Security and Privacy for Data Mining of RFID-Enabled Product Supply Chains

Xiaoming Yao, Wencai Du, Xiaoyi Zhou
College of Information Science and Technology
Hainan University
Haikou, China
xiaomingyao@163.com

Jixin Ma
School of Computing and Mathematical Science
University of Greenwich
London, UK
J.ma@gre.ac.uk

***Abstract*—The e-Pedigree used for verifying the authenticity of the products in RFID-enabled product supply chains plays a very important role in product anti-counterfeiting and risk management, but it is also vulnerable to malicious attacks and privacy leakage. While the radio frequency identification (RFID) technology bears merits such as automatic wireless identification without direct eye-sight contact, its security has been one of the main concerns in recent researches such as tag data tampering and cloning. Moreover, privacy leakage of the partners along the supply chains may lead to complete compromise of the whole system, and in consequence all authenticated products may be replaced by the faked ones! Quite different from other conventional databases, datasets in supply chain scenarios are temporally correlated, and every party of the system can only be semi-trusted. In this paper, a system that incorporates merits of both the secure multi-party computing and differential privacy is proposed to address the security and privacy issues, focusing on the vulnerability analysis of the data mining with distributed EPCIS datasets of e-pedigree having temporal relations from multiple range and aggregate queries in typical supply chain scenarios and the related algorithms. Theoretical analysis shows that our proposed system meets perfectly our preset design goals, while some of the other problems leave for future research.**

Keywords—supply chain; e-pedigree; multi-party security; anti-counterfeiting; differential privacy

I. INTRODUCTION

Counterfeiting has been growing greatly in the decade and penetrating into various industries such as food, drugs, high-tech products and luxury goods by altering or misrepresenting an individual product for economic gain, shaking the public confidence in the ability of manufacturers and governments to assure the safety of food and other products [1-3]. With the merits of the radio frequency identification (RFID) technology such as automatic wireless identification without direct eye-sight contact, the e-pedigree of RFID-enabled supply chain has been taken as the promising tool of anti-counterfeiting and brand protection by tracking and/or tracing the historical movement of the specific product. It can be authenticated at each node of the supply chain for data consistence with those pre-stored in the EPC information system (EPCIS) repositories [4-6], according to the architecture specified by EPCglobal [7], an organization dedicated to promote the global standardization of the electronic product code (EPC) that is used to uniquely identify single products.

By EPCglobal standard of pedigree 1.0 released in 2007 [8], a pedigree is a certified record that contains information about each distribution of the product to be protected. It records the sale of an item by a manufacturer, any acquisitions and sales by authenticated wholesalers or distributors, and final sale to a customer who buys this product. The pedigree contains information about the product, transaction, authenticated distributor, the recipients, and related signatures.

The typical scene of e-pedigree processing can be simply modeled as the iterative two-party protocol between two sides from the manufacturers to the customers: one is the sender who distributes the product, while the other is the product recipient. The typical UML sequence diagram of e-pedigree flow can be illustrated in Fig.1.

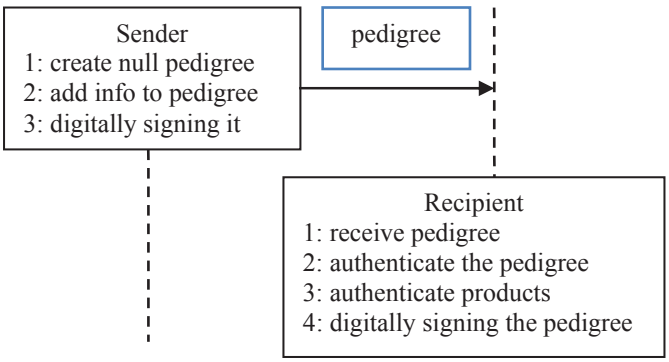

Fig. 1. UML sequence of e-pedigree flow: note that the product is manually verified with authentication of the respective transactions

After the recipient signed the pedigree, the ownership is then transferred, and its identity will be changed into the sender, and next shipping process starts.

The typical implementation of this system assumes the following requirements: each node of the supply chain maintains and updates its local EPCIS with EPC data and the related events connected to their e-pedigrees, which can also provide some more advanced services of range and aggregate queries via internet for further inference or sales promotion support under the EPCglobal networked framework. Even if all nodes of the supply chain are authenticated and trustworthy according to the law, it has been shown and reported in many literatures that such queries from distributed sources with

outputs of "true answers" or even some of the intermediate data mining reports may result in privacy leakage of individual records [9]. Since any product has to move from one to another before it arrives at the hand of the final buyer, its relevant historical records must be shared by those intermediate nodes. And at least partial information of one node are always shared by its several neighboring nodes.

The challenge is obvious: if privacy leakage due to data mining of the information service released datasets occurs at any time, the attacker will get the information of the specific product, and further compromise the X.509 based signature, and the e-pedigree will then be compromised. Besides, the RFID tags also suffer many security vulnerabilities.

Quite different from other conventional databases, datasets in supply chain scenarios are temporally correlated, and every party of the system can only be semi-trusted. To address this problem in this paper, a system that incorporates merits of both the secure multi-party computing [10] and differential privacy [11] is proposed, focusing on the vulnerability analysis of the data mining with distributed EPCIS datasets of e-pedigree having temporal relations from multiple range and aggregate queries in typical supply chain scenarios and the related algorithms.

The paper is organized as follows: Next the related work and the background is discussed, including the potential vulnerabilities and threats, the multiparty security computing protocol and differential privacy. Then our problems are carefully formulated in section III. Our proposed system is presented and theoretically analyzed along with key issues of its implementation in section IV. It is concluded in section V.

II. Related Work and Background

Typical system architecture of a product supply chain can be illustrated in Fig.2.

