Private Information Extraction over Online Social Networks

Huang Lin*, Yuguang Fang*, Zhenfu Cao†
*Department of Electrical and Computer Engineering, University of Florida,
†Department of Computer Science & Engineering, Shanghai Jiao Tong University

***Abstract*—Due to the popularity of online social networks (OSNs), various online surveys have been done over OSNs to help researchers extract information of human behaviors on various aspects, ranging from online purchasing to disease epidemic patterns. Since online surveys usually attempt to extract the statistical features over a large population, the more participants respond, the more accurate the results will be, and hence the greater the social utility of such survey will be. Despite the growing importance of online surveys in modern social life, people are generally reluctant to respond to an online survey due to the possible privacy leakage especially when the questionnaire in the survey is related to sensitive personal information such as the health related issues, or even if they choose to respond to the survey, people might deliberately twist their responses to protect their privacy. On the other hand, low response rate of online surveys will lead to biased or even unqualified results. How to find an effective way to increase the response rate while preserving responders' privacy to some extent is a challenging problem. In this paper, we design two privacy-preserving online survey protocols which enable the inquirer to extract two most important statistical data: the intersection and the union information of responders' choices. The intersection information implies the common choice selected by all the responders while the union information corresponds to the preference of each choice in the survey. We formally prove that the proposed schemes are secure. We have also carried out extensive study and shown that the proposed schemes are more efficient than the related works. Moreover, we also discuss how to make the two basic schemes accommodate dynamic group formation and extend our schemes without central key authority.**

I. Introduction

With the increasing popularity of online social networks and mobile computing, online survey involving a large number of people has become an important application to facilitate different aspects of social utility. Nowadays, various organizations distribute surveys among popular social networks such as Facebook, Myspace, or LinkedIn for various purposes. Telemarketing could use online surveys to gather consumption structure and marketing interest information among different categories of consumers. For instance, researchers [1] have been conducting surveys on the typology of online purchasing behavior characterized by psychographic and behaviorial variables through online questionnaires published over OSNs. Questionnaires have also been posted over OSNs to study health related issues. Online surveys [2], [3] have been used as a tool to make disease related event detection and prediction and the study has shown many subtle behaviorial changes among the affected individuals due to illness. Such behaviorial changes can be used to recommend the infected individual for a hospital visit before they get seriously ill. Through timely survey on a large group of high risk population on their behaviorial changes such as the increased consumption of over-the-counter medications and/or changes in mobility patterns can enable caregivers to timely detect the emerging disease burst and pinpoint the affected subset of the population.

While apparently online surveys could benefit our society in various ways, there is also high risk of privacy breach for the respondents to those online surveys. Those who submit responses to the online commercial study may face the leakage of their personal online purchasing habit information and behaviorial structure to the inquirer, which could possibly further lead to the leakage of one's financial status. Similarly, people who participate in online surveys related to medical or disease issue may risk leaking their personal health information, which is considered as one of the most private information in their personal life. Research findings in [4], [5], [6] have shown that privacy concern has been the major cause of low response rate or unreliable responses, which is one of the leading factor of biased or even unqualified results in online surveys. Despite the growing importance and popularity of online survey, little efforts have been made in improving the privacy preservation of participants in online surveys. Although there are some works [7], [5] focusing on improving the design of questionnaires themselves to prevent privacy breach, there are few works concerning the improvement of online survey protocol itself. Most related works (see Sec. III) either cannot truly solve the privacy issue, or simply too inefficient to be feasible in practice.

Meanwhile, we have observed that the target information that the inquirer is concerned with is generally some statistical information. In other words, there is huge information redundancy for the inquirer in the responses provided by the participants. Sometimes, people might be willing to reveal these target statistical information to serve their own benefit while not disclosing other sensitive personal information. For instance, suppose that the investigators in [8] of recent Escherichia coli (E. coli) outbreak in Europe may want to find out which type of the following vegetables to be blamed for the outbreak: lettuce, tomatoes, cucumbers or bean sprout. The only information they are concerned with is the intersection of the vegetables the infected people all have eaten. Any other information is irrelevant to the study and should be considered as privacy breach in this case. Obviously, this intersection information is very helpful in figuring out the positive correlation among different choices in the questionnaires. How to design an online survey protocol to guarantee that the inquirer can only obtain the intersection information without leaking any other information remains an open question. This paper attempts to provide a solution to this interesting problem.

Intersection information can only provide limited information for the inquirer, and sometimes this might not be sufficient. The union information of an online survey which composes of the count of responders selecting each choice listed in a posted questionnaires is another important tool for data analysis [9] and data mining. Actually it contains all the information needed for various data analysis such as max/min, average, or histogram while at the same time it properly protects the privacy for the participants by hiding the linkage between the response and those who submit the response. Traditional approaches to dealing with similar problem such as secure online voting include using anonymous channel or homomorphic encryption to make the voter untraceable. However, those approaches suffer from either inefficiency or weak robustness and thus cannot fully solve the privacy problem. This paper also provides a secure online survey protocol which can enable the inquirer to extract union information from the collected responses without leaking any other information.

A. Our Contributions

In this paper, we have made the following few contributions:

(1). This paper formally defines the requirement for a secure online survey protocol. Two secure online survey protocols are designed. One is to guarantee that the inquirer can only obtain the intersection information without leaking any other information while the other one is to make sure that the only information the inquirer get is the union information. Our scheme is efficient in the sense that users only need to encrypt their responses and submit their responses to the inquirer for once. This paper also provides comparison between our solution and several other related works. The analysis shows that our scheme is much more efficient.

(2). Our basic scheme for extracting intersection information or union information has a trusted key distributor responsible for distributing secret keys for the inquirer and individual users. This paper provides another approach to remove the central key distributor, which means that there is no privacy breach even without any trusted authority in the system. The basic scheme requires each individual user to submit his answer to the inquirer to guarantee that the inquirer can successfully extract the target private information. By incorporating secret sharing technique, the basic scheme can further guarantee that even some responses are missing, the inquirer can still extract the target information from the rest remaining responders.

II. Preliminaries

Before we present our protocols, we need some preliminaries used in our design. The system to be considered consists of three kinds of parties or entities: the key authority A, the inquirer I, and the system users (participants or responders in an online survey) $u_1, \cdots, u_n$. At the system initialization, A is responsible for distributing secret keys to the users and the inquirer. The system lifetime is divided into several time periods. At the beginning of each time period t, I publishes his online questionnaire. The questionnaire is composed of several multiple choice questions, although these questions may not necessarily be provided in the form of multiple choice questions. For instance, a typical survey question about the age can still be posed as a multiple choice question, say, whether the response is a choice in the interval $[1, 150]$, assuming the maximum age is 150. All the possible choices of those questions constitute a choice universe U, which will be published along with the questionnaire. We notice that the questionnaire should also include the target information of an online survey, which would specify the category of target private information which the inquirer intends to extract from the response data. Each user makes his own choices from U and encrypts the choices to make sure that I can only extract the target information. Thus, users return their generated individual ciphertext to the inquirer, and the inquirer decrypts the received data using his secret key to obtain the target information.

In this paper, we design two schemes which enable I to extract two different kinds of target information as follows.

Target intersection information: The first kind of target information is the intersection information, and we call a scheme which can enable I extract the intersection information without learning other information about the responders an *intersection information extractor*. Consider the whole online survey as a unique multiple choice question and let the corresponding choice universe be $U = \{a_j\}_{j=1}^{m}$. For instance, suppose there are 10 five-choice questions in an online survey, then the choice universe is considered as $U = [1, 50]$. The users in our proposed scheme can not only choose any combination of the choices in U, but also can determine the rank of each choice. This would certainly make a more flexible survey since the rank of a choice could represent some other related information. Take the Escherichia coli investigation as an example, suppose a choice represents the responser eats tomato on a daily base, then it implies that the user may eat at least two pounds of tomato daily assuming the user chooses rank 2 for the choice tomato. The intersection information indicates the identical choices among the returned response from n system users. Suppose there are ten users in the system, then I should be able to identify which choices $a_i \in U$ are chosen by all these users. Better than this, the intersection information also tells I the minimum rank of a_i among the users in our system.

Target union information: The second kind of target information is the union information. Consider a system where the inquirer publishes an online survey with m choices, which means $U = \{a_j\}_{j=1}^{m}$. The union information denotes the collection of the number of times c_i choosing choice a_i by all users. The inquirer I obtains no other information on the individual choice, which implies that there is no way for him to figure out the response of any proper subset of the n system users.

We assume the network environment is an OSN such as Facebook. The protocol assumes an underlying secure channel for the secret key distribution from the key authority A to the other parties. This could be established easily when each of the engaging parties owns a public/private key pair. We are concerned about the insiders, the participants of the protocol, either the key authority A, the inquirer I, or any proper subset of the system users $\{u_i\}_{i=1}^{n}$ because the insider attackers are much more powerful than those outside with much less useful information. This paper concentrates on the semi-honest adversary, which are assumed to follow their prescribed actions. We also assume all the users only provide one correct response in each time period.

