

Scalable Industry Data Access Control in RFID-Enabled Supply Chain

Saiyu Qi, Yuanqing Zheng, *Member, IEEE, ACM*, Mo Li, *Member, IEEE, ACM*,
Yunhao Liu, *Fellow, IEEE, ACM*, and Jinli Qiu

Abstract—By attaching RFID tags to products, supply chain participants can identify products and create product data to record the product particulars in transit. Participants along the supply chain share their product data to enable information exchange and support critical decisions in production operations. Such an information sharing essentially requires a data access control mechanism when the product data relate to sensitive business issues. However, existing access control solutions are ill-suited to the RFID-enabled supply chain, as they are not scalable in handling a huge number of tags, introduce vulnerability to the product data, and perform poorly to support privilege revocation of product data. We present a new scalable industry data access control system that addresses these limitations. Our system provides an item-level data access control mechanism that defines and enforces access policies based on both the participants' role attributes and the products' RFID tag attributes. Our system further provides an item-level privilege revocation mechanism by allowing the participants to delegate encryption updates in revocation operation without disclosing the underlying data contents. We design a new updatable encryption scheme and integrate it with ciphertext policy-attribute-based encryption to implement the key components of our system.

Index Terms—Industry data, access control, RFID, supply chain.

I. INTRODUCTION

IN SUPPLY chain data management, participants can attach RFID tags to products, automatically identify products and create product data to record the product particulars in transit.

Manuscript received March 29, 2015; revised October 31, 2015; accepted February 1, 2016; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor H. Zheng. Date of publication March 25, 2016; date of current version December 15, 2016. This work was supported in part by The Key Program of NSFC under Grant U1405255, the National High Technology Research and Development Program (863 Program) under Grant 2015AA011704 and Grant 2015AA016007, Nanyang Technological University Nanyang Assistant Professorship under Grant M4080738.020, the Singapore Ministry of Education within the Academic Research Fund under Grant MOE2013-T1-002-005, The Key Program of NSFC-Guangdong Union Foundation under Grant U1135002, and the Hong Kong Early Career Scheme under Grant PolyU 252053/15E.

S. Qi is with the School of Cyber Engineering, Xidian University, Xi'an 710071, China (e-mail: syqi@connect.ust.hk).

Y. Zheng is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csyqzheng@comp.polyu.edu.hk).

M. Li is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: limo@ntu.edu.sg).

Y. Liu is with the Tsinghua National Laboratory for Information Science and Technology, School of Software, Tsinghua University, Beijing 100084, China (e-mail: yunhaoliu@gmail.com).

J. Qiu is with the Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: jinliqiu1984@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNET.2016.2536626

The product data can thus be shared among participants, which facilitates information exchange and supports critical decisions in production operations [1], [2].

To facilitate data management, a service provider is usually appointed to coordinate the data sharing between participants. Each participant only needs to communicate directly with the service provider for data submission/retrieval. In such a paradigm, each product is associated with an RFID tag with a unique ID that can be used to index its relevant product data managed by the service provider. When a participant receives a tagged product, it can use the tag ID as an index to submit its own product data or retrieve the corresponding product data submitted by other participants. In practice, the service provider could be a commercial supply chain manager such as GT Nexus [3] or a certain participant in the supply chain.

Product data stored in the service provider can be closely related to sensitive business issues and thus the data privacy becomes a natural concern. Strict data access control should be applied to disallow unauthorized data access from other entities or even the service provider itself. For instance, the pharmaceutical supply chain tracks tagged drugs and establishes a pedigree for each drug, e.g., counterfeit certificate, time of delivery, manufacturers, etc [4]. The drug pedigrees should be accessed by retailers and consumers to check whether the drugs are from trustworthy participants [5]. However, drug pedigrees may contain sensitive business matters subject to various malicious accesses. Drug counterfeiters and competitive manufacturers can exploit software vulnerabilities to gain unauthorized access to the service provider; a curious service provider may actively explore the content of its managed drug pedigrees; and any attackers with physical access to the service provider can access all the drug pedigrees in memory. Therefore, participants highly desire the data confidentiality and access control based on participant defined policies.

A straightforward but inefficient approach is to encrypt product data using cryptographic primitives and distribute decryption keys only to authorized participants. A large scale supply chain, however, needs to handle millions of products going through many participants. The products can spread worldwide and thus the intermediate participants as well as the final consumer of a product are generally unknown in advance. Therefore, it is hard to manage and distribute decryption keys in advance.