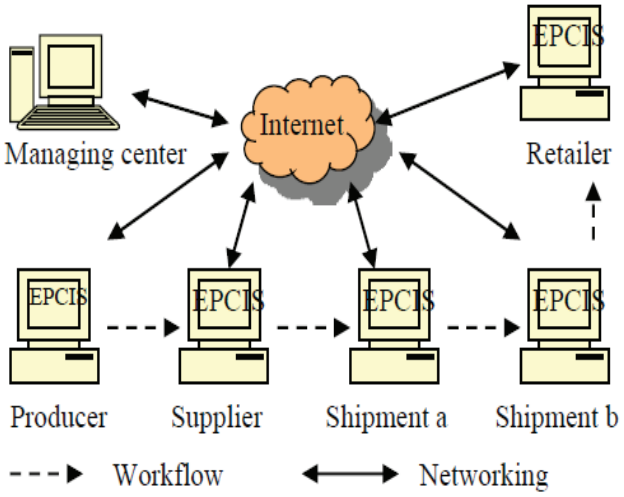

Fig. 2. Typical system architecture of a product supply chain

The products are first manufactured by specific producers and attached with respective RFID tags whose data will be read out and stored in the EPCIS databases. As mentioned above, the e-pedigree will be created, filled with its respective data, signed and stored simultaneously into the EPCIS databases. Then along with the products entering into another node of the supply chain, a supplier, or a distributor during the shipment, the receiver of the products after the RFID tags data of the products automatically read and sanitized and finally stored into the EPCIS databases, will manually examine the products and authenticate the e-pedigree as shown in Fig.1. Then the e-pedigree will be authenticated and signed again to guarantee the consistence of the actual objects with their respective data. This process repeats until the products arrive at the hands of the buyer, when the e-pedigree received can be acted as the certificate of the authenticity.

A. *E-pedigree verification process with a centrilised TTP*

Based on the above-stated background, the typical process for e-pedigree verification via a centralized TTP can be shown by its UML sequence diagram of message flow in Fig.3.

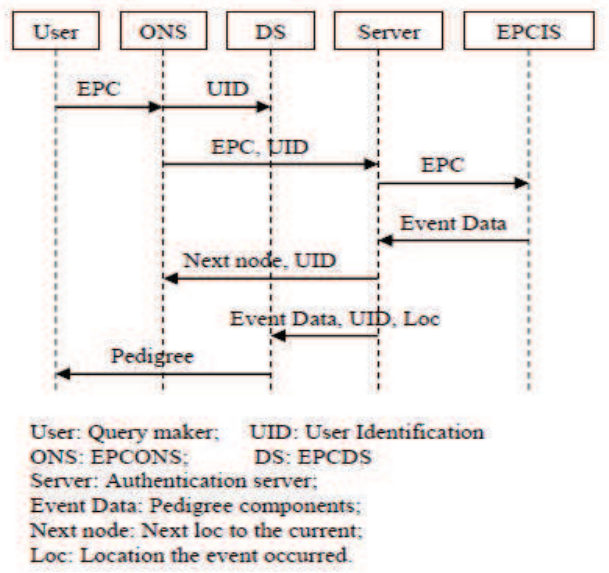

Fig. 3. UML sequence diagram of message flow for e-pedigree verification

B. *Data queries for data mining*

There are generally two types of data mining tasks in this system: one of the data mining task is called *exception detection*, the objectivity of which is to extract the normal patterns of the events' activities when the product moves from the producer to the buyer by data mining, and capture the point when exceptions are detected, so that the risk of threats from daily working fallacy or counterfeiting intentions can be greatly reduced [12].

Another task of the data mining in the system is for each node of the supply chain to discover its knowledge of some type of products from the daily, weekly, monthly and even yearly counting statistics, which helps to support decision making of specific networking node in its marketing promotion and internal management.

Both types of data mining tasks have shared some data features in common.

Dynamic: products of same type enter and exit to the same node are continuously updating, therefore, if products of a type are found in a fixed number for a relatively persistent period, there is supposed to be an exception.

Coocurrence: Products in the warehouse of a node are usually dependent due to in the same pallet, package. In fact, useful knowledge could be discovered on this relationship. This feature is also described as its *spatial relationship*.

Connections: Links between the prior node and the next node of a product plays important role on enhancing the cooperative relationship among the neighboring nodes, based on the knowledge of their link statistics. This feature is called the *temporal relationship of the product*.

Sensitivity: Both types are dealing with sensitive data, and the results including the intermediate results of the data mining might influence the leakage of the privacy.

Similarly, Both types of data mining tasks have their special features that make them different from each.

Purpose: the purposes for both types of data mining are different because the first one is trying to detect the exceptions so that the e-pedigree can be authenticated with high degree of trust, instead the second type is to find out the normal relations between different strategies so that the efficiency and benefits of the business can be improved with the knowledge.

Spread: The range that the data spread in both types is significantly different. The first type requires nearly all data in all nodes taken into consideration, while the second type only requires that within its node.

Parties: The computation of the first type is a multi-party computing, while the second type can do it by each node. Therefore, they make a big difference in considering security and privacy.

C. Potential vulanerbilities and threats

Potential vulnerabilities and threats related to the business processing of the RFID-enabled product supply chain are well studied in [12], which are summarized as shown in Fig.4.

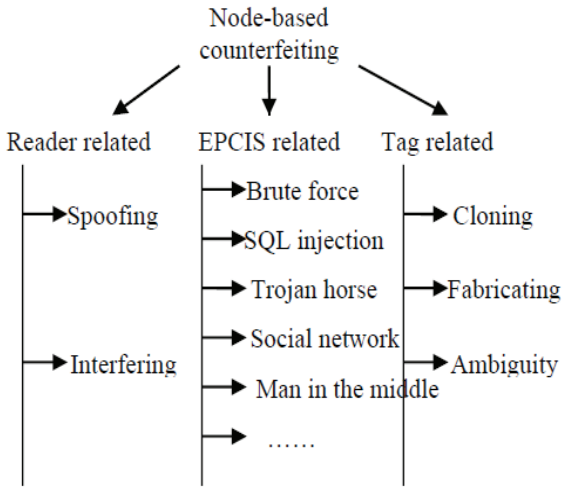

Fig. 4. Classification of node-based potential vulnerabilities and threats

However, there are potential vulnerabilities and threats related to the data release of the data mining or knowledge discovery. The main concern related to the data mining is the privacy preserving, which is deeply intrigued with security issues.