Roughly speaking, the security requirement ensures that the attackers gain no extra useful information except what is allowed by the protocol. In other words, I can only obtain the target information from the returned answer and nothing else. I cannot learn any useful information from a proper subset of the users' returned data. Any user without the capacity of I cannot learn anything about the target information, even when several users form a coalition against the other users. If I colludes with a subset of the users, then I can learn the target information of the remaining users. However, I cannot learn any additional information about the honest users' data. This is a reasonable security requirement because when some users collude with the inquirer, these users should be considered as part of the inquirer

I, and hence it is reasonable to assume the inquirer should obtain the target information of the remaining users. The other way to justify this security requirement is somehow target information dependent. For the intersection information targeting case, all the colluding users can simply select all the choices in the universe U, then apparently I can obtain the intersection information S of the remaining honest users because I should be able to get the target information $S \cap U$=S. For the union information targeting case, I can get the union information of all the users, and the union information of the honest users set can be obtained by subtracting the union information of the colluding users from the whole union information. From the above discussion we can see that the target information of the honest users is implied by the information provided by the colluding users and gathered by I, and hence is inevitably leaked when the collusion occurs.

So far we have not mentioned anything about the attack launched by the key authority A. It seems that a corrupted key authority is unstoppable since all the encrypted data from each user can be recovered by I if it colludes with A. However, we can always argue that in reality the key authority should be a highly trusted third party (just as the key authority in the identity-based encryption (IBE) mechanism [10]) independent of any online survey provided by the inquirers. Besides, we also provide a scheme that can remove A from the system. In this extended system, the users and the inquirer can collaboratively distribute secret keys to themselves without relying on a trusted third party.

Another interesting problem is that the basic private information extractors can only allow I to successfully get the target information when all the n users reply to the inquiry. This would render the basic schemes suitable only for the static users. However, it is always possible that some users in the system choose not to reply or simply ignore the inquiry. Therefore, this paper provides a scheme to guarantee that even if only a subset of users responds to the inquiry, the target information of these responders can still be extracted by I.

III. Related Work

Multi party computation and *Private set intersection*: Multi party computation enables n users with inputs $(x_1, \cdots, x_n)$ to privately compute $(f(x_1), \cdots, f(x_n))$. Although it seems possible to design a private information extractor protocol based on multi party computation protocol, most of the existing multi party computation protocols require each participant to interact with each other, which would be too costly both in computation and communication. Private set intersection (PSI) scheme is a special multi party protocol. The concept is first proposed by Freedman, Nissim and Pinkas [11]. It is a two-party protocol between a client C (Initiator) with an input set $X = \{x_1, \cdots, x_c\}$ and a server S with an input set $Y = \{y_1, \cdots, y_s\}$. At the end of the protocol operation, C learns the intersection of X and Y (i.e., $X \cap Y$) while S learns nothing. It could be generalized into multi-party case where each engaging party learns the intersection information of their inputs. Later, in the protocol proposed by Kissner and Song [12], the participants of PSI protocol can be enabled to obtain function evaluations on either the intersection or the union of the input sets. Although it appears the PSI protocol is closely related to intersection information extractor protocol, it cannot be directly applied to our proposed scenario due to the following reasons: first, there is a conceptual difference between PSI protocol and information extractor scheme. The PSI protocol is designed to enable each participant of the protocol to obtain the target information while our proposed scheme aims to provide the target information to the inquirer only. Second, most existing PSI protocols require each participant to interact with each other, which is considered highly inefficient in our scenario. Third, there is no trivial way to modify the existing PSI protocols to satisfy the requirement for our scenario. Take the most closely related PSI protocol by Kissner and Song as an example, one cannot simply add an inquirer with the group decryption key to obtain the target intersection or union information. The reason is that it seems impossible to prevent those with the decryption key from using it to decrypt the intermediate encrypted information exchanged between system users, which would imply that the inquirer may obtain far more extra information beyond the defined target information.

Secure online voting and *Secure data aggregation*: The proposed union information extractor scheme shares some similarities to secure online voting protocol [13], which considers the following scenario: there are L candidates in the protocol, and each user in the system votes for one. The candidate who receives the most votes wins. An important aspect of privacy requirement in a secure online voting protocol is the unlinkability between individual vote and its voter. The current solutions for this requirement can be divided into two categories: the first is based on a untraceable anonymous channel such as mix net or dining cryptographers network [14]. The underlying idea of the mix net is that the votes are received, transformed after either encryption or decryption, permuted and parallely transferred on each mix node of the mix net. However, the anonymous channel based solution suffers from inefficiency [14] due to massive encryption or decryption operations for a system with a huge number of users. Plus, the robustness is also a problem since a corrupted mix node may try to modify its input or a faulty node might simply fail to perform its operation. It can further cause disruption which would require the repetition of the entire mixing, resulting in a degradation of efficiency. The second category is based on homomorphic encryption, where each user encrypts his choice under the vote aggregator's public key using homomorphic encryption scheme. However, the aggregator not only can decrypt the union information, but also can decrypt the individual choice. Most schemes falling into this category usually address the privacy issue by simply distributing trust over multiple aggregators. Threshold cryptography is adopted [15], [16] to guarantee that only when a majority of the aggregators cooperates with each other can decrypt the encrypted individual vote. However, this solution leads to a huge communication overhead during the decryption process. Besides, most of these systems fail to protect individual privacy whenever those aggregators collude with each other to decrypt an individual encrypted vote. The disadvantage of the current homomorphic encryption based solution is due to the fact that the secret keys are generated by the aggregators themselves, who are also responsible for extracting the target union information. On the other hand, our design divides the responsibility of distributing secret keys and extracting union information among the key authority and the inquirer respectively. The design guarantees that no information except the target union information can be obtained by the inquirer when a third independent and trusted authority serves as the key authority. Although there is still a chance for the collusion between the trusted key authority and the inquirer, extra mechanism can be brought in to reduce the trust on the key authority or even remove the key authority. Reducing trust on the key authority [17], [18] in some closely related encryption mechanism such as IBE system has been a well investigated topic, which leaves us plenty of tools to resolve this issue.

There has been a large body of works related to secure data aggregation, especially in the area of sensor networks [19], [20], [21] or private data mining [22], [23], [24]. However, most of those solutions either treat the data aggregator as a trusted authority or can only securely extract some very simple statistics such as summation or product of individual messages. Shi *et al.* [21] recently proposed a privacy preserving data aggregation scheme for time series data, which focuses on providing the data aggregator with the summation or product statistics of individual message. The security requirement guarantees that the data aggregator can only extract the summation or product information and nothing else on individual data. Although it seems this scheme can be adopted to provide a union information extractor, the number of invocation of the underlying scheme would be linearly dependent on m, which is the number of choices in U. This implies a significant amount of communication and computation overhead for both the data aggregator and individual users. On the other hand, our proposed union information extractor only requires constant computation and communication overhead for each user, which significantly improves the efficiency. Besides, we design the first scheme in supporting intersection information extraction while there is no trivial approach to enable their scheme to extract such a more complicated but powerful statistical information. Besides, their scheme also does not support dynamic user population while our improved scheme does. Indeed, designing a scheme which supports richer statistics and dynamic user population are open problems posed in [21]. Our proposed scheme can be viewed as a positive answer to those open problems. Recently, Chow *et al.* [24] also proposed a protocol aiming to provide a querier with the intersection or union information of different database. Their scheme relies on the help of a trusted randomizer and a computing machine. However, since all the databases use the same randomness, it would be easy for the computing machine (corresponding to inquirer in our system) to determine the content of each database (corresponding to user) if the randomness is known to the computing machine. In other words, their scheme cannot resist the collusion attack between even one user and the inquirer while our basic scheme can resist the collusion between the inquirer and any $n-1$ users. Besides, there are at least two communication rounds between each user and the other authorities to complete a single protocol instance while our basic schemes only require one communication round. Since we are dealing with an online survey protocol which might involve a large population, the heavy communication round might render their scheme less desirable.

IV. Intersection Information Extractor

Design overview: In the scheme proposed by Shi *et al.* [21], each user u_i periodically submits data x_i to an aggregator. At the end of each period t, the aggregator obtains no other information except the summation of these data, i.e., $\sum_{i=1}^{n} x_i$. In order to guarantee the individual data privacy, at the system initialization, each user u_i and the aggregator are assigned by a trusted authority with a random secret key s_i or s_0, respectively, with the condition that $\sum_{i=0}^{n} s_i = 0$. Each user u_i uses the secret key s_i to encrypt his data x_i to generate individual ciphertext $g^{x_i} H(t)^{s_i}$, where $H(t)$ is an evaluation of a public hash function $H(\cdot)$ on t. When the aggregator gathers all the individual ciphertext, he then computes $H(t)^{s_0} \prod_{i=1}^{n} g^{x_i} H(t)^{s_i}$=$\prod_{i=1}^{n} g^{x_i} H(t)^{\sum\limits_{i=0}^{n} s_i}$=$g^{\sum_{i=1}^{n} x_i}$ and obtains $\sum_{i=1}^{n} x_i$ by brute force searching over all the possible values of $\sum_{i=1}^{n} x_i$ assuming that the message space is small. The aggregator cannot obtain any useful information on the individual message of the honest users even if he colludes with some users assuming that H is modeled as a random oracle and DDH assumption holds.

