A Verification Scheme For Data Aggregation in Data Mining

Kilho Shin
Carnegie Mellon CyLab Japan
yshin@cmuj.jp

Justin Zhan
The Heinz School
Carnegie Mellon University
justinzh@andrew.cmu.edu

Abstract

To conduct data mining, we often need to collect data from various data owners. Privacy concerns may prevent the parties from directly sharing the data and some types of information about the data. To conduct data mining without compromising data privacy, we propose a verification scheme to ensure that the collected data follow the requirements of data miners, which is one of the important issues in privacy-preserving data mining systems.

Key Words: Data Collection, Data Mining, Verification, Cryptography.

1 Introduction

Data mining is a powerful technology in extracting the hidden predictive information from large databases. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analysis offered by data mining move beyond the analysis of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

Data mining techniques are the result of a long process of research and product development. Recent advances in data collection, data dissemination and related technologies have inaugurated a new era of research where existing data mining algorithms should be reconsidered from the point of view of privacy preservation.

In this paper, we would like to focus on data collection. In particular, we consider the following scenario: There are a number of data owners. Each of them retains or generates as needed certain data sets. Those data sets are usually related to activities of an individual, and therefore may include information that enables identification of the individual. On the other hand, a data aggregator collects such data sets, then forward them to data analysts.

In order to obtain correct analyzing results, a practical model is that the data analysts inform data owners of their requirements regarding what information should be included in data sets. This indicates that the model should be equipped with some methods to make data owners honestly fulfill the requirements.

To address this problem, the paper assumes certain devices on hand at data owners' points that guarantee the support of the requirements and the accuracy of data sets on behalf of data analysts. As an implementation in the real world of such devices, referred to as *Observers* in the paper, we have IC-chip-based credit cards in mind. They not only have access to *accurate* personal information of customers but also provide secure environments to process data sets without dishonest interference by customers and others. Moreover, it is important to note that credit card companies are trustworthy from standpoints of customers and data analysts, and they certainly have motivation to play the role on behalf of data analysts. In consideration of these points, we strongly believe the framework with Observers for accurate data collection is realistic and feasible.

However, from a privacy preservation point of view, the framework stated above would certainly raise problems. Since data sets may include information that would reveal the identities of customers or would allow trace of their activities, they must not be disclosed to anyone but data analysts. Although this request can be supported by encrypting data sets, the encryption should be performed by Observers since, otherwise, customers have opportunities to submit inaccurate information. However, if Observers perform encryption, customers have serious concerns that Observers disclose sensitive information of customers exceeding their consents. Thus, it is critical to support this seemingly contradictory requirements. The main contribution of this paper is to propose a verification scheme to achieve such a goal.

0-7695-3032-X/07 $25.00 © 2007 IEEE
DOI 10.1109/GrC.2007.121

2 Framework and Fundamental Requirements

2.1 Players

In the process of data collection, we observe three fundamental players, namely *data owners* who retain or generate as needed data sets to be collected, *data analysts* who draw some conclusion through analysis of collected data, and *data aggregators* who collect data sets on behalf of data analysts.

Further, we will classify data owners into two categories. One consists of *privacy subjects* (*Subject*), while the other consists of *reporters* (*Reporter*). A *Subject* (*e.g.* a customer) is an individual such that collected data sets may disclose his/her identity and/or may enable third parties to track him/her. On the other hand, a *Reporter* (*e.g.* a shop, restaurant, virtual mall) provides information relating to *Subject*'s activities and/or interaction with the *Reporter*.

Thus, a data set comprises the part that a *Subject* provides and the part that zero or more *Reporter*'s provide. The former part includes information that third parties can never get to know unless the *Subject* intentionally discloses it. In the rest of the paper, we refer to this part of data as *non-observable* private data. In contrast, the later part may include private information such that the *Reporter* obtains through observation of activities of the *Subject*. We refer to this part of data as *observable* private data. Table 2.1 shows examples of non-observable and observable private data.