Since the RFID tags can also be read by unknown party nearby with more powerful readers, if the links connected with those tags are disclosed by such data mining process, then the related e-pedigrees, even in their encrypted format, are under the threat of known plaintext-cyphertext pair, which is very easy to be compromised.

Therefore, more sophisticated scheme should be proposed to address this challenging issue that combining security with privacy. In this paper, we propose a system that combines secure multiparty computing with differential privacy to address this issue, and theoretically prove its effectiveness.

D. Secure Multiparty Computing protocols

There are scenarios of data mining when the data is divided among two or more different parties and the aim being to run a data mining algorithm on the union of the parties' databases without allowing any party to view another individual's private data [13,14,15]. This might happen between untrusted financial organizations who plan to cooperatively work on a project for their mutual benefit without disclosing their private requirements, or even among competitors who want to know whether their regions overlap without giving away their private location information. Secure multiparty computing protocol is designed for this purpose.

A simple implementation of this protocol can be achieved with two separate processing layers: one layer as the trusted third party (TTP) is responsible for the joint computation for the datasets collected separately from the other parties, while the other layer is responsible for the secret data exchange between the TTP and each of the parties who needs the result of the computation from the TTP under the framework of PKI infrastructure.

Each party including TTP maintains initially two keys: one is the secret key, the other the public key. Then they need to exchange for each other's public key. The secure computing starts after that. As we can prove, each party shares its secret data with TTP, the later after collecting the data from all the parties, generates the computation results and secretly feedback to each party.

This simple implementation only considers one TTP which might be not so trustworthy. For instance, the sever that the TTP runs may be corrupted by the adversaries, and the whole system then be compromised afterwards.

Mishra et al. [13] proposed an extended encrypto_random (EER algorithm) to address this problem. Their algorithm can be rewritten and illustrated as follows:

Inputs: $P_1, P_2, \cdots, P_n$ as parties; D as function pool;

$f_1, f_2, \cdots, f_n$ as encrypting functions;

R_f as a randomization function

Step 1: for party P_1 to P_n do

Break data block into packets for P_i :$(P_iK_1, P_iK_2, \cdots, P_iK_r)$

Step 2: Select f_j from D, and compute the encrypting values for each packet: $V_1, V_2, \cdots, V_r$.

Step 3: Compute $S_{ik} = P_iK_k + V_kf_j$;

/* where, *k* denotes the index of specific packet */

Step 4: for *i=1 to n*, do

for *k=1 to r,* do

send S_{ik} randomly to P_i;

Step 5: Choose TTP by using R_f;

Step 6: for *i=1 to n*, do

send S_{ik} to TTP chosen;

Step 7: TTP decodes S_{ik} using f_j from D; and rearrange P_iK_k into data blocks;

Step 8: TTP computes and announces result to each party.

Since each party doesn't know what encrypting function may be selected, the privacy for each party can be preserved from each other, while TTP with knowledge of these functions can easily reassemble the data packets to form the whole data blocks, but in no case can relate any data block thus formed to the certain party.

However, this protocol has obvious limitations too. In many scenarios of product supply chains the neighboring nodes have shared data, since in the above-stated protocol these shared data after reassembly by TTP are expressed in their plaintext format, therefore, it is easy for TTP to disclose.

One powerful tool of cryptography in dealing with this kind of problem is the famous Paillier encryption scheme [14] based on the following homo-morphic properties.

Let E_{pk} be the encrypting function with public key *pk* given by (*N,g*), where *N* is a product of two large primes and *g* is a generator in $\mathbb{Z}^*_{N^2}$; And let D_{sk} be the encrypting function with secret key *sk.* Given two plaintexts *x*, *y*∈ $\mathbb{Z}_N$, then we have:

Homomorphic addition:

$$E_{pk}(x + y) \leftarrow E_{pk}(x) * E_{pk}(y) \bmod N^2;$$

Homomorphic multiplication:

$$E_{pk}(x * y) \leftarrow E_{pk}(x)^y \bmod N^2;$$

Semantic security: this scheme is shown semantically secure [15]. In other words, no additional information about the plaintexts can be deducted from a set of ciphertexts given.

Since all information processed in TTP are in their cipher format, even if TTP can reassemble the data blocks from each party and compute them as a whole to generate the expected results for each party, it is unable to disclose any information that is referred to any specific party, not to mention its relation.

E. Differential privacy

Scenarios of product supply chains are too complicated to just use such protocols to achieve sound security and privacy at the same time. In fact, for each node of the supply chain, it is its duty to promote its services and release some of its counting statistics to the public. Obviously, all these data reported in public could not be encrypted, which makes the auxiliary data known to the malicious adversaries.

A paradigm called "differential privacy" was first proposed in 2003 by researchers collaborated as a team such as Dinur, Nissim, and Dwork [16,17], which is completely different from the conventional ways on that it concerns nothing about the sensitive data but the feature of the existence of the data, which can be easily formulized in mathematics as follows[17,18]:

As briefly described in [18], A randomized function *K* gives ε -differential privacy if for all data sets *x* and *x'* differing on at most one element, and all $S \subseteq \text{Range}(K)$,

$$\Pr[K(x) \in S] \leq exp(\varepsilon) \times \Pr[K(x') \in S]$$

Therefore, when ε is a small number, the probabilities for both data sets *x* and *x'* are indistinguishable, so that the difference between both data sets is unable to be leaked.

ε -differential privacy can be achieved by adding a random noise whose magnitude is chosen as a function of the largest change a single participant could have on the output to the query function. This quantity which determines the noise magnitude is called "sensitivity of the query function" [18]:

For a query function $f: D \rightarrow R^d$, the L1-sensitivity of *f* is

$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1$$

for all D_1, D_2 differing on at most one element.

Then the noise of scaled symmetric exponential distribution with variance σ^2 can be determined by letting $\sigma \geq \epsilon/\Delta f$ [19].