In 2005, Kissner and Song [12] proposed a polynomial representation technique for private set intersection. Given a multiset $S_1 = \{a_j\}_{1 \le j \le k}$ (a multiset is a set in which an element can appear multiple times), S_1 can be represented as a polynomial $f_1(x) = \prod\limits_{1 \le j \le k} (x - a_j)$ in a polynomial ring $R(x)$ consisting of all polynomials with coefficients from the ring R. In here, a appears in the multiset b times iff $(x-a)^b | f_1(x) \wedge (x-a)^{b+1} \nmid f_1(x)$. The intersection of two multisets $S_1 \cap S_2$ is defined as the multiset in which each element a that appears in S_1 $k_1 > 0$ times and in S_2 $k_2 > 0$ times, respectively, will appear in the resulting multiset k=min$\{k_1, k_2\}$ times. We refer this case as that a has a k intersection rank. Let S_1 and S_2 be two multisets of equal size, and f_1 and f_2 be their polynomial representations, respectively. The polynomial representation of $S_1 \cap S_2$ as: $f_1 g_1 + f_2 g_2$ where $g_1, g_2 \leftarrow R^{\ge \deg(f_1)}[x]$, where $R^{\ge \deg(f_1)}[x]$ is the set of random polynomials with degree no lower than the degree of f_1. It is proven in Theorem 2 of [12]: Suppose each player P_i inputs a polynomial f_i representing P_i's multiset S_i, then the mere information the third party can extract from the polynomial $\sum_{i=1}^{n} f_i g_i$ is the intersection information $S_1 \cap \cdots \cap S_n$, where g_i is a random polynomial with degree $\ge \max_i \deg(f_i)$.

In our scheme on extracting intersection information, each user u_i selects a multiple choice set $S_i = \{a_j\}$ from the universe U and also determines the rank k_j of each choice $a_j \in S_i$. Then this individual choice set is represented as a polynomial $f_i = \prod\limits_{a_j \in S_i} (x - a_j)^{k_j}$. Then each polynomial will be multiplied by a random polynomial g_i with a degree greater than deg(f_i) to generate a randomized polynomial $f_i g_i$. User u_i will encrypt the polynomial $f_i g_i$ in a similar way as x_i is encrypted in the scheme proposed by Shi *et al.*

The security of the distributed aggregation technique in [21] can ensure that the inquirer obtains only the exponential summation of all the randomized polynomial $g^{F(x)}$=$g^{\sum\limits_{i=1}^{n} f_i g_i}$ at most. According to the security of polynomial representation technique [12] for set intersection, the summation $F(x)$ only contains the intersection information of the individual choice. The intersection information polynomial representation $F(x)$ is equal to $gcd(f_1, f_2, \cdots, f_n) * u(x)$, where gcd is the greatest common divisor and $u(x)$ is a uniformly distributed polynomial [12], and hence the message space for $g^{F(x)}$ is too large for the brute force searching method, which is how Shi *et al.* computes $\sum_{i=1}^{n} x_i$ given $g^{\sum_{i=1}^{n} x_i}$. This is one of the major technical challenges, but we will show how the inquirer can still be able to effectively extract the intersection information in the our proposed scheme.

Intersection information extractor

Our proposed extractor consists of three algorithms: `Setup`, `Individual data generation` and `Information extraction`. The key authority A generates and publishes public parameters param and distributes the secret keys to the inquirer I and all the users in the Setup algorithm, which takes the security parameter λ as input. The choice universe U is also posted by the inquirer I in the Setup algorithm, but it can also be posted at the beginning of each period since U corresponds to various online surveys posted in the respective period. Then in each period each user u_i determines his choice set $S_i = \{a_j^{(k_j)}\} \subseteq U$ and encrypts S_i to generate his individual ciphertext $C_i = \{C_{ij}\}$ by running the Individual data generation algorithm, which takes system parameter param, user secret key SK_i, time period t, and S_i as input. All the ciphertexts for the n users will be delivered to I who runs Information extraction algorithm, which takes param, I's secret key SK_0, time period t, and the collected ciphertext $\{C_i,\ i \in [1, n]\}$ as input, to extract the respective target information.

Let $\mathbb{G}$ denote a cyclic group of prime order p in which Decisional Diffie-Hellman is hard. Let $H : Z \to \mathbb{G}$ denote a public hash function.

Setup(1^λ): In our scheme, each user is allowed to make choices from an universe $U = \{a_i\}_{i=1}^m$ periodically, where C denotes the maximum rank for each $a_j \in U$. The inquirer I defines and publishes the choice universe U. The key authority A chooses a random generator $g \in \mathbb{G}$, and $(n + 1) \times (2mC + 1)$ random secrets $s_{0j}, \cdots, s_{nj}, j \in [0, 2mC]$ such that for $\forall j, \sum\limits_{i=0}^{n} s_{ij} = 0$. A also needs to choose a monic irreducible polynomial $f(x) = \sum\limits_{i=0}^{2mC+1} d_i x^i$ of degree $2mC$+1 over the underlying field $\mathbb{F}_p$ ([25]).

The public parameters are param=$\{g,\ f, U\}$. The inquirer I gets secret key $\{s_{0j}\}_{j=0}^{2mC}$ while each user u_i gets his secret key $\mathsf{SK}_i = \{s_{ij}\}_{j=0}^{2mC}$.

Individual data generation(param, SK_i, t, $S_i = \{a_j^{(k_j)}\}$): For user u_i who chooses $S_i = \{a_j^{(k_j)}, j \in [1, m]\}$ (where $k_j \leq C$ is the rank of a_j), user u_i first computes the polynomial $f_i = \prod\limits_{a_j \in S_i} (x - a_j)^{k_j}$. Then u_i chooses a random mC-degree polynomial g_i over field $\mathbb{F}_p$ and multiplies two polynomials $f_i g_i \bmod f = \sum\limits_{j=0}^{2mC} c_{ij} x^j$. The degree of $f_i g_i$ is always smaller than deg(f)=$2mC + 1$, but the user can simply let the coefficient c_{ij} for the missing terms be 0. We also have $f_i g_i$ mod f=$f_i g_i$ due to the fact that deg($f_i g_i$) <deg(f) always holds.

Then for each coefficient $c_{ij}, j \in [0, 2mC]$, u_i computes the following ciphertext $C_{ij} = g^{c_{ij}} H(t)^{s_{ij}}$. At the end of each time period t, the individual ciphertext $C_i = \{C_{ij},\ j \in [0, 2mC]\}$ will be submitted to the inquirer I.

Information extraction(param, SK_0, t, $\{C_i, i \in [1, n]\}$): At the end of each period, the inquirer I collects all the individual ciphertext $C_i = \{C_{ij} = g^{c_{ij}} H(t)^{s_{ij}}, j \in [0, 2mC]\}$. For each $j \in [0, 2mC]$, it is easy to compute $H(t)^{s_{0j}} \prod\limits_{i=1}^{n} C_{ij} = g^{\sum\limits_{i=1}^{n} c_{ij}}$. The target polynomial is $F(x) = \sum\limits_{i=1}^{n} f_i g_i$= $\sum\limits_{j=0}^{2mC} \sum\limits_{i=1}^{n} c_{ij} x^j = \sum\limits_{j=0}^{2mC} e_j x^j$, instead now I has the exponents of all the coefficients, i.e., g^{e_j}.

As mentioned before, $F(x)$ corresponds to the target intersection information. If everyone selects a choice $a \in U$ at least k times, we would have $(x - a)^k | F(x)$. In order to find out which element $a \in U$ is the intersection choice, I only needs to check whether $(x - a)|F(x)$ holds as follows: for each $a \in U$, given $g^{e_j}, j \in [0, 2mC]$, I check whether $g^{F(a)} = \prod\limits_{j=0}^{2mC} (g^{e_j})^{a^j}$= 1 holds. Indeed, $g^{F(a)} = g^0 = 1$ holds whenever $(x - a)|F(x)$ holds.

Assume choice a's intersection rank is k and thus $(x - a)^k | F(x)$ holds, then let $F(x) = G(x)(x - a)^k$ where $(x - a) \nmid G(x)$. Because deg($f_i g_i$) <deg(f), we have deg($F(x) = \sum\limits_{i=1}^{n} f_i g_i$) <deg($f$), and thus deg($F(x)/(x - a)^j$) <deg($f$) for $0 < j$. Therefore, $F(x)(x - a)^{-j} \bmod f$=$F(x)/(x - a)^j$ holds for any $0 < j \leq k$. Since $(x - a)|G(x)(x - a)^{k-j}$=$(x - a)|F(x)/(x - a)^j$ holds for any $0 < j < k$ and $(x - a) \nmid G(x)$=$(x - a) \nmid F(x)/(x - a)^k$ holds, we can conclude that $F(a)(a-a)^{-j} \bmod f$=$F(a)/(a-a)^j$=0 holds for all $0 < j < k$, and $F(a)(a-a)^{-k} \bmod f$=$F(a)/(a-a)^k \neq$0 holds. Therefore, in order to calculate choice a's intersection rank k, I can iteratively compute $g^{F(x)(x-a)^{-j} \bmod f}$ for $j = 1, 2, \cdots$ until he finds[1] the k satisfying $g^{F(a)(a-a)^{-k} \bmod f} \neq 1$. Now, I needs to figure out how to compute $g^{(F(x)(x-a)^{-j}) \bmod f}$ by running the following steps:

(1). Compute $(x - a)^{-1} \bmod f$, which can be effectively computed by running the extended Euclidean algorithm [25]. Assume $(x - a)^{-1} \bmod f = \sum_{i=0}^{l} c'_i x^i$

(2). The inquirer I computes $g^{(F(x)(x-a)^{-1}) \bmod f}$ given $g^{F(x)}$ and $(x - a)^{-1} \bmod f$ in this step. Since both $g^{F(x)}$ and $(x - a)^{-1} \bmod f$ are known to A, then it is trivial to compute $g^{F(x)((x-a)^{-1} \bmod f)}$=$\left(g^{\sum\limits_{j=0}^{2mC} e_j x^j} \right)^{\sum_{i=0}^{l} c'_i x^i}$=$g^{\sum\limits_{i=0}^{l+2mC} b_i x^i}$. It is easy to see that $(F(x)(x-a)^{-1}) \bmod f$=$F(x)((x-a)^{-1} \bmod f)) \bmod f$ holds. Now the problem is reduced to compute $g^{F(x)((x-a)^{-1} \bmod f)) \bmod f}$ given $g^{F(x)((x-a)^{-1} \bmod f)}$ and f. In other words, A needs to compute $g^{\sum\limits_{i=0}^{l+2mC} b_i x^i \bmod f}$ given $g^{\sum\limits_{i=0}^{l+2mC} b_i x^i}$ and f. A needs to determine the remainder polynomial $g^{r(x)}$=$g^{\sum\limits_{i=0}^{l+2mC} b_i x^i \bmod f(x)}$ such that $g^{f(x)q(x)+r(x)}$ =$g^{\sum\limits_{i=0}^{l+2mC} b_i x^i}$ for $r(x)$ with deg($r(x)$) <deg($f(x)$).