Data type	Examples
Non-observable	Name, Age, Day of Birth, Address, Telephone Number, Social Security Number, Passport Number, Nationality, Religion
Observable	Gender, Race, Height

Table 1. Non-observable and observable private data

2.2 An Issue of Accuracy and A Solution By the *Portal*

From *Analyst*'s standpoint, the most important requirement for the data sets that it is going to collect must be their accuracy, in particular, the accuracy of *non-observable information*— *non-observable information* is such a kind of information that the *Subject* may be motivated to hide from *Analyst*'. Hence, it is likely that the *Subject* submits incorrect or totally different information as his/her *non-observable information*. To avoid such frauds by the *Subject*, and to support the *Analyst*'s requirement for the accuracy of the data sets, this paper proposes the introduction of *Portal*.

A *Portal* is a small portable device such that a *Subject* carries with himself/herself. A *Portal* stores *Subject*'s *non-observable information* in such a manner that even the *Subject* cannot change it.

Considering a credit card as a candidate implementation of the *Portal*, the proposal is realistic. — A credit card is a very small portable device, and most of people retain at least one for each. Also, a credit card is used as a method of proof of cardholder's identity in daily life.

From security point of view, a *smart-chip-equipped* credit card [1] has plural desirable properties. First, it provides fully access-controlled memories to protect stored confidential information. This functionality not only protects cardholder's secrets for cardholder's sake, but also prevents a dishonest cardholder from tampering with data stored in it. Technically speaking, the smart-chip is equipped with a micro computer (for this reason, it is called smart), and the micro computer controls all the accesses from the outside to the data and the programs stored inside. Against attempts to bypass the access control by the micro computer, its tamper-resistant hardware implementation defeats such attempts. In addition, the smart chip executes any confidential calculation (*e.g.* cryptographic calculation using secret keys) by itself, and hence confidential data consistently stay in the smart chip.

A smart-chip-equipped credit card has advantages not only from security point of view but also from practice point of view. One of them is the derived information from the fact that credit card companies, namely the providers of credit cards, have been handling cardholders' private information. In this sense, credit card companies have already acquired reasonable confidence in society to play the role of *Aggregator*. Consensus to use credit cards as the *Portal* would be created without great difficulties.

Taking advantage of its security features, on input of *observable information* from *Reporter*'s, a *Portal* combine the *non-observable information* stored in itself with the input *observable information* into a data set, and digitally signs it to guarantee the accuracy.

2.3 An issue of privacy and our contribution

From a privacy protection point of view, a *Subject* should be able to know the contents of data sets in advance that they are submitted to *Analyst*'s, and should retain the rights to reject requests to disclose his/her private information at will. Moreover, the *Subject* must require that his/her *non-observable information* should be disclosed only to the intended *Analyst*, and, in particular, should be hidden from

