INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL. 10, NO. 1, MARCH 2005, 77-85

Mining Inconsistent Secure Messages Toward Analyzing Security Protocols

Chengqi ZHANG, Yi-Ping Phoebe CHEN, Shichao ZHANG and Qingfeng CHEN

Abstract- **Traditional approaches such as theorem proving and model checking have been successfully used to analyze security protocols. Ideally, they assume the data communication is reliable and require the user to predetermine authentication goals. However, missing and inconsistent data have been greatly ignored, and the increasingly complicated security protocol makes it difficult to predefine such goals. This paper presents a novel approach to analyze security protocols using association rule mining. It is able to not only validate the reliability of transactions but also discover potential correlations between secure messages. The algorithm and experiment demonstrate that our approaches are useful and promising.**

***Index Terms*—Security protocol, association rule mining, secure message, inconsistency**

1. INTRODUCTION

The rapid growth of Electronic commerce (e-commerce) not only plays a nontrivial role in global economy but also pose a challenge to security in e-commerce. Plenty of security protocols have been developed to ensure data integrity and confidentiality [10-12] in e-commerce. However, the design of security protocols is error-prone [1], and missing and spurious data are prevalent in data gathering today.

Traditional methods to analyzing security protocols are mainly implemented by theorem proving [1, 5, 6, 7] and model checking [5, 8]. They have been successful in modeling the behavior of a protocol and mathematically verifying that the protocol design and implementation satisfy protocol safety properties. Usually, the verification starts from the assumption, via intermediate formulae, to the authentication goal predefined by users [8]. In other words, users should have clear ideas about what the suspectable problems are. Nevertheless, it becomes infeasible in case of a complicated security protocol. Also, some hidden security problems are not easy to be detected by users.

Traditional approaches unrealistically assume that the data communication is reliable. However, the transmitted secure messages in hostile environment can bring out missing or inconsistent values such as a user's identity, credit card number and password in transaction databases [3]. These inconsistent secure messages are in fact interactional. For example, a tampered user's *password* may imply the potential divulgence of user's *credit card number* and *identity*. Therefore, there is an urgent need to deal with the analysis of increasingly complicated security protocols and identify the potential correlation between secure messages.

In the recent years, data mining techniques emerged as a means of identifying patterns and trends from large quantities of data [9, 13, 16]. Among them, association rule mining is a popular summarization and pattern extraction algorithm to identify the correlations between items in transactional databases [2]. Unlike the conventional idea of association rule discovery, we extend the original idea to *association rule mining of inconsistent secure messages*. In order to conform to the new idea, we need to make some extension to the original setting. It is briefly summarized below.

- Firstly, the missing message in itemsets should be taken into account. For example, in A' = {*expiration date, password*$_2$*, account number*}, it misses message *name*. Therefore, itemsets A = {*expiration date, password*$_2$*, account number, name*} and A' are regarded as two different itemsets.
- Even all messages in two itemsets are corresponding, they must be consistent, and otherwise they are deemed to be inconsistent itemsets. For example, since the *password*$_2$ in A' is not equal to *password*$_1$ in A, they are inconsistent.

During transactions, if messages in itemset A were lost or tampered it will lead to the decrease of the number of occurrences of this itemset, and in this way, the degree of support and confidence for related rules will decrease. And thereby, it is reasonable to say this transaction is insecure in case that the support and confidence of the rule is smaller than specified *minimum support* and *minimum confidence* by users or experts.

This paper presents how to use association rule mining to analyze security protocols and identify the potential correlation between secure messages. Unlike traditional market basket data, the freshness of secure messages and validity of public keys must be examined before extracting frequent itemsets. In particular, missing and inconsistent items in transaction databases are converted to operable data for data mining.

The rest of this paper is organized as follows. Section 2 briefly summarizes the related work. Section 3 presents some basic concepts. Section 4 presents how to analyze inconsistent secure messages using association rule mining.

Manuscript received September 10, 2004; revised December 13, 2004. This research is partially supported by Discovery Grants from the Australian Research Council (DP0559251, DP0449535).

Prof. Chengqi Zhang and Dr. Shichao Zhang are with Faculty of Information Technology, University of Technology Sydney, PO Box 123, Broadway NSW 2007, Australia ({chengqi, zhangsc}@deakin.edu.au). Asso Prof. Phoebe Yi-ping Chen and Dr. Chen are with School of Information Technology, Deakin University, VIC 3125, Australia (e-mail: {phoebe, qifengch}@deakin.edu.au).

Section 5 presents algorithms and experiments. Finally, we conclude this paper in Section 6.

2. RELATED WORK

Security protocols have become the requisite of e-commerce systems. However, they easily suffer from malicious attacks for their implicit specifications, and their designs are a difficult and error-prone task [4, 7]. A variety of methods and tools have been developed to verify these protocols [1, 5, 6]. Among them, theorem proving and model checking have gained many attentions.