I first lets the quotient polynomial as $g^{q'(x)}$= $g^{b_{l+2mC} x^{l+2mC-2mC-1}}$ where $g^{b_{l+2mC}}$ is the leading coefficient of $g^{\sum\limits_{i=0}^{l+2mC} b_i x^i}$, then computes the remainder polynomial as:

[1] Kissner and Song calculate the rank k of a by simply testing whether both $(x - a)^k | F(x)$ and $(x - a)^{k+1} \nmid F(x)$ hold. We cannot directly accomplish this test because it is difficult to run a division test on the exponential polynomial $g^{F(x)}$, which is why we introduce the irreducible polynomial $f(x)$ such that we can compute the inverse $(x - a)^{-k} \bmod f(x)$ to further compute $g^{(F(x)(x-a)^{-k}) \bmod f(x)}$=$g^{(F(x)/(x-a)^k)}$.

$$
\begin{aligned}
& g^{r_1(x)} \\
&= g^{\sum\limits_{i=0}^{l+2mC} b_i x^i - q'(x)f(x)} \\
&= \frac{g^{\sum_{i=0}^{l+2mC} b_i x^i}}{g^{b_{l+2mC} x^{l+2mC-2mC-1} f(x)}} \\
&= \frac{g^{\sum_{i=0}^{l+2mC} b_i x^i}}{g^{\sum_{i=0}^{2mC+1} d_i x^i b_{l+2mC} x^{l-1}}} \\
&= g^{\sum_{i=0}^{l+2mC-1} u_i x^i}.
\end{aligned}
$$

We note that the degree of $r_1(x)$ is $l+2mC-1$ because $f(x)$ is monic and thus the leading coefficient of the denominator is $d_{2mC+1} \times b_{l+2mC} = b_{l+2mC}$, which is equal to that of the numerator.

Then I further lets the quotient polynomial as $g^{q''(x)}$=$g^{u_{l+2mC-1} x^{l+2mC-2mC-2}}$=$g^{u_{l+2mC-1} x^{l-2}}$ where $g^{u_{l+2mC-1}}$ is the leading coefficient of $g^{r_1(x)}$, and computes the remainder polynomial similarly to $g^{r_2(x)}$=$g^{r_1(x)-f(x)q''(x)}$=$\frac{g^{\sum_{i=0}^{l+2mC-1} u_i x^i}}{g^{\sum_{i=0}^{2mC+1} d_i x^i u_{l+2mC-1} x^{l-2}}}$=$g^{\sum_{i=0}^{l+2mC-2} u'_i x^i}$ Now we have a remainder polynomial $r_2(x)$ of degree $l+2mC-2$, which is reduced by 1 compared with previous remainder polynomial $r_1(x)$. By running similar steps at least l times until $g^{r(x)}$ of degree deg$(r(x))$ <deg$(f(x))$ is found, then we have $g^{r(x)}$=$g^{(F(x)(x-a)^{-1} \bmod f(x))}$.

(3). Given $g^{(F(x)(x-a)^{-1} \bmod f)}$ and $(x-a)^{-1} \bmod f$, I can compute $g^{(F(x)(x-a)^{-2} \bmod f)}$ by running the second step. All I has to do is to treat $(F(x)(x-a)^{-1} \bmod f)$ as a new $F(x)$ in the second step. I can iteratively compute in this way and finally obtain $g^{(F(x)(x-a)^{-j} \bmod f)}$.

Security analysis: Intuitively, the inquirer I cannot obtain any useful information on the individual choice because it is randomized by the individual secret key. For any proper subset $S \subset [1,n]$, the respective data $g^{\sum\limits_{i\in S} c_{ij}}$ would be randomized by $H(t)^{\sum\limits_{i\in S} s_{ij}}$, where $\sum\limits_{i\in S} s_{ij} \neq 0$ is unknown to the aggregator and thus $H(t)^{\sum\limits_{i\in S} s_{ij}}$ is uniformly distributed in the respective group if H is modeled as a random oracle and DDH assumption holds. Suppose I colludes with a subset S' of users, then since both $g^{F(x)}$ and $g^{F'(x)}$ ($F'(x)$ corresponds to the integrated polynomial for the subset S') are known to I, I can find the intersection information on the rest of users by running the `Information extraction` algorithm on $g^{F(x)-F'(x)}$. However, no extra information on the choices of the honest user set S is leaked to I or the corrupted users due to the above mentioned reason.

Indeed, the security of the proposed scheme is summarized in the following theorem with the outline of the proof, and the detailed proof can be found in a more detailed version [26].

Theorem 1: Assuming that the Decisional Diffie-Hellman assumption holds in the group $\mathbb{G}$, and that the hash function H is a random oracle and the underlying polynomial representation securely represents the intersection information, then intersection information extractor is secure.

Outline of the proof is given as follows: the proposed scheme can be viewed as running $2mC+1$ independent instances of the distributed aggregation system [21] simultaneously because for each dimension $j \in [0, 2mC]$, independent secret key set $s_{0j}, ..., s_{nj}$ is used. According to Theorem 1 in [21], the inquirer (corresponding to aggregator in [21]) should only gain the information on the integrated polynomial $F(x)$ at most. According to Theorem 2 in [12], $F(x)$ leaks no more information than the intersection information, hence conclude our proof.

Performance analysis: The proposed scheme is fairly efficient. The major time consuming operation is the modular exponentiation computation in the Diffie-Hellman group. Each user is required to complete at most $(2mC+1)*2$ modular exponentiation operations in the `Individual data generation` algorithm. For the inquirer I, in order to find the intersection choice a, he at most needs to complete $(m+1)(2mC+1)$ modular exponentiation operations. To further calculate a's rank k, he at most needs to complete $2C(2mC+1)^2$ modular exponentiation operations. Using high speed elliptic curve such as "curve 25519", it takes roughly 0.3ms to compute a modular exponentiation over a classic Diffie-Hellman group modular a 1024-bit prime on a modern 64-bit computer [27], [21]. Assuming an inquirer publishes an online questionnaire consisting of 60 five-choice question and one question on user's age. Then this questionnaire can be viewed as a multiple choice question with a choice universe U of $m = 60*5+100 = 400$ choices assuming the user ages range from 1 to 100. Let the maximum rank C=4. Then each individual user will spend roughly $(2mC+1)*2*0.3$=$(2*400*4+1)*2*0.3$ms, which is roughly 2s to generate its individual ciphertext. It takes roughly $(m+1)(2mC+1)*0.3$=385s for the inquirer to find the intersection choice a. I needs to spend roughly $2C(2mC+1)^2*0.3$=24591s≈6.83 hours to further calculate the rank k assuming $k = C$. From this example, it is fair to say that the computation cost of our proposed scheme is acceptable. It is somehow counterintuitive that the computation cost is not dependent on n, but the maximum rank C. We also note that the maximum rank C is closely related to how the questionnaire is designed. Indeed, C can be set to 1 in most cases since most of the related information of a choice can be expressed as extra choices by simply increasing m rather than C. However, C is especially useful when all choices have a similar maximum rank. The communication complexity of the proposed scheme is also low. Each individual ciphertext consists of $2mC+1$ group elements, and hence at the end of each period, the inquirer will obtain $(2mC+1)n$ group elements from all the n users.

V. Union Information Extractor

In this section, we consider a system where each user is handed with an online survey of m choices. Therefore, the choice universe is $U = \{a_j\}_{j=1}^m$, and each user can only make one unique choice from U, and we will discuss how to extend the basic scheme to accommodate a multiple choice questionnaire later. The union information $\{c_j\}_{j=1}^m$ is the collection of the number of times c_j for each choice $a_j \in U$ chosen. A trivial union information extractor can be constructed from the basic scheme by Shi *et al.* [21]. At the beginning, A distributes m secret keys $\{s_{ij}, i \in [0,n], j \in [1,m]\}$ satisfying $\sum_{i=0}^n s_{ij} = 0$ for I and each user $u_i, i \in [1,n]$, respectively. Each user can encrypt m single bits to indicate his choice for each choice $a_j \in U$, 1 in dimension j indicating positive choice for a_j, 0 for the negatives. Then I can simply calculate the count of 1 in each dimension j, which will be the exact count information c_j.

