular, note that all polynomial arithmetics should be performed on ciphertexts. It is well known that one can multiply two polynomials over R of degree at most k using $\mathcal{O}(k^{\log_2 3})$ multiplications in R. In particular, we apply the well-known Karatsuba method for a *polynomial expansion* in a linear factored form using at most sub-quadratic multiplications in R.

The *polynomial evaluation* requires the heaviest operation in set operation protocol. Even using Horner's scheme, which clearly helps to eliminate large exponents, the total overhead still remains to be quadratic in the size of input multiset k. One might ask for fast Fourier transformation (FFT) which allows a fast polynomial evaluation with at most $\mathcal{O}(k \log k)$ computational complexity [58]. However, the FFT is more sophisticated and even requires to alter the set operation protocols. The requirement of FFT is the knowledge of n-th root of unity. This is main reason that FFT cannot be used for secure set operations. We design an efficient polynomial evaluation algorithm which has sub-quadratic complexity using the divide-and-conquer based division algorithm.

The rest of the chapter is organized as follows. In Section 2 we state prior set operations in the literature, and recall essential backgrounds of our work. We propose the techniques for efficient computation of set operations in Section 3. In Section 4, we give a computational cost analysis of basic set operations and analyze the asymptotic complexity of our proposed polynomial arithmetics. Concluding remarks and future work are given in Section 5.

4.2 Our construction

In this section we discuss a way for speeding up the polynomial arithmetics in case of the computational complexity of being quadratic in the size of input multiset, k. This complexity completely resorts on the polynomial arithmetic: polynomial expansion, polynomial multiplication, and polynomial evaluation. Therefore we focus on reducing the number of multiplications in the polynomial arithmetic. Note that throughout this section we write $\mathcal{O}(k^2)$ for $\mathcal{O}(N^2 k^2 \log n)$ (*i.e.*, we write the computational cost omitting the N factor and the modulus n.).

4.2.1 Polynomial multiplication

The schoolbook polynomial multiplication method has an asymptotic complexity $\mathcal{O}(k^2)$ with respect to computation. Instead recursive application of the Karatsuba algorithm leads to a computational cost of $\mathcal{O}(k^{\log_2 3})$ [64].

Homomorphic encryption allows us to multiply a polynomial by an encrypted polynomial. Now we describe an algorithm for polynomial multiplication on ciphertexts (PMoC). Let $r_{i,j}$ be a random polynomial and $\mathcal{E}_{pk}(f_j)$ an encrypted polynomial in Step 4 of Set Intersection for all $i, j \in [1, N]$, $r_{i,j} := \displaystyle\sum_{l=0}^{k} r_{i,j}[l]x^l$ and $\mathcal{E}_{pk}(f_j) := \displaystyle\sum_{l=0}^{k} \mathcal{E}_{pk}(f_j[l])x^l$. If $g_j = r_{i,j} \cdot f_j$, we obtain the encryption of polynomial g_j of degree $2k$.

It is straightforward to verify that Algorithm 1 runs correctly.

Algorithm 1 Polynomial multiplication on Ciphertexts (PMoC)

```
Require: a random polynomial $r_{i,j}$ and an encrypted polynomial $\mathcal{E}_{pk}(f_j)$
Ensure: $\mathcal{E}_{pk}(g_j) = r_{i,j} \boxdot \mathcal{E}_{pk}(f_j) = \sum_{l=0}^{2k} g_j[l]x^l$
for $l = 1$ to $k$ do
    $h_l \leftarrow r_{i,j} \boxdot \mathcal{E}_{pk}(f_j[l])$
end for
for $l = 1$ to $2k - 1$ do
    for $s, t$ such that $s + t = l$ and $t > s \geq 0$ do
        $h_{st} \leftarrow (r_{i,j}[s] + r_{i,j}[t]) \boxdot \mathcal{E}_{pk}(f_j[s]) \boxplus \mathcal{E}_{pk}(f_j[t])$
    end for
end for
$g_j[0] \leftarrow h_0$
$g_j[2k] \leftarrow h_k$
if $l$ is odd then
    $g_j[l] \leftarrow \sum_{s+t=l} h_{st} - \sum_{s+t=l}(h_s + h_t)$
else
    $g_j[l] \leftarrow \sum_{s+t=l} h_{st} - \sum_{s+t=l}(h_s + h_t) + h_{l/2}$
end if
return $\sum_{l=0}^{2k} g_j[l]x^l$
```

4.2.2 Polynomial expansion

The polynomial expansion (PE) could be easier than the polynomial multiplication as the former does not require the homomorphic property.

Before we turn to the main subject we need to restate a theorem to analyze it running time. (it is also known as the Master Theorem.)

Theorem 1 *[55], Sec7.3: Let $k = p^m$ and m be a positive integer. Let assume that an increasing function M satisfies the recurrence relation $M(k) = aM\left(\frac{k}{b}\right) + ck^d$ where $a \geq 1$, $b > 1$ is an integer and c, d are real numbers with c positive and d nonnegative. Then*

$$M(k) = \begin{cases} \mathcal{O}(k^d) & \text{if } a < b^d \\ \mathcal{O}(k^d \log_b k) & \text{if } a = b^d \\ \mathcal{O}(k^{\log_b a}) & \text{if } a > b^d \end{cases}$$

The polynomial expansion helps a user represent the input multiset into a polynomial in Step 1 of set operation protocols. However, it is not necessary to encrypt the polynomial f. Let the polynomial $f(x) = (x - a_1)(x - a_2) \cdots (x - a_k)$ where all a_j is in the multiset for all $j \in [1, k]$. To simply, assume that $k = 2^m$ for a positive integer m.

The key idea is to apply the divide-and-conquer based Karatsuba strategy to expanding $f(x)$. We first write $f(x) = (x - a_1) \cdots (x - a_{k/2}) \cdot (x - a_{k/2+1}) \cdots (x - a_k)$.

Then we recursively apply the same strategy to $f_L(x)$ and $f_R(x)$ until $f_L(x)$ and $f_R(x)$ are in the form of product of two linear factors. Let denote $KM_{d_1,d_2}(\alpha \cdot \beta)$ the Karatsuba multiplication for two polynomials (denoted by α, β) of degree d_1, d_2 respectively. Our strategy is presented in Algorithm 2.

Algorithm 2 Polynomial Expansion (PE)

```
Require: $f(x) = f_L(x) \cdot f_R(x)$ where $f_L(x) = \prod_{j=1}^{k/2}(x - a_j)$, $f_R(x) = \prod_{j=k/2+1}^{k}(x - a_j)$, and $L = R = k$
Ensure: $f(x) = \sum_{j=0}^{k} f[j]x^j$
if $L = 2$ then
    return $KM_{1,1}(f_1 \cdot f_1)$
end if
if $R = 2$ then
    return $KM_{1,1}(f_1 \cdot f_1)$
end if
$f_L \leftarrow KM_{L/2,L/2}(f_{L/2} \cdot f_{L/2})$        ▷ left half
$f_R \leftarrow KM_{R/2,R/2}(f_{R/2} \cdot f_{R/2})$        ▷ right half
return $KM_{L,R}(f_L \cdot f_R)$
```

The algorithm 2 has the computational complexity of $\mathcal{O}(k^{\log_2 3})$. We analyze the asymptotic complexity of result at Section 4.

4.2.3 Polynomial evaluation

Finishing a group decryption in Step 5 (Step 4 for set union), each user gets a polynomial $p(x)$ ($P(x)$ for set union) of degree $2k$ ($k(N + 1)$ for set union). But the common factor is to evaluate the polynomial p (P for set union) at all elements in the input multiset.

Set intersection

Let p be a resulting polynomial and f a polynomial representing a user's input multiset. As pointed out before, even though Horner's method clearly helps to reduce the number of multiplications, the asymptotic complexity is still $\mathcal{O}(k^2)$ given k points.

To overcome this barrier, we again take advantage of the divide-and-conquer strategy. To do this, we first divide f into α polynomials, $f_1, f_2, \ldots, f_\alpha$, of degree $\frac{k}{\alpha}$ and then apply the division algorithm.

Assume $p(x) = \sum_{i=0}^{2k} p[i]x^i$ and $f(x) = \prod_{j=1}^{k}(x - a_j)$. By the division algorithm, there exists $q, r \in R[x]$ such that $p = q \cdot f + r$ and $r = 0$ or $deg(r) < deg(f)$. It is straightforward to check that $p(a) = r(a)$ where a is a root of f. Now we provide the efficient algorithm for polynomial evaluation in set intersection (PE/SI).

Set union

In case of Set Union, the target polynomial of evaluation is $P(x) = \sum_{i=1}^{N}\sum_{l=0}^{t-1} p^{(l)} \cdot F_l \cdot r_{i,l}$ where $deg(P) = k(N + 1)$. Let f_{SI} denote the polynomial for set intersection that represents a user's in-

Algorithm 3 Polynomial Evaluation and Set Intersection (PE/SI)

```
Require: $X = \{a_1, \ldots, a_k\}$ and $p(x) = \sum_{i=0}^{2k} p[i]x^i$
Ensure: $\langle p(a_1), \ldots, p(a_k) \rangle$
fix $\alpha$
$\beta \leftarrow k/\alpha$
$f \leftarrow \prod_{i=1}^{\alpha} f_i(x)$        ▷ $deg(f_i) = \beta$
for $i = 1$ to $k$ do
    $r_i \leftarrow p - q_i \cdot f_i$
    $p(a_i) \leftarrow r_i(a_i)$
end for
return $\langle p(a_1), \ldots, p(a_k) \rangle$
```

put multiset, and f_{SU} for set union.

Note that we may assume $deg(f_{SI}) = deg(f_{SU})$. Then we observe that the computational complexity is the function of $deg(f_{SI})$. We discuss this in next section in detail. Therefore we can reach the conclusion that in the Set Union case, the polynomial evaluation (PE/SU) can be performed by using at most $\mathcal{O}(k^{1.7})$ multiplications. The details of complexity is given next section. Moreover with a slight modification of the PE/SI algorithm we may get PE/SU.

4.3 Analysis

In this section we give an abstract description for basic set operation protocols and analyze the arithmetic cost of polynomials using KS protocol. This comprehensive assessment show that general set operation protocols require $\mathcal{O}(k^2)$ computational complexity. We also analyze the performance of the proposed polynomial arithmetics.

4.3.1 Complexity of basic set intersection protocol

Description

First we give an abstraction description for set intersection in the presence of malicious users. Passive secure (a.k.a., honest-but-curious, see [23], *Sec. 7* for details) protocols can be easily obtained by omitting tools for preventing malicious users from deviating the protocols.

In addition to set intersection, we can construct a protocol for cardinality of set intersection. However, it is a straightforward variant of set intersection problem as follows: all users compute $|X_1 \cap \cdots \cap X_N|$ on multisets.

The followings are the description of set intersection protocol based on polynomial representation.

Description of set intersection

Notations

- pk: public key for an additive homomorphic encryption scheme
- k: the maximum cardinality of input multiset

Phases

1. Each user u_i $(1 \leq i \leq N)$ computes a polynomial representation f_i of his multiset X_i
2. A user encrypts f_i and sends the encryptions to all other users with proofs that show the user knows coefficients of f_i
3. For all encrypted polynomials, each player chooses a random polynomial $r_{i,j}(1 \leq j \leq N)$ and commits its encryptions.
4. All users computes the encryption of $p = \sum_{i=1}^{N} \sum_{j=1}^{N} r_{i,j} \boxdot \mathcal{E}_{pk}(f_j)$
5. All users perform group decryption to get the polynomial p
6. Each user u_i finds $a_{i,j} \in X_i$, such that $(x - a_{i,j})|p$ for all $j \in [1, k]$.

KS set intersection and its analysis

Now we give a specific algorithm using the notations in [40]. However, there is no need to utilize the Paillier scheme as an additive homomorphic cryptosystem suggested in [39], [40]. Note that the homomorphic cryptosystem should allow efficient zero-knowledge proofs.

As we describe the basic set intersection in previous section, KS set intersection protocol consists of the following six phases.

Phase 1. In order to transform each user's multiset into the corresponding polynomial representation, a user first computes the polynomial expansion. Let X_i be the multiset of u_i whose order is k. Then the user calculates the polynomial f_i whose k roots are the elements in X_i as

$$f_i = \prod_{j=1}^{k} (x - a_{i,j}) = (x - a_{i,1}) \cdots (x - a_{i,k})$$

where all $a_{i,j} \in X_i$ for all $j \in [1, k]$.

Phase 2. Each user encrypts all coefficients of f_i and outputs $\mathcal{E}_{pk}(f_i)$. Instead of polynomial arithmetic, this step just requires modular multiplications and exponentiations. To prevent from joining with a random polynomial, a user is required to compute zero-knowledge (ZK) proofs by which a user proves to know the coefficients of polynomial at Step 4.

One can find the specific ZK proof protocols in [12] for the Paillier cryptosystem and in [17] for the ElGamal variants.

Phase 3. For all $j \in [1, N]$ each user u_i does the followings:

3-(i) Choose uniformly a random polynomial $r_{i,j} \in R^k[x]$,

3-(ii) Output the commitment to $\gamma_{i,j}$ where $\gamma_{i,j} = \mathcal{E}_{pk}(r_{i,j})$. Thus this step also does not require the polynomial arithmetic.

Phase 4. At first each user u_i should open the commitments and verify whether the proof from Step 2 holds or not. If not, terminate the protocol. A user multiplies each received encrypted polynomial f_j by j-th random polynomial $r_{i,j}$ for all $j \in [1, N]$ as

$$\mathcal{E}_{pk}(p_i) = \sum_{j=1}^{N} (r_{i,j} \boxdot \mathcal{E}_{pk}(f_j))$$

and, at the same time, computes ZK proofs that shows this multiplication is correct. We need to compute the polynomial multiplication.

On obtaining all other encrypted polynomials $\mathcal{E}_{pk}(p_1), \ldots, \mathcal{E}_{pk}(p_N)$, each user computes

$$\mathcal{E}_{pk}(p) = \sum_{i=1}^{N} \mathcal{E}_{pk}(p_i)$$

Phase 5. In order to obtain the polynomial p, all users engage in the group decryption. This step does not require any polynomial arithmetic.

Phase 6. Finally each player u_i $(1 \leq i \leq N)$ may construct an intersection I as follows: for all $a_{i,j} \in X_i$, a user verifies $(x - a_{i,j})|p$. If $(x - a_{i,j})$ divides the polynomial p, with overwhelming probability the root is in the intersection I. This corresponds to the polynomial evaluation at all elements in the multiset.

In summary, except for modular arithmetics in homomorphic encryptions and ZK proofs its inefficiency in the set intersection protocols arises from inefficiency of polynomial arithmetics.

We look into the computational complexity of these protocols for KS set intersection protocol.