As to theorem proving, BAN logic [1] is the representative work among them, from which a number of approaches are developed [4, 6]. It expresses the assumption and goal as statements in a symbolic notation so that the logic can proceed from a known state to one where it can ascertain whether the goal is in fact reached. Therefore, it is not surprising that the predetermination of authentication goals can become difficult for increasingly complicated security protocols. Additionally, some latent flaws are not easy to be detected. Model checking is another kind of approaches aiming at automated verification of security protocols. Lowe [5] used Failures Divergences Refinement Checker (FDR) to debug and validate the correctness of Needham-Schroeder protocol and Heintze [8] used FDR to verify NetBill and a simplified digital cash protocol. However, they ideally assume that the communication channel and principal are secure and trustworthy. The inconsistency between secure messages is partially neglected.

There have been many approaches in tackling the inconsistency in knowledge bases, such as arbitration based information merging [17] and majority based information merging [18]. Nevertheless, they focus on the handling of incoherence in knowledge base rather than the inconsistency in secure messages. A logical framework to merging inconsistent secure messages was presented in [4]. However, all of them are unable to identify potential associations between secure messages.

Data mining, with its potential to discover hidden and valuable information from large databases has been successfully used to identify patterns and trends from large quantities of data [13 16]. Among them, association rule mining plays an important role in identifying correlations between items in transactional databases [2]. Although data mining techniques have recently touched on privacy issues, they put emphasis on how to protect sensitive knowledge before sharing. For example, a two-party algorithm [19] is presented to efficiently discover frequent itemsets with minimum support levels, without either revealing individual transaction values. Stanley and Osmar proposed a new framework for enforcing privacy in mining frequent itemsets, in which it combines techniques for efficiently hiding restrictive patterns [20]. Unfortunately, no work has been closely related to analyzing security protocols using data mining.

3. BASIC CONCEPTS

Suppose that L denotes a set of proposition formulae formed in the usual way from a set of atom symbols A. In particular, A can contain α and $\neg\alpha$ for some atom α. $\wedge$, $\neg$ and $\rightarrow$ denote logical connectives. We use X, Y and P for principals, CA for Certificate Authority, and m, $\alpha \in A$ for messages in general. Let k be a key and $Cert(X)_{CA}$ be X's certificate signed by CA. $K_p(X)$ and $K^{-1}(X)$ represent the public/private pair of X respectively. $S(m, k)$ represents the signed message m by key k. Let $T_{expiration}$ be expiration time of message and T be timestamp.

Cryptography [14] is an essential tool to achieve data security such as authentication, integrity and confidentiality in e-commerce systems. In general, it is classified into asymmetric cryptography and symmetric cryptography. Symmetric cryptography uses the same key (the secret key) to encrypt and decrypt a message, and asymmetric cryptography use one key (the public key) to encrypt a message and another key (the private key) to decrypt it.

Example 1. Alice can use a shared symmetric key k to encrypt a letter and send it to *Bob*. *Bob* can use the same key to decrypt it. If *Alice* uses her private key $K^{-1}(Alice)$ to sign the letter, *Bob* uses $K_p(Alice)$ to open it and knows it was really signed by *Alice*.

In addition to the cryptographic strength of cryptography algorithms, the security of e-commerce systems also depends on the reliability of security protocols that cover the full range of administrative and technical measures that need to fulfill corporate security objectives. The following two fundamental elements are often emphasized in the analysis of security protocols:

- replay of messages that presents a message from different context is used into the intended context to fool the honest participant into thinking they have successfully completed the protocol run; and
- correct correlation of cryptography keys with specified principals.

The timestamp has been proved to be an efficient way to ensure the freshness of secure messages in [15]. It plays an important role in preventing the replays of former transmitted secure messages.

Definition 1. *Let T be a timestamp attached to message m. If* $|Clock - T| < \Delta t_1 + \Delta t_2$ *regarding received messages or* $T < T_{expiration}$ *regarding generated messages then m is fresh; otherwise m is viewed as a replay.*

In this definition, *Clock* is the local time, Δt_1 is an interval representing the normal discrepancy between the server's clock and the local clock, and Δt_2 is an interval representing the expected network delay time [15]. In addition, $T_{expiration}$ denotes the expiration time, which is designated to messages when they are generated.

Example 2. If the attached timestamp T is *30 Nov 2004 18:35:20 +1000*, and the specified *expiration time* $T_{expiration}$ is *30 Nov 2004 18:34:28 +1000*, we can say the message is not fresh for $T > T_{expiration}$.

Authentication is further strengthened by the use of certificates that are digitally signed by recognized certificate authorities (*CA*) and used to specify the affiliation between public keys and principals. Figure 1 describes a public-key infrastructure (PKI) [11] that provides the issue, management and use of public keys and certificates for authentication, privacy and other security properties. A certificate is verified following the trust tree to a known trusted party.

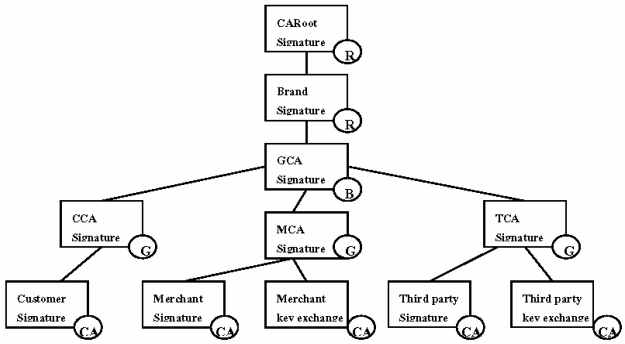

Fig. 1. PKI tree of Trust.