This trivial scheme can be viewed as running m independent instances of the system by Shi *et al.* simultaneously. It seems there is no trivial approach to further improve the efficiency. Each user has to encrypt all the single bit answer for each choice in U no matter whether it is 1 or 0. Otherwise, it will trivially leak the user's individual choice. The user has to use distinct secret keys to generate ciphertext for different dimensions. Otherwise, suppose a user uses an identical key s for two different dimensions j and k, then the attacker might be able to find out the bit answer of the respective dimension as follows: if the two choices are different, then with the ciphertext $C_{ij} = g^0 H(t)^{s_{ij}} = H(t)^s$ and $C_{ik} = g^1 H(t)^{s_{ik}} = gH(t)^s$, the attacker can compute $C_j/C_k = g^{-1}$ and $C_k/C_j = g$ and thus know the respective choices. In this trivial scheme, both the computation and communication overhead of each user have to be linearly dependent on m. In the following section, we will employ the prime representation technique to develop an union information extraction scheme where both the computation and communication overhead are constant.

Design overview: The union of two multisets $S_1 \cup S_2$ is defined as the multiset where each element a that appears in S_1 $k_1 > 0$ times and S_2 $k_2 > 0$ times appears in the resulting multiset $S_1 \cup S_2$ $k_1 + k_2$ times in [12]. Assuming the choice set S_i of user u_i is represented by a polynomial f_i, then Theorem 1 in [12] has shown that the inquirer cannot learn more information about $\{S_i\}_{i=1}^{n}$ from the polynomial multiplication $\prod\limits_{i=1}^{n} f_i$ except the union set $\bigcup\limits_{i=1}^{n} S_i$, which basically corresponds to our definition of union information. However, it is difficult to simply adopt polynomial representation method because it is hard to find an effective way to randomize the individual polynomial representation f_i. Suppose each user u_i chooses a_i, and the corresponding polynomial representation is $f_i = x - a_i$. There are two possible approaches to randomizing these polynomials: by adding or multiplying a random polynomial $g_i = c_i x + d_i$ to f_i. First, it is infeasible to employ the addition randomization technique because it is impossible to recover the target multiplication polynomial $\prod_{i=1}^{n} f_i$=$\prod_{i=1}^{n}(x - a_i)$ from $\prod_{i=1}^{n}(f_i+g_i)$=$\prod_{i=1}^{n}(x-a_i+c_i x+d_i)$. Neither the multiplication randomization is an option since the attacker can still know the choice a_i from the randomized polynomial $(x - a_i)(c_i x + d_i)$.

We observe that there is a correspondence between monic irreducible polynomial in an extension field and a positive prime in a finite field. Most theorems on these two objects can be mutually transferable [25]. In fact, it is trivial to prove the following theorem, which is a prime representation based counterpart of Theorem 1 in [12]:

Theorem 2: For each $a_j \in U = \{a_j\}_{j=1}^{m}$, choose a unique prime p_j to represent a_j. Let TTP1 be a trusted third party which receives the private input multiset S_i from user u_i for $i \in [1, n]$, and then returns to the inquirer the union set directly. Let TTP2 be another trusted third party, which receives the private input multiset S_i from user u_i for $i \in [1, n]$, and then: (1) calculates the prime representation $\prod\limits_{a_j \in S_i} p_j$ for each S_i; (2) computes and returns to the inquirer $\prod\limits_{i=1}^{n} \prod\limits_{a_j \in S_i} p_j$. There exists a probabilistic polynomial time translation algorithm such that, to each player, the results of the following two scenarios are statistically identical: (1) applying translation to the output of TTP1; (2) returning the output of TTP2 directly.

Proof: The above theorem basically states that given a prime product representation $\prod\limits_{i=1}^{n} \prod\limits_{a_j \in S_i} p_j$, it is impossible for the attacker to deduce any other useful information on individual prime representation set $\left\{ \prod\limits_{a_j \in S_i} p_j \right\}_{i=1}^{n}$ except the union information. It is observed that any combination of n factors of $\prod\limits_{i=1}^{n} \prod\limits_{a_j \in S_i} p_j$ is a legitimate prime representation set, and thus any combination of n individual multisets which can constitute the given union set is equally likely. In other words, the attacker gains no extra information other than the union information. ■

Since the goal of prime representation based union information extraction is to deliver $\prod\limits_{i=1}^{n} \prod\limits_{a_j \in S_i} p_j$ to the inquirer without leaking any additional information on individual representation $\prod\limits_{a_j \in S_i} p_j$, then a secure distributed aggregation technique protecting the privacy of the product statistic is sufficient to serve as a underlying tool. Shi *et al.* [21] claimed that their scheme can be directly applied to protect the privacy of product statistic. Therefore, a union information extractor can be provided directly from Shi *et al.*'s scheme as follows.

Union information extractor

Let $\mathbb{G}$ denote a cyclic group of prime order p in which Decisional Diffie-Hellman is hard. For the simplicity of analysis, we let this cyclic group be $\mathbb{Z}_p$. Let $H : Z \to \mathbb{Z}_p$ denote a public hash function.

`Setup`(1^λ): The key authority A chooses a random generator $g \in \mathbb{G}$, and $n + 1$ random secrets $s_0, \cdots, s_n$ such that for $\sum\limits_{i=0}^{n} s_i = 0$. Each user can make a single choice from the universe U=$\{a_j\}_{j=1}^{m}$, which is published by the inquirer I. For each choice $a_j \in U$, A chooses a unique prime P_j for representation. The optimal way to is to let the first m primes $p_1 = 2, p_2 = 3, p_3 = 5, \cdots$ to be the representation of the first m choices, the reason is given later. The public parameters are param=$g, \{p_i\}_{i=1}^{m}, U$. The inquirer I gets secret key $\mathsf{SK}_0 = s_0$ while the participant u_i gets its secret key $\mathsf{SK}_i = s_i$.

`Individual data generation`(param, SK_i, t, a_j): For user u_i to encrypt his unique choice a_j, he computes the following ciphertext $C_i = p_j^{(i)} H(t)^{s_i}$=$p_j^{(i)} H(t)^{s_i}$, where $p_j^{(i)}$ denotes the prime representation p_j picked by u_i. We note the individual message $p_j^{(i)}$ is an integer and the respective ciphertext C_i is a group element. We will show how to extract the product of those integer primes from the collection of those individual ciphertext in the next algorithm.

`Information extraction` (param, SK_0, t, $\{C_i\}_{i=1}^{n}$): Compute $H(t)^{s_0} \prod\limits_{i=1}^{n} C_i$ =$\prod\limits_{i=1}^{n} p_j^{(i)} \bmod p$. The union information $\{c_j\}_{j=1}^{m}$ can be computed by deciding the prime expression of an integer $\prod\limits_{i=1}^{n} p_j^{(i)} = \prod_{i=1}^{m} p_i^{c_i}$=$\prod\limits_{i=1}^{n} p_j^{(i)} \bmod p$ when $[p_1^n, p_m^n] \subseteq [0, p-1]$. The reason is shown as follows: we first note p_1^n, p_m^n and $p - 1$ are just plain integers rather than group elements here. Since we choose the first m primes to represent the first m choices, respectively, We can guarantee $p_i < p_{i+1}, i \in [1, m-1]$ and $p_m^n \leq p - 1$ for as large n as possible. Therefore, we have $p_1^n \leq \prod\limits_{i=1}^{n} p_j^{(i)} \leq p_m^n \leq p - 1$, which implies that the group elements $\prod\limits_{i=1}^{n} p_j^{(i)} \bmod p$=$\prod\limits_{i=1}^{n} p_j^{(i)}$.

A more efficient approach to determining the prime representation of $\prod\limits_{i=1}^{n} p_j^{(i)}$ is to use the trial-and-error approach:

Firstly, for each prime p_i, compute $p_i^{\frac{n}{2}}$ since n is the maximum chosen times for a_i. Check whether $p_i^{\frac{n}{2}} \mid \prod_{i=1}^{n} p_j^{(i)}$ or not. If it does, then try $p_i^{\frac{3n}{4}}$ and continue the similar procedure as mentioned in the above, otherwise try $p_i^{\frac{n}{4}}$. Run the similar process iteratively until c_i is found.

Security analysis: Intuitively, the individual prime representation $p_j^{(i)}$ of user u_i is randomized by its secret key $H(t)^{s_i}$ and hence is unknown to I even I colludes with the other users according to Theorem 1 in [21]. Thus, the only information I can obtain is $\prod_{i=1}^{n} p_j^{(i)} \bmod p$, which is the union information. The security can be formally stated in the following theorem, which is basically a corollary of Theorem 1 in [21] and Theorem 2 given before. The concrete proof is omitted here due to space limit.

Theorem 3: Assuming that the Decisional Diffie-Hellman assumption holds in the group $\mathbb{G}$, and that the hash function H is a random oracle and the underlying prime representation securely represents the union information, then our union information extractor is secure.

Correctness: The inquirer can determine the set of unique chosen times due to the uniqueness of the prime representation of integer $\prod_{i=1}^{n} p_j^{(i)}$.

Multiple choice case: The above scheme can be easily extended to accommodate a multiple choice survey. There is no significant difference between the multiple choice and single choice scheme in the `Setup` and `Individual data generation` algorithms except that u_i will choose multiple primes $\{p_j^{(i)}\}_{a_j \in S_i}$ to represent his choice set S_i and generate his individual ciphertext as $H(t)^{s_i} \prod_{a_j \in S_i} p_j^{(i)}$. For the `Information extraction` algorithm, the condition on successful extraction is modified as $\left[p_1^n, \prod_{i=1}^{m} p_i^n\right] \subseteq [0, p-1]$. I can compute the union information following the similar steps as done in the single choice scheme.