In **Phase 1** one needs to transform the multiset, X, into the polynomial representation, f. First a user writes for this $f(x) = (x - a_1)(x - a_2) \cdots (x - a_k)$ for all $a_i \in X$, $i \in [1, k]$. However, in order to encrypt all coefficients one is required to expand f in the form of k linear terms. $y_1 := (x-a_1)\times(x-a_2)$ requires one multiplication, $y_2 := (x-a_1)\times(x-a_2)\times(x-a_3) = y_1\times(x-a_3)$ requires two multiplications, and so on. At the end, Step 1 requires $\frac{k(k-1)}{2}$ multiplications in a polynomial expansion.

The computation cost in **Phase 2** is dominated by the additive homomorphic encryption and ZK proofs. In case of Paillier's cryptosystem, encryption requires a modular exponentiation as pointed out by [50], *Sec. 7*. ZK proofs can be instantiated with the Σ-protocol in [11, Sec. 8]; this also requires a modular exponentiation. Therefore this step requires $4k \log n$ multiplications in total.

Phase 3 consists of generating a sequence of random polynomials and committing to its encryptions. Without loss of generality, although the complexity depends on the baseline commitment scheme, we may expect the total complexity to be constant times of $N(2k \log n + k)$.

In **Phase 4**, each user multiplies the encrypted polynomial by N random polynomials. In general, multiplication of two arbitrary polynomials of degree k requires $(k+1)^2$ multiplications. Thus this step requires $N(k+1)^2$ multiplications per a user. In addition, each user calculates ZK proofs for showing the multiplication correct. The Σ-protocol requires 4 exponentiations and 2 multiplications for computing the proofs and 6 exponentiations and 4 multiplications for verifying the proofs. Thus ZK proofs require $N(20k \log n + 8k)$ multiplications. Finally, this step totally requires $N((k+1)^2 + 20k \log n + 8k)$ multiplications in total.

Phase 5 requires all users to participate in a group decryption protocol. Especially the threshold version of the Paillier cryptosystem requires one exponentiation and one multiplication. Other threshold decryptions also are as efficient as that of threshold Paillier's scheme.

Finally, the most expensive work in **Phase 6** is the polynomial evaluation at k points. As pointed out by [58], given a polynomial $f \in R[x]$ of degree k and an element $a \in R$, Horner's rule is an elegant and efficient way of computing $f(a)$. It needs $(k+1)$ multiplications for a point; therefore this step totally requires $k(k+1)$ multiplications. Our analysis is summarized in Table 1 below.

Table 4.1: Computational cost in set intersection

	Complexity
Step 1	$Nk(k-1)/2$
Step 2	$4Nk \log n$
Step 3	$N(2k \log n + k)$
Step 4	$N^2((k+1)^2 + 20k \log n + 8k)$
Step 5	$N(2k \log n + k)$
Step 6	$Nk(k+1)$
Total	$\mathcal{O}(N^2k^2)$

4.3.2 Complexity of basic set union protocol

Description

Now we review the protocol for set union, or more precisely, a threshold set union.

Intuitively we can define the union of multisets $X_i \cup X_j$ as the multiset such that each element a that appears $t_i(\geq 0)$ times in X_i, $t_j(\geq 0)$ times in X_j, and $(t_i + t_j)$ times in $X_i \cup X_j$. For polynomials f_i, f_j which are the polynomial representation of X_i and X_j respectively, $f_i \cdot f_j$ is a polynomial representation of $X_i \cup X_j$.

In addition, for completeness we need to define the operation of element reduction. While the multiplication of polynomials implies the union of multisets, the formal derivative enables us to control a threshold saying how many times some element appears in a resulting multiset. We define t-element reduction on a multiset X as follows: for each a that appears s times in X, the element appears $\max\{s-t, 0\}$ times in the resulting multiset, which is denoted by $\partial_t(X)$. More formally, for a given polynomial f of degree k, r_l randomly chosen from $R^k[x]$, and polynomial F_l of degree l such that $F_l(a) \neq 0$ and $gcd(F_0, \ldots, F_t) = 1$ for all a in an input domain and $l \in [0, t]$,

$$\partial_t(X) := \sum_{l=0}^{t} f^{(l)} \cdot F_l \cdot r_l$$

Using the threshold set union protocol, each user learns which elements appear at least a threshold value t times in the resulting multiset without revealing any other information. This can be expressed as $\partial_{t-1}(X_1 \cup \cdots \cup X_N)$.

Description of set union

NOTATIONS

- pk: public key for an additive homomorphic encryption scheme
- k: the maximum coalition size
- t: the maximum threshold of repetitions of an element appearing in the resulting set
- $F_0, \ldots, F_{t-1}$: fixed polynomials such that they have no common roots in input domain.

PHASES

1. Each user u_i $(1 \leq i \leq N)$ obtains the polynomial representation f_i of the input multiset X_i.
2. All users then compute encryption of polynomial $p = \prod_{i=1}^{N} f_i$, which corresponds to the polynomial representation of $X_1 \cup \cdots \cup X_N$.
3. Each user chooses uniformly at random $r_{i,l} \in R^k[x]$ for all $l \in [0, t-1]$ and calculates the encryption of polynomial $\sum_{l=0}^{t-1} p^{(l)} \cdot F_l \cdot r_{i,l}$.
4. Each user then computes the encryption of polynomial $P = \sum_{i=1}^{C+1} \sum_{l=1}^{t-1} p^{(l)} \cdot F_l \cdot r_{i,l}$ and performs a decryption to get P where P is a polynomial representation of $\partial_{t\grave{}1}(X_1 \cup \cdots \cup X_N)$
5. Each user may determine if the element $a_{i,j}$ appears in the combined multiset t times by evaluating the polynomial.

KS set union and its analysis

We give a specific set union protocol [40]. As we recall the description of the basic set union, KS set union protocol consists of the following five phases.

Phase 1. In order to transform each user's multiset into the corresponding polynomial representation, a user first computes the polynomial expansion. Let X_i be the multiset of u_i whose order is k.

Then the user calculates the polynomial f_i whose k roots are the elements in X_i as

$$f_i = \prod_{j=1}^{k}(x - a_{i,j}) = (x - a_{i,1}) \cdots (x - a_{i,k})$$

where all $a_{i,j} \in X_i$ for all $j \in [1, k]$. Each user needs to expand the polynomial representing input multiset.

Phase 2. In order to obtain the encryption of the polynomial p, each user u_i receives the encryption of neighbor $u_{i`1}$'s polynomial $f_{i`1}$, computes $f_i \boxdot \mathcal{E}_{pk}(f_{i`1})$, and sends it neighbor u_{i+1}. At the end of this step, all users have the encryption of the same polynomial $p = \sum_{i=1}^{N} f_i$. During this step, all users perform the polynomial multiplication N times along with computing ZK proofs to prevent malicious users from deviating from the protocol.

Phase 3. For all $i \in [1, C+1]$, each user computes the l-th derivative of $\mathcal{E}_{pk}(p)$ for all $l \in [1, t-1]$, chooses random polynomials $r_{i,0}, \ldots, r_{i,(t-1)} \in R^{N_k}[x]$, and then outputs

$$\mathcal{E}_{pk} = \sum_{i=0}^{t-1}(r_{i,l} \cdot F_l) \boxdot \mathcal{E}_{pk}(p^{(l)}).$$

This step also consists of repeated polynomial multiplications.

Phase 4. In this step all users just perform polynomial additions to obtain the encryption of polynomial

$$\mathcal{E}_{pk}(P) = \sum_{i=1}^{N} \mathcal{E}_{pk}(P_i)$$

Completing the addition, each user engages in the group decryption to get the polynomial,

$$P = \sum_{i=1}^{N} \sum_{l=0}^{t-1} p^{(l)} \cdot F_l \cdot r_{i,l}.$$

Phase 5. Suppose that each user u_i's multiset $X_i = \{a_{i,1}, \ldots, a_{i,k}\}$. By evaluating the polynomial P at all elements $a_{i,j} (1 \leq j \leq k)$, each user easily determine which elements appear at least t times in the resulting multiset V as follows:

$$\forall j \in [1, k], \begin{cases} a_{i,j} \in V & \text{if } P(a_{i,j}) = 0 \\ a_{i,j} \notin V & \text{otherwise} \end{cases}$$

In addition to this operation, although we can consider several variants of set union, we may see that a slight modification allows us to construct a corresponding protocol. In all variants the modification arises in Step 5. As an example one might want to know which elements appear in the combined multisets at least t times as well as the number of times these elements appeared in the multiset. Note that despite of such this modification, there is no need for additional polynomial arithmetic.

Regarding set union, we arrive at the same conclusion since this operation is also constructed using polynomial arithmetic. We can easily derive the asymptotic complexity of set union algorithm is $\mathcal{O}(k^2)$.

4.3.3 Analysis of our proposed algorithms

In this section, we consider the computational complexity of our proposed algorithms in Section 3. At first we give an asymptotic complexity of Algorithm 1 (PMoC).

Proposition 1 *Algorithm 1 has an asymptotic complexity of $\mathcal{O}(k^{\log_2 3})$.*

Proof. To simplify the analysis of the complexity of Algorithm 1, we assume that the number of coefficients k is a power of some integer, *i.e.*, $k = p^m$.

We see that the number of h_l is k and the number of h_st is equal to that of all possible pairs out of k coefficients, i.e., $\frac{k(k-1)}{2}$. Therefore the total number of multiplications is

$$\left(\frac{1}{2}p^2 + \frac{1}{2}p\right)^m = \left(\frac{1}{2}p^2 + \frac{1}{2}p\right)^{\log_p k} = k^{\log_p\left(\frac{1}{2}p^2+\frac{1}{2}p\right)}$$

If $p = 2$ then we obtain the complexity of $k^{\log_2 3}$. This completes the proof. ■

Now we analyze the Algorithm 2 (PE).

Proposition 2 *Algorithm 2 has an asymptotic complexity of $\mathcal{O}(k^{\log_2 3})$.*

Proof. At lines 2 and 5 in Algorithm 2, the Karatsuba algorithm performs one multiplication. At line 7, we perform Karatsuba multiplication for two polynomials of degree $k/4$ whose computational complexity is $\mathcal{O}((k/4)^{\log_2 3})$. We have the same result at line 8. Therefore we can write

$$M(k) = 2M\left(\frac{k}{2}\right) + 2\left(\frac{k}{2}\right)^{\log_2 3}$$

By Theorem 1, when $a = 2, b = 2, c = 2/9$, and $d = \log_2 3$, the total complexity $M(k) = \mathcal{O}(k^{\log_2 3})$ since $a < b^d$ ■

Proposition 3 *The new PE/SI algorithm in Algorithm 3 evaluates a polynomial a k points at most using $\mathcal{O}(k^{1.7})$ multiplications*

Proof. For the sake of simplicity, we assume $k = 2^m$. Let $p(x) = \sum_{i=0}^{2k} p[i]x^i$ and $f(x) = \prod_{j=1}^{k}(x - a_j)$. By the division algorithm, we know that there exist $q, r \in R[x]$ such that $p = q \cdot f + r$ and $r = 0$ or $deg(r) < deg(f)$. Furthermore we see that $p(a_j) = r_i(a_j)$ for all $j \in [1, k]$ and $i \in [1,]$.

In line 1 we expand each polynomial f_i using Algorithm 2. By **Proposition 2**, the New PE algorithm ensures that it has an asymptotic complexity of $\beta^{\log_2 3}$.

It remains to compute a remainder polynomial, r_i, which requires to perform polynomial division. When we divide p by f_i if we employe the Karatsuba algorithm for division, line 5 requires at most $\alpha\beta^{\log_2 3-1}$ multiplications [32]. Then we evaluate the polynomial r_i at a_j. If we use Horner's rule we only need to perform at most β^2 multiplications. Since we have to repeat such a polynomial evaluation α times, the total computational complexity amounts to $\alpha(\alpha\beta^{\log_2 3-1}+\beta^2)$. If let $\beta = k^\epsilon$ for some real number $\epsilon > 0$, then $\alpha = k^{1-\epsilon}$ so that $\alpha(\alpha\beta^{log_2 3-1}+\beta^2) = k^{1-\epsilon}(k^{1+\epsilon(\log_2 3-2)}+k^{2\epsilon})$. If we let $\epsilon = 0.28$ then we have $\mathcal{O}(k^{1.28})$ complexity in terms of multiplication and the proof is complete. ■

From the above Propositions, we can derive the following Corollary.

Corollary 2 *Using* PMoC , PE , *and* PE/SI *or* PE/SU *algorithms, set operations have an asymptotic complexity bound to $\mathcal{O}(k^{\log_2 3})$ in total for the cardinality of input multiset k.*

Now we compare our proposed algorithms with previous proposed set operations in computational complexity. As we analyzed in this section, previously proposed algorithms have $\mathcal{O}(k^2)$ complexity in total. But from the **Proposition 1**, **Proposition 2**, **Proposition 3** and **Corollary 2**, our construction can reduce the total complexity of set operations from $\mathcal{O}(k^2)$ to $\mathcal{O}(k^{\log_2 3})$.

4.4 Summary and discussion

We have shown a naive polynomial arithmetic for set operations results in a total of $\mathcal{O}(N^2k^2)$ computational overhead in terms of multiplication. For the legacy polynomial arithmetics, we have proposed that better efficiency can be gained by applying the well-known Karatsuba scheme and divide-and-conquer based division algorithm. This results in reducing the total computational overhead from $\mathcal{O}(N^2k^2)$ to $\mathcal{O}(N^2k^{\log_2 3})$ multiplications.

Our proposed method can be useful building block for constructing privacy preserving set operations based on polynomial representation and make an efficient implementation of the most of set operation schemes via the polynomial representation.

We conclude this chapter with commenting further work of our work. In the polynomial multiplication case, fast Fourier Transformation (FFT) is optimal complexity, $\mathcal{O}(k \log k)$. But FFT cannot be applicable for encrypted polynomial, since the FFT requires whole n-th root of unity. So, the finding the quasi-quadratic arithmetic algorithms for set operations are open problems in next step.

Chapter 5. Generic Construction of Privacy-Preserving Top-k Query using Homomorphic Encryption

In this chapter we reconsider the privacy-preserving top-k query (PPT-k) problem and introduce the formal model of PPT-k scheme with novel security notion from the point of owner privacy and user privacy. To the best of our knowledge, this is the first attempt to measure the PPT-k protocol in formal security model using homomorphic encryption. Based on our proposed PPT-k model, we give generic construction of provably-secure and privacy-preserving top-k query algorithm. We analyze the complexity of our construction, and show that it is better than the other work in computation rounds.

5.1 Introduction

Privacy-preserving is one of the most important issue in data aggregation methods by multiparty under different computing domain environments. There proposed plenty of research results of privacy-preserving methods in various area, top-k query algorithms [5, 62, 67], secure voting systems [31, 19, 48], and collaborative data aggregation and mining [27, 3]. Among these researches, there is some common security requirements for privacy-preserving as follows:

- *Confidentiality of set elements*: which requires that no entity or traitor should learn anything about honest users' inputs except what can be trivially derived from the output itself.
- *User Privacy*: for each element in an aggregated set, no user should distinguish a honest user who is the owner of elements in the union.
- *Owner Privacy*: no user or traitors should identify his/her or their elements from the aggregated set.