In e-commerce systems, the authentication may occur in different transactions and places. According to the source of transaction data, secure messages are classified into three categories:

- D_S represents the transaction database where messages are generated and sent to the receiver;
- D_R represents the transaction database where messages are received and authenticated; and
- $D_T = \{D_{T1}, \ldots, D_{Tm}\}$ $(1 \le m)$ represents all relevant transaction databases from the third party such as financial institutions and certificate issuers.

The transmitted secure message stored in above transaction databases is actually the data set used for data mining. However, unlike the traditional market basket data, the inherent property of secure messages such as freshness requires us to make additional authentication of their validity before going to data mining. Correspondingly, a supporting relationship $\models$ is used to authenticate secure messages.

(1) $\models_D \alpha$ iff $\alpha \in D$, and α is *fresh*;

(2) $\models_D K_p(X)$ iff $K_p(X) \in D$, and $K_p(X)$ is authenticated to be valid.

Here α indicates a secure message, $D \in \{D_S, D_R, D_T\}$, and X *re*esents the principal who sent messages to *receivers*.

However, there may exist conflicting supports among different transaction databases. For example, $\models_{Ds} \alpha$ and $\models_{DR} \neg\alpha$ present discrepant supports between D_S and D_R. This phenomenon in fact indicates the possibility of potential security flaws in security protocols. As mentioned above, each transaction can be viewed as a association rule containing correlated secure messages. Therefore, the reliability of a transaction can be determined in virtue of the support and confidence of the corresponding association rule.

4. ANALYZING SECURITY PROTOCOLS USING ASSOCIATION RULE MINING

This section extends the original association rule mining to analyze inconsistent secure messages and identify the potential correlations between secure messages.

4.1 The Basics of Association Rule Mining

Let $I = \{i_1, \ldots, i_n\}$ be a set of items and D be a collection of transactions, called transaction database. Each transaction $T \in D$ consists of a collection of items. Let $A \subseteq I$ be an itemset. We can say that a transaction T contains A in case $A \subseteq T$. An itemset A in a transaction database D has a *support*, denoted as *supp*(A). Hence we have

$$supp(A) = |T_A| / |D| \%$$

where T_A presents transactions in D, which contain the itemset A.

An itemset A in D is called a *frequent itemset* if its support is equal to, or greater than, the given frequency threshold λ (*minimum support*) by user or experts, namely $supp(A) \ge \lambda$. An association rule is an implication of the form, $A \rightarrow B$, where A and B are frequent itemsets, and $A \cap B = \varnothing$. For each association rule, we can use the support-confidence framwork [13] to measure it. A rule $A \rightarrow B$ is valid if

(1) $supp(A \cup B) \ge minsupp$
(2) $conf\ (A \rightarrow B) = supp(A \cup B) / supp(A) \ge minconf$.

where *minsupp* and *minconf* are designated by users or experts. Association rule provides a simple but efficient form of rule patterns for data mining. It is briefly summarized as follows:

- First, we need to extract all frequent itemsets from the transaction database.
- Second, we determine valid rules from discovered frequent itemsets according to the support and confidence of this rule.

Additionally, according to Piatetsky-Shapiro's argument, a rule $A \rightarrow B$ can be extracted as a valid rule of interest if it satisfies minimum interest *mininterest*. It will be discussed

in Section 4.3.

4.2 Data Preparation

Traditional association rule mining does not take into account the missing and inconsistent data. Unlike the market basket data, secure messages have some properties as mentioned above. Additionally, the local support from individual transaction databases needs to be integrated to obtain a global support from whole databases. Therefore, it is necessary for us to extend the original association rule mining to analyze inconsistent secure messages.

Secure messages in transaction databases are regarded as items. Suppose $D = \{D_1, \ldots, D_k\}$ is a set of transaction databases. Each transaction database consists of a collection of secure messages. Let $I = \{x \mid x \in D_i, 1 \le i \le k\}$ be a set of items. $A \subseteq I$ and $B \subseteq I$ are itemsets. An association rule is an implication of the form $A \rightarrow B$ where $A \cap B = \varnothing$. The rule $A \rightarrow B$ has *support*, s in the set of transaction databases if s% of transaction databases contains $A \cup B$. The association rule has *confidence*, c in the set of transaction databases if c% of transaction databases containing A contains $A \cup B$.

Example 3. To register an account, the cardholder needs to fill out the registration form from *CA* with information such as the *cardholder's name*, *date of birth*, *expiration date* and *account billing address*. Let I = {*cardholder's name*, *date of birth*, *expiration date*, *account billing address*} be the set of items. Hence we can say {*cardholder's name*, *date of birth*} and {*expiration date*, *account billing address*} are itemsets as usual.

Suppose that a transaction $T = D_1 \cup D_2 \cup \ldots \cup D_n$ comprises n transaction databases, in which D_i $(1 \le i \le n)$ may be sender, receiver or the third party. If D_i contains φ and believes its freshness or validity when φ is a public key, itemset φ has local support from D_i, namely $\models_{D_i} \varphi$. The global support of itemset φ actually integrates the local support from all transaction databases in T. Additionally, the missing item is assigned *null* value, and the inconsistent item is denoted by inserting $\neg$ symbol. For simplicity, all itemsets in the rest of this article are assumed to be valid (freshness of messages or validity of keys) in the corresponding transaction databases.