In practice, instead of designating a particular partner, a participant may want to provide access to the ones with

certain attributes. For instance, a USA drug manufacturer may allow access to the ones that have both ‘USA’ and ‘FDA’ attributes, or the ones who are ‘Retailer’ or ‘Consumer’. Recently, a new cryptographic primitive called Ciphertext Policy-Attribute Based Encryption (CP-ABE) [9] is proposed to provide attribute based access control. With CP-ABE, a user can specify access policies based on logical expressions over user attributes for data encryptions before uploading to a third-party database. A key authority assigns each user a credential corresponding to the set of attributes that describe the user’s features (e.g., company nationality, business domain, etc). CP-ABE ensures that only users with attributes satisfying the logical expression can decrypt the data. In addition, as the database only stores encrypted data, even the service provider cannot learn any nontrivial information about the data.

Despite those benefits, CP-ABE is ill-suited to secure the product data of RFID-enabled supply chains for three main reasons. First, CP-ABE was designed to protect data associated with users, providing data access control in user-level; it is thus not scalable to protect product data associated with products, which requires item-level data access control. Second, CP-ABE relies on a key authority to manage all the credentials of attributes, which is a target for compromise. If the key authority is compromised or the credentials are occasionally exposed, the product data of the whole supply chain can be decrypted. Third, CP-ABE introduces substantial credential issuing overhead for supply chain participants to revoke access privilege of product data in item-level.

In this paper, we develop a scalable industry data access control system for RFID-enabled supply chain that (1) provides item-level data access control and (2) enforces item-level privilege revocation.

Our system provides a new item-level data access control mechanism, which defines and enforces role attributes (for participants) and tag attributes (for products) to regulate item-level data access control. A participant may define an access policy to be a logical expression over the role attributes and a tag attribute. An authorized partner must have the credentials of both satisfactory role attributes and the tag attribute so as to access the corresponding data of the particular product, which preclude adversaries outside the supply chain (who lack the credential of tag attributes) or unauthorized participants within the supply chain (who lack the credential of satisfactory role attributes). More importantly, the credentials of role attributes for participants are issued by a key authority while the tag attributes and their credentials can be locally generated by participants and stored in tags for distribution. Our system uses this mechanism to address the first two limitations of CP-ABE. First, the mechanism scales to support millions of products where the generation and maintenance of their tag attributes and credentials do not have to go through the key authority. Second, since the key authority only manages the credentials of role attributes, compromising it does not expose the product data, which is encrypted by both role attributes and tag attributes.

Our system provides a new item-level privilege revocation mechanism in a way that overcomes the inefficiency hurdles of CP-ABE. Generally, our revocation mechanism consists of

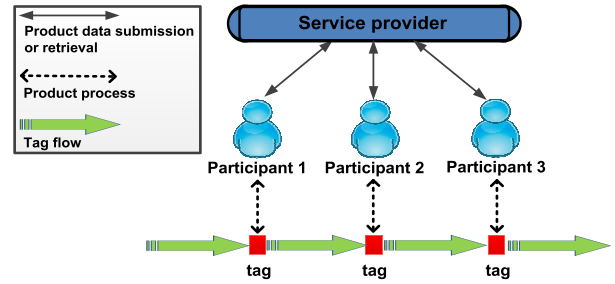


Fig. 1. An example of an RFID-enabled supply chain consisting of three participants. A tagged product flows through the chain and is sequentially processed by the participants. The participants rely on the service provider to share their product data.

two steps. When a participant wants to revoke the data access privilege of some participants for a product, it first updates the tag attribute/credential pair stored in the tag. It then delegates the service provider to update the tag attribute contained in the access policies of the corresponding encrypted product data without revealing any additional information about the data. Our system uses this mechanism to address the third limitation of CP-ABE. The mechanism does not involve the key authority to issue any credentials to participants and only incurs minimal computation and communication overhead. We believe that our item-level privilege revocation mechanism could greatly enhance the flexibility and security of our system. For instance, an intermediate participant may leave the supply chain after completing its production tasks or be temporally compromised by an attacker. In both cases, the downstream participant of the leaved participant can immediately revoke the leaved one’s data access privilege about the products.

We devise a new encryption scheme called updatable encryption and integrate with Ciphertext Policy-Attribute Based Encryption (CP-ABE) [9] to enforce the key components of our system. With complete design, we further evaluate the performance of our system through large-scale experiments to show its efficiency and scalability.