In this chapter, we consider how to assure the security in top-k query algorithm. At first, we recall two main problems in top-k data aggregation.

Problem definition. Let n users be denoted by $u_i (1 \leq i \leq n)$ and each of them has a private (multi)set of order r, A_i, where $A_i = \cup_{j=1}^{r} \{\alpha_{i,j}\}$ and $\alpha_{i,j}$ is the j-th private element of u_i. There may exist one or more elements such that $\alpha_{i,j} = \alpha_{i,j'}$ for some $1 \leq j < j' \leq r$. Let us denote $C(A)$ for the collection of frequencies for all elements in the multiset A, here the frequency of an element refers to how many times the element appears in the multiset A.

- **Private distribution problem:** All users learn the distribution $C(U)$ of a joint multiset $U = \bigcup_{i=1}^{n} A_i$, without revealing any other information.

- **Privacy-preserving top-k (PPT-k) problem:** All users learn the most frequent top-k items of all private input multisets without revealing any other information.

Our main contribution is to deal with the PPT-k algorithm from the sense of provable security at the first time. A formal construction of PPT-k query protocol for rigorous security analysis will be described based on the two privacy issues, *user privacy* and *owner privacy*. We also describe adversary's behavior for breaking *user privacy* and *owner privacy*. Based on our formal model of PPT-k algorithm, we propose provably-secure PPT-k algorithm with security proof.

The remainder of chapter is organized as follows: A brief survey on the related work and cryptographic tools is discussed in Section 2, and we define the formal construction of PPT-k algorithm and its security notion in Section 3. Our proposed scheme is presented in detail in Section 4. Section 5 analyzes the security and performance of our scheme and gives a comparison with previous work. Finally, we summarize and conclude in Section 6.

5.2 Provably secure top-k queries

In this section, we define the formal construction of PPT-k scheme and its security goal and adversarial model in detail.

5.2.1 PPT-k protocol

A privacy-preserving top-k query protocol consists of five probabilistic polynomial-time algorithms (Setup, Encrypt, Shuffle, Aggregate, Reveal).

- Setup(1^λ). The setup algorithm takes as input the security parameter λ and outputs a pair of private and public key (sk_i, pk_i) for each user u_i. Setup() also generates a secret information s and computes the its shares s_i.

- Encrypt(pk_i, A_i). Encryption algorithm takes as input the user's public key pk_i and a private ordered multiset A_i of the user where $A_i = \{a_{i,j} | j = 1, 2, \ldots, r\}$ and outputs the encrypted multiset $\mathcal{E}(A_i)$ can be defined as $\mathcal{E}(A_i) = \{Enc(pk_i, a_{i,j}) | a_{i,j} \in A_i\}$. $Enc()$ can be any encryption algorithm which is secure under adaptively chosen cipher text attack (IND-CCA2).

- Shuffle($s_i, U, \sigma_i(\cdot)$). The blinding of the original owner of each element in the ordered multiset U algorithm Shuffle() takes as input a shared secret s_i, an ordered multiset U, and private permutation algorithm $\sigma_i(\cdot)$ and outputs shuffled multiset U_i^s. U is an ordered multiset, so U has an identical vector $\vec{u}$. The Shuffle() algorithm reorders the elements of $\vec{u}$ using σ_i and outputs a vector $\vec{u}_s$ and its identical ordered multiset U_i^s.

- Aggregate($U, s_i, Find(\cdot)$). The information aggression algorithm Aggregate() takes as input an ordered multiset U, a secret share s_i, and elements distribution counting algorithm $Find(\cdot)$ and outputs a subset U_k of U which contains the top-k occurred elements in U. The distribution means the number of occurrence of elements in the set. Based on the distribution result, Aggregate() generates a U_k.

- Reveal($U_k, s_i, Dec(sk_i, m)$). The reveal algorithm of PPT-k Reveal() takes as input a top-k subset U_k, a user's shared secret s_i and a user's secret key sk_i and outputs the cleartext of top-k elements among the each users's private multiset. Reveal() recovers the secret information s from s_i using threshold scheme and finding the cipher text of top-k elements. Then Reveal() outputs the cleartext of top-k elements using threshold decryption algorithm.

Using the above algorithms, PPT-k protocol can be operated as follows:

1. TTP runs Setup(1^λ) and sends its outputs, public and secret parameters to all user.

2. A user u_i generates own private multiset A_i and computes encrypted multiset $\mathcal{E}(A_i)$. Then the user u_i broadcasts the encrypted set to other participants.

3. After receiving all encrypted multiset, the user u_i generate the aggregated multiset U_i using and blinding the order of elements using Shuffle. Then u_i sends the U_i^s to u_{i+1}. This process can be done in a recursive manner.

4. Using Aggregate() algorithm, each user finds the distribution of elements in the aggregated multiset and achieves U_{top-k}.

5. PPT-k participants recover the top-k elements using Reveal() algorithm.

We give a generic construction of provably-secure and privacy-preserving top-k query protocol in Section4.

5.2.2 Security requirements

Based on the common security requirements of privacy-preserving algorithms in open literature, we can derive the four security requirements for PPT-k as follows:

- *Correctness*: Top-k elements among participants's private set must be found by Reveal().

- *Confidentiality of non top-k element*: Only top-k elements are decrypted from an aggregate multiset and there must be no information leakage on non top-k element, $\overline{T_k}$.

- *Owner privacy*: Given a top-k element T_k, identifying the actual owner is computationally difficult.

- *User privacy*: For each element in an aggregated set, no user should distinguish an honest user who is the owner of elements in the union set.

5.2.3 Adversarial model

Adversarial power

Lindell and Pinkas [44] categorized the adversaries based on the strategy of corruption and its allowable behavior. There are two types of corruption model, static corruption model and adaptively corruption model. We use the adaptively corruption model which allows capability of corrupting parties during the computation except target participants. The sequence of corruption can be solely determined by the adversary in adaptively.

Also, we assume the adversary behaves like a malicious way in general. There are two main types of adversary, semi-honest adversary and malicious adversary. Semi-honest adversary is also named 'honest-but-curious' or passive adversary and malicious adversary is also called the active adversary.

Accessible oracles

In PPT-k protocol, we define three oracles, *user oracle* ($\mathcal{O}^{user}$), *owner oracle* ($\mathcal{O}^{own}$) , and *decryption oracle* ($\mathcal{O}^{dec}$) which can be accessed by an adversary. *User oracle* outputs any user's private shared information, secret information and computation results in Setup, Encrypt and Shuffle phases. But *user oracle* queries are not allowed to entire participants but a specific user. *Owner oracle* outputs the actual owner from a given element of an aggregated set except target element. *Decryption oracle* outputs the plaintext from a queried ciphertext.

Now we give the details of adversary to disturb owner privacy and user privacy.

Fig. 1 describes the concept of adversarial behavior. In PPT-k protocol, the adversary can utilize $\mathcal{O}^{own}$, $\mathcal{O}^{user}$ and $\mathcal{O}^{dec}$ in adaptively.

Adversarial model for owner privacy

The adversary for owner privacy's behavior is as follows.

1. The adversary $\mathcal{A}$ gets a top-k element α and an aggregated set U to break.

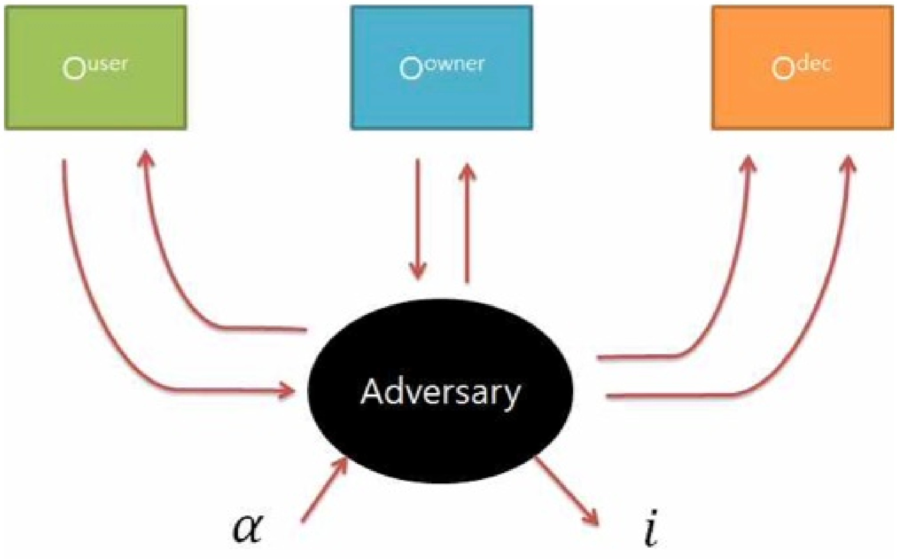

Figure 5.1: Adversarial model of PPT-k

2. The adversary is free to perform any number of additional computations and queries to the $\mathcal{O}^{own}$, $\mathcal{O}^{user}$ and $\mathcal{O}^{dec}$ in adaptively.

3. Finally, the adversary outputs a guessed user i as the owner of target element.

The adversary, $\mathcal{A}$ can be modeled as a probabilistic polynomial time algorithm. The input of $\mathcal{A}$ is a top-k element, α and outputs is an actual owner of the element. We now define the advantage of distinguishing the actual owner of top-k element.

$$\mathsf{Adv}_{i.j}^{\mathcal{O}^{own},\mathcal{O}^{user},\mathcal{O}^{dec}}(\mathcal{A},\alpha) = |\mathsf{Pr}\left[\mathcal{A}(\alpha \in U) = u_i\right] - \mathsf{Pr}\left[\mathcal{A}(\alpha \in U) = u_j\right]|$$

If the advantage $\mathsf{Adv}_{i.j}^{\mathcal{O}^{own},\mathcal{O}^{user},\mathcal{O}^{dec}}(\mathcal{A},\alpha)$ is negligible for each i and j, then actual owner can be indistinguishable.

Adversarial model for user privacy

The adversary for user privacy's behavior is as follows:

1. The adversary $\mathcal{B}$ gets an aggregated set and a target element β of the aggregated set U to break.

2. The adversary is free to perform any number of additional computations and queries to the $\mathcal{O}^{own}$, $\mathcal{O}^{user}$ and $\mathcal{O}^{dec}$ in adaptively.

3. Finally, the adversary outputs a guessed user i as owner of the actual private element.

The adversary, $\mathcal{B}$ can be modeled as a probabilistic polynomial time algorithm. The input of $\mathcal{B}$ is an element β in an aggregated set and the output are an actual owner of the element. We now define the advantage of distinguishing the actual owner of aggregated set's element.

$$\mathsf{Adv}_{i.j}^{\mathcal{O}^{own},\mathcal{O}^{user},\mathcal{O}^{dec}}(\mathcal{B},\beta) = |\mathsf{Pr}\left[\mathcal{B}(\beta \in U) = u_i\right] - \mathsf{Pr}\left[\mathcal{B}(\beta \in U) = u_j\right]|$$

If the advantage $\mathsf{Adv}_{i.j}^{\mathcal{O}^{own},\mathcal{O}^{user},\mathcal{O}^{dec}}(\mathcal{B},\beta)$ is negligible for each i and j, then private set can be indistinguishable.

For confidentiality, IND-CCA2 model is sufficient to deal with the privacy issue.

5.3 Generic construction of PPT-k protocol

In this section we describe our construction for PPT-k protocol in detail. We use a fully homomorphic encryption scheme and general secret share methods as cryptographic tools in our construction.

Let there be n users, denoted by $u_{i\in[1,n]}$ and PS_i be a private set of each user where $PS_i = \cup_{j=1}^{r}\alpha_{i,j}$. We also denote PS_i^s as $\{\alpha_{i,1}^s, \alpha_{i,2}^s, \ldots, \alpha_{i,r}^s\}$.

Setup

Each user agrees on a fully homomorphic encryption with a public/private key pair (pk, sk). In addition, all users are distributed a share s_i such that $s = \prod_{i=1}^{n} s_i$ and $s \cdot s' = 1 \pmod{p}$ *i.e.*, s' is an inverse element of s under the multiplication. Notice that in threshold decryption scheme, user generally produce shares of the decrypted element; if a user sends a uniformly generated share instead of a valid one, the decrypted element is uniform. Also, if the decrypted element is uniform, this reveals no information to the users.

Encrypt

Every user u_i encrypts his private dataset A_i using the fully homomorphic encryption E. Let user's encrypted dataset $\mathcal{E}_{pk}(A_i)$ be $\mathcal{E}_{pk}(A_i) := \{E_{pk}(\alpha_{i,1}), \ldots, E_{pk}(\alpha_{i,r})\}$. Each user computes his own private dataset. Then u_i broadcasts $\mathcal{E}_{pk}(A_i)$ to all other users $u_l(l \in I - \{i\})$.

Shuffle

Every user u_i receives $\mathcal{E}_{pk}(A_1), \ldots, \mathcal{E}_{pk}(A_{i-1}), \mathcal{E}_{pk}(A_{i+1}), \ldots, \mathcal{E}_{pk}(A_n)$ and then using his secret share, s_i, computes

$$\mathcal{E}_{pk}(A_l)^{s_i} := \{(E_{pk}(\alpha_{l,1}))^{s_i}, \ldots, (E_{pk}(\alpha_{l,r}))^{s_i}\}$$

for all $i \in I$. Using a private permutation σ_i, u_i randomly permutes the list of mixed ciphertexts as follows. Let $c_{\sigma_i(l,j)}$ denotes $(E_{pk}(\alpha_{\sigma_i(l,j)}))^{s_i}$.

$$\mathcal{C} = \{c_{\sigma_i(1,1)}, \ldots, c_{\sigma_i(1,r)} \cdots c_{\sigma_i(n,1)}, \cdots c_{\sigma_i(n,r)}\}$$

where $|C| = nr$. The user u_i sends $\mathcal{C}$ to u_{i+1} in a recursive manner.

Aggregate

Let the composition of users' private permutation σ be $\sigma = \sigma_{i-1} \circ \cdots \circ \sigma_i$. At the end of the Shuffle() phase, each user has

$$\mathcal{E}_{pk}(U)^s = \left\{(E_{pk}(\alpha_{\sigma(1,1)}))^s, \ldots, (E_{pk}(\alpha_{\sigma(n,r)}))^s\right\}$$

Performing a group decryption by all users allows u_i to have

$$U^s = \left\{\alpha^s_{\sigma_{(1,1)}}, \ldots, \alpha^s_{\sigma_{(n,r)}}\right\}$$

Reveal

From the $C(U^S)$, each participant learns the frequency distribution of aggregated multiset. For finding top-k elements the predicate φ is needed. If an element α^S is one of the top-k, then $\varphi(\alpha^S) = 1$. Otherwise $\varphi(\alpha^S) = 0$. The Reveal() recovers decrypts the elements whose predicate's output is 1. And then recovers the secret information s from s_i using threshold scheme. Finally Reveal() outputs the cleartext of top-k elements using threshold decryption algorithm.