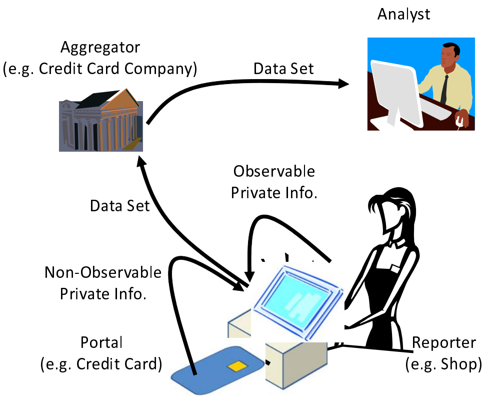

Figure 1. The Proposed Architecture

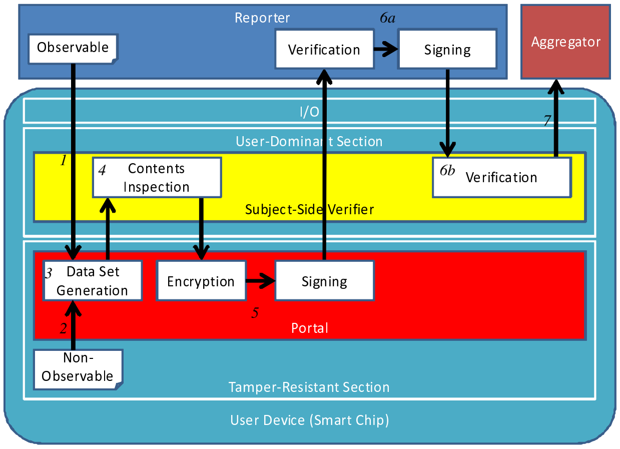

Figure 2. An architecture of observers

*Aggregator*and *Reporter*. This requirement for privacy protection implies the following.

1. The data sets must be encrypted so that only the intended *Analyst* can decrypt it.

2. The *Subject* must be able to convince himself/herself that the encryption is executed properly.

The simplest way to fulfill the above is that the *Subject* encrypts data sets using *Analyst*'s public key. However, this method apparently contradicts the requirement for the accuracy of data sets stated in the previous section — a dishonest *Subject* may encrypt incorrect or fake data sets.

Thus, to fulfill the accuracy requirement, it is necessary that the encryption is executed by a *Portal*. On the other hand, to fulfill the privacy protection requirement, we need an appropriate verification scheme with which the *Subject* can inspect whether the *Portal* properly encrypts the data sets that the *Subject* intends to submit to the *Analyst* (*e.g.* whether the *Portal* did not encrypt information exceeding what the *Subject* consents to disclose, whether the encryption is executed using *Analyst*'s public key). The objective and the main contribution of this paper is to present such a verification scheme.

2.4 Requirements

Based upon the arguments in the previous sections, Table 2.4 clarifies requirements from the viewpoints of the *Subject*, *Reporter*, *Analyst* and *Aggregator*respectively.

2.5 The Proposed Architecture

Fig. 2 provides a big picture of our proposed architecture. A user device (*e.g.* a smart chip of a credit card) comprises two section. — One is the tamper-resistant section, while the other is the user-dominant section. In the tamper-resistant section, the *non-observable information* of the device-holder is stored, and the *Portal* program is installed as well. Due to the tamper-resistant feature of the device, even the device-holder cannot tamper with the *non-observable information* or interfere in the processes of the *Portal*. On the other hand, the user-dominant section is laid under the total control by the device-user. — The device-user can store any kinds of data and install arbitrary programs at will. The *Subject-Side Verifier* program, which controls all the inputs and the outputs of the *Portal*, and, in particular, inspects the contents of data sets as well as the outputs of the *Portal*, is presented in this section.

Fig. 2 also depicts the data flow and the procedures of the cooperation of the *Portal* the *Subject-Side Verifier* and the other entities.

The Information Flow in Fig. 2

1. The *Reporter* inputs an instance of *observable information* into the *Portal* via the *Subject-Side Verifier*.

2. The *Portal* retrieves the necessary part of the *non-observable information* stored in the tamper-resistant section.

3. The *Portal* prepares a data set by combining the *observable information* and the *non-observable information*, and then outputs the generated data set to the *Subject-Side Verifier*.

4. The *Subject-Side Verifier* inspects whether its contents are appropriate. If possible, the *Subject-Side Verifier* may present the contents to the *Subject* for his/her consent using a displaying device (*e.g.* LCD). Unless the *Subject-Side Verifier* is satisfied with the contents, the *Subject-Side Verifier* discards the data set and aborts the process.

5. Responded by the *Subject-Side Verifier*, the *Portal* encrypts the data set using *Analyst*'s public key, and then signs the encrypted data set.