Parameter setting and efficiency comparison: We will first present the performance analysis on the single choice case, and the analysis on the multiple choice case can be accomplished in a similar way, which is omitted. Assuming a Diffie-Hellman group modular a 1024-bit prime is used and we consider a questionnaire consisting of 50 five-choice questions, all of which are single choice questions. Let $m = 5$ and run 50 union information extractors simultaneously, each of which corresponds to one question. In order for the inquirer to successfully extract the union information, the maximum system user number n should satisfy $p_m^n < p < 2^{1024}$. The system can accommodate almost n =300 users if $p_5 = 11$, which is good enough for most surveys. If a larger modular such as a 4096-bit prime is chosen, then the system can accommodate up to more than a thousand people. Indeed, one can always choose to use a larger prime to accommodate a larger n or m. In the above example, suppose we set $m = 10$ and run 25 extractors, each of which corresponds to 2 five-choice questions. The successful extraction condition is modified as $p_5^n p_{10}^n = 11^n \times 29^n < 2^{\lambda}$ (since each user only makes a single choice in the first and second five questions in a universe consisting of 10 questions, the maximum of the prime representation product is $p_5^n p_{10}^n$), where λ is the bit length of the modular prime. Assuming there are n =400 users involved in the survey, then we can set $\lambda = 4096$ because $p_5^n p_{10}^n$=$11^{400} \times 29^{400} < 2^{4096}$. We also note that the time cost of exponentiation operation, which is the most time consuming operation in our system, is mainly determined by the exponent rather than the modular, which implies a larger modular prime will have a negligible influence on time cost.

In the single choice scheme, each user only needs to complete two exponentiation operations and delivers a group element while he needs to complete m exponentiation and deliver m group elements in both the trivial solution assuming a unique security parameter is used in the trivial solution and our scheme. Our proposed scheme improved the individual computation and communication overhead also reduced the communication cost for the inquirer by m times. I in the trivial scheme needs to at least complete n modular exponentiation operations when the brute force search method is used since the counting number c_i satisfies $\sum_{i=1}^{m} c_i = n$. However in our system, I only needs to accomplish nearly $m \log n$ exponentiation operations (we note simple integer exponentiation operation is more efficient than modular exponentiation operation since the cost of modular computation is saved). Since n is equal to the number of system users, which is generally much larger than $m \log n$, our system also saves the computation overhead for the inquirer.

VI. Extensions

A. Dynamic extractor

Polynomial interpolation based secret sharing (SS) scheme was first proposed by Shamir [28]. There is a trusted dealer and n users in a SS scheme. The trusted dealer is responsible for generating the master key s and distributing a secret share s_i for s to individual user $u_i, i \in [1, n]$. For a (n, k)-threshold secret sharing scheme, the master key can be deduced from any k secret shares. Our proposed schemes can be considered as a $(n+1, n+1)$-threshold cryptosystem because only when all the $n+1$ users (the inquirer is treated as a user here) pool together their secret keys $H(t)^{s_i}, i \in [0, n]$ can $H(t)^{\sum_{i=0}^{n} s_i} = H(t)^0 = 1$ be calculated. This is the only way the inquirer can remove the randomization factor to extract the target information. The key authority in our system can be viewed as the trust dealer in the SS scheme and the master key is 0 since the $n+1$ secret keys satisfy $\sum_{i=0}^{n} s_i = 0$. In order to construct a dynamic information extractor scheme, the inquirer will define a tolerable number of responses r, which is the smallest number of users who respond to the online survey. In the following, we will describe how to design a dynamic intersection information extractor, and the dynamic union information extractor can be designed in a similar way.

A runs the $(n+1, k)$-threshold SS scheme and delivers each user with the secret share of 0 as their respective secret keys. A will choose $h_j(x), j \in [0, 2mC]$ of degree $k-1$ in F_p, which are polynomials with no constant term because the master secret is zero. Then A generates the secret keys for each user as $s_{ij} = h_j(i), i \in [0, n]; j \in [0, 2mC]$ which satisfy $h_j(0) = \sum_{u=1}^{k} h_j(i_u) \prod_{\substack{1 \le v \le k \\ u \ne v}} \frac{0 - i_v}{i_u - i_v}$=0, where $\prod_{\substack{1 \le v \le k \\ u \ne v}} \frac{0 - i_v}{i_u - i_v}$ is Lagrange coefficient.

The `Individual data generation` algorithm is modified as follows: in order to cancel the influence of the multiplication by Lagrange coefficient during the reconstruction step, each user has to pre-process his raw data by multiplying the inverse of its Lagrange coefficient. The ciphertext is formed as follow: $C_{ij} = g^{c_{ij}[\prod_{1\leq v\leq k, i\neq i_v}^{\frac{0-i_v}{i-i_v}}]^{-1}} H(t)^{s_{ij}}$. In the `Information extraction` algorithm, the inquirer can get the final result in the following way: in the first scheme, for those who agrees to respond, I computes[2]

$$\prod_{i=1}^{k} g^{c_{ij}[\prod_{1\leq v\leq k, i\neq v}^{\frac{0-v}{i-v}}]^{-1}[\prod_{1\leq v\leq k, i\neq v}^{\frac{0-v}{i-v}}]} H(t)^{s_{ij}[\prod_{1\leq v\leq k, i\neq v}^{\frac{0-v}{i-v}}]}$$

$$= \prod_{i=1}^{k} g^{c_{ij}} H(t)^{h_j(i)[\prod_{1\leq v\leq k, i\neq v}^{\frac{0-v}{i-v}}]} = g^{\sum_{i=1}^{k} c_{ij}}$$

One might wonder how user u_i gets his Lagrange coefficient because he does not know the indices of the other responders. One way to deal with this problem is to let I first collect and publish all the indices for the k users who are willing to respond to the survey beforehand. This will add another communication round to the proposed scheme, which is an additional cost due to the dynamic property. Besides, each user u_i should encrypt their individual ciphertext under I's public key such that only the inquirer can decrypt those ciphertexts. The security analysis of this extended system is similar to that of the basic system, which is omitted here.

B. Removing key authority A

In order to remove the key authority during the protocol operations, we have to incorporate an extra scheme as an underlying tool: joint zero secret sharing (JZSS) scheme, which is first proposed by Gennaro $et\ al.$ [18]. In the JZSS protocol, all the users cooperatively generate secret shares for a $(n+1, n+1)$-threshold secret sharing of zero without a trusted dealer. At the end, each user u_i will get a secret share s_i, which is the secret key in our proposed system (or simply run $2mC + 1$ JZSS protocol instances to get $2mC + 1$ secret keys in the intersection information extractor case). The scheme is distributed in the sense that each user acts as a trusted dealer and thus there is no room for a central trusted dealer. The basic idea of JZSS protocol is that each user u_i picks a random polynomial $f_i(x)$ of degree n over the finite field $\mathbb{F}_p$ with zero constant term. Then user u_i distributes $f_i(j)$ to other user $u_j, j \in [0, n] \setminus \{i\}$. Each user u_i collects all the polynomial values $\{f_j(i), j \in [0, n] \setminus \{i\}\}$ from the other users. Then he calculates his own secret key as $s_i = \sum_{j=0}^{n} f_j(i)$. The master polynomial is set as $\sum_{j=0}^{n} f_j(x)$ of degree n. Since each user gets an evaluation of this polynomial, then only when $n + 1$ users pool together their secret keys can they recover the master polynomial. The advantage of using JZSS protocol to replace the key authority is that the only way for the inquirer to obtain useful information on the secret keys of the honest users is to corrupt all the n users in the system, which is apparently impossible and meaningless. In other words, the trust on a single central key authority is distributed among all the individual users in the system.

VII. Conclusions

In this paper, we propose two schemes to enable the inquirer who conducts an online survey to extract mere intersection and union information from the submitted data of the participants. Our schemes ensure that the inquirer can only obtain these target information from the received data while preserving the individual privacy. The privacy guarantee of the proposed schemes can stimulate the users' participation over online social network in online surveys and hence enhance the social utility provided by such online surveys.

[2] Without loss of generality and for the ease of exposition, we assume the first k users form the response group

References

[1] bada.hb.se/bitstream/2320/5178/1/2009mf26.pdf. pages 1–88, 2009.

[2] Sharique Hasan, George T. Duncan, Daniel B. Neill, and Rema Padman. Automatic detection of omissions in medication lists. *JAMIA*, 18(4):449–458, 2011.

[3] Daniel B. Neill and Gregory F. Cooper. A multivariate bayesian scan statistic for early event detection and characterization. *Machine Learning*, 79(3):261–282, 2010.

[4] Martyn Denscombe. Web-based questionnaires and the mode effect. In *Social Science Computer Review*, pages 246–254, volume 24 Number 2, Summer, 2006.

[5] Mette T. J. Sijtsma Jerry J. Vaske, Maarten H. Jacobs and Jay Beaman. Canweighting compensate for sampling issues in internet surveys? In *Human Dimensions of Wildlife*, pages 16:200–215, 2011.

[6] Stephen R. Porter and Michael E. Whitcomb. Non-response in student surveys: The role of demographics, engagement and personality. In *Human Dimensions of Wildlife*, pages vol. 46, No.2 127–152, March 2005.

[7] Liao Li and Jinbao Zhang. Intervals design for economic variable's range and its application in the survey of household finance. In *International Conference on Management Science and Engineering (ICMSE)*, pages 1131–1136, 24-26 Nov. 2010.

[8] http://abcnews.go.com/health/coli-outbreak-sprouts-german-investigators/story?id=13812166.