5.4 Security and performance analysis

5.4.1 Proof of security

In this section, we show our proposed scheme satisfies whole security requirement we have claimed in Section 3. First, we show that all participants get the distribution of joint multiset without knowing any information of each elements in multiset.

Proposition 4 *After* Aggregation*() phase is over, every user learns the joint set distribution of all user's private inputs.*

Proof. Each user gets a permuted joint multiset $U^s = \{\alpha^s_{\sigma(1,1)}, \ldots, \alpha^s_{\sigma(n,r)}\}$. Each element in U^s is the output of composition of private permutations of union of all participants' private set.

So there is a unique index $(i^*, j^*) = \sigma^{-1}(i, j)$ is a unique shuffled elements of $\alpha_{i^*,j^*} \in \bigcup_{i=1}^{n} A_i$ where $\sigma = \sigma_{i-1} \circ \cdots \circ \sigma_i$. From the multiset U^s, it is trivial that every user learns the distributions of union of private sets. ■

From the above proposition, we can easily derive the following corollary:

Corollary 3 *Every HBC user learns the top-k items of all users' private inputs with overwhelming probability.*

Now we show that our proposed protocol satisfies the privacy requirements in the HBC model. Let TTP be a trusted third party in the ideal world which receives the private input multiset A_i of size k from user u_i for $i \in [1,n]$, and then returns to every user the joint multiset distribution $\{F(\alpha)\}$ for all $\alpha \in \cup_{i=1}^{n} A_i$. Here we use $F(\alpha)$ to denote a frequency of an element α in a multiset A and $F(A)$ a collection of frequencies for all elements in the multiset A.

Proposition 5 *Assume that the homomorphic encryption $E_{pk}(\cdot)$ is semantically secure. Our proposed construction after* Aggregate() *phase, any coalition of less than n HBC users learns no more information than would be given by using the same private inputs in the ideal-worlds model with the TTP.*

Proof. We assume that the homomorphic encryption scheme is semantically secure, and so each user learns only $\mathcal{E}_{pk}(A_i), \mathcal{E}_{pk}(A_{i-1})^{s_{i-1}}, \ldots, \mathcal{E}_{pk}(A_{i+1})^{s_{i-1}\cdots s_{i+1}}$ during an execution. At the end of the protocol all users further know $\mathcal{E}_{pk}(U)^s$ where

$$\mathcal{E}_{pk}(U)^s = \left\{(E_{pk}(\alpha_{\sigma(1,1)}))^s, \ldots, (E_{pk}(\alpha_{\sigma(n,r)}))^s\right\}$$

Note that σ is a composition of random permutations and unknown to all users, as the maximum coalition size is smaller than n. That is, if there exists at least an honest user, then a composition of random permutations $\sigma = \sigma_{i-1} \circ \cdots \circ \sigma_i$ is a random permutation because at least a permutation $\sigma_i \in [1,n]$ is secure. What is more, note that s is uniformly distributed and unknown to all users for the same reason. As s is uniformly distributed for any user inputs and σ is random, no user or coalition can learn more than a set of re-randomized the give homomorphic encryptions. As s is uniformly distributed, a group decryption of the homomorphic encryptions reveals no more than $\{\alpha_{i,j}^s\}_{i\in[1,n],j\in[1,k]} = \cup_{i=1}^{n} A_i^s = U^s$. We know the fact that $F(\alpha) = F(\alpha^s)$ for two multisets A and $A^s \in (\mathbb{G}_q)^k$, for all $s \in \mathbb{Z}_q$ and for all $\alpha \in A$. Hence we see that $F(U^s) = F(\cup_{i=1}^{n} A_i^s) = F(\cup_{i=1}^{n} A_i) = F(U)$, which can be derived from the output that would be returned by the TTP in the ideal-world model. This completes the proof. ■

Now, we claim that our construction satisfies the owner privacy and user privacy.

Proposition 6 *If the homomorphic encryption is semantically secure, our proposed scheme provides owner privacy and user privacy against any coalition of less than n users.*

Proof. We assume that there is at least one honest user. After aggregation phase, every user obtains a multiset $U^s = \{\alpha_{\sigma(1,1)}^s, \ldots, \alpha_{\sigma(n,r)}^s\}$, where $\sigma = \sigma_i \circ \cdots \circ \sigma_{i+1}$ and $s = \prod_{i=1}^{n} s_i$ consider the worst case, *i.e.* $n-1$ user coalition and let denote the honest user as u_h. Since the given homomorphic encryption is semantically secure, the probability of finding permutation is exactly $\left(\frac{1}{n!}\right)$ and computing α^{s_h} is computational difficult, the coalition cannot learn any useful information without knowing u_h's private permutation and shared secret. This implies that $\left|Pr\left[\mathcal{A}(\alpha \in U) = u_i\right] - \frac{1}{n!}\right| < \frac{\epsilon}{2}$ and

$\left|Pr\left[\mathcal{B}(\alpha \in U) = u_i\right] - \frac{1}{n!}\right| < \frac{\epsilon}{2}$ for all $1 \leq i \leq n$ and $\epsilon > 0$. By the triangle inequality,

$$\mathsf{Adv}_{i.j}^{\mathcal{O}^{own},\mathcal{O}^{user},\mathcal{O}^{dec}}(\mathcal{A},\alpha) = |\mathsf{Pr}\left[\mathcal{A}(\alpha \in U) = u_i\right] - \mathsf{Pr}\left[\mathcal{A}(\alpha \in U) = u_j\right]| < \epsilon$$

and

$$\mathsf{Adv}_{i.j}^{\mathcal{O}^{own},\mathcal{O}^{user},\mathcal{O}^{dec}}(\mathcal{B},\alpha) = |\mathsf{Pr}\left[\mathcal{B}(\alpha \in U) = u_i\right] - \mathsf{Pr}\left[\mathcal{B}(\alpha \in U) = u_j\right]| < \epsilon$$

is satisfies for all i and j. This completes the proof. ■

5.4.2 Performance evaluation

In this section, we give a detailed analysis of the running time and space requirements.

Computational complexity of our scheme

- Setup(): The secret key sk is shared among $u_1, u_2, \ldots, u_n$ using a polynomial f of degree $n-1$ over Z_q such that u_i holds a share $sk_i = f(i)$. The round complexity is the same as that of Shamir's secret sharing. Further, each user should be distributed a share of the secret information s. So, communication complexity of user is $O(1)$ in this phase.

- Encrypt(): Each user encrypts his multiset and broadcasts the set of ciphertexts. So, the computational complexity is $O(r)$ where r is the order of private multi set and the communication complexity is $O(1)$. We have $O(nr)$ total computational complexity.

- Shuffle(): Each user shuffles the encrypted set received from other users and randomly mixed the list. Thus, each user should compute $O(nr)$ operations in n rounds for each user's mixed set. In sum, the computational complexity is $O(n^2r)$ operations in $O(n)$ rounds.

- Aggregate(): Each user performs a threshold decryption. So, each user proceeds one broadcasting and computes decryption for each encrypted element. The total computational complexity is $O(nr)$ in constant rounds.

- Reveal(): This phase requires the same computation and round complexity as those of Shuffle phase. Therefore, the total computational complexity is $O(n^2r)$ along with $O(n)$ round complexity.

In summary, the total computational complexity of our construction is $O(n^2r)$ in $O(n)$ rounds.

We make a comparison between our scheme and Burkhart and Dimitropoulos (BD) scheme [5]. BD scheme has $\mathcal{O}(n^3H)$ round complexity where n is the number of participants, v is and H is the size of hash table which is same role as the private set's order r in our construction. In contrast, our scheme has $\mathcal{O}(n^2r)$ round complexity. Please refer the original paper [5] for the details. We $r \approx H$, our proposed scheme is more efficient that BD scheme.

5.5 Summary and discussion

Privacy-preserving top-k queries played a crucial role in various applications which used in distributed network. But, to the best of our knowledge, there is no formal security model for this protocol. In this chapter we have defined the formal model of privacy-preserving top-k query, and introduced the new security notion of owner privacy which is useful security requirements in the various applications. Based on our formal model, we proposed the first provably secure privacy preserving top-k query algorithm based on homomorphic encryption scheme and secret sharing method. We also have proved the security of our proposed scheme in the case of owner privacy and user privacy. Moreover our proposed scheme has better performance in computation rounds compared to prior solutions for the top-k query problem.

Our construction is based on honest-but-curious model. We remain the provably secure PPT-k scheme under malicious model as future work.

Chapter 6. A Scalable Privacy-preserving Authentication Protocol for Secure Vehicular Communications

In this chapter, we provide the first scalable privacy-preserving authentication protocol for VANETs without participation of the nearby RSU. Existing authentication methods for VANETs require the participation of the nearby RSUs. So, bottleneck problem can be occurred as increasing the number of vehicles. Also, the time delay to authenticate the nearby vehicle will increase. In order to minimize the participation of the nearby RSU, we propose a verification of the authenticated vehicle, which only requires two modular exponentiations. Our verification methods uses homomorphic encryption algorithm and keyword searching on encrypted data algorithm as cryptographic tools. Through this verification, the vehicle i can verify whether the nearby vehicle j is authenticated by the nearby RSU. As a result, our solution overcomes the inefficiency and bottleneck problem of previous approaches. Our construction of privacy-preserving authentication for VANETs provides better transmission delay between nearby RSU and vehicle.

6.1 Introduction

A Vehicular Ad-hoc NETwork (VANET) is a type of Mobile Ad-hoc NETwork (MANET) that is used to provide communications among the nearby vehicles, and between vehicles and fixed infrastructure on the roadside. VANET allows a driver in the vehicle to collect dynamic traffic information and sense various physical conditions related to traffic distribution with very low cost and high accuracy. Since VANET has a great potential to revolutionize driving environment and will undoubtedly play an important role in the future transportation system [66], we should address security and privacy problems in the VANET.

We can classify the communications in VANET into Vehicle-TO-Vehicle (V2V) and Vehicle-TO-Infrastructure (V2I). V2V indicates the communications between On-Board Units (OBUs) in vehicles while Vehicle-TO-Infrastructure (V2I) denotes the communications between OBUs and Road-Side Units (RSUs), which is fixed equipment on the road. Through V2I communication, the driver in a vehicle can identify the road condition, the road traffic, and the estimated time to the destination. Since the nearby vehicles can propagate the emergency warning message to the driver's vehicle by V2V, 60% roadway collisions can be avoided [65].

Figure 6.1 shows the typical system architecture consisting of vehicle, RSU, and Certificate Authority (CA). The vehicle A communicates with the CA through the nearby RSU. The RSU gathers the communication messages within its communication range and forwards them to the CA. To support sufficient bandwidth, the RSU should have wireless communication capability such as 3GPP. In addition, the RSU has the permanent power supply in order to satisfy scalability and easy maintenance. Because the battery-powered RSU may not available in case of emergency due to the numerous accesses of the nearby vehicles. Also, frequent battery replacement causes more maintenance cost. That's why the RSU has the permanent power supply. The CA, believed to be trusted third party.

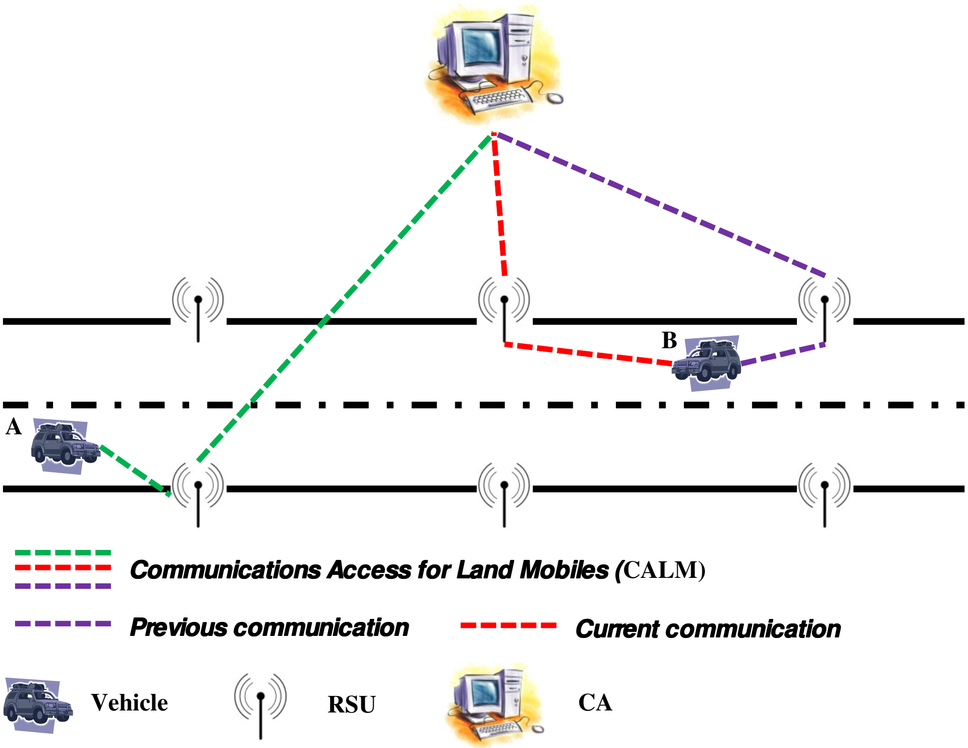

Figure 6.1: System architecture

To provide the privacy of the drivers in VANET (Vehicular Ad-hoc Network), various anonymous authentication protocols [45, 43, 70] have been proposed. However, these protocols can allow the target vehicle to communicate with the nearby vehicles through the trusted authority. As the number of the vehicles in certain location area increases, these protocol may suffer the bottleneck problem in the nearby RSUs. For instance, during lunar holidays in Asian countries, as the Asian people visit their hometown using public transportation (*i.e.*, car and train), the traffic on highways is heavy. Due to the number of vehicles in the certain area, the nearby RSUs cannot support the numerous number of authentication requests. Also, the time delay to authenticate the nearby vehicle will increase.

Recently there are several approaches which solve the security weakness in ad-hoc networks and heterogeneous wireless sensor networks [30, 68, 34]. But these results cannot be applied to VANET in direct. Kim *et al.* proposed to provide an efficient re-authentication protocol for wireless sensor network [35]. Through a keyword search on encrypted data, the end-user can authenticate himself/herself to the deployed sensor node. While the proposed idea can be applied to anonymous V2V authentication, each end-user has the same token having the different form. If a malicious driver sends the fake warning message, the CA cannot track who is the malicious driver.

In this research, we propose a scalable privacy-preserving authentication protocol for secure vehicular communications. Through the verification of the service subscribers, the proposed authentication protocol allows the vehicle A to authenticate itself to the nearby vehicles without any participation of the nearby RSU. If the vehicle A has authenticated with the nearby RSU, the vehicle can obtain the token, authenticating the vehicle to the nearby vehicles, from the CA. The verification of the service subscribers enable the nearby vehicles to verify whether the vehicle A has valid token or not. Therefore, the nearby vehicles can check whether the vehicle A has authenticated with the nearby RSU or not. Compared to the previous approaches [43, 70], the proposed protocol reduces computational overhead in V2V authentication process in order to support better scalability.