Definition 2. *Let D_i be a transaction database and I be a set of items. Suppose $\phi = \{m_1, m_2, \ldots, m_k\} \subseteq I$ is an itemset. Then,*

$$\models_{D_i} \phi \ \ \textit{iff} \ \ \models_{D_i} \forall\, m_i \in \phi$$

Example 4. Let *ID* be identity number and *AC* be account number. Table 1 indicates secure messages derived from different transaction databases. As to the item *account number*, it is contained in D_1, D_2 and D_4 but missed in D_3. Thus, $\models_{D_1} AC$, $\models_{D_2} AC$, $\models_{D_4} AC$ and $\models_{D_3} \neg AC$. We observe that item *AC* and *ID* are missing in D_3 and D_4 respectively. Besides, D_3 contains an inconsistent item $\neg ID$. Consequently, we have $\models_{D_1} AC \cup ID$, $\models_{D_2} AC \cup ID$ due to $\models_{D_1} ID$ and $\models_{D_2} ID$, whereas, $\not\models_{D_3} AC \cup ID$ and $\not\models_{D_4} AC \cup ID$ due to $\not\models_{D_3} AC$, $\models_{D_3} \neg ID$ and $\not\models_{D_4} ID$.

Table 1: Secure Message Sources

Database	*Account number*	*Identifier*	*Key*
D_1	*AC*	*ID*	$K^{-1}(X)$
D_2	*AC*	*ID*	$K_p(X)$
D_3	*null*	$\neg ID$	$K_p(X)$
D_4	*AC*	*null*	$\neg K_p(X)$

From the observation, secure messages are different from traditional market basket data in terms of their security properties. Although we assume that secure messages are fresh and there is correct association between the public key and principal, it cannot exclude the possibility of inconsistency, including missing and inconsistent values, between secure messages due to the possible *message loss*, *communication block* and *broken cipher*. Moreover, the inconsistency has an effect on measuring the reliability of transactions as mentioned above.

In particular, keys are unlike ordinary secure messages. For example, in public-key cryptography, each principal has a pairs of related keys: a public key and a private key. Usually, the public key is known to everyone but the private key is known only by the recipient of the message. Nobody can forge his/her signature without knowledge of his/her private key. Therefore, for each public/private key pairing, *private key* and *public key* are viewed as identical items when computing the corresponding support and confidence. The traditional association rule mining should be extended to cope with the public/private key pairs and the inconsistency between secure messages.

(1) *Missing Item*. If an itemset misses some items, it will be viewed as a different itemset from the original one.

For example, in Table 1, itemsets {*AC*, *ID*} in D_1 and D_2, and {*AC*, *null*} in D_4 are viewed as different itemsets. The item *ID* is missed in D_4.

(2) *Tampered Item*. If some items are tampered in an itemset, the itemset is regarded as a different itemset from the original one.

For example, in Table 1, the item $\neg ID$ in D_3 is a tampered item in contrast to item *ID* in D_1 and D_2. And thereby, itemset {*null*, $\neg ID$} in D_3 and itemset {*AC*, $\neg ID$} in D_1 and D_2 are viewed as different itemsets.

(3) *Itemsets with Public/Private Key Pairs*. For any two itemsets, an itemset contains *private key* and the other contains *public key*. If the public key and private key are correctly matched pairs and the other items are identical, they are regarded as identical itemsets.

For example, in Table 1, the public key $K_p(X)$ in D_2 and D_3 correctly match with the private key $K^{-1}(X)$ in D_1.

Nevertheless, $\neg K_p(X)$ in D_4 is a tampered public key. Therefore, itemset $\{AC, K^{-1}(X)\}$ in D_1 and itemset $\{AC, K_p(X)\}$ in D_2 are viewed as identical itemsets, whereas, itemset $\{AC, K_p(X)\}$ in D_2 and itemset $\{AC, \neg K_p(X)\}$ in D_4 are viewed as inconsistent itemsets owing to the inconsistency between $K_p(X)$ and $\neg K_p(X)$.

4.3 Identifying Association Rules of Interest

During a transaction, if secure messages in an itemset are lost or tampered or public/private key pairs are not correctly matched it will result in the decrease of the number of occurrences of this itemset. Consequently, the degree of support and confidence on the relevant association rule will decrease. Therefore, if the rule is not of interest, it is reasonable to say the corresponding transaction of this rule is insecure.

In order to compute the support of itemsets, we need to extend the original association rule mining.

Definition 3. *Suppose D_i, $1 \le i \le$ n, is a transaction database in transaction T. Let φ be an itemset and $\varphi(D_i) = \{D_i \text{ in } T \mid D_i \text{ contains } \varphi\}$. Let the global support of φ be supp(φ).*

$$supp(\varphi) = \sum_{i=1}^{n} |\varphi(D_i)| / |T| \qquad (1)$$

Example 5. In Table 1, item *AC* is contained in D_1, D_2 and D_4 but not in D_3. Consequently, we have $|AC(D_1)| = 1$, $|AC(D_2)| = 1$, $|AC(D_4)| = 1$ and $|AC(D_3)| = 0$. In the same way, we have $|\{AC \cup ID\}(D_1)| = 1$, $|\{AC \cup ID\}(D_2)| = 1$, $|\{AC \cup ID\}(D_4)| = 0$ and $|\{AC \cup ID\}(D_3)| = 0$ for *AC* and *ID* are missing in D_3 and D_4 respectively. Therefore, $supp(AC) = 3/4 = 0.75$ and $supp(AC \cup ID) = 2/4 = 0.5$.