This sounds very good philosophy of privacy preserving, and even more, McSherry and Talwar [20] extends ε -differential privacy to cases when the output is not easily "perturbed" with noise by a utility function $u(X, y)$ that measures the quality of an output *y* for data sets *X*. It is shown that y is output with probability proportional to $\exp(-\epsilon u(X, y)/2)$, which ensures ϵΔu-differential privacy, or ϵ -differential privacy whenever $\Delta u \leq 1$. Here Δu is defined as the maximum possible change to the value of *u* caused by changing the data of a single row instead of removing or adding a row. This so-called "exponential mechanism", addresses privacy preserving for structured data sets and/or strategy data sets.

III. PROBLEM FORMULATIONS

A. Data types for data mining queries

There are several different problems that have to be solved, which will be formulated respectively in this section.

The datasets for queries that the data mining process want to draw can be classified as the following different data types:

Data type I: Since e-pedigree of a product records its history of movement from the manufacturer to the buyer, the basic pathways of its shipment or traffic based on its spatial relations (locations) and temporal relations (the order when

those locations are passed) builds up the pattern for the producer to transfer its product to the consumers. There do exist some variations in the middle when some special events occur with reasonable interpretations. While exceptions of the pattern are detected without reasonable interpretations, some attacks are thus detected.

For data mining the data type I, the basic dataset is the temporal set of locations for the product (denoted with its epc), which makes very instance of a pathway for a specific epc. It can be expected that for the same producer and buyer, collections of datasets for similar epcs can be used for data mining this normal pattern, which can be used for exception detection without reasonable interpretations. More formally, Let $\mathbf{D_I}$ be a set of tuples, $\{(epc_i, loc_j, t_k)|i,j,k \in \mathbb{Z}\}$, where t_k denotes a continuous time period from the startpoint when the epc of the product is generated and written into the RFID tag attached to it to the endpoint when the buyer authenticates his/her receiving the product, and i, j, k, are used to distinguish the relevant epc, location and time.

Sometimes when the above-stated data mining results are inconsistent with the flexibility of daily operations, there are too many exceptions with reasonable interpretations, reducing the working efficiency as a whole. In this case, local patterns can help for improvement.

Similarly, the data type is also composed of a collection of tuples, (epc, loc, t), in a limited continuous time period. That is to say, suppose t_0 be the time when the epc of the product is generated, t_n be the time when the buyer receives and authenticates the product, then $\mathrm{t} \in \mathrm{T} \subset [\mathrm{t_0}, \mathrm{t_n}]$.

To put these situations into considerations together, we should formulize our expression of $\mathbf{D_I}$ into more generalized form as follows:

$$\boldsymbol{D_I} \coloneqq \{(epc_i, loc_j, t_k)|i,j,k \in \mathbb{Z}, t_k \subseteq [\mathrm{t_0}, \mathrm{t_n}]\}$$

A sample dataset of data type I is shown in Table I:

TABLE I. A SAMPLE DATASET OF DATA TYPE I

Epc code	Loc_id	T_entry	T_exit
1.2.3.4	12101	2015-07-13 13:55	2015-07-14 09:15
1.2.3.4	12102	2015-07-14 15:30	2015-07-15 03:45
1.2.3.4	12103	2015-07-15 10:20	2015-07-15 20:50
1.2.3.4	12104	2015-07-16 17:30	2015-07-25 06:50
1.2.3.4	12105	2015-07-27 18:30	2015-09-10 14:20

According to EPCglobal standard, the epc code can be designed as a 96-bit code which basically includes the manufacturer_id, product_id, product_type, and product's serial number.

For easy understanding of the problem that might be encountered in typical RFID-enabled product supply chain scenarios, a simple sample location coding table for loc_id is illustrated in Table II.

TABLE II. A SAMPLE LOCATION CODING TABLE

Loc_id	Loc_name	owner
12101	Production Line E	Manufacturer J
12102	Warehouse B	Supplier K
12103	Shipping Port A	Carrier G
12104	Shipping Port F	Carrier G
12105	Retailer Store C	Retailer C

By both Table I and II, the movement of product with its 96 bits' epc of 1.2.3.4 can be semantically interpreted.

Data type II: Usually each node does the work of data mining to improve its management and optimize its marketing strategies, some knowledge of which needs to be discovered this way.

From the viewpoint of a supply chain node manager, the most frequently asked questions should be: "which links make the most profitable part of our business?", "which type of packages takes the shortest time and which type the longest time?", "what is the statistics of those products?", and so on. The answers of these questions are of range or aggregate queries of the related datasets.

For each node of the supply chain, it receives products from different sources, and sends them to the designated different destinations. For simplicity, it can be assumed that each node has the link information of the product that can explicitly express its previous and next location. Let loc' be the previous location, $loc"$ be the next location, this type of datasets can be defined as $\mathbf{D_{II}}$:

$$\boldsymbol{D_{II}} \coloneqq \{(epc_i, loc'_i, loc"_i, t_k)|i,j,k \in \mathbb{Z}, t_k = [t_{k-}, t_{k+}]\}$$

Where t_{k-} denotes when the product arrives at this node, t_{k+} denotes the time of leaving for the next stop. It is pointed out that in real daily practice, both loc' and $loc"$ have multiple options, therefore both have different values for each epc.

Data type III: Since data mining on Data type I and II is a common business processing, the intermediate results and the final results of this processing form another data type, which is denoted as Data type III.

This type of datasets is vulnerable to security threats and privacy leakage because there is some hidden connections between Data type I and II. To show this, we can look at the following simple example.

Suppose by computing Table I which is completely from Data type I dataset, the duration of product 1.2.3.4 in 12101,12102,12103,12104, and 12105 will be 20hrs, 12hrs 15min, 34hrs 30min, 229hrs20min, 1075hrs50min. Moreover, the statistics of all epcs that passed by these 5 locations can be computed, and one of the intermediate results is: "the average duration in a shipping port is within the range of 45hrs-62hrs".

Meanwhile, in location 12104, by computing statistics of the records completely from Data type II, suppose the average duration in this location of computing is approximately 50hrs.

Suppose those results are disclosed and get known by the public, in a worst-case when the adversary knows all information except the duration of one epc, this value can be accurately reconstructed with the following formula:

$$x_i = n * \mu - \sum_{i \neq j} x_j$$

Where n is the sum of the products in this location, μ is the average duration of all products in this location.