The rest of this paper is organized as follows. We describe the system model and problem in Section II. We present the design details of our system in Section III. In Section IV, we conduct theoretical analysis to prove the security properties of our schemes. In Section V, we examine the efficiency of our system. At last, we review the related works in Section VI and conclude this paper in Section VII.

II. SYSTEM MODEL AND PROBLEM DESCRIPTION

A. RFID-Enabled Supply Chains

An RFID-enabled supply chain is typically composed of three components: RFID tags (attached to products), readers (owned by participants for tag interrogation) and a database (at the service provider for data management). Supply chain participants (e.g., manufacturers, deliverers, and retailers) create, store, and share product data through the service provider and leverage the RFID tags to track the product data in the database.

Figure 1 depicts an illustrative example, where the supply chain consists of three participants. When a product enters the

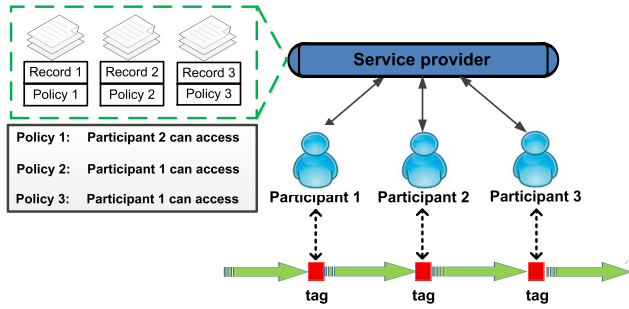


Fig. 2. An example of data access control in an RFID-enabled supply chain consisting of three participants. For a tagged product, each participant generates a product data record, associates the record with an access policy, and submits it to the service provider for sharing purpose.

supply chain, the product is labeled with a tag. The tagged product then flows through the supply chain and is sequentially processed by the participants. Each participant generates its own data record about the product. The three participants rely on a service provider to share their product data, i.e., each participant can submit its product data record to the provider or retrieve the product data records submitted by others from the provider.

As for the supply-chain structure, we assume that participants may only know their direct upstream and downstream participants and may continuously join and leave the chain. For ease of presentation, we assume that one centralized service provider is appointed for data management. In practice, the service provider can be deployed in a distributed manner or in a cloud.

B. Security Model

In this paper, we consider honest but curious service provider, i.e., the service provider follows our proposed system in general, but may try to explore as much product data as possible based on participant inputs. We also consider that participants may try to access product data outside the scope of their access privileges independently or cooperatively. Communication channel between the participants and the service provider are assumed to be secured using security protocols like SSL.

C. Design Requirements

We aim to develop a scalable industry data access control system for RFID-enabled supply chain. Figure 2 shows an example of data access control in an RFID-enabled supply chain consisting of three participants. When a tagged product flows through the supply chain, each participant generates a product data record (participant i for record i) and submits the record to the service provider. As these records can be closely related to sensitive business issues, each participant wants to specify an access policy for its record so that only certain authorized partners within the supply chain can access its record. For instance, policy 1 specified by participant 1 states that “Participant 2 can access”, which means that participant 2 is the only authorized partner to record 1.

In the following, we identify four design requirements that a scalable industry data access control system in RFID-enabled supply chain should support:

- **Data privacy:** The basic design requirement is to prevent unauthorized entities from learning any nontrivial information about the product data submitted by participants. As the product data may contain highly sensitive information about products (e.g., counterfeit information, business relationship between participants, etc.), we need to strictly ensure data privacy.
- **Item-level access control:** When a tagged product flows through a supply chain, each participant should be able to specify an access policy to its own data record about the product. The policy should be fine-grained so that the participant can precisely define the authorized partners.
- **Item-level privilege revocation:** When a participant receives a tagged product, it can (optionally) choose to: 1) revoke the access privilege of its upstream participants to the product data records generated by its downstream participants, and 2) at the same time preserve the access privilege of the downstream participants to the product data records generated by the upstream participants.
- **Scalability and efficiency:** We want to achieve item-level access control and privilege revocation with high scalability. Also, we want to incur small memory cost that is affordable for low cost RFID tags and prevent implementing complex cryptographic algorithms on tags.

D. Limitations of CP-ABE

CP-ABE fails to satisfy the above design requirements with efficiency and security guarantees mainly due to following three reasons.