Player	Requirements
Subject	Only the *non-observable information* that the *Subject* consents to disclose shall be disclosed only to the *Analyst* to whom the *Subject* intends to disclose the *non-observable information*. In particular, the *non-observable information* shall be hidden from *Aggregator*'s and *Reporter*'s sights.
Reporter	The *observable information* submitted to an *Analyst* shall be the same as what the *Reporter* intends to submit to the *Analyst*. In particular, the *Subject* shall not be able to tamper with the *observable information*.
Analyst	A received data set shall be accurate in the following sense. — The *non-observable information* included in the data set shall be of the sorts that the *Analyst* requested and shall not be tampered with; The *observable information* included in the data set shall be the same as what the *Reporter* intends to submit to the *Analyst*.
Aggregator	On receipt of a data set, and in advance of submitting it to the *Analyst* the *Aggregator* shall be able to verify that it is accurate in the sense stated in the previous row.

Table 2. Requirements from the viewpoint of each player

6. The encrypted and signed data set is sent to the *Subject-Side Verifier*, and the *Subject-Side Verifier* executes the following.

   (a) The *Subject-Side Verifier* sends the received data to the *Reporter*. The *Reporter* verifies that the received data is right encryption of the same *observable information* as the *Reporter* submitted in Step 1. If, and only if, the verification succeeds, the *Reporter* signs the data and returns it to the *Subject-Side Verifier*.

   (b) The *Subject-Side Verifier* verifies that the received data is right encryption of exactly the same data set as the *Subject-Side Verifier* inspected in Step 4. If the verification fails, the *Subject-Side Verifier* discards the data and aborts the process.

   The *Subject-Side Verifier* outputs the encrypted and doubly-signed data set.

7. Finally, the encrypted data set is sent to the *Aggregator*, who collects data sets on behalf of *Analyst*.

2.6 Introducing Problem

The requirements of the *Analyst* and the *Aggregator* will be supported by the procedures illustrated in the previous section. In fact, the *Analyst* and the *Aggregator* have only to verify the signatures of the *Portal* and the *Reporter*.

To support the requirements of the *Subject* and the *Reporter*, we need an additional verification scheme that should be used in Step 6b and Step 6a. The objective of the paper is to propose such a verification scheme. The difficulty in designing such a verification scheme is that neither the *Subject* nor the *Reporter* can decrypt the encrypted data set, because it is encrypted using *Analyst*'s public key.

The problem that we are struggling with in the subsequent sections is to engineer a verification scheme such that the *Subject* and the *Reporter* can verify that an encrypted data set is to encrypt the data that they expect without accessing *Analyst*'s private key.

3 Verifiable Entrusted Encryption

3.1 Definition

We consider a public key encryption scheme $\mathcal{E}$ with the following features.

- The scheme involves three players, an entruster, a trustee and a recipient.
- The trustee receives a clear text m from the entruster, and encrypts it into c using the recipient's public key R and the entrustee's public key E_R.
- The entruster verifies that c is a right encryption of m using its private key.
- The recipient decrypts c into m using its private key.

Definition 1 *Let m be an arbitrary message that an entruster tries to make a trustee to encrypt. While c is right encryption of m with respect to R and E_R, let $\bar{c}$ be arbitrary data such that it is not right encryption of m. Furthermore, let $\tilde{c}$ be right encryption of m but using the public key of a different entruster. If $\mathcal{E}$ satisfies the conditions of* Completeness, Soundness *and* Secrecy, *it is called a* verifiable entrusted encryption scheme.

- **Completeness:** *The entruster accepts c.*
- **Soundness:** *The entruster accepts $\tilde{c}$ only with a negligible probability.*
- **Secrecy:** *The entruster distinguishes between $\bar{c}$ and $\tilde{c}$ only with a negligible advantage.*

In the subsequent clauses, we propose an instance of the verifiable entrusted encryption scheme by presenting its key generation, encryption, decryption and verification steps separately. Further, we will give a proof for its completeness, soundness and secrecy.

3.2 Key Generation

The public key pairs of the recipient and the entruster are generated using a common cyclic group $\mathcal{G}$. In the remainder of this paper, the operation of $\mathcal{G}$ is denoted by the additive operator $+$ for simplicity. Moreover, we assume the following.