Similarly, with the knowledge discovered by data mining from data type I, several nodes can collude to deduce the data of the other nodes as competitors. For instance, according to statistics of shipping nodes, some shipping nodes can work together to inference the related values of the other shipping nodes.

B. System assumptions

To achieve an enhanced guarantee of security and privacy, we need to set up a set of system assumptions.

- We assume that it is the responsibility of the producer to guarantee the authenticity of its products and prevent its brand from corruptions by risk management of choosing the authenticated the shipping pathways according to the related laws. Therefore, all partners along with this supply chain should provide this producer with the information related to its products under its queries while keeping the rights of privacy related to that of other producers.
- We also assume that the communications between each producer with its partners are based the common PKI infrastructure. It is reasonable to assume that each producer should be responsible for the e-pedigree of its own products; therefore, each of the producers should maintain and update the key rings of the PKI infrastructures for the verification of its certificates of its e-pedigrees.
- Each producer works as a semi-trusted TTP based on the contracts with its cooperative partners along the supply chain. It computes the data mining results from the output of its queries to its partners. The basic purpose of this data mining process is for exception detections and verifying the authenticity of its e-pedigrees, while it also has a tendency to inference the efficiency of its partners which is the privacy its partners might be reluctant to disclose. Therefore, while each partner submits the basic location information of the product in its encrypted form to its producer, since the duration information of the product will directly show some features of the node to the producer, it is required that while the data in the set keep unvaried, the order of the data should be randomly decided so that the producer should be unable to inference by data mining and disclose the relations of the data with the data owners.
- Each node has the right to do data mining for improving its own management and marketing promotions only based on the data sets stored in its own database, with assistance of some publicly released data sources from other nodes. However, since the data of each node are linked with that of its neighboring nodes, the leakage of its data will directly be followed with the leakage of that of its neighboring nodes, and leading to the leakage of the datasets in the whole chain. Thus, we assume that for specific node, the knowledge discovered during the data mining processing should be classified into types: one type is *public*, the other is *private*. For the private type, the access to the result is strictly restricted and used only for internal decision support. For the public type, the results of data mining can be released but the related privacy preserving strategy must be taken into considerations.
- Furthermore, we assume that the processing of data mining by the producers is completely independent of that data mining by each node in the whole chain. In practical applications, more often than not, we can usually observe that while there are obvious links among the datasets between the producers' data mining and their partners' data mining, there can be some inconsistence between them due to inconsistent updating or different processing time. This observation can strongly justify our assumption. Basically, the result of any node's data mining should not depend directly on any of other node's results, although there may be influential factors related to its neighboring nodes only on that they may show some similarities in some aspects. For simplicity, we make this assumption.
- We assume the expectations from the public to the accuracy of the data mining reports for the e-pedigree's verifications and the evaluations of the supply chain are completely different: for the first, the more accurate the e-pedigree's verifications, the more authentic the product, and in return, the more beneficial for the producers; while for the other, the accuracy of those published reports is only to show claims from the chain partners, which can be less trustworthy to the public.

In summary, there are two main roles of two *different and also independent* data mining processes: one is the *curator* who tries to collect the related data sets and learn out the exceptions that are most likely referred to counterfeiting activities, so that some measures can be taken to reduce the risks that may follow up. The other is the *promoter* who tries to dig out the best side of its business activities, so that more opportunities can be found by its internal attractions. Both roles are partially connected by their internally correlated datasets. We need to point out that although producers also dig out to promote their products, this behavior is nothing with the supply chain, thus not the competing source for any part of the supply chain, therefore, we don't consider it in this study. Therefore, in this paper, producers are only considered as the curators, while all shipping and distributing parts of the chain are taken as the promoters. Both the curator and its related promoters communicate with each other under the common PKI

infrastructure maintained and supported by the specific curator.

C. Problem formulations

Now we can formulate our problem as follows.

System model: This system is modeled as a distributed environment consisting of N databases across the network. In this system, there are three types of users: the producers or the curators, who start the records of the databases; the distributors or the promoters, who update the records of the databases in the middle; and the customer, or the buyer, who requires the proof of the e-pedigree of the products to be bought. For simplicity, we can only pay attention to the first two types of users. In general scenarios, one curator has to distribute its goods with the assistance of many promoters, and different curators will independently manage this one-to-many relations based on their collaboration contracts. Thus, we can only take one curator and its respective promoters into our considerations.

Databases in the middle are maintained and updated by the respective distributors, as shown in Fig.2.

Suppose $c_i \in$ C (i=1,2,…,n), n is the total number of the producers in the supply chain. $p_{ij} \in$ P (j=1,2,…,m), m is the total number of the promoters collaborated with c_i. Let epc_{ik} be product manufactured by c_i, $k \in [1, K]$, K is the total number of the products made by c_i. Then the history of the records for epc_{ik}, $k \in [1, K]$, can be denoted as a collection of data sequence,

$$D_i \coloneqq \{(\mathrm{epc}_{ik}, p_{ij}, t_{ij}) | k \in [1, K], j \in [1, m]\}$$

Where t_{ij} determines the temporal order of the data sequence.

Obviously, we can prove that the pair of (epc_{ik}, p_{ij}) share the same temporal order of the data sequence with that of t_{ij}.

As previously mentioned, t_{ij} also denotes the entry time and exit time of the product, therefore, the duration of the product in the node can be easily derived, which characterizes some part of the node, especially on its operational efficiency that this node is reluctant to disclose.

Problem formulation I: is there a secure way that c_i can collect D_i, without losing the information of its temporal order, but keeping the order of t_{ij} as a secret to c_i?

Let f_s be a scrambling function that can scramble the order of t_{ij}, then we have,

$$D^*{}_i = f_s(D_i) = f_s(\{(\mathrm{epc}_{ik}, p_{ij}, t_{ij}) | k \in [1, K], j \in [1, m]\}) = \{(\mathrm{epc}_{ik}, p_{ij}, f_s(t_{ij})) | k \in [1, K], j \in [1, m]\}$$

Therefore the critical part of the solution to problem I is to find a reasonable scrambling function that can meet the requirements of the related privacy preserving and security.