The remainder of this chapter is organized as follows: A brief survey on the related work is conducted in Section 2, and our proposed scheme is presented in detail in Section 3. Section 4 analyzes the security and performance of our scheme and gives a comparison with previous work. Finally, we summarize and conclude our chapter in Section 5.

6.2 Our protocol

In this section, we explain our protocol consisting of vehicle registration, V2I authentication, and V2V authentication in detail. In order to reduce the computational overhead in V2I authentication and V2V authentication, we proposed the verification of the authenticated vehicle. This idea is based on the following fact: When the vehicle A has authenticated itself with the CA, the vehicle and CA can share the secret information. Hence, the nearby vehicles can verify whether the vehicle A has been authenticated with the base station or not.

Before describing our protocol, we summarize our notations used throughout this chapter in Table 6.1.

We assume that a driver can control the source addresses of the outgoing Medium Access Control (MAC) frames since this assumption is a prerequisite for anonymous communications. A detailed method for this modification is covered by Gruteser *et al.* [22]. The CA issues SID, a polynomial $f(x)$ with degree p, access key ak_i, $E[i + r, PK_{BGN}, G_1]$, and ID_i to a driver i. Using the received infor-

Table 6.1: Notations

CA / RSU / V	Certificate Authority / Road-Side Unit / Vehicle
$Credential$ / ID_A / n	A ticket for entity authentication / An identifier of entity A / A user's access frequency
$CertA$ / VSS	A certificate that binds entity A with A's public key / Verification of the service subscriber
MT	A ticket for VSS which indicates subscribers of the target SP
PK_A / SK_A	A public key of entity A / A private key of entity A
PK_{BGN}	A public key under BGN encryption scheme [6] owned by AS
S	A set of selected numbers where $\mid S \mid \geq 2n$
SID	A service type identifier describes a selected subset of the available service pool and includes an polynomial identifier for membership test
SK_{BGN}	A private key under BGN encryption scheme [6], which is owned by AS and distributed to DS for membership test
C^i or C^i_A, $i = 0, 1, \cdots$	A series of authorized credentials by entity A
j^i or j^i_A, $i = 1, 2, \cdots$	A series of a user's number selections by entity A
$K_{A,B}$	Shared secret key between entities A and B
$E\{m, K_A\}$	A message m is encrypted by a symmetric key K_A
$E[m, PK_A]$ or $D[m, SK_A]$	A message m is encrypted by an entity A's public key or signed by an entity A's private key
$E[m, PK_{BGN}, G]$	A message m is encrypted by the public key PK_{BGN} on cyclic group G and the ciphertext is g^m
$H(m)$	A hash value of message m using SHA-1 or other cryptographic strong hash functions
R^i or R^i_A, $i = 1, 2, \cdots$	A series of nonces generated by entity A where $\mid R^i \mid \geq$ 64-bit.

mation, the driver i can generate his/her MT. The CA stores the coefficients of the given polynomial $f(x)$, (i.e., a_0, $\cdots$, a_p), in its database after encrypting the coefficients using BGN encryption [6]. The administrator in the CA cannot obtain any relationship between the authorized credential and driver A. That's why the CA encrypts the coefficients of the given polynomial $f(x)$. Finally, PK_{CA}, ID_{CA}, PK_{BGN}, and SK_{BGN} are assumed to be known to all entities.

6.2.1 Verification of the authenticated vehicle

Using the idea used in keyword search on encrypted data, we can preserve the privacy of the driver A while allowing the driver B to authenticate the driver A without any help of the nearby RSUs. Since the driver A should submit the proper trapdoor with the encrypted identifier, the verifier (*i.e.*, RSUs or CA) can compare the computation result with the verification value. If the result is the same as the verification value, the verifier believes that the end-user has proper access permission on the

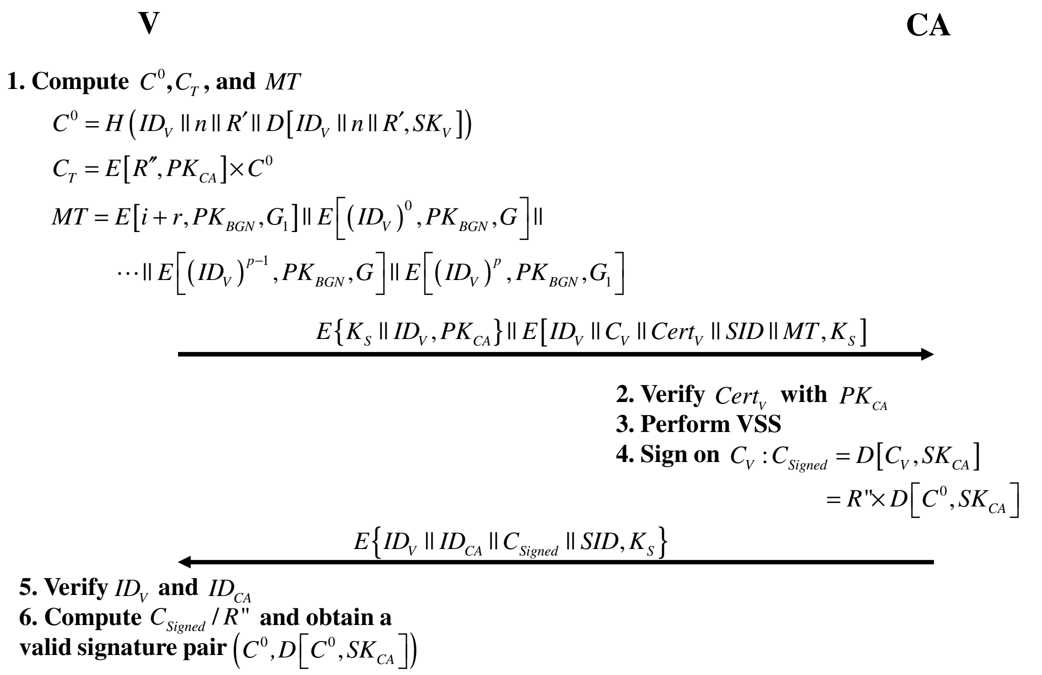
V
CA
1. Compute C^0, C_T, and MT
$C^0 = H\left(ID_V \| n \| R' \| D[ID_V \| n \| R', SK_V]\right)$
$C_T = E[R'', PK_{CA}] \times C^0$
$MT = E[i + r, PK_{BGN}, G_1] \| E\left[(ID_V)^0, PK_{BGN}, G\right] \|$
$\cdots \| E\left[(ID_V)^{p-1}, PK_{BGN}, G\right] \| E\left[(ID_V)^p, PK_{BGN}, G_1\right]$
$E\{K_S \| ID_V, PK_{CA}\} \| E[ID_V \| C_V \| Cert_V \| SID \| MT, K_S]$
2. Verify $Cert_V$ with PK_{CA}
3. Perform VSS
4. Sign on $C_V : C_{Signed} = D[C_V, SK_{CA}]$
$= R'' \times D\left[C^0, SK_{CA}\right]$
$E\left\{ID_V \| ID_{CA} \| C_{Signed} \| SID, K_S\right\}$
5. Verify ID_V and ID_{CA}
6. Compute C_{Signed} / R'' and obtain a
valid signature pair $\left(C^0, D\left[C^0, SK_{CA}\right]\right)$

Figure 6.2: Vehicle registration

service. Note that anyone can take a role of the verifier in keyword search on the encrypted data if the entity knows the stored vale and PK_{BGN}. Using this novel property, we can allow B can authenticate the other driver A without communicating with the nearby RSUs. However, in the existing approach by Kim *et al.*, the verifier requires $(p + 1)$ pairing operations where p is the number of the service subscribers. As the number of the nearby vehicle increase, the verification time will be increased.

Authorized token Generation phase

To address the above problem, we employ the following approach. When the nearby RSU receives the authentication request of a driver A, the RSU forwards the request and R_{RSU} to the CA. If the driver is a legitimate entity, having proper access permission on the service, the CA issues the verification value $E[\gamma, PK_{BGN}, G]$ and trapdoor $E[R_{RSU} + \beta, PK_{BGN}, G]$ to the vehicle. Note that $\gamma = \beta + R_{RSU}$. The verification value and trapdoor are encrypted by PK_{BGN} so that only the CA knows the exact value. By sending $E[-R_{RSU} + \beta, PK_{BGN}, G]$, the driver can authenticate himself/herself to the other drivers.

Verification phase

If the other drivers have been authenticated with the nearby RSU, they should have $E[\gamma, PK_{BGN}, G]$. Using the received information, R_{RSU} and β, the other drivers can verify whether the drive A has been authenticated by the nearby RSU.

Then, the nearby vehicles performs the following steps:

1. Set V to $E[-R_{RSU} + \beta, PK_{BGN}, G]$.

2. If $V^{SK_{BGN}} = (E[\gamma, PK_{BGN}, G])^{SK_{BGN}}$, return TRUE.

3. Return FALSE.

According to the property of BGN encryption, V is the same as $g^{-R_{RSU}+\beta}h^{r_1} = g^{-R_{RSU}+\beta+\mu q_2}$. Similarly, $(E[\gamma, PK_{BGN}, G])$ is $g^{\gamma}h^{r_2} = g^{\gamma+\mu q_2}$. Note that μ is a random integer less than n and q_2 is a large prime number where the order n of the given group G is $q_1 q_2$. By computing modular exponentiation $SK_{BGN} = q_1$, $V^{SK_{BGN}} = g^{-R_{RSU}+\beta}$. As a result, the driver B can identify whether the driver A has authenticated with the nearby RSU. This idea will be used to support V2V authentication.

6.2.2 Vehicle registration

In vehicle registration phase, each vehicle register its credential with the CA. Only if the driver of the vehicle belongs to one of the service subscribers, indicating that the driver has valid certificate issued by the CA and proper access permission on the service, the CA authorizes the received credential. In order to verify proper access control on the service, we employ Verification of the Service Subscribers (VSS), which is proposed by Kim *et al.*. If the driver has his/her identifier and valid access permission, $E[i+r, PK_{BGN}, G]$, VSS returns the index of the driver among the legitimate service subscribers. To illustrate VSS, let assume that $f(x)$ is the polynomial representation of the given legitimate service subscribers. The evaluation result of $f(ID_V)$ will be $-r$ if ID_V is an identifier of the legitimate service subscribers. Using $f(ID_V)$ and $i + r$, the CA can verify that the driver is one of the legitimate service subscribers without identifying the driver.

The driver A of the vehicle computes initial credential C^0 using his/her vehicle registration number ID_V, access frequency n, random number R'. This credential will be used to prove whether the driver A has valid ID_V, Certificate and SK_v. Without knowing SK_v and ID_v, the malicious driver cannot generate the above credential. For VSS, the driver A generate $MT = E[i{+}r, PK_{BGN}, G_1]||E[(ID_V)^0, PK_{BGN}, G$ $||E[(ID_V)^{p-1}, PK_{BGN}, G]\ ||E[(ID_V)^p, PK_{BGN}, G_1]$.

In order to verify whether the driver A is one of the legitimate service subscribers, the CA performs the following steps:

1. Set $z = 1$.

2. Compute $C = \prod_{n=1}^{p-1} e(E[a_n, PK_{BGN}, \mathbb{G}]\ , E[(ID_V)^v, PK_{BGN}, \mathbb{G}])$.

3. Compute $C' = C \cdot E[a_0, PK_{BGN}, \mathbb{G}_{\mathbb{1}}] \cdot E[(ID_V)^t, PK_{BGN}, \mathbb{G}_{\mathbb{1}}]\ \cdot E[i + r, PK_{BGN}, \mathbb{G}_{\mathbb{1}}]$

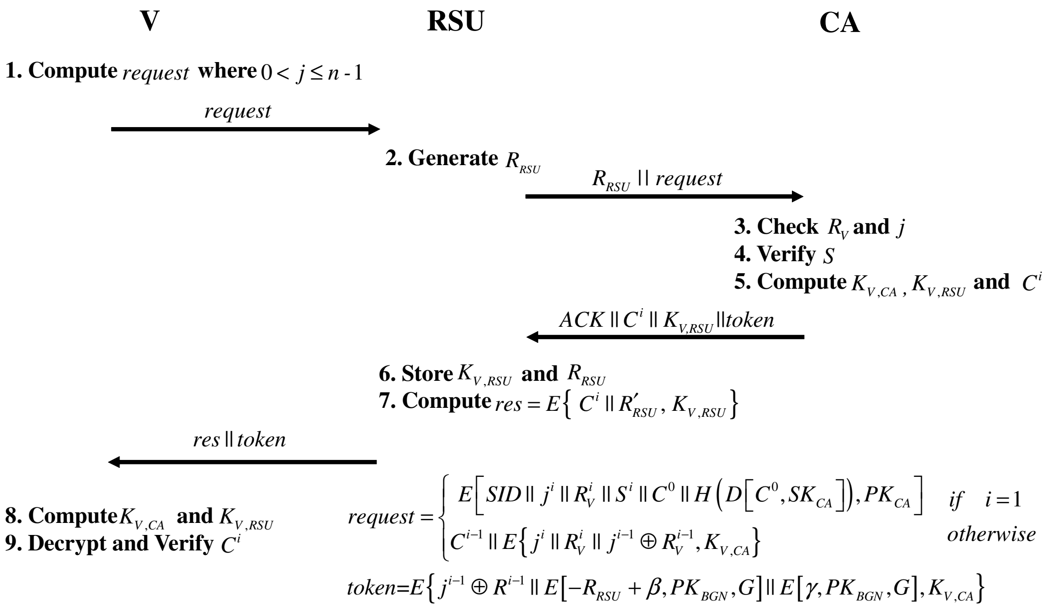
V
RSU
CA
1. Compute $request$ where $0 < j \leq n - 1$
$request$
2. Generate R_{RSU}
$R_{RSU} \,||\, request$
3. Check R_V and j
4. Verify S
5. Compute $K_{V,CA}$, $K_{V,RSU}$ and C^i
$ACK \,||\, C^i \,||\, K_{V,RSU} || token$
6. Store $K_{V,RSU}$ and R_{RSU}
7. Compute $res = E\{\, C^i \,||\, R'_{RSU}, K_{V,RSU}\}$
$res \,||\, token$
8. Compute $K_{V,CA}$ and $K_{V,RSU}$
9. Decrypt and Verify C^i
$request = \begin{cases} E\left[SID \,||\, j^i \,||\, R_V^i \,||\, S^i \,||\, C^0 \,||\, H\left(D\left[C^0, SK_{CA}\right]\right), PK_{CA}\right] & if \quad i = 1 \\ C^{i-1} \,||\, E\{j^i \,||\, R_V^i \,||\, j^{i-1} \oplus R_V^{i-1}, K_{V,CA}\} & otherwise \end{cases}$
$token{=}E\{j^{i-1} \oplus R^{i-1} \,||\, E[-R_{RSU} + \beta, PK_{BGN}, G] \,||\, E[\gamma, PK_{BGN}, G], K_{V,CA}\}$

Figure 6.3: V2I Authentication

4. Repeat the following steps until $z \leq p$.

   (a) If $C'^{(SK_{BGN})} = e(g,g)^{(z \cdot SK_{BGN})}$, return z.