An association rule is the implication χ: $A \to B$, where $A \cap B = \varnothing$. Therefore, the confidence of the rule $A \to B$ is

$$conf(A \to B) = \frac{\sum_{i=1}^{n} |\chi(D_i)| / |T|}{supp(A)} \qquad (2)$$

Example 6. In an online booking, the user needs to fill out a form with *credit card number*, key *k*, *amount* and *address*. They are encrypted and sent to merchant *Y*. Initially, *Y* needs to authenticate the received message via the third party such as financial institutions. Suppose $D_1 = \{card\_number, k, amount, address\}$, $D_2 = \{card\_number, k, amount\}$ and $D_3 = \{card\_number, k, address\}$ are transaction databases. Let *minsupp* = 50% and *minconf* = 60%. We have $supp(\{card\_number, k, amount\}) = 2/3 > 0.5$, $supp(\{card\_number, k\}) = 1 > 0.5$ and $conf(\{card\_number, k\}) \to \{amount\}) = 2/3 >$ o.6. $\{card\_number, k\} \to \{amount\}$ can be extracted as a valid rule.

Nevertheless, according to Piatetsky-Shapiro's argument, a rule $X \to Y$ can be extracted as a valid rule of interest if

- $|\frac{supp(X \cup Y)}{supp(X) * supp(Y)} - 1| \ge mininterest$,
- $supp(X \cup Y) \ge minsupp$, and
- $conf(X \to Y) \ge minconf$.

Let *mininterest* = 0.07. $\{card\_number, k\} \to \{amount\}$ is not a rule of interest due to $|\frac{supp(card\_number, k, amount)}{supp(card\_number, k) * supp(amount)} - 1| = 0 < mininterest$.

The derived rule also indicates that this transaction is unreliable and insecure and the number of lost and tampered messages is high. On the contrary, if the inconsistency between secure messages is low, the support and confidence of association rules will turn to be high. Hence we have

1. *belief* $(A \to B)$ = "*secure*", if $supp(A \cup B) \ge minsupp$, $conf(A \to B) \ge minconf$ and $|\frac{supp(A \cup B)}{supp(A) * supp(B)} - 1| \ge mininterest$;
2. *belief* $(A \to B)$ = "*insecure*", otherwise.

The belief in the association rule $A \to B$ determines the reliability of transaction *T*. For the rule that satisfies the above conditions, the corresponding transaction is believed to be secure; if at least one condition is not satisfied, the transaction will be treated as being insecure. The parameters including *minsupp*, *minconf* and *mininterest* can be regulated to achieve different levels of security. In general, the lager their values are, the higher the degree of security is. Using association rule mining, the belief in transaction can be transferred to the belief in corresponding association rules. Therefore, it provides a novel and efficient way to analyze security protocols.

5. ALGORITHMS AND EXPERIMENTS

5.1 Algorithms

Identifying frequent itemsets is one of the key issues in discovering association rules. There have been a number of algorithms developed for mining frequent itemsets in databases. Among them, *Apriori* is a widely-used algorithm for extracting frequent itemsets. In [2], it developed a new algorithm to identify frequent itemsets from transaction data, in which Piatetsky-Shapiro's argument is taken into account. However, the secure messages are different from traditional data for their security properties. Hence this algorithm is inappropriate for mining inconsistent secure messages. In this article, we design an algorithm to identify frequent itemsets from secure messages based on Frequent Patterns (FP) tree algorithm [9], in which the properties of secure messages are taken into account.

Firstly, we need to generate all frequent items, which are supported globally by whole transaction databases rather than supported locally by a local database. The *missing message*, *tampered message* and *private/public key pairs* mentioned above are considered to correctly compute the *support* and *confidence* of items. We assume the secure messages are fresh and generated and sent by the sender and received and seen by whom it claims to be, and the keys have been authenticated to be valid and issued by the correct authorities.

Algorithm 5.1 *Mining Inconsistent Secure Messages*

begin

Input: D : data set; minsupp: minimum support; minconf: minimum confidence; mininterest: minimum interest;

Output: frequent itemsets;

//Select the candidate transaction database.

let $D_c \leftarrow$ candidate transaction database with private key;

//Convert inconsistent secure messages.