First, CP-ABE is not scalable to protect product data associated with products, which requires item-level data access control. Suppose a tagged product enters the supply chain and a participant wants to share its data record about the product with its partners. As the data record is associated with the product, the participant regulates: (1) who are retailers in USA or France *and* (2) who have the privilege to process the product (within the same supply chain) can access this data record. Note that the condition (2) is necessary to preclude a retailer in USA but in another supply chain to access this record.

To enforce such a regulation, a strawman CP-ABE solution is to define a small set of role attributes (e.g. ‘retailer’, ‘USA’, ‘France’) to describe the features of participants and a large set of tag attributes to identify each tagged product, and appoint a key authority to issue credentials of proper attributes to each participant. In this way, the participant can specify an access policy (‘retailer’ AND (‘USA’ OR ‘France’) AND ‘TagAtt’) for its product data record; and a participant with the credential of attributes {‘retailer’, ‘USA’, ‘TagAtt’} can decrypt the data. However, this means that when a tagged product flows through the supply chain, each participant within the chain has to apply a credential for its role attributes as well as the tag attribute, which poses heavy computation and communication overhead on the key authority when handling millions of tagged products.

As the overhead of a credential is proportional to the number of its attributes, a better way is to manage role attributes and tag attributes separately. A participant applies from the key authority a credential of its role attributes at once and credentials of tag attributes for tagged products when needed. Unfortunately, CP-ABE disallows a participant to jointly use two different credentials to decrypt ABE-encryptions. This is due to the collusion-resistance property of CP-ABE, which states that multiple credentials can only be able to decrypt an encryption if at least one of the credentials could decrypt it on its own. Originally, this property is used to prevent large-scale data leakage from an attacker that manages to get a hold of a few credentials.

Second, all the participants have to trust the key authority to securely manage their credentials, which introduces vulnerability to their product data records. The credentials may be leaked due to ill-managements. An adversary can exploit software vulnerabilities to gain unauthorized access to servers; curious or malicious administrators at a hosting can snoop on the credentials; and attackers with physical access to servers can access all credentials in memory. The leaked credentials could be used to decrypt the product data of the whole supply chain stored in the service provider. Trusting a key authority to securely manage their credentials is often unacceptable to supply chain participants whose product data is highly sensitive.

Third, CP-ABE introduces substantial credential issuing overhead to support item-level revocation. To revoke the data access privilege of its upstream participants for a product, a participant has to inform the key authority to issue two credentials with the old tag attribute and a new tag attribute (respectively) for each of its downstream participants. For instance, suppose the revoked participants use the tag attribute 'TagAtt' to regulate the access policies of their product data records. A downstream participant with role attributes 'retailer' and 'USA' has to acquire a credential of attributes {'retailer', 'USA', 'TagAtt'} and a credential of attributes {'retailer', 'USA', 'New-TagAtt'} from the key authority. With the two credentials, the downstream participant can then: (1) use the new tag attribute 'New-TagAtt' to regulate the access policy of its product data record to preclude the access privilege of the revoked participants; (2) still use the former credential to access the product data records of the revoked participants; and (3) use the latter credential to access the product data records of other downstream participants.

III. SYSTEM DESIGN

A. Design Principle

1) *Item-Level Data Access Control*: We use two types of attributes: role attributes and tag attributes to specify data access policies. Role attributes generally refer to different properties of participants. For instance, 'USA' describes the location of participants; 'drug' describes the production type of manufacturers. Role attributes alone however do not provide sufficient granularity to specify item-level data access policies. We thus introduce tag attributes to identify tagged products. A tag attribute is similar in essence to a unique tag ID,

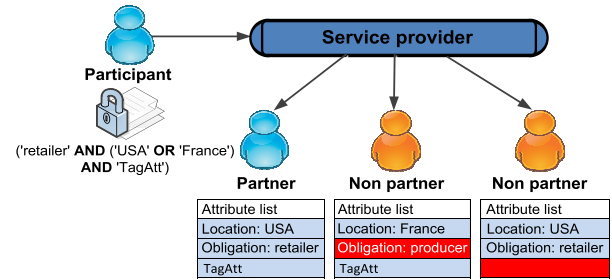


Fig. 3. An example of item-level data access control. Participants with satisfiable attributes regarding the policy are partners and can access the record, while the ones with unsatisfiable attributes regarding the policy are non partners and cannot access the record.

but is carefully designed to enforce cryptographic functionality. We use 'TagAtt' to denote a tag attribute.