   (b) z = z + 1

5. Return 0.

Using $f(x) = a_t \cdot x^t + a_{t-1} \cdot x^{t-1} + \cdots + a_1 \cdot x + a_0$ and the homomorphic properties of the BGN encryption scheme, we can change $\prod_{v=1}^{t-1} e(E[a_v, PK_{BGN}, \mathbb{G}], \, E[(w_j)^v, PK_{BGN}, \mathbb{G}])$ to C in the above procedure. Assuming that a_t and a_0 are both 1, C' in the above step (3) is the same as $E[i + r, PK_{BGN}, G_1] \cdot E[f(ID_V), PK_{BGN}, G_1] = E[(i + r) + f(ID_V), PK_{BGN}, G_1]$. If the driver is one of the legitimate service subscribers, the above computation $E[(i + r) + f(ID_V), PK_{BGN}, G_1]$ $= E[i, PK_{BGN}, G_1]$. Therefore, the CA can verify that the driver has proper access permission without identifying the driver. Only if the driver has proper access permission, the CA signs on C_V. As applying blind signature technique [15] to C_V, the CA cannot identify C^0. The detailed procedure is illustrated in Figure 6.2.

To verify whether the driver A has proper certificate $Cert_V$, its corresponding private key SK_V, and MT, the CA can request the driver A to send $D[C_T || K_S, SK_V]$. Because the legitimate driver A, having proper SK_V and performing vehicle registration, only can generate $D[C_T || K_S, SK_V]$. Although this approach can allow the CA to distinguish the received registration request with the eavesdropped registration request, this approach requires additional computation and communication overhead. When the CA receives the frequent vehicle registration from the same driver A, we recommend this approach.

6.2.3 V2I Authentication

In V2I authentication phase, the vehicle authenticates itself by sending the one-time credential, authorized by the CA, to the nearby RSU. We adopt entity authentication in [36], since the approach supports various security features (*i.e.*, mutual authentication, non-linkability, and enhanced level of security) with less computational overhead and communication cost. Figure 6.3 depicts this phase.

In the V2I authentication, each entity establishes $K_{V,AS}$ and $K_{V,RSU} = H(K_{V,CA}||C^i||R_{RSU})$.

$$K_{V,CA} = \begin{cases} H(C^0||PK_{CA}||R^1||j^1||SID) & \text{if } i = 1 \\ H(C^0||C^{i-1}||SID) & \text{otherwise} \end{cases}$$

To provide accountability of the authorized credential, we adopt a set of selected numbers S, which is l-bit array. In the first access request, a vehicle generates the set randomly. Whenever sending an i^{th} authentication request, the vehicle generates a fresh nonce R^i_{Entity} and selects one random number j between 0 to $l-1$ until $j-th$ value of S is 0. Since the set is only known to the vehicle and CA, the adversary without knowing S cannot generate the authentication request. Therefore, we believe that our protocol can enhance security level. Note that $C^i = H(C^0||j^i||R^i)$. For V2I authentication, the CA performs the following verification procedure:

1. 1^{st} request: After decrypting the request message, the CA computes $H(D[C^0\ ,SK_{CA}])$ and compares the result with the received $H(D[C^0, SK_{CA}])$. Only if the result is same, the CA believes that the entity has an authorized credential and computes $C^1 = H(C^0||j^1||R^1)$ and stores SID, S^1, C^0, and C^1 in the database. Otherwise, the CA discards the request.

2. i^{th} request: The CA finds C^0, $S^{(i-1)}$ and SID in the database using the received $C^{(i-1)}$ and decrypts the received message with $K_{V,CA}$. Next, the CA verifies that the entity has the same set of selected numbers and j^i is not in the set. Only if the result is correct, the CA stores the received C^i and S^i. Otherwise, the CA discards it. If the entity is a legal one with proper access permission, C^i and S^i are stored in the database. As a result, the CA can verify whether the entity has an authorized credential using the received $C^{(i-1)}$.

When the vehicle is a legitimate entity, the CA issues trapdoor $E[-R_{RSU} + \beta, PK_{BGN}, G]$ and $E[\gamma, PK_{BGN}, G]$ to the vehicle. Based on the received acknowledge, the nearby RSU receives sends $E\{C^i||R'_{RSU}, K_{V,RSU}\}$ and the proper *token*, to the vehicle. *token* consists of $E\{j^{i-1} \oplus R^{i-1}\ ||E[-R_{RSU} + \beta, PK_{BGN}, G]\ ||E[\gamma, PK_{BGN}, G], K_{V,CA}\}$. Since $E[-R_{RSU} + \beta, PK_{BGN}, G]$ and $E[\gamma, PK_{BGN}, G]$ are encrypted by the shared key with the vehicle $K_{V,CA}$, only the legitimate vehicle can obtain this information. Note that the CA generates β and γ which have the relation as $-R_{RSU} + \beta = \gamma$. In order to support non-linkability, the nearby RSU should generates a random R_{RSU} to each vehicle. Therefore, the CA should generate different β so that the same group has the same γ. This property can allow us to support numerous vehicles in the same group while reducing the computation and communication cost for VSS. From this point, we believe that our protocol can satisfy scalability requirement.

6.2.4 V2V Authentication

In V2V authentication phase, the vehicle m sends its $TICKET_m$ to the nearby vehicle n. Using $E[-R_{RSU} + \beta, PK_{BGN}, G]$, the vehicle n verifies whether the vehicle m has been authenticated with the CA. Figure 6.4 illustrates this phase. Through V2I authentication phase, the legitimate vehicle can obtain $E[-R_{RSU} + \beta, PK_{BGN}, G]$, which have the relation as $-R_{RSU} + \beta = \gamma$. In order to share a fresh session key $K_{m,n}$, the vehicle m selects a random point a on cyclic group G and sends g^a with $TICKET_m$.

After verifying $TICKET_m$, the vehicle n generates $TICKET_n$, selects a random point b on cyclic group G, computes $K_{m,n}$, and forwards $TICKET_n$ to the vehicle m. Although our protocol supports non-linkability, we can prevent misbehavior of a vehicle having the authorized credential. In order to execute V2V authentication, each vehicle should prove that it has been authenticated with the CA using $E[-R_{RSU} + \beta, PK_{BGN}, G]$, $H(R^{i-1} \oplus j^{i-1})$, and C^{i-1}. However, the adversary cannot generate the different $E[-R_{RSU} + \beta, PK_{BGN}, G]$ without knowing the actual value of β or γ. Therefore, the adversary should reuse his/her $E[-R_{RSU} + \beta, PK_{BGN}, G]$ to deceive the nearby vehicles.

The above authentication is useful when two-way communication between the vehicle m and n is required. However, when we want to send the emergency warning message, one-way communication from the vehicle m to n is better. Because one-way communication requires only one message transmission which reduces the propagation time of the emergency message. Then, the vehicle m sends its $TICKET'_m = C_m^{i-1} || MSG || E\{MSG || g^a || C_m^{i-1} ||\ H(R_m^{i-1} \oplus j_m^{i-1}) || RT_m, K_m\}$.

In addition, for stable operation of our proposed, we suggest combining our proposed scheme and Policing Traffic Management (PTM) [52] algorithm as rate control algorithm.

6.3 Analysis

6.3.1 Performance analysis

In this section, we analyze our protocol in detail. Table 6.2 shows the computational overhead in each phase. Note that $1/n$ indicates that one operation is required during n sessions and p is the

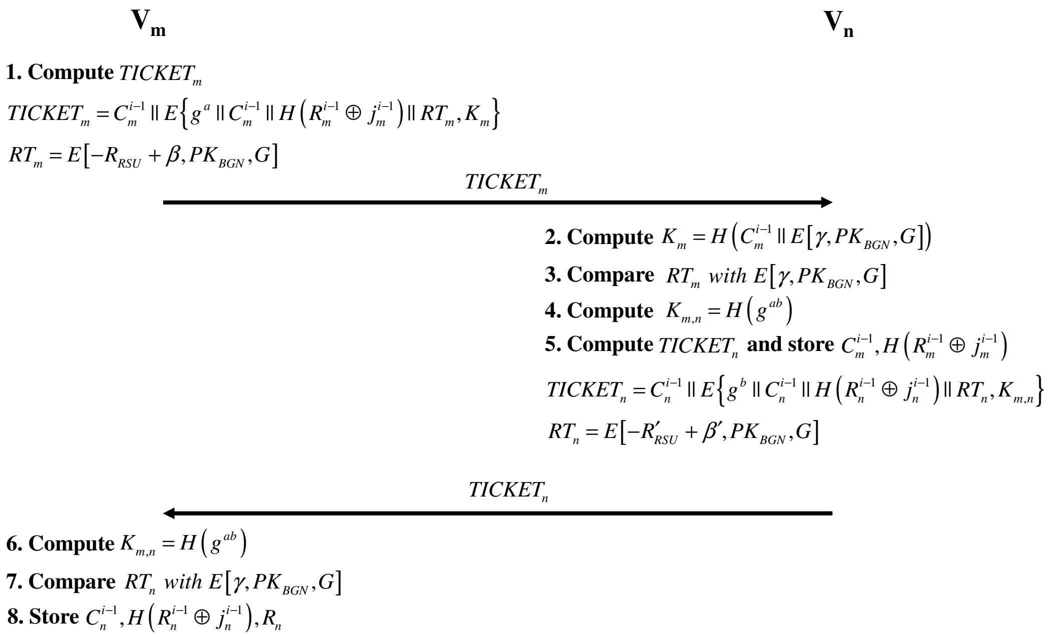
$\mathbf{V_m}$
$\mathbf{V_n}$
1. Compute $TICKET_m$
$TICKET_m = C_m^{i-1} \,\|\, E\left\{g^a \,\|\, C_m^{i-1} \,\|\, H\left(R_m^{i-1} \oplus j_m^{i-1}\right) \|\, RT_m, K_m\right\}$
$RT_m = E[-R_{RSU} + \beta, PK_{BGN}, G]$
$TICKET_m$
2. Compute $K_m = H\left(C_m^{i-1} \,\|\, E[\gamma, PK_{BGN}, G]\right)$
3. Compare RT_m *with* $E[\gamma, PK_{BGN}, G]$
4. Compute $K_{m,n} = H\left(g^{ab}\right)$
5. Compute $TICKET_n$ **and store** $C_m^{i-1}, H\left(R_m^{i-1} \oplus j_m^{i-1}\right)$
$TICKET_n = C_n^{i-1} \,\|\, E\left\{g^b \,\|\, C_n^{i-1} \,\|\, H\left(R_n^{i-1} \oplus j_n^{i-1}\right) \|\, RT_n, K_{m,n}\right\}$
$RT_n = E[-R'_{RSU} + \beta', PK_{BGN}, G]$
$TICKET_n$
6. Compute $K_{m,n} = H\left(g^{ab}\right)$
7. Compare RT_n *with* $E[\gamma, PK_{BGN}, G]$
8. Store $C_n^{i-1}, H\left(R_n^{i-1} \oplus j_n^{i-1}\right), R_n$

Figure 6.4: V2V Authentication

degree of the polynomial $f(x)$ used to enforce proper access permission. In our protocol, a vehicle m only requires two modular exponentiations to share a fresh session key with the nearby vehicle n. In our approach, vehicle registration phase is important to support anonymous communication in V2I and efficient verification of the authenticated vehicle. During vehicle registration, the CA can verify whether the driver has registered as a legitimate subscriber. Only if the driver is one of the legitimate service subscribers, the CA issues the authorized credential to the driver. Using the credential, the driver A can authenticate himself/herself with the nearby RSU. Although the computation overhead for vehicle registration is additional cost, vehicle registration can be done in the driver A's home. Hence, we believe that this phase does not affect the actual performance of our protocol. Also, through the verification of the authenticated vehicle, the vehicle m can authenticate itself to the vehicle n.

To illustrate the efficiency of our protocol, we compare our protocol with the existing approaches [43, 70] in Table 6.3 and 6.4. Through verification of the proposed algorithms in the existing approaches [43, 70], we can derive the computation overhead of their algorithms. Also, we refer to the computation overhead of VSS in V2I authentication as shown in [36]. Although our protocol and the approach by Yim *et al.* [70] can support mutual authentication, the protocol proposed by Lu *et al.* [43] only provide one-way authentication. Compared to the previous protocols [43, 70] requiring several pairing operations, our protocol needs 3 hash operations and 3 secret key operations as online computation, and $1/n$ hash operations and $1/n$ public key operation as off-line computation.

In addition, our protocol can support V2V communication using 3 hash operations, 1 secret key operation and 2 modular exponentiations as online computation, and 1 secret key operation as off-line computation. During V2V communication, the vehicle A does not need to communicate with the nearby RSUs. However, the previous protocols [43, 70] need heavy computation such as several public key operations and pairing operations.

From these points, we believe that our protocol has reduced the processing delay time for vehicle authentication. Through the reduced time, the nearby RSUs can authenticate more vehicles within the fixed time period. Therefore, our protocol can support better scalability than the previous protocols [43, 70].

Table 6.2: Computational overhead in each phase

	Registration		**V2I Auth.**			**V2V Auth.**	
	V	CA	V	RSU	CA	V_m	V_n
Public key Oper.	(2)† + 1	2	(1/n)†	0	2/n	0	0
Hash Oper.	0	0	(1/n)† + 3	0	(1/n)† + 3	3	3
Secret Key Oper.	2	2	3	1	2	(1)† + 1	(1)† + 1
Pairing Oper.	0	p-1	0	0	0	0	0
Modular Exp.	(2p+3)†	2	0	0	0	2	2
Modular Addition	(p+2)†	p+2	0	0	0	0	0

† : Precomputation Auth.: Authentication

Oper.: Operation Exp.: Exponentiation

6.3.2 Security analysis

Our protocol provides the following security-related features.

Mutual authentication: The vehicle authenticates the CA through a public key of the CA and knowledge of the corresponding private key. Also, the CA authenticates the end-user using an authorized credential of the vehicle.

Data confidentiality and integrity: All communications are protected by a shared session key or the receiver's public key. In this point, our protocol supports data confidentiality. Although we do not explain explicitly how to generate a key for integrity check, vehicle, RSU, and CA can derive the key using the shared information such as a fresh session key (or the receiver's public key) and exchanged nonce. By applying HMAC with the derived key, our protocol can support data integrity.