//$1 \le i \le n, \le j \le m$, in which n and m denotes the number

//and cardinality of databases respectively.

forall $D_i \in D - D_c$ **do**

forall $I_{ij} \in D_i$ **do**

if item $I_{ij} \neq null$ **then**

if I_{ij} is a *key* **then**

{

if $I_{ij} = I_{cj}$ or matches with I_{cj} in D_c **then**

$I_{ij} = K(D_i)$;

$I_{cj} = K(D_i)$;

}

end

end

//Construct FP-tree.

forall $I_{ij} \in D$ **do**

$F \leftarrow \{\}$;

$F \leftarrow F \cup$ frequent items of D_i;

sort items in F according to the frequency;

add items to FP-tree;

end

//Mining frequent patterns from FP-tree.

forall $node_i \in FP\text{-}tree$ **do**

process one node each time from bottom to the root;

output frequent itemsets;

end

end

This algorithm is used to extract frequent itemstes from transaction databases. Before extracting frequent itemsets, the public/private key pairs in transaction database need to be converted into a common assumed key $K(X)$ if it equals the private key of candidate database or matches it. FP-tree algorithm in [9] is used to mine frequent itemsets. It consists of two steps: (1) constructing FP-tree; and (2) mining frequent itemsets from FP-tree. The output only contains frequent itemsets so it is effcient than traditional *Apriori* algorithm. In particular, the message losing, message tampering and private/public key pairs are taken into account to correctly compute the frequency of itemsets.

Preprocessing of secure messages needs $O(nm)$ time. As mentioned in [9], the search time of inserting a transaction *Trans* into the FP-tree is $O(|freq(Trans)|)$, where $freq(Trans)$ is the set of frequent items in *Trans*. The worst case is $|freq(Trans)|$ equals m. Thus, FP-tree construction needs $O(nm)$ time. The mining of frequent patterns phase is $O(m)$ time. Hence, our algorithm has the worst-case $O(nm + m)$.

5.2 EXPERIMENTS

In this section, we study the efficiency and performance of our algorithm presented in Section 5.1 by verifying our algorithm against an assumed dataset. The dataset are derived from a merchant's payment authorization process in SET protocol. The merchant authorizes the transaction during the processing of an order from a cardholder. *Third parties* here include the financial institutions and the processor of the transactions. SET aims at providing confidentiality of information, ensuring payment integrity, and authenticating both merchants and payment gateway. For simplicity, authorization request and authorization request processing are considered only.

At the beginning, the merchant needs to generate an authorization request and send it to payment gateway. When the payment gateway receives the authorization request, it verifies transaction identifier, and forwards an authorization request to the issuer through a payment system. The transactions details are listed below:

- *Authorization Request*. In order to authorize a transaction, the merchant generates an authorization request *AuthReq*, which includes the amount to be authorized, the transaction identifier from the *OI*. It is then combined with the transaction identifiers *TransIDs* and the hashing of the order information *OI*. *M* signs *AuthReq* and encrypts it with a randomly generated symmetric key k_2. This key is then encrypted with the gateway's public key $K_p(P)$. Finally, the merchant transmits the authorization request to the payment gateway *P*.
- *Processing Authorization Request*. The gateway decrypts the symmetric key k_2, and then decrypts authorization request using k_2. It uses the merchant public signature key $K_p(P)$ to verify the merchant digital signature. The gateway also verifies the merchant signature certificate and cardholder signature certificate to ensure that they have not expired. Then the gateway decrypts k_1 and cardholder account information with gateway private key $K^{-1}(P)$, and decrypts the *PI* using k_1. The gateway also verifies the transaction identifier received from the merchant by comparing it with the identifier in cardholder payment request.

The extracted secure messages from above transaction databases include *OI*, *PI*, *AuthReq*, k_2, $K^{-1}(P)$, $K_p(P)$, $K_p(M)$, $K^{-1}(M)$, and k_1. They are stored in the following transaction databases.

- $D_M = \{OI\ PI, AuthReq, K^{-1}(M), K_p(P), k_2\}$
- $D_P = \{OI, PI, AuthReq, K_p(M), K^{-1}(P), k_2\}$
- $D_T = \{OI, PI, AuthReq, K_p(M), K_p(P), k_2\}$

where key k_1 is not included in the above transaction databases for it is encrypted by $K_p(P)$ and unknown to the merchant. Initially, each item has a value corresponding to the record in transaction databases. On the other hand, transaction databases contain some inconsistent items such as missing items, tampered items and unmatched private/public key pairs. Before going to mine frequent itemsets, we need to convert the original item into operable data. Table 2 presents the items of transaction databases.

Table 2: Payment Authorization in SET

	OI	*PI*	*K*(*M*)	*K*(*P*)	k_2	*AuthReq*
D_M	*OI*	¬*PI*	$K^{-1}(M)$	$K_p(P)$	k_2	*AuthReq*
D_P	*OI*	*PI*	$K_p(M)$	$K^{-1}(P)$	k_2	¬*AuthReq*
D_{T1}	*OI*	*PI*	$K_p(M)$	$K_p(P)$	k_2	*AuthReq*
D_{T2}	¬*OI*	*PI*	$K_p(M)$	$K_p(P)$	¬k_2	*AuthReq*

Suppose that the minimum support *minsupp* = 50%, minimum confidence *minconf* = 60%, and minimum interest *mininterest* = 0.07. Let *i-itemset* be the frequent itemset that contains *i* items. Hence we can get *frequent items* = $\{OI^3, PI^3, K(M)^4, K(P)^4, k_2^3, AuthReq^3\}$, in which the superscript presents the frequency of items. For example, *supp*(*OI*) = (1 + 1 + 1)/4 = 75% > *minsupp* and *supp*(*PI*) = (1 + 1 +1)/4 = 75% > *minsupp*.