Similarly, for specific $p_{ij} \in$ P (j=1,2,…,m), the history of the records for epc_{ik}, $k \in [1, K]$ from different producers or curators $c_i \in$ C (i=1,2,…,n) can be collected automatically and stored in its database, which can be denoted as,

$$D_j \coloneqq \{(\mathrm{epc}_{ik}, p_{ij}, t_{ij}) | k \in [1, K], i \in [1, n]\}$$

Where i=1,2,…,n and acts as the pointer to its related producer c_i.

Most of the knowledge discovered by data mining from datasets of this type in a specific node is related to statistics of its neighboring relations, which will be kept as the secret of this company. However, there are some statistics of this node such as some sums of the product counts from some producers, which might become part of the auxiliary information for the adversary to guess out the related sensitive data just as we have mentioned previously.

Problem formulation II: is there a secure way that p_{ij} can collect D_j, and do its data mining, but keeping the privacy of the statistics of the sum of counts from queries of the databases?

Let f_p be a perturbing function that can add noise of some random distributions to the respective results, then we have,

$$\begin{aligned} &\sum D^*{}_i = \sum f_p(D_i) \\ &= \sum f_p(\{(\mathrm{epc}_{ik}, p_{ij}, t_{ij}) | k \in [1, K], j \in [1, m]\}) \\ &= \left\{\left(f_p\left(\sum \mathrm{epc}_{ik}\right), f_p\left(\sum p_{ij}\right), f_p\left(\sum t_{ij}\right)\right) | k \in [1, K], \right. \\ &\left. j \in [1, m]\right\} \end{aligned}$$

The most important part of problem II requires that while the outcome of the data mining is perturbed by some random noises, the utility must be preserved to prevent it from being useless.

Design goals: Specifically for the security and privacy of the data mining activities, several design goals must be defined in advance.

- *Confidentiality*: The data that each node submits to the curator for the verification of e-pedigree should be kept unknown to the other nodes. The curator, however, should also be restricted with its access to the data which might be related to partial business secret of those distributors.
- *Integrity*: Since data tampering is a very important way of counterfeiting, data exchange between the curator and those distributors should be kept intact.
- *Availability*: We assume that each node has its special data center to ensure its data availability, and the related access control strategies, which will not be discussed in this paper.
- *Non-repudiation*: Each party of the supply chain is not wholly trustable, thus some cryptographically based methods such as the digital certificate should be used to address this issue. This is usually accompanied with the process of verifying the e-pedigrees.
- *Privacy preserving*: The most innovative part of our system is its privacy preserving design of data mining processing. There are two different types of problem

formulations. For problem I, not only the security issues of multiparty data exchange should be taken into considerations, but also the privacy related to some of the data should be kept intact, leading to a complicated situation of combining the secure multiparty computing with privacy preserving. For problem II, while most of the output of the data mining results should be kept in secret for administrative decision making under the role-based access policy, some of the data publically released for marketing promotion must take its privacy into considerations, especially for those eligible for deducting the sensitive information of its partners as auxiliary information.

Threat model: The widely accepted semi-honest adversary model [21] will be used in our system. In this adversary model, all parties, even corrupted ones, follow the protocol honestly to obtain their "legal verifications" through the system, but the adversary will try to obtain the unauthorized data by exploiting the vulnerabilities of the system such as data mining. An adversary is adaptive if it is able to choose the specific parties to corrupt during the computation, rather than having a fixed set of corrupted parties.

An adversary can be either the miner or a database analyzer, who will try to learn both the accurate statistics of its own database and the statistics of sum of counts from all other databases if available.

D. *Our contributions*

In this paper, we focus our attentions on data mining of databases of a sequence of datasets, where the data records of one database has internally temporal and spatial relations with another, while guaranteeing the security and privacy of those databases. In comparison with conventional schemes, our system has the following contributions:

Trade-off between privacy and utility: In a secure multiparty computing environment, the privacy of each party is fully taken into considerations by choosing some scrambling or perturbing functions. The parameters for trade-off between privacy and utility are not only dependent on the databases but also on the situations to be considered. It shows that our system has much better adaptability.

Universality: To our best knowledge, most of current works dealing with similar issues in very restricted scenarios such as queries of statistical databases with independent data or at most limited correlations between those data which are previously fixed before queried. However, in a typical supply chain, those data are correlated with each other and build up dynamically a special data sequence. Obviously, leakage of any part of the data chain will result in the compromise of the whole chain. Therefore, new algorithms must be proposed to address this issue.

Attributes based access control (ABAC): As mentioned above, supply chain databases have some unique features, one of which is that, different data attribute may play different role in the privacy preserving process under the different requirement of the roles such as the data miners and the promoters. In our system, these factors have been carefully considered, and new access control mechanisms are proposed to adapt to this goal.

IV. Our Proposed System

A. *Our proposed system*

In general, the basic way to design such a system is to set up a trust boundary with security policies(SP) or mechanism to protect the databases, so that we can ensure that the data miner will neither disclose the sensitive data nor leave some of the databases unprotected, as shown in Fig.5.

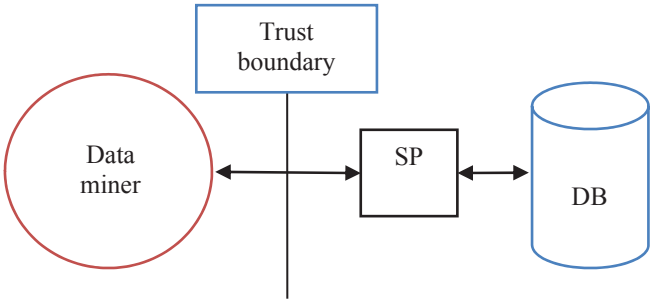

Fig. 5. General security model for data mining of databases

Therefore, the key part of the system design is the security mechanism that includes security policies, methods and rules of access control. The basic attribute-based access control method discussed in [22] will be used in our system, the model structure of which is shown in Fig.6.