Table 6.3: Computational overhead comparison for V2I authentication

	Ours			**Yim *et al.***			**Lu *et al.***		
	V	RSU	CA	V	RSU	CA	V	RSU	CA
Public key Oper.	$(1/n)^\dagger$	0	2/n	2	2	1	0	0	0
Hash Oper.	$(1/n)^\dagger$	0	$(1/n)^\dagger$	2	1	0	1	1	2
Secret Key Oper.	3	1	2	1	1	0	4	1	1
Pairing Oper.	0	0	0	0	0	0	1	3	0
Modular Exp.	0	0	0	9	6	1	4	2	3
Modular Addition	0	0	0	3	2	0	2	1	0

† : Precomputation

Oper.: Operation Exp.: Exponentiation

Table 6.4: Computational overhead comparison for V2V authentication

	Ours		**Yim *et al.***		**Lu *et al.***	
	V_m	V_n	V_m	V_n	V_m	V_n
Public key Oper.	0	0	3	3	0	0
Hash Oper.	3	3	3	3	3	3
Secret Key Oper.	$(1)^\dagger + 1$	$(1)^\dagger + 1$	1	1	2	2
Pairing Oper.	0	0	0	0	9	9
Modular Exp.	2	2	12	12	24	24
Modular Addition	0	0	4	4	3	3

† : Precomputation

Oper.: Operation Exp.: Exponentiation

Non-linkability: Non-linkability means that, for insiders (*i.e.*, RSU and nearby vehicle) and outsiders, 1) neither of them can ascribe any session to a particular driver, and 2) neither of them can link two different sessions to the same driver [?]. More precisely, non-linkability needs to prevent insiders and outsiders from obtaining an driver's private information. Our protocol can achieve non-linkability with respect to both insiders and outsiders. First, the information to distinguish each driver is never transmitted in a plaintext form. As a result, outsiders cannot associate a session with a particular driver and ascribe two sessions to the same driver. Second, outsiders and insiders cannot find any relationship between the exposed credentials due to the one-way hash function. Finally, all communications are protected by a fresh session key.

Scalability: Since our protocol reduces the computational overhead compared to the previous approaches [43, 70], our protocol allows one RSU to authenticate the more vehicles within the certain time period. Moreover, our protocol does not require the participation of the nearby RSU in V2V authentication.

6.4 Summary

In this chapter, we have presented a scalable privacy-preserving authentication protocol for secure vehicular communications. Compared to the previous approaches [43, 70], our protocol excludes the participation of the nearby RSU so that the nearby vehicle from a vehicle A require less the time delay authenticating the vehicle A. As the nearby vehicles can authenticate the vehicle A without the help of the nearby RSU, we can save the transmission delay for sending authentication request of the vehicle A. In addition, our protocol requires less computational cost in V2I authentication and V2V authentication. From these points, our protocol increases the number of the vehicles which can be authenticated by one RSU.

Chapter 7. Concluding remarks

7.1 Summary

In this thesis, we focused on how to share private information and fast implementation methods of polynomial representation based set operations. We have shown a naive polynomial arithmetic for set operations results in a total of $\mathcal{O}(k^2)$ computational overhead in terms of multiplication. For the legacy polynomial arithmetics, we have proposed that better efficiency can be gained by applying the well-known Karatsuba scheme and divide-and-conquer based division algorithm. This results in reducing the total computational overhead from $\mathcal{O}(k^2)$ to $\mathcal{O}(k^{1.7})$ multiplications. Our proposed method can be useful building block for constructing privacy preserving set operations based on polynomial representation and make an efficient implementation of the most of set operation schemes via the polynomial representation.

In the literature, there proposed some solutions for solving privacy-preserving top-k problem. But, to the best of our knowledge, there is no formal security model for this protocol. In this research we have defined the formal model of privacy-preserving top-k query, and introduced the new security notion of owner privacy which is useful security requirements in the various applications. Based on our formal model, we proposed the first provably secure privacy preserving top-k query algorithm based on homomorphic encryption scheme and secret sharing method. We also have proved the security of our proposed scheme in the case of owner privacy and user privacy. Moreover our proposed scheme has better performance in computation rounds compared to prior solutions for the top-k query problem.

We also have presented a scalable privacy-preserving authentication protocol for secure vehicular communications. Compared to the previous approaches [43, 70], our protocol excludes the participation of the nearby RSU so that the nearby vehicle from a vehicle A require less the time delay authenticating the vehicle A. As the nearby vehicles can authenticate the vehicle A without the help of the nearby RSU, we can save the transmission delay for sending authentication request of the vehicle A. In addition, our protocol requires less computational cost in V2I authentication and V2V authentication. From these points, our protocol increases the number of the vehicles which can be authenticated by one RSU.

7.2 Further work

We conclude this thesis with commenting further work of our work. In the polynomial multiplication case, fast Fourier Transformation (FFT) is optimal complexity, $\mathcal{O}(k \log k)$. But FFT is hard to apply for encrypted polynomial, since the FFT requires whole n-th root of unity. So, the finding the quasi-quadratic arithmetic algorithms for set operations are open problems in next step.

Our proposed provably-secure PPT-k protocol was designed under the honest-but-curious (HBC) model. HBC model requires strong restriction that whole participants communicate each other following correct way. We remain the provably-secure PPT-k protocol under malicious model as future work.

Summary

Privacy-Preserving Information Aggregation and Authentication in the Distributed and Mobile Networks

컴퓨터 시스템 및 네트워크의 발전은 정보기술 분야에 있어 다양한 어플리케이션을 제공할 수 있도록 했다. 정보자원의 교류가 활발해짐에 따라 프라이버시 보호기능은 정보공유에 있어서 필수적인 요소로 자리잡고 있다. 본 연구에서는 이러한 정보보호의 기술 중, 분산환경과 이동환경에서 안전한 자료의 수집에 관한 주제와 다수 사용자간의 효율적인 인증방식에 대한 주제에 대한 해결을 시도하였다.

첫번째는 분산환경에서 프라이버시 보호 기능을 유지하며 자료의 수집과 공유에 관한 연구다. 이를 해결하기 위한 방법으로 분산환경에서의 안전한 집합연산에 관한 연구가 다수 진행되었다. 우리는 우선 기존 기법 중 다항식 재표현 기반의 집합연산의 속도를 개선하기 위한 암호화된 다항식에서의 다항식 연산 알고리즘을 개발하였다. 카라추바가 제시한 기법과 분할-탐색 방식을 적용하여 다항식 곱셈, 확장, 함수값 계산 방식을 제안하였고, 사용자의 비밀집합의 개수가 k개 일 때, 기존 방식의 계산 복잡도 $\mathcal{O}(k^2)$ 으로부터 $\mathcal{O}(k^{\log_2 3})$ 으로 낮출 수 있었다.

분산환경 다자간 통신 환경에서 제시된 문제 중 상위 k개 원소를 탐색하는 기법을 제시하였다. 기존의 연구결과들은 엄밀한 공격목표와 안전성목표가 제시되지 않아서, 증명가능 안전성을 제시하지 못했다. 본 연구에서는 상위 k개 원소 탐색 기법에 최초로 증명가능 안전성 모델을 적용하였다. 새로운 안전성 개념인 소유자 보호 개념을 정의하였다. 또한 제안한 안전성 모델에 기반하여 안전성이 증명가능한 상위 k개 원소 탐색 기법을 제시하고, 그 안전성을 확인하였다.

두번째 연구주제는 이동 네트워크에서 (특별히 차량 애드혹 네트워크) 인증 기법을 설계했다. 기존의 익명성을 제공하는 기법이 가지고 있는 다수 차량이 집중되는 상황에서의 인증 속도 개선을 연구했으며, 이를 위해 빠른 검증 과정과 노변기기의 개입을 최소화 한 기법을 제시하였다. 호모몰픽 암호기법 중 BGN암호 방식을 사용하여 충반한 속도 개선을 이루었으며, 노변기기의 개입이 줄이고, 인증된 차량간의 상호 검증하는 기법을 제시하였다. 제안 기법은 다수 차량이 밀집하게 되는 경우 인증 시간이 오래걸리는 문제를 효과적으로 해결할 수 있었다.

References

[1] J. Al-Muhtadi, R. Campbell, A. Kapadia, D. Mickunas, and S. Yi, Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing, Proc. ICDCS, Vienna, Austria, 2002, pp. 65–74.

[2] J. Al-Muhtadi, A. Ranganathan, R. Campbell and M. Mickunas, A Flexible, Privacy-Preserving Authentication Framework for Ubiquitous Computing Environments, Proc. 22nd International Conference on Distributed Computing Systems (ICDCS), 2002, pp. 771–776.

[3] B. Applebaum, H. Ringberg, M. Freedman, M. Caesar, and J. Rexford, Collaborative, privacy-preserving data aggregation at scale. In *Privacy Enhancing Technologies (PET 2010)*, LNCS 6205, pages 56–74, 2010.

[4] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, Public key encryption with keyword search, Advances in Cryptology (EUROCRYPT '04), LNCS 3027, pp. 506-522, 2004.

[5] M. Burkhart and X. Dimitropoulos, Fast privacy-preserving top-k queries using secret sharing. In *Proceedings of International Conference on Computer Communications and Networks (ICCCN 2010)*, 2010.

[6] D. Boneh, E.-J. Goh and K. Nissim, Evaluating 2-dnf formulas on ciphertexts, Theory of Cryptography (TCC 2005), LNCS 3378, pp. 325-341, 2005.

[7] M. Bodrato, Towards optimal toom-cook multiplication for univariate and multivariate polynomials in characteristic 2 and 0, in WAIFI 2007, ser. LNCS, C. Carlet and B. Sunar, Eds., vol. 4547. Springer-Verlag, 2007, pp. 116–133.

[8] J. Baek, R. Safavi-Naini and W. Susilo, Public Key Encryption with Keyword Search Revisited, Cryptology ePrint Archive, Report 2005/191.

[9] D. Boneh, G. Segev, and B. Waters, Targeted malleability: homomorphic encryption for restricted computations, Cryptology ePrint Archive, Report 2011/311, 2011.

[10] M. Burnside et al., Proxy-Based Security Protocols in Networked Mobile Devices, Proc. ACM SAC, Madrid, Spain, 2002, pp. 265–272.

[11] Z. Brakerski and V. Vaikuntanathan, Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages, In *Advances in Cryptology-CRYPTO 2011*, LNCS 6841, pages 505–524, 2011.

[12] R. Cramer, I. Damgård, and J. B. Nielsen, Multiparty computation from threshold homomorphic encryption, in Advances in Cryptology-EuroCrypt 2001, ser. LNCS, B. Pfitzmann, Ed., vol. 2045. Springer-Verlag, 2001, pp. 280–299.

[13] S. Creese, M. Goldsmith, B. Roscoe, and I. Zakiuddin, Authentication for Pervasive Computing, Proc. Security in Pervasive Computing 2003, 2004, vol. 2802, pp.116–129.

[14] D. Chaum and E. van Heijst, Group signatures, in Proc. Advances in Cryptology - Eurocrypt '91, LNCS, 196(1984), 257-265.

[15] D. Chaum, Untraceable Electronic Mail, Return Address, and Digital Pseudonyms, Communications of the ACM, vol. 24, no. 2, pp. 84–88, 1981.

[16] J. Coron, A., D. Naccache and M. Tibouchi, Fully Homomorphic Encryption over the Integers with Shorter Public Keys, In *Advances in Cryptology-CRYPTO 2011*, LNCS 6841, pages 487–504, 2011.

[17] J. Camenisch and M. Stadler, Proof systems for general statements about discrete logarithms, Department of Computer Science, ETH Zürich, Tech. Rep. TR 260, 1997.

[18] J. Camenish and G. Zaverucha, Private intersection of certified sets, in Financial Cryptography 2009, ser. LNCS, R. Dingledine and P. Golle, Eds., vol. 5628. Springer-Verlag, 2009, pp. 108–127.

[19] Y. Desmedt and K. Kurosawa, How to break a practical mix and design a new one. In *Advances in Cryptology-EUROCRYPT 2000*, LNCS 1807, pages 557–572, 2000.

[20] D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung, Efficient robust private set intersection, in ACNS 2009, ser. LNCS, M. Abdalla, D. Pointcheval, P.-A. Fouque, and D. Vergnaud, Eds., vol. 5536. Springer-Verlag, 2009, pp. 126–142.

[21] M. J. Freedman, K. Nissim and B. Pinkas, Efficient Private Matching and Set Intersection, Advances in Cryptography (EUROCRYPT '04), LNCS 3027, pp. 1-19, 2004.

[22] M. Gruteser and D. Grunwald, Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis, Mobile Networks and Applications, vol. 10, no. 3, pp. 315–325, 2003.

[23] O. Goldreich, The Foundations of Cryptography. Cambridge University Press, 2004, vol. 2.

[24] P. Golle, J. Staddon and B. Waters, Secure Conjunctive Search over Encrypted Data, Proc. Applied Cryptography and Network Security (ACNS '05), LNCS 3089, pp. 31-45, Jun. 8-11, China.

[25] Homomorphic encryption, `http://en.wikipedia.org/wiki/Homomorphic_encryption` (accessed 05/04/11).

[26] D. Hankerson, J. Löpez, and A. Menezes, Software implementation of elliptic curve cryptography over binary fields, in CHES 2000, ser. LNCS, Çetin Kaya Koç and C. Paar, Eds., vol. 1965. Springer- Verlag, 2000, pp. 1–24.

[27] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. F. Abdelzaher, Pda: Privacy-preserving data aggregation in wireless sensor networks. *In Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, pages 2045–2053. IEEE, 2007.

[28] W. G. Horner, A new method of solving numerical equations of all orders, by continuous approximation, Philosophical Transactions Royal Society of London, 1819, pp. 308—335.

[29] Q. He, D. Wu and P. Khosla, Quest for Personal Control over Mobile Location Privacy, IEEE Commun. Mag., vol. 42, no. 5, pp. 130–136, May 2004.

[30] M. Imani, M. Taheri, and M. Naderi, Security enhanced routing protocol for ad hoc networks, Journal of Convergence (JoC), 1:1(2011), 43-48.

[31] M. Jakobsson, A practical mix. In *Advances in Cryptology- EUROCRYPT'98*, LNCS 1403, pages 448–461, 1998.

[32] T. Jebelean, Practical integer division with Karatsuba complexity," in ISSAC, W. Kuchlin, Ed. ACM Press, 1997, pp. 339–341

[33] U. Jendricke, M. Kreutzer and A. Zugenmaier, Pervasive Privacy with Identity Management, in Proc. 1st Workshop Security, Ubicomp, 2002.

[34] D. Kumar, T. C. Aseri, R.B. Patel, Multi-hop communication routing (MCR) protocol for heterogeneous wireless sensor networks, International Journal of Information Technology, Communications and Convergence (IJITCC), 1:1(2011), 130-145.