Based on *frequent items*, we can find out all *frequent i-itemsets*($i \geq 2$) using the algorithm mentioned in Section 5.1 where no candidate itemset needs to be generated. The second step is to extract all association rules from the derived frequent itemsets. Some derived association rules are listed below. The rule corresponding to a transaction can be used to verify its reliability. On the other hand, additional rules can be used to detect potential correlations between secure messages, which possibly imply potential security flaws. For simplicity, we present the findings of association rules in relation to *5-itemsets* only.

(1) $supp(OI \cup PI \cup K(M) \cup K(P) \cup k_2) = 50\% \geq minsupp$
(2) $supp(OI \cup PI \cup K(M) \cup K(P) \cup AuthReq) = 25\% < minsupp$
(3) $supp(OI\ [\ PI \cup K(M) \cup k_2 \cup AuthReq) = 25\% < minsupp$
(4) $supp(OI\ [\ PI \cup K(P) \cup k_2 \cup AuthReq) = 25\% < minsupp$
(5) $supp(OI \cup K(M) \cup K(P) \cup k_2 \cup AuthReq) = 50\% < minsupp$
(6) $supp(PI \cup K(M) \cup K(P) \cup k_2 \cup AuthReq) = 25\% < minsupp$

From the observation, only (1) and (5) are frequent itemsets, from which several rules can be derived. For example, $supp(\{OI, K(M), K(P), k_2\}) = 3/4 = 75\% > minsupp$, $conf(\{OI, K(M), K(P), k_2\} \rightarrow AuthReq) = 67\% > minconf$ and $\left|\frac{supp(OI, K(M), K(P), k_2, AuthReq)}{supp(OI, K(M), K(P), k_2) * supp(AuthReq)} - 1\right| = 0.11 > mininterest$. Therefore, $\{OI, K(M), K(P), k_2\} \rightarrow AuthReq$ can be extracted as a valid rule of interest. This rule indicates that *AuthReq* is believed to be secure for *OI*, *K*(*M*), *K*(*P*), and k_2 are reliable. This rule in fact corresponds to a transaction in payment authorization.

On the other hand, some discovered rules can imply potential correlations between secure messages. For example, $supp(\{OI, PI\}) = 0.5 > minsupp$, $supp(K(M), K(P), k_2) = 0.75 > minsupp$, $conf(\{K(M), K(P), k_2\} \rightarrow \{OI, PI\}) = 0.67 > minconf$ and $\left|\frac{supp(OI, PI, K(M), K(P), k_2)}{supp(K(M), K(P), k_2) * supp(OI, PI)} - 1\right| = 0.33 > mininterest$. Therefore, $\{K(M), K(P), k_2\} \rightarrow \{OI, PI\}$ is a valid rule of interest. This rule indicates the correlation between $\{K(M), K(P), k_2\}$ and $\{OI, PI\}$. In other words, the reliability of $\{K(M), K(P), k_2\}$ has an effect on $\{OI, PI\}$. Therefore, if *PI* and *OI* are found to be inconsistent in transaction databases, *K*(*M*), *K*(*P*) and k_2 are suspectable to be attacked.

As to (2), (3), (4) and (6), they are not frequent itemsets. Hence no valid rule can be derived from them. In other words, there are latent security problems in regard to these itemsets.

6. CONCLUSIONS

Traditional approaches used to analyze security protocols require users to predetermine authentication goals. They assume that secure messages are secure and reliable. However, inconsistent secure messages have been a big challenge to the reliability of electronic transactions. This paper proposes a novel method to analyze security protocols using association rule mining. The properties of secure messages are included to examine secure messages before data mining. In particular, it takes into account the missing and inconsistent secure messages. The discovered association rule is able to not only verify the reliability of corresponding transactions but also discover potential association between secure messages. We demonstrate our methods by using the proposed algorithms and conducting the experiments.

ACKNOWLEDGEMENT

The authors would like to thank the instructive and constructive comments provided by the anonymous reviewers and editor, which help the authors improve this paper significantly.