[9] Cynthia Dwork. Differential privacy: A survey of results. In *TAMC*, pages 1–19, 2008.

[10] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[11] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In *EUROCRYPT*, pages 1–19, 2004.

[12] Lea Kissner and Dawn Xiaodong Song. Privacy-preserving set operations. In Shoup [29], pages 241–257.

[13] Diplomova praca. Electronic voting schemes. *MS. thesis.*

[14] Krishna Sampigethaya and Radha Poovendran. A framework and taxonomy for comparison of electronic voting schemes. *Computers & Security*, 25(2):137–153, 2006.

[15] Benaloh J. Verifiable secret-ballot elections. *Ph.D. thesis, Yale University.*

[16] Junji Nakazato, Kenji Fujimoto, and Hiroaki Kikuchi. Privacy preserving web-based questionnaire. In *AINA*, pages 285–288, 2005.

[17] Vipul Goyal. Reducing trust in the pkg in identity based cryptosystems. In *CRYPTO*, pages 430–447, 2007.

[18] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Robust threshold dss signatures. *Inf. Comput.*, 164(1):54–84, 2001.

[19] Marian Kamal Iskander, Adam J. Lee, and Daniel Mossé. Privacy and robustness for data aggregation in wireless sensor networks. In *ACM Conference on Computer and Communications Security*, pages 699–701, 2010.

[20] Jing Shi, Rui Zhang, Yunzhong Liu, and Yanchao Zhang. Prisense: Privacy-preserving data aggregation in people-centric urban sensing systems. In *INFOCOM*, pages 758–766, 2010.

[21] Elaine Shi, T-H.Hubert Chan, Eleanor Rieffel, Richard Chow, and Dawn Song. privacy preserving aggregation of time series data. In *NDSS*, 2011.

[22] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *SIGMOD Conference*, pages 735–746, 2010.

[23] Emmanouil Magkos, Manolis Maragoudakis, Vassilios Chrissikopoulos, and Stefanos Gritzalis. Accurate and large-scale privacy-preserving data mining using the election paradigm. *Data Knowl. Eng.*, 68(11):1224–1236, 2009.

[24] Sherman S. M. Chow, Jie-Han Lee, and Lakshminarayanan Subramanian. Two-party computation model for privacy-preserving queries over distributed databases. In *NDSS*, 2009.

[25] Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, 2006.

[26] Huang Lin, Yuguang Fang, and Zhenfu Cao. Private information extraction over online social networks. In *Eprint*, pages 1–11, 2011.

[27] D.j.Bernstein and T. L. ebacs: Ecrypt benchmarking of cryptographic systems. In *http://bench.cr.yp.to.*, 7 March, 2011.

[28] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[29] Victor Shoup, editor. *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*. Springer, 2005.

[30] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Shoup [29], pages 258–275.

Appendix

A. Inquirer Oblivious (IO) security

The following Inquirer Oblivious (IO) security game is first defined in [21]. The original game is concerning summation statistics on one dimension noisy message, while our proposed extractors are dealing with intersection information on $2mC+1$-dimensional message without adding noise. Therefore, we extend the original game to accomodate our proposed scenario.

Setup. Challenger runs the **Setup** algorithm, and returns the public parameters param to the adversary.

Queries. The adversary makes the following types of **queries** adaptively under the restriction to be further specified later.

1) **Encrypt**. The adversary may specify (u_i; t; S_i), and ask for the ciphertext. The challenger returns the ciphertext `Individual data generation`(param; SK_i; t; S_i) to the adversary.
2) **Compromise**. The adversary specifies an integer $i \in [0, n]$. If $i = 0$, the challenger returns the aggregator capability SK_0 to the adversary. If $i \neq 0$, the challenger returns SK_i, the secret key for the u_i, to the adversary.
3) **Challenge**. This query can be made only once throughout the game. The adversary specifies a set of participants V and a time t^*. Any $u_i \in V$ must not have been compromised at the end of the game. For each user $u_i \in V$, the adversary chooses a S_i. The challenger flips a random bit b. If $b = 0$, the challenger computes $\forall u_i \in V$: `Individual data generation`(SK_i, t^*, S_i), and returns the ciphertexts to the adversary. If $b = 1$, the challenger computes and returns a random ciphertext for each $u_i \in V$ instead. [3]

Guess. The adversary outputs a guess of whether b is 0 or 1. We say that the adversary wins the game if she correctly guesses b and the following condition holds. Let $K \subseteq \{u_1, \cdots, u_n\}$ denote the set of compromised participants at the end of the game (not including the aggregator). Let $Q \subseteq \{u_1, \cdots, u_n\}$ denote the set of participants for whom an **Encrypt** query has been made on time t^* by the end of the game. Let $V \subseteq \{I, u_1, \cdots, u_n\}$ denote the set of (uncompromised) participants or aggregator specified in the Challenge phase. If $V = \overline{K \cup Q} := \{u_1, \cdots, u_n\} \setminus (K \cup Q)$, and the adversary has compromised the aggregator capability, the following condition must be met:

$$\cap_{u_i \in V} S_i = \cap_{u_i \in V} S'_i$$

, where S'_i is the choice set for the returned random ciphertext when $b = 1$.

Definition 1: (Inquirer oblivious security). An intersection information extractor scheme is inqurier oblivious, if no probabilistic polynomial-time adversary has more than negligible advantage in winning the above security game

Explanation Suppose that the adversary has compromised the aggregator capability SK_0. In addition, for every participant $u_i \notin V$, the adversary knows a ciphertext $C_i = \{C_{ij}, j \in [0, 2mC]\}$ for the time t^* as well as the corresponding plaintext S_i. Such an adversary is able to use the `Information extraction` function to learn the intersection information over the subset V as mentioned in Sec. II. Note that the adversary may be able to learn a plaintext and ciphertext pair for $u_i \notin V$ in two ways. The adversary can either make an Encrypt query for $u_i \notin V$ and time t^*, or compromise the secret key of participant u_i so that it is able to produce the ciphertexts on its own. Therefore, when the aggregator capability has been compromised and $V = \overline{K \cup Q}$, we require that apart from the intersection information over the subset V, the adversary is unable to infer additional information about the individual choice set of the honest participants in V. This means that the adversary in the above security game is unable to distinguish which plaintext vector $\{S_i | u_i \in V\}$ or $\{S'_i | u_i \in V\}$ the challenger encrypted, as long as $\{S_i | u_i \in V\}$ and $\{S'_i | u_i \in V\}$ are equivalent with respect to the intersection information.

On the other hand, under the following conditions, the adversary learns nothing from the challenge ciphertexts corresponding to the set V of participants. 1) The adversary has not compromised the aggregator capability; or 2) $V \neq \overline{K \cup Q}$, i.e., there exists at least one $u_i \notin V$ for whom the adversary does not know a ciphertext for time period t^*. Under these situations, for arbitrary choices of $\{S_i | u_i \in V\}$ and $\{S'_i | u_i \in V\}$ that the adversary submits in the **Challenge** phase, the adversary is unable to distinguish which one the challenger encrypted.

The proposed schemes also make the **Encrypt-once security** assumption, which states each honest participant only encrypts once in each time period and it can be formally reflected in the following game.

Definition 2: (Encrypt-once security) The above game is inquirer oblivious in the encrypt-once model, if no PPT adversary has more than negligible advantage in the above game, and in addition, the following constraint holds: $\forall u_i \in V, \forall S_i$: the tuple (u_i, t^*, S_i) must not have appeared in any **Encrypt** query.

Similar to the proof of Theorem 1 in [21], we also need to prove the following intermediate game is difficult to win given the DDH problem is hard.

Setup Let $\mathbb{G}$ be a group of prime order p, the challenger picks random generators $g, h \in \mathbb{G}$, and random $\alpha_{0j}, \alpha_{1j}, \cdots, \alpha_{nj}, j \in [0, 2mC]$ from $\mathbb{F}_p^{(n+1)\times(2mC+1)}$ such that $\sum_{i=0}^{n} \alpha_{ij} = 0$ for each $j \in [0, 2mC]$. The challenger gives the adversary g, h, $\{g^{\alpha_{ij}}, i \in [0, n], j \in [0, 2mC]\}$.

Queries The adversary can make "compromise queries" adaptively and ask for the value of $\alpha_{ij}, j \in [0, 2mC]$. The challenger returns $\alpha_{ij}, j \in [0, 2mC]$ to the adversary when asked. We note either the adversary obtains all $2mC + 1$ secrets $\alpha_{ij}, j \in [0, 2mC]$ or nothing on u_i.

Challenge The adversary specifies an uncompromised set $V \subseteq \{I, u_1, \cdots, u_n\}$. The challenger flips a random coin b. If $b = 0$, the challenger returns to the adversary $\{h^{\alpha_{ij}} | u_i \in V\}$. If $b = 1$, the challenger picks $|V| \times (2mC+1)$ random elements $\{h'_{ij} | u_i \in V\}$ from the group $\mathbb{G}$ such that

$$\prod_{u_i \in V} h'_{ij} = \prod_{u_i \in V} h_{ij}^{\alpha_{ij}}$$

for each $j \in [0, 2mC]$.