[35] J. Kim, J. and T. Shon, An Efficient and Scalable Re-authentication Protocol over Wireless Sensor Network, IEEE Transactions on Consumer Electronics, Vol 57, Issues 2, May 31, 2011, pp 516-522, ISSN: 0098-3063

[36] J. Kim, J. Baek, J. Zhou, K. Kim, and T. Shon, An Efficient and Secure Service Discovery Protocol for Ubiquitous Computing Environments, in Proc. of 7th European Workshop on Public Key Services, Applications and Infrastructures (EuroPKI 2010), LNCS, 6711(2010), 45-60.

[37] D. Knuth, The Art of Computer Programming: Seminumerical Algorithms, 3rd ed. Addison-Wesley, 1998, vol. 2.

[38] A. A. Karatsuba and Y. Ofman, Multiplication of multidigit numbers on automata, in Soviet Physics Doklady, no. 7, 1963, pp. 595–596.

[39] L. Kissner and D. Song. Privacy-preserving set operations. In V. Shoup, editor, In *Advances in Cryptology-CRYPTO 2005*, LNCS 3621, pages 241–257, 2005.

[40] L. Kissner and D. Song, Privacy-preserving set operations, Carnegie Mellon University, Tech. Rep. CMU-CS-05-113, 2006.

[41] M. Langheinrich, A Privacy Awareness System for Ubiquitous Computing Environments, Proc. UbiComp, 2002, vol. 2498, pp. 237–245.

[42] J. Lopez and R. Dahab, High-speed software multiplication in $\mathbb{F}_{2^m}$, in Indocrypt 2000, ser. LNCS, B. K. Roy and E. Okamoto, Eds., vol. 1977. Springer-Verlag, 2000, pp. 203–212.

[43] R. Lu, X. Lin, H. Zhu, P. Ho, and X. Shen, ECPP: Efficient Conditional Privacy Preservation Protocol for Secure Vehicular Communications, in Proc. of The 27th IEEE Conference on Computer Communications (INFOCOM 2008), (2008), 1229-1237.

[44] Y. Lindell and B. Pinkas. Secure multiparty computation for privacy-preserving data mining, In *the Journal of Privacy and Confidentiality*, 1(1):59–98, 2009.

[45] X. Lin, X. Sun, P.H. Ho, and X. Shen, GSIS: A Secure and Privacy-pserving Protocol for Vehicular Communications, IEEE Transactions on Vehicular Technology, 56(2007), 3442-3456.

[46] R. Li and C. Wu, An unconditionally secure protocol for multi party set intersection, in ACNS 2007, ser. LNCS, J. Katz and M. Yung, Eds., vol. 4521. Springer-Verlag, 2007, pp. 226–236.

[47] P. L. Montgomery, Five, six, and seven-term Karatsuba-like formulae, in IEEE Transactions on Computers, vol. 54. IEEE Computer Society, 2005, pp. 362–369.

[48] C. Neff, A verifiable secret shuffle and its application to e-voting. In *ACM Conference on Computer and Communications Security (ACM-CCS 2001)*, pages 116–125, 2001.

[49] K. Nahanishi, J. Nakazawa, and H. Tokuda, LEXP: Preserving User Privacy and Certifying Location Information, Proc. 2nd Workshop Security Ubicomp, 2003.

[50] P. Paillier, Public-key cryptosystems based on composite degree residuosity public-key cryptosystems based on composite degree residuosity classes, in Advances in Cryptology-EuroCrypt'99, ser. LNCS, J. Stern, Ed., vol. 1592. Springer-Verlag, 1999, pp. 223–238.

[51] A. Patra, A. Choudhary, and C. P. Rangan, Information theoretically secure multi party set intersection re-visited, in Selected Area of Cryptography 2009, ser. LNCS, M. J. J. Jr., V. Rijmen, and R. Safavi-Naini, Eds., vol. 5867. Springer-Verlag, 2009, pp. 71–91.

[52] S. Prahmkaew, Performance Evaluation of Convergence Ad Hoc Networks, Journal of Convergence (JoC), 1:1(2011), 101-106.

[53] K. Ren, W. Lou, Privacy Enhanced Access Control in Ubiquitous computing Environments, 2nd International Conference of Broadband Networks 2005, Vol. 1, pp. 356–365, 3-7 Oct. 2005.

[54] K. Ren, W. Lou, K. Kim and R. Deng, A Novel Privacy Preserving Authentication and Access Control Scheme for Pervasive Computing Environments, IEEE Transactions on Vehicular Technology, vol. 55, no. 4, pp. 1373–1384, July 2006.

[55] K. Rosen, Discrete Mathematics and Its Applications, 6th ed. McGraw-Hill Higher Education, 2007.

[56] A. Shamir, How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[57] A. Shamir, Identity-based cryptosystems and signature schemes, in Proc. Advances in Cryptology - Crypto '84, LNCS, 8(1984), 47-53.

[58] V. Shoup, A computational introduction to number theory and algebra, 2nd ed. Cambridge University Press, 2008.

[59] Y. Sang, H. Shen, Y. Tan, and N. Xiong, Efficient protocols for privacy preserving matching against distributed datasets, in ICICS 2006, ser. LNCS, P. Ning, S. Qing, and N. Li, Eds., vol. 4307. Springer-Verlag, 2006, pp. 210–227.

[60] Y. Sang and H. Shen, ,Privacy preserving set intersection protocol secure against malicious behaviors, in PDCAT 2007. Washington, DC, USA: IEEE Computer Society, 2007, pp. 461–468.

[61] Y. Sang and H. Shen, Privacy preserving set intersection based on bilinear groups, in ACSC 2008, vol. 74. Darlinghurst, Australia: Australian Computer Society, 2008, pp. 47–54.

[62] J. Vaidya and C. Clifton, Privacy-preserving top-k queries. In *Proceedings of the 21st International Conference on Data Engineering (ICDE2005)*, pages 545–546, 2005.

[63] M. Wu and A. Friday, Integrating Privacy Enhancing Services in Ubiquitous Computing Environments, Workshop on Security in Ubiquitous Computing, 4th International Ubicomp, 2002.

[64] A. Weimerskirch and C. Paar, Generalizations of the Karatsuba Algorithm for Efficient Implementations, IACR eprint archive, `http://eprint.iacr.org/2006/224`, 2006.

[65] C. D. Wang and J. P. Thompson, Apparatus and method for motion detection and tracking of objects in a region for collision avoidance utilizing a real-time adaptive probabilistic neural network, 1997. US. Paten No. 5,613,039.

[66] F.Y. Wang, D. Zeng, and L. Yang, Smart Cars on Smart Roads: An IEEE Intelligent Transportation Systems Society Update, IEEE Pervasive Computing, 5:4(2006), 68-69.

[67] L. Xiong, S. Chitti, and L. Liu, Top k queries across multiple private databases. In *Proceedings of International Conference on Distributed Computing Systems (ICDCS 2005)*, pages 145–154, 2005.

[68] B. Xie, A. Kumar, D. Zhao, R. Reddy, and B. He, On secure communication in integrated heterogeneous wireless network, International Journal of Information Technology, Communications and Convergence (IJITCC), 1:1(2011), 4-23.

[69] A. Yao, Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (FOCS '82)*, pages 160–164, 1982.

[70] J. Yim, I. Choi, and K. Kim, An Efficient Anonymous Authentication Protocol in Vehicular Adhoc Networks, in Proc. of The 10th International Workshop on Information Security Applications (WISA 2009), Aug. 24-26, (2009).

[71] S. S. Yau and Y. Yin, Controlled Privacy Preserving Keyword Search, Proc. ACM Symposium on Information, Computer & Communication Security (ASIACCS '08), pp. 321-324 , Mar. 18-20, 2008, Tokyo, Japan. ETRI Journal, Vol. 32, No. 5, October 2010, pp. 704-712.

[72] A. Zugenmaier and A. Hohl, Anonymity for Users of Ubiquitous Computing, Proc. Security Workshop in Ubicomp, Seattle, Washington, Oct. 2003.

Curriculum Vitae

Name : Kim, Zeen

Date of Birth : August 30, 1977

Domicile : 983-302, Gwanpyoung Dong, Yuseong Gu, Daejeon, 701-801 KOREA

Address : 983-302, Gwanpyoung Dong, Yuseong Gu, Daejeon, 701-801 KOREA

E-mail : zeenkim@kaist.ac.kr

Educations

2001. 2. B.S. in Mathematics at Korea Advanced Institute of Science and Technology, Korea

2004. 2. M.S. in Engineering at Information and Communications University, Korea

2012. 8. Ph. D. in Information and Communications Engineering at Korea Advanced Institute of Science and Technology, Korea

Publications

1. Zeen Kim, Jangseong Kim, Doyoung Chung, Kwangjo Kim, Taeshik Shon, "A Scalable and Efficient Privacy-preserving Authentication Protocol for Secure Vehicular Communications", INFORMATION - International Interdisciplinary Journal (SCIE), Accepted, 2012
2. Zeen Kim and Kwangjo Kim, Generic polynomial arithmetics for secure set operations with subquadratic complexity, Submitted to IEEE Transactions on Computers, 2012
3. Zeen Kim and Kwangjo Kim, Provably-secure and privacy-preserving top-k queries using homomorphic encryption, Preprint, will be submitted to IEICE Transactions on Fundamentals, 2012
4. Zeen Kim, Junhyun Yim, Jangseong Kim, Kwangjo Kim, and Taeshik Sohn, "Traceable Anonymous Authentication Scheme for Vehicular Ad-hoc Networks", Proceedings of IEEE ISPA 2011, Busan, May 26-28, 2011
5. Zeen Kim, Jangseong Kim, Kwangjo Kim, Imsung Choi, and Taeshik Shon, "Untraceable and Serverless RFID Authentication and Search Protocols", Proceedings of IEEE ISPA 2011, Busan, May 26-28, 2011

6. Hyewon Park, Zeen Kim, and Kwangjo Kim, ”Forward Secure ID-based Group Key Agreement Protocol with Anonymity”, The Third International Conference on Sensor Technologies and Applications (SENSORCOMM 2009), June 18-23, 2009, Athens/Glyfada, Greece. -Best Paper Award

7. Imsung Choi, Zeen Kim, and Kwangjo Kim, ”DoS-Resilient Authenticated Key Agreement Scheme between Actor and Sensor nodes in Wireless Sensor and Actor Network”, Joint Workshop on Information Security 2009, Aug. 6-7, 2009, Kaohsiung, Taiwan.

8. Zeen Kim and Kwangjo Kim, ”Mutually Authenticated Key Exchange Protocol For Computationally Limited Devices”, Triangle Symposium on Advanced ICT 2008 (TriSAI 2008), Oct. 6-9, 2008, Daejeon, Korea.

9. Sungbae Ji, Zeen Kim, and Kwangjo Kim, ”Design of an RFID-embedded e-ID System for Privacy Protection”, Proc. Of SCIS 2008, Jan. 22-25, 2008, Miyajaki, Japan.

10. Jangseong Kim, Zeen Kim, and Kwangjo Kim, ”A Lightweight Privacy Preserving Authentication and Access Control Scheme for Ubiquitous Computing Environment”, The 10th International Conference on Information Security and Cryptology, LNCS 4817, pp.37-48 , Nov. 29-30, 2007, Seoul,Korea.

11. Sungchul Heo, Zeen Kim, and Kwangjo Kim, ”Certificateless Authenticated Group Key Agreement Protocol for Dynamic Groups”, The 50th IEEE Global Telecommunications Conference, Nov. 26-30, 2007, Washington, D.C., USA.

12. Divyan M. Konidala, Zeen Kim and Kwangjo Kim, ”A Simple and Cost-effective RFID Tag-Reader Mutual Authentication Scheme”, Pre-Proc. of International Conference on RFID Security 2007 (RFIDSec 07), pp.141-152, July 11-13, 2007, Malaga, Spain.

13. Zeen Kim, Jangseong Kim, and Kwangjo Kim, ”Key Predistribution Scheme for Wireless Sensor Networks with Higher Connectivity”, Proc. Of SCIS 2007, Abstracts pp.235, Jan. 23-26, 2007, Sasebo, Japan.

14. Dang N. Duc, Kyusuk Han, Zeen Kim, and Kwangjo Kim, ”A New Transitive Signature Scheme based on RSA-based Security Assumptionss”, Proc. of Industrial and Short-Papers Track in ACNS2005, pp.165-175, June 7-10, 2005, NY, USA.

15. Jaemin Park, Zeen Kim, and Kwangjo Kim, ”State-Based Key Management Scheme for Wireless Sensor Networks”, Proc. of WSNS2005, pp.819-825, Nov. 7, 2005, Washington, D.C., USA.

16. Dang Nguyen Duc, Zeen Kim, and Kwangjo Kim, ”A New Provably Secure Transitive Signature Scheme”, Proc. of SCIS 2005, Jan. 25-28 Maiko Kobe, Japan.

17. 김진, 김광조, ”Li 등의 ID기반 방송형 Signcryption기법의 안전성 분석”, CISC-S’10 Proceedings, pp.11-13, 2010.6.18, POSTECH, 포항.

18. 최임성, 김진, 김광조, ”무선센서 네트워크에서 안전하고 효율적인 방송형 인증 기법 연구, CISC-W’09 Proceedings, pp.206-213, 2009.12.5, 연세대학교, 서울.

19. 최임성, 김진, 김광조, ”무선 센서 네트워크에서 안전한 클러스터링 프로토콜들의 안전성 분석”, 2008년도 한국정보보호학회 충청지부 학술발표회 논문집, pp.85-91, 2008.10.17, 배재대학교, 대전.

20. Divyan M Konidala, Zeen Kim, Chan Yeob Yeun, Jin Li, and Kwangjo Kim, ”Secure Approach to Deploy RFID-based Applications in Smart Home Environment”, CISC-W’07 Proceedings , pp.717-720, 2007.12.1,상명대학교, 서울.

21. 김진, 김장성, 강영두, 정충희, 김광조, ” 원자력 발전소 디지털 시스템의 보안 요구 사항 ”, CISC-W’07 Proceedings, pp.248-251, 2007.12.1, 상명대학교, 서울.

22. Zeen Kim, Jangseong Kim, Youngdoo Kang, Kwangjo Kim, Dai I. Kim and Choongheui Jeong, ”Guideline of Cyber Security Policy for Digitial I&C Systems in Nuclear Power Plant”, in Transactions of the Korean Nuclear Society Autumn Meeting, Oct. 25-26, 2007, PyeongChang, Korea.

23. 김진, 김중만, 김광조, ”VOIP 키복구 시스템의 보안 요구사항”,2006년도 정보보호학술발표회논문집, pp. 127-133, 2006. 9.29-30, 목원대학교, 대전.

24. 이상신, 김진, 김광조, ”저가형 RFID를 위한 효율적인 프라이버시 보호 기법”, 2005년 한국정보보호학회 하계정보보호학술대회 논문집, pp.569-573, 2005.6.3, 조선대학교, 광주.

25. 박재민, 김진, 김광조, ”State-Based Random Key Pre-Distribution Scheme for Wireless Sensor Networks”, 2005년 한국정보보호학회 하계정보보호학술대회 논문집, pp.37-42, 2005.6.3, 조선대학교, 광주.