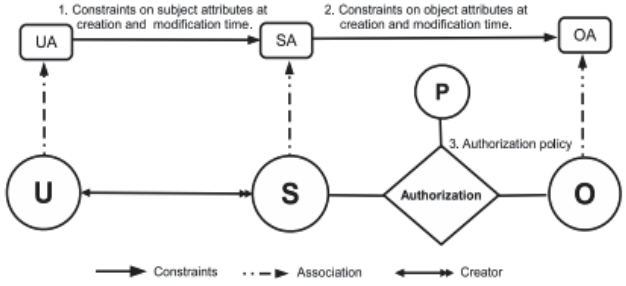

Fig. 6. Unified ABAC model structure, adapted from [22]

In our proposed system, there are mainly two kinds of users, one is the curator, and the other is the internal users of this chain node.

For different users (U) with different attributes (UA), some constrains on its related subjects (S) and objects (O) can be determined when it is created in its initialization phase.

Based on this model structure, we can further determine the security policies for the related queries and functions to process the datasets. The basic architecture of our proposed system is shown in Fig.7.

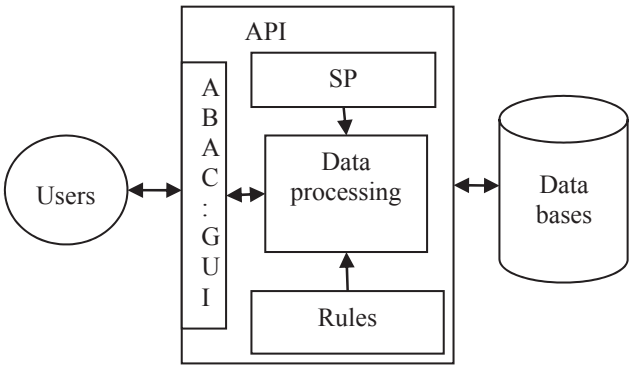

Fig. 7. The basic architecture of our proposed system

From Fig.7, each user is connected with a series of rules (i.e. protocols to get those data ready for queries of its user) and security policies (i.e. the strategies that tells how the data should be processed) based on its attributes and constraints, which can select the appropriate functions to map the raw databases to the related datasets and process the data to the form that can meet our security and privacy requirements.

B. Key issues of its implementation

The core of the "data processing" modular in Fig.7 attempts to process the datasets according to the rules and the security policies for specific users. There are chiefly two types of data processing in our system: one is the random distributions of temporal dataset based on a secure multiparty computing protocol (TDSMUP), and the other is noise perturbation of the dataset based on differential privacy (NPDDP).

TDSMUP algorithm: we draft this algorithm as a secure multiparty computing protocol by re-designing Mishra's EER algorithm [13].

Inputs: $P_1, P_2, \cdots, P_{\mathrm{n}}$ as parties; D as function pool;

$f_1, f_2, \cdots, f_{\mathrm{m}}$ as encrypting functions;

Key_{pub}, Key_{pri} as the PKI key pair of the curator;

Key_{ipub}, Key_{ipri} as the PKI key pair of P_i;

R_f as a randomization function

Initialization:

for party $P_1$to P_{n} , do

Break data block into packets for P_i :$(P_iK_1, P_iK_2, \cdots, P_iK_r)$

for the curator, do

Generate a random vector with R_f: $N = (r_1, r_2, \cdots, r_n)$.

Select f_k from D by letting $k = r_j$,

Encrypt (r_j, f_k) with Key_{ipub} , send Key_{ipub} (r_j, f_k) to P_i.

P_i:

Decrypt Key_{ipub} (r_j, f_k) with Key_{ipri};

Encrypt P_iK_r with f_k , i.e. $V_r = f_j(P_iK_r)$;

Encrypt (V_r, r_j) with Key_{pub};

Send $Key_{pub}(V_r, r_j)$ to the curator.

The curator:

Decrypt Key_{pub} (V_r, r_j) with Key_{pri};

Decrypt V_r with f_k from D by letting $k = r_j$;

Re-assemble P_iK_r back to its original data block.

NPDDP algorithm: this algorithm is specially designed for those statistical queries that can be used for knowledge discovery of statistics in marketing promotion use, where we assume that removal or addition of any one of the records should not affect the outcome of the queries, therefore the privacy of particular sensitive data could be preserved this way. In our system, we choose to use the Laplacian mechanism, and the parameters that determine the Laplace distribution will be computed according to the results published in [23].

Inputs: E as the essential cost; B as the budget; α as the required accuracy and T as the required error threshold; $D = \{D_0, D_1, \cdots, D_n\}$ as the data sets.

Step 1: Compute number of records that can meet the requirement of privacy preserving N, such that

$$\mathrm{N} = \frac{12}{\mathrm{T}^2} \ln \frac{3}{\alpha}$$

Step 2: Compare N with the actual counts of records N', if N<N', then halt, giving the message of privacy warning.

Step 3: Otherwise, Compute t1 and t2 such that

$$\mathrm{t1} = \frac{\mathrm{T}}{6}; \quad \mathrm{t2} = \ln\left(1 + \frac{\mathrm{B}}{\mathrm{EN}}\right).$$

Step4: Choose ε=min{t1,t2}.

Step 5: Compute the noise of Laplace distribution such that

$$\hat{\mathrm{x}} = -\varepsilon \, \mathrm{sgn}(\mathrm{U}) \ln(1 - 2|\mathrm{U}|)$$

Where U is a random variable of uniform distribution within the interval $\left[-\frac{\mathrm{N}\prime}{2}, \frac{\mathrm{N}\prime}{2}\right]$.

Step 6: add $\hat{\mathrm{x}}$ to the output data.

Step 7: Return $D' = \{D'_0, D'_1, \cdots, D'_n\}$

C. Theoretical analysis

For TDSMUP algorithm, it is easy to prove that any other parties are unable to know the data sent to P_i because it is encrypted with Key_{ipub}, which can only be decrypted with Key_{ipri}, the private key of P_i. Similarly, any other parties are unable to know the data sent to the curator by eavesdropping because it is encrypted with Key_{pub} which can be decrypted only with Key_{pri}, the private key of the curator.

Furthermore, we also can prove that the curator should be unable to distinguish the source of the data, therefore having the privacy of the data preserved. The reason is, the curator randomly selects the encryption functions with the index and send to the respective parties in a temporal order, the curator itself is not permitted to remember this random numbers. In consequence, the curator can re-assemble the data blocks just

with the random numbers, losing the information of their temporal order, i.e. the source of the data.