- $\mid \mathcal{G} \mid= p$ for a prime number p.
- G is a fixed generator of $\mathcal{G}$.
- Decision Diffie-Hellman problems (DDHP) are computationally intractable over $\mathcal{G}$. Given a quadruple $(X, Y, Z, U) \in \mathcal{G}^4$, the corresponding DDHP is the problem to determine whether $Y = aX$ and $U = aZ$ hold for some $a \in [0, p)$.

The public key pair of the recipient is a pair (R, ρ) such that $R = \rho G$. The recipient randomly selects $\rho \in [0, p)$, and submits R as its public key.

On the other hand, the public key pair of the entruster is a pair $(E_R, (x, y))$ such that $E_R = xG + yR$. The recipient randomly selects $(x, y) \in [0, p)^2$, and submits E_R as its public key.

Note that E_R is dependent on R, and, therefore, the entruster has to prepare as many public key pairs as the recipients that it likes to communicate.

3.3 Encryption

The scheme that we propose in this paper is based on the ElGamal public key encryption. The trustee encrypts a clear text m as follows.

1. Select a random $\beta \in [0, p)$.
2. Calculate $A = \beta E$, $B = \beta G$, and $c = \beta R \oplus m$.
3. Output the triplet (A, B, c) as the cipher text.

3.4 Decryption

Since (B, c) is in accordance with the ordinary ElGamal public key encryption, the recipient who retains the private key ρ corresponding to the public key R can decrypts m as follows.

$$m = c \oplus \rho B \oplus c$$

3.5 Verification

The entruster doesn't know the private key ρ, which is a critical component to decrypt c. However, by verifying that the following equality holds, it can verify the correctness of (A, B, c) without decrypting c.

$$A = xB + y(c \oplus m) \tag{1}$$

3.6 Proof of Security

We will prove the completeness, soundness and secrecy of the proposed scheme.

The completeness immediately follows from the definition.

The soundness is proved as follows. When α, ϵ and ξ satisfy $A = \alpha G$, $E_R = \epsilon G$ and $c \oplus m = \xi G$, the fact that the entruster successfully verifies (1) indicates that the equalities of (2) and (3) hold at the same time.

$$\epsilon = x + y\rho \tag{2}$$

$$\alpha = x\beta + y\xi \tag{3}$$

If $\xi \neq \beta\rho$, only a single instance of (x, y) satisfies (3) and (2). Therefore, the probability that the trustee, who does not know the value of x or y, can present A such that (3) holds is only $\frac{1}{p^2}$.

Lastly, we will prove the secrecy of the proposed scheme. While $(G, R, \tilde{B}, \tilde{c} \oplus m)$ is a DDH quadruple but $(G, R, \bar{B}, \bar{c} \oplus m)$. If the entruster distinguishes between them with a significant advantage, it can also solve DDHP's.

4 Solving the Problem

To solve the problem stated in Section 2.6, it suffices that the *Portal* plays the role of the trustee, and encrypts the *non-observable information* and *observable information* in accordance with the verifiable entrusted encryption scheme presented in the previous section.

In the following, let π and δ denote the *non-observable information* and the *observable information*, respectively. The *Portal* uses *Analyst*'s public key R, *Subject*'s public key E_R and *Reporter*'s public key $\bar{E}_R$. In Step 5, the *Portal* encrypts the *non-observable information* and *observable information* as follows.

1. Generate random $\beta, \beta' \in [0, p)$ and let B and B' be βG and $\beta' G$.
2. Encrypt δ and π into (B, c) and (B', c')
$$c = \beta R \oplus \delta \quad c' = \beta' R \oplus \pi$$
3. Calculate A, A' and $\bar{A}$ as follows.
$$A = \beta E_R, \quad A' = \beta' E_R, \quad \bar{A} = \beta \bar{E}_R$$
4. The cipher text for δ is $(A, \bar{A}, B, c)$, while that for π is (A', B', c').