REFERENCES

1. Burrows M., Abadi M., Needham R., "A logic for Authentication", *ACM Transactions on Computer Systems*, 8(1), pp 18-36, February 1990.
2. Chengqi Zhang, and Shichao Zhang., "*Association Rule Mining: Models and Algorithms"*, LNAI 2307, Springer-Verlag, Germany, 2002.
3. Qingfeng Chen and Shichao Zhang, "Dealing with Inconsistent Secure Messages", *Proceeding of 8th Pacific Rim International Conference on Artificial Intelligence*, LNCS 3157, pp 33-42, Auckland, New Zealand, August 2004.
4. Needham R. and Schroeder M., "Using Encryption for Authentication in Large Networks of Computers", *Comm. of the ACM*, 21(12), pp 993-999, Dec 1978.
5. Lowe G., "Breaking and fixing the Needham-Schroeder public-key protocol using FDR", *In Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, vol 1055, pp 147-166. Springer-Verlag, Berlin Germany, 1996.
6. Dolev D. and Yao A., "On the Security of Public Key Protocols", *IEEE Transaction on Information Theory*, 29(2), pp 198-208, 1983.
7. Gritzalis S., "Security Protocols over open networks and distributed systems: Formal methods for their Analysis", Design, and Verification, *Computer Communications*, 22(8), pp 695-707, May 1999.
8. Heintze N., Tygar J., Wing J., and Wong H., "Model Checking Electronic Commerce Protocols", *Proceedings of the 2nd USENIX Workshop on Electronic Commerce*, pp 147-164, Oakland, California November, 1996.
9. Han J., Pei J. and Yin Y., "Mining frequent patterns without candidate generation", *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, USA, pp 1-12, 2000.
10. http://wp.netscape.com/eng/ssl3/ssl-toc.html.
11. SET Secure Electronic Transaction Specification: *A Programmers Guide*, http://www.setco.org/.
12. ITU-T Recomendation X.509 (1993) Also ISO/EC 9594-8 *Information Technology - Open Systems Interconnection - The Directory*: Authentication Framework, 1995.
13. Agrawal R., Imielinski T., and Swami A. "Database mining: A performance perspective". *IEEE Transaction. Knowledge and Data Eng*., 5(6), pp 914-925, 1993.
14. Menzies A.J, von Oorschot P.C, and Vanstone S.A., *Handbook of Applied Cryptography*, CRC Press, New York, 1996.
15. Denning D. and Sacco G., "Timestamp in Key Distribution Protocols", *Communications of ACM*, 24(8), pp 533-536, August 1981.
16. Srikant R. and Agrawal R., "Mining quantative association rules in large relational tables", *Proceeding of SIGMOD'96*, pp1-12, Canada, 1996.
17. Liberatore P. and Schaerf M., "Arbitration (or How to Merge Knowledge Bases)", *IEEE Transaction on Knowledge and Data Engineering*, 10(1), pp 76-90, 1998.
18. Lin J., and Mendelzon A.O., "Knowledge base merging by majority", *In Dynamic Worlds: From the Frame Problem to Knowledge Management*, Kluwer, 1999.
19. Vaidya J. and Clifton C., "Privacy preserving association rule mining in vertically partitioned data", *Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 639-644, Edmonton, Alberta, Canada, July 2002.
20. Stanley R. M. Oliveira and Osmar R. Zaiane, "Privacy preserving frequent itemset mining", *Proceeding of IEEE ICDM Workshop on Privacy, Security and Data Mining*, vol. 14, pp 43-54, Maebashi , Japan, 2002.

Professor Chengqi Zhang received a BSc degree from Fudan University, Shanghai, an MSc degree from Jilin University, Changchun, a PhD degree from the University of Queensland, Brisbane, and a DSc degree (Higher Doctorate) degree from Deakin University, Geelong, all in Computer Science.

He has been a Lecturer at the University of New England in Australia from January 1990 and was promoted to Senior Lecturer in January 1994, and then to Associate Professor in January 1998. He moved to Deakin University as an Associate Professor from January 1999 and moved to the University of Technology, Sydney as a Research Professor from January 2002. He is now a member of Smart eBusiness Systems Laboratory.

His research interests include "Multi-Agent Systems, Data Mining, Information Retrieval on Internet, Knowledge-based Decision Support Systems, Distributed Artificial Intelligence, Cooperation under Uncertainty, and Distributed Expert Systems". He is an author or co-author of more than 200 research papers. Some of these papers have been published in renowned international journals which include "*Artificial Intelligence*" and "*IEEE Transactions*". He is also a co-author of three monographs and a co-editor of nine books.

Associate Professor (Reader) Yi-Ping Phoebe Chen joined the Deakin University in 2003. She is the Director of Research of School of Information Technology, a Chief Investigator of ARC Centre in Bioinformatics and also head of Multimedia Stream. Dr. Chen received her BInfTech degree with First Class Honours and PhD in Computer Science from the University of Queensland. From 1999 to 2003, she worked as an Associate Lecturer/Lecturer/Senior Lecturer in Queensland University of Technology. She is steering committee chair of Asia-Pacific Bioinformatics Conference (founder) and Multimedia Modelling. Her research interests include bioinformatics, multimedia databases and technology, visual query, web information systems, machine learning and data mining. Dr. Chen has published more than 80 refereed publications. She is an associate editor for a number of journals.

Dr. Shichao Zhang is a principal research fellow in the Faculty of Information Technology at the University of Technology Sydney, and a professor at Guangxi Normal University. He received his PhD degree in computer science from Deakin University, Australia. His research interests include data analysis and smart pattern discovery. He has published about 30 international journal papers (including 5 in IEEE/ACM Transactions, 2 in Information Systems, 6 in IEEE magazines) and over 30 international conference papers (including 2 ICML papers and 3 FUZZ-IEEE/AAMAS papers). He has won 4 China NSFC/863 grants, 2 Australian large ARC grants and 2 Australian small ARC grants. He is a senior member of the IEEE, a member of the ACM, and serving as an associate editor for Knowledge and Information Systems and The IEEE Intelligent Informatics Bulletin.

Dr. Qingfeng Chen received a BSc and MSc degree from Guangxi Normal University, China, and a PhD degree from the University of Technology Sydney in Computer Science.

He is now a postdoctoral research fellow at School of Information Technology, Deakin University, Australia. He has published 16 refereed papers, including International Journal of Data Mining and Knowledge Discovery and a number of first class journals of China. One of his papers "Dealing with Inconsistent Secure Message" won the Best Paper Award in the 8th Pacific Rim International Conference on Artificial Intelligence, New Zealand, 2004 and was invited to publish in the journal of *Artificial Intelligence*.