[3] We consider a slightly different definition on the **Challenge** phase since there should be a trivial attack if this phase were defined as in [21]. In the original definition, the adversary could simply provide the challenger with two plaintext-randomness pairs $(0, 0)$, (x, r) for one user $u_i \in V$ and choose the other pairs carefully to keep the respective summation condition specified in equation (1) satisfied. Then for u_i, the challenger will return to the adversary $g^{0+0}H(t)^{\mathsf{SK}_i}$ when $b = 0$ and $g^{x+r}H(t)^{\mathsf{SK}_i}$ when $b = 1$. By dividing $g^{x+r}H(t)^{\mathsf{SK}_i}$ to $g^{0+0}H(t)^{\mathsf{SK}_i}$, the adversary can trivially guess out b without solving any hard problem. The success of this trivial attack is due to the fact that the challenger has to encrypt the message twice for one period in the original definition of **Challenge** phase, which is somehow against the **Encrypt-once security** principle to be introduced. Our definition on the **Challenge** phase emulates that of the broadcast encryption system [30], which can successfully express the security requirement while avoiding the above conflict.

The challenger returns $\{h'_{ij}|u_i \in V\}$ to the adverary.

More Queries The adversary can make more ”compromise” queries

Guess The adversary guesses either $b = 0$ or $b = 1$.

The adversary wins the game if he has not asked for any α_{ij} for $u_i \in V$, while successfully guessing b. We also require that $|V| \geq 2$, since otherwise, the distribution of the outputs of the challenger when $b = 0$ and $b = 1$ are trivially indistinguishable.

Lemma 1: The above game is difficult for PPT adversaries assuming DDH problem is hard for group $\mathbb{G}$

Proof of Lemma 1: This proof can be viewed as a hybrid argument with $(2mC+1)\times|V|$ game sequences. The proof can run the proof of Lemma 2 in [21] as the underlying routine. The $(2mC+1)\times|V|$ game sequences can be divided into $(2mC+1)$ groups of game sequences, each group of which proves $\{h^{\alpha_{ij}}|u_i \in V\}$ and $\{h'_{ij}|u_i \in V\}$ such that $\prod_{u_i \in V} h'_{ij} = \prod_{u_i \in V} h_{ij}^{\alpha_{ij}}$ are computationally indistguishable. In other words, the above lemma can be proven by directly repeating $2mC+1$ underlying proofs. ■

Proof of Theorem 1: We will make the same modification as in the proof of Theorem 1 in [21] in the sense that we will treat the **Encrypt** queries on t^*, which is the time step specified in the **Challenge** phase, as **Compromise** queries too.

The security game is also divided into two cases. Let $K \subseteq \{u_1, \cdots, u_n\}$ denote the set of compromised participants. Let $\overline{K} := \{u_1, \cdots, u_n\} \setminus K$ denote the set of uncompromised participants.

- **Case 1**. $V \neq \overline{K}$ or the aggregator capability has not been compromised. In other words, either there exists at least an uncompromised participant $u_i \notin V$ at the end of game or the aggregator capability has not been compromised. In this case, it suffices to show that the adversary cannot distinguish between real or random, that is, whether the challenger returns a faithful encryption of the plaintext submitted in the challenge stage, or a random tuple picked from the appropriate group.
- **Case 2**. $V = \overline{K}$ and the aggregator capability has been compromised. In this case, we show that the adversary cannot distinguish whether the challenger returns a faithful encryption of the plaintext submitted in the challenge stage, or a random tuple with the same product. Given an adversary $\mathfrak{A}$ who can break the IO game with non-negligible probability, we construct an algorithm $\mathfrak{B}$ who can solve the above intermediate problem with non-negligible probability.

Setup. $\mathfrak{B}$ obtains from its challenger $\mathfrak{C}$ the following tuple $g, h \in \mathbb{G}$, and random $\alpha_{0j}, \alpha_{1j}, \cdots, \alpha_{nj}, j \in [0, 2mC]$ from $\mathbb{F}_p^{(n+1)\times(2mC+1)}$. $\mathfrak{B}$ implicitly sets α_{0j} to be I's capability, and $\alpha_{1j}, \cdots, \alpha_{nj}$ to be the secret keys of participants u_1 through u_n respectively. $\mathfrak{B}$ also needs to randomly select a choice universe and a monic irreducible polynomial $f(x)$ of degree $2mC+1$. The public params is $g, f(x)$ and the choice universe. The algorithm $\mathfrak{B}$ makes a random guess as to whether Case 1 or Case 2 will happen, and if the guess turns out to be wrong, the simulator simply aborts. Moreover, if $\mathfrak{B}$ guesses Case 1, then $\mathfrak{B}$ will randomly guess a participant (or aggregator) $u_{j^*} \in (\overline{K} \setminus V) \cup \{0\}$ that remains uncompromised at the end of the game. If the guess turns out to be wrong later, $\mathfrak{B}$ aborts. Let q_H denote the total number of oracle queries made by the adversary $\mathfrak{A}$ and by the algorithm $\mathfrak{B}$ itself. $\mathfrak{B}$ guesses at random an index $k \in [1, q_H]$. Suppose the input to the $k-$th random oracle query is t^*. The algorithm $\mathfrak{B}$ assumes that t^* will be the challenge time step. If the guess turns out to be wrong later, $\mathfrak{B}$ simply aborts.

Hash Function Simulation. The adversary submits a hash query for the integer t. $\mathfrak{B}$ first checks the list L to see if t has appeared in any entry $(t;\ z)$. If so, $\mathfrak{B}$ returns g^z to the adversary. Otherwise, if this is not the $k-$th query, $\mathfrak{B}$ picks a random exponent z and returns g^z to the adversary, and saves $(t;\ z)$ to a list L. For the $k-$th query, $\mathfrak{B}$ returns h.

Queries.

- **Encrypt**. The adversary $\mathfrak{A}$ submits an **Encrypt** query for the tuple $(u_i;\ S_i;\ t)$. As mentioned above, in the modified version of the game, we ensure that $t \neq t^*$, since otherwise, we simply treat it as a **Compromise** query. $\mathfrak{B}$ checks if a hash query has been made on t. If not, $\mathfrak{B}$ makes a hash oracle query on t. As a result, $\mathfrak{B}$ knows the discrete log of $H(t)$. Let $H(t)$= g^z, then $\mathfrak{B}$ knows z. Since $\mathfrak{B}$ also knows $g^{\alpha_{ij}}$, $\mathfrak{B}$ can compute the ciphertext $g^{c_{ij}}(g^z)^{\alpha_{ij}}$ as $g^{c_{ij}}(g^{\alpha_{ij}})^z$.
- **Compromise** $\mathfrak{B}$ forwards $\mathfrak{A}$'s query to its own challenger $\mathfrak{C}$ and forwards the answer $\{\alpha_{ij}, j \in [0, 2mC]\}$ to $\mathfrak{A}$.

Challenge. The adversary $\mathfrak{A}$ submits a set V and a time t^*, as well as plaintexts $\{S_i|u_i \in V\}$. If t^* does not agree with the value submitted in the $k-$th hash query, then $\mathfrak{B}$ aborts. If $\mathfrak{B}$ has guessed Case 1 at the beginning of the game, then it submits the set $V \cup \{u_{j^*}\}$ in a Challenge query to its own challenger $\mathfrak{C}$. As a result, it obtains a tuple $\{T_{ij}\}_{u_i \in V, j \in [0,2mC]}$, $\{T_{j^*j}\}_{j \in [0,2mC]}$.

If $\mathfrak{B}$ has guessed Case 2, then it simply submits the set V in a **Challenge** query to its own challenger. As a result, it obtains a tuple $\{T_{ij}\}_{u_i \in V, j \in [0,2mC]}$. In both cases, the challenger returns the following ciphertexts to the adversary:

$$\forall i \in V, j \in [0, 2mC] : g^{c_{ij}} \cdot T_{ij}$$

.

More queries. Same as the **Query** stage.

Guess. If the adversary $\mathfrak{A}$ guesses that $\mathfrak{B}$ has returned a random tuple then $\mathfrak{B}$ guesses b' = 1. Otherwise, $\mathfrak{B}$ guesses that b' = 0.

- Case 1. If the challenger $\mathfrak{C}$ returns to $\mathfrak{B}$ a faithful Diffie-Hellman tuple $\forall u_i \in V, j \in [0, 2mC] : T_{ij} = h^{\alpha_{ij}}$, and T_{j^*j}=$h^{\alpha_{j^*j}}$, then the ciphertext returned to the adversary $\mathfrak{A}$ is a faithful encryption of the plaintext submitted by the adversary. Otherwise, if the challenger returns to $\mathfrak{B}$ a random tuple under the product constraint, then the ciphertext returned to $\mathfrak{A}$ is a random tuple.
- Case 2. If the challenger $\mathfrak{C}$ returns gives $\mathfrak{B}$ a faithful Diffie-Hellman tuple $\forall u_i \in V, j \in [0, 2mC] : T_{ij} = h^{\alpha_{ij}}$, then the ciphertext returned to the adversary $\mathfrak{A}$ is a faithful encryption of the plaintext submitted by the adversary. Otherwise, if the challenger returns to $\mathfrak{B}$ a random tuple under the product constraint, then the ciphertext returned to $\mathfrak{A}$ is a random tuple under the product constraint.

■

Proof of Theorem 3: It is implied in Theorem 1 of [21] that our proposed union information extractor is inquirer oblivious with respect to the product statistics, i.e., $\prod_{i=1}^{n} p_j^{(i)} \bmod p$ in our system. According to Theorem 2, the mere information which can be deduced from $\prod_{i=1}^{n} p_j^{(i)}$=$\prod_{i=1}^{n} p_j^{(i)} \bmod p$ is the union information. Therefore, the proposed union information extractor is inquirer oblivious. ■