For NPDDP algorithm based on the principle of differential privacy, it can be easily proved as in [23] that the privacy of the data can be well preserved within the range of the required accuracy and perturbed error.

Therefore, our design goals are achieved. However, in this algorithm, there are places that need to be improved. (1) It is simply assumed that the number of functions for encryption is no less than that of the parties, which might be untrue in practical applications. (2) What if the curator remembers the random numbers since it is semi-trusted?

Although there are some seemingly simple ways to get these problems solved, there are also some other factors to be considered to have the system easily maintained. We leave them to be studied in the near future.

V. CONCLUSIONS

In scenarios of a complicated RFID-enabled product supply chain, the e-pedigree of the product plays a very important role in verifying the authenticity of the product, thus reducing greatly the risk of faked products. However, since the data of the e-pedigree is essentially a set of a temporal sequence which is correlated with each other, the publishing or release of their data mining results might cause severe leakage of the privacy, leading to the complete compromise of the whole supply chain. Therefore, the security and privacy of this kind of data turn out to be a very challenging issue in this field.

To address this issue, we proposed a system that combines the secure multi-party computing with the differential privacy techniques on the basis of the attribute based access control mechanism of the databases distributed along the supply chain, achieving a better trade-off between the privacy and utility with a better universality.

Although our proposed algorithms look well fitted for our preset requirements, we also noticed that some problems still exist such as the curator might remember the random numbers and disclose the privacy of the related parties, which will leave just for future research.

ACKNOWLEDGMENT

This survey is supported by Natural Science Foundation of China under the grant no. 61462023.

REFERENCES

[1] A. Musa, A. Gunasekaran, and Y. Yusuf, " Supply chain product visibility: Methods, systems and impacts", Expert Systems with Applications, vol.41, pp.176-194, 2014

[2] A. Marucheck, N. Greis, C. Mena, and L. Cai, " Product safety and security in the global supply chain: Issues, challenges and research opportunities," Journal of Operations Management, vol. 29, pp.707-720, 2011

[3] F. Dabbene, P. Gay, and C. Tortia, " Traceability issues in food supply chain management: A review," Biosystems Engineering, vol. 120, pp.65-80,2014

[4] EPCglobal, "Pedigree Ratified Standard version 1.0", http://www.gs1.org/gsmp/kc/epcglobal/pedigree/pedigree_1_0-standard-20070105.pdf , 2007, pp.1-138

[5] M.. Schapranow, J. Muller, A. Zeier, and H. Plattner, " Costs of authentic pharmaceuticals: research on qualitative and quantitative aspects of enabling anti-counterfeiting in RFID-aided supply chain," Pers. Ubiqui. Comput. Vol.16, pp.271-289, 2012

[6] S. H. Choi, and C. H. Poon, " An RFID based anti-counterfeiting system," IAENG Int. J. of Compu. Sci., vol.35, 2008, pp.1-12, http://www.iaeng.org/IJCS/issues_v35/issue_1/IJCS_35_1_12.pdf

[7] EPCglobal, Inc. "EPC information services v.1.1," http://www.gs1.org/gsmp/kc/epcglobal/epcis/epcis_1_1-standard-20140520.pdf, pp.1-169, May 2014

[8] EPCglobal,Inc. "Architecture Framework v.1.6," http://www.gs1.org/docs/gsmp/architecture/GS1_System_Architecture.pdf, pp.1-71, April 14, 2014

[9] N. Zhang, M. Li, and W. Lou. Distributed data mining with differential privacy. In Proc. of ICC 2011, Kyoto, Japan, June 2011

[10] Y. Lindell and B. Pinkas. " Privacy preserving data mning." In Journal of Cryptology, Springer-Verlag, pp.36-54, 2000

[11] M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. Arxiv preprint arXiv:1012.4763, 2011

[12] X.Yao, X. Zhou, and J. Ma. "Object Event Visibility for Anti-Counterfeiting in RFID-Enabled Product Supply Chains", in proc. Of SAI2015, July 28-30, London, UK, 2015

[13] D. Mishra, N. Koria, N. Kapoor, and R. Bahety. "A Secure Multi-Party Computation Protocol for Malicious Computation Prevention for Preserving Privacy during Data Mining." (IJCSIS) International Journal of Computer Science & Information Security, Vol.3, No.1, July 2009, pp 79-85.

[14] B. Samanthula, W. Jiang, and E. Bertino. Privacy-preserving complex query evaluation over semantically secure encrypted data. In ESORICS, pp. 400–418, 2014

[15] O. Goldreich. The Foundations of Cryptography, volume 2, chapter Encryption Schemes, pages 373–470. Cambridge University Press, Cambridge, England, 2004.

[16] I. Dinur and K. Nissim. Revealing information while preserving privacy. In Proceedings of the Association for Computing Machinery SIGACTSIGMOD-SIGART Symposium on Principles of Database Systems, pages 202–210. 2003.

[17] C. Dwork. Differential privacy: A survey of results. In Proc. 5th TAMC, pages 1–19. Springer, 2008.

[18] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In Proceedings of the 3rd Theory of Cryptography Conference, pages 265–284. Springer, 2006.

[19] T. Steinke and J. Ullman, Between pure and approximate differential privacy, Arxiv preprint arXiv:1501.06095, 2015

[20] F. McSherry and K. Talwar. Mechanism design via differential privacy. In Proceedings of the 48th Annual Symposium on Foundations of Computer Science, pages;94-107, 2007

[21] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacypreserving data mining," Journal of Privacy and Confidentiality, vol. 1, no. 1, pp. 59–98, 2008

[22] X. Jin. "Attribute-based access control models and implementation in cloud infrastructure as a service", Ph.D dissertation, dept. of computer science, college of sciences, The University of Texas at San Antonio, May, 2014

[23] J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, Benjamin C. Pierce, and A. Roth. Differential privacy: An economic method for choosing epsilon. In Proceedings of 27th IEEE Computer Security Foundations Symposium (CSF), 2014