In Step 6b of the information flow architecture, the *Subject-Side Verifier* inspects (A, B, c) and (A', B', c') based on the knowledge of δ and π informed in Step 4. In the same way, in Step 6a, the *Reporter* inspects $(\bar{A}, B, c)$ based on the knowledge of δ that it submitted.

5 Discussion

To protect actual data from being disclosed, one approach is to alter the data in a way that actual individual data values cannot be recovered, while certain computations can still be applied to the data. Due to the fact that the actual data are not provided for the mining, the privacy of data is preserved. This is the core idea of randomization-based techniques. Randomization approaches were first proposed by Agrawal and Srikant [2] to solve the privacy-preserving data mining problem. Specifically, they addressed the following question. Since the primary task in data mining is the development of models about aggregated data, can they develop accurate models without access to precise information in individual data records? The underlying assumption is that a person will be willing to selectively divulge information in exchange of useful information that such a model can provide. Du and Zhan [3] proposed a technique for building decision trees using randomized response techniques which were developed in the statistics community for the purpose of protecting surveyees privacy. The randomization-based methods have the benefits of efficiency. However, the drawbacks are that post-randomization data mining results are only an approximation of pre-randomization results. Encryption is a well-known technique for preserving the confidentiality of sensitive information. Comparing with other techniques described, a strong encryption scheme can be more effective in protecting the data privacy. An encryption system normally requires that the encrypted data should be decrypted before making any operations on it. For example, if the value is hidden by a randomization-based technique, the original value will be disclosed with certain probability. If the value is encrypted using a semantic secure encryption scheme [4], the encrypted value provide no help for attacker to find the original value. One of the schemes is the homomorphic encryption which was originally proposed in [5] with the aim of allowing certain computations performed on encrypted data without preliminary decryption operations. To date, there are many such systems. Homomorphic encryption is a very powerful cryptographic tool and has been applied in several research areas such as electronic voting, on-line auction, etc. [6] is based on homomorphic encryption where Wright and Yang applied homomorphic encryption to the Bayesian networks induction for the case of *two* parties. Zhan et. al. [7] proposed a cryptographic approach to tackle collaborative association rule mining among multiple parties.

To our best knowledge, most of the previous works dealing with data mining computation lack of works coping with verification in data aggregation which is one of critical steps in data mining systems. In this paper, we have proposed a verification scheme for data aggradation in data mining process. We have shown that our scheme guarantees that data analysts can verify whether the collected data follow the requirements without actually disclosing the original data. In the future, we would like to extend the work to cope with the verification of other steps in data mining systems.

References

[1] EMVCo, EMV Integrated Circuit Card Specification for Payment Systems — Common Payment Application Specification, December, 2005, http://www.emvco.com

[2] R. Agrawal and R. Srikant, Privacy-preserving data mining, Proceedings of the ACM SIGMOD Conference on Management of Data, 439–450,May, 2000, Dallas, Texas

[3] W. Du and Z. Zhan, Using Randomized Response Techniques for Privacy-Preserving Data Mining, Proceedings of The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, DC, USA. August 24 - 27, 2003

[4] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, Advances in Cryptography - EUROCRYPT '99, pp 223-238, Prague, Czech Republic,1999

[5] R. Rivest and L. Adleman and M. Dertouzos, On data banks and privacy homomorphisms, Foundations of Secure Computation, eds. R. A. DeMillo et al., Academic Press, pp. 169-179, 1978

[6] R. Wright and Z. Yang, Privacy-Preserving Bayesian Network Structure Computation on Distributed Heterogeneous Data, Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD),2004

[7] J. Zhan and S. Matwin and L. Chang, Privacy-Preserving Collaborative Association Rule Mining, 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Nathan Hale Inn, University of Connecticut, Storrs, CT, U.S.A., August 7-10, 2005