Access policy to a product data record is defined as a logical expression over role attributes AND the associated tag attribute. As a result, a participant can specify access policy in item-level by incorporating the tag attribute in the policy when encrypting its product data record. Figure 3 illustrates an example where a participant shares its product data record with certain participants named partners. In this example, the partners are retailers in USA and France who have rights to process the product. The participant specifies the access policy as ('retailer' AND ('USA' OR 'France') AND 'TagAtt') and submits the policy enforced record to the service provider. By doing so, the participant with role attribute 'producer' and the one without the tag attribute 'TagAtt' are non partners and are precluded from the data record, while the participant with attributes satisfying the access policy is a partner and thus can access the data record. The service provider is also excluded from the data record since it does not have satisfiable attributes.

The data access privileges of participants are regulated by attribute credentials, i.e., an authorized participant must have the credentials of satisfiable attributes to decrypt product data records. Our system manages the credentials of attributes in a distributed manner. First, we use a key authority to manage role attributes and their credentials. We rely on the key authority to check if the role attributes match the properties of the participants when issuing the role attribute credentials. As the cardinality of role attributes is small and can be fixed in the system initialization, this management only requires the key authority to issue lightweight role attribute credentials to participants once in the system initialization. Second, we put the management of tag attribute credentials in the hands of the participants. As the cardinality of tag attributes is large (million-level) and cannot be fixed in the system initialization, this management relieves the key authority from issuing large amounts of tag attribute credentials during the production and disabling its ability to decrypt the product data records.

A participant encrypts its product data records with access policies defined from role attributes and tag attributes and uploads the policy enforced encryptions to the service provider. When a downstream participant receives a tagged product, it can retrieve the policy enforced encryptions (submitted by upstream participants) about the product from

the service provider. Without credential of the tag attribute, however, the participant cannot decrypt. As large number of tagged products can flow through many participants within the supply chain, it is hard to distribute the credentials of tag attributes for all the participants in advance. To solve this problem, we leverage tags as a natural medium to distribute the credentials of their attributes to all the participants (within the supply chain). Although commodity passive tags only have very limited memory (e.g., 512 bits in ALN-9640 tags), our compact design of tag attribute credentials can fit in such a small space.

Enforcement of Distributed Credential Issuing: We enforce the above distributed credential issuing mechanism by combining symmetric encryption scheme with CP-ABE [9]. We apply CP-ABE to encode the credentials of role attributes into CP-ABE decryption keys. Initially, the key authority generates a public/master key pair for a list of role attributes and publishes the corresponding public key. Each participant then requests the key authority for a credential of a set of role attributes which the participant possesses.

To incorporate tag attributes, we encode credentials of tag attributes into symmetric keys. When a tagged product enters the supply chain, the first participant of the chain generates a tag attribute and a symmetric key. The tag attribute-symmetric key pair forms a tag token and is stored in the attached tag. To prevent the tag token to be stolen by an outsider of the supply chain, a participant can use lightweight encryption schemes such as AES to encrypt the tag token and share the decryption key with its direct upstream and downstream participants.

We use double encryption paradigm to generate policy enforced encryption. Given a tagged product, a participant first encrypts to a logical expression over role attributes using the CP-ABE scheme and then encrypts the result by using the symmetric key contained in the tag token. Such a paradigm precisely enforces the access policy desired in our system (a logical expression over role attributes AND a tag attribute). Participants can decrypt the policy enforced encryption only if they possess the credentials of both satisfiable role attributes and the tag attribute.

2) *Item-Level Privilege Revocation:* Item-level privilege revocation means that a participant could revoke the data access privileges of its upstream participants for a product. To preserve regular data access control after the revocation, the participant should be able to complete two tasks: 1) revoke the access privileges of its upstream participants to the product data records generated by its downstream participants, and 2) meanwhile preserve the access privileges of the downstream participants to the product data records generated by the upstream participants.

To complete task 1, the participants can locally update the tag token. To complete task 2, the participant can download the policy enforced encryptions of its upstream participants, use the old tag tokens to decrypt them and the new tag tokens to re-encrypt them, and submit them to the service provider. Such a solution, however, incurs high computation and communication overhead. Instead, our revocation mechanism allows the participant to delegate the service provider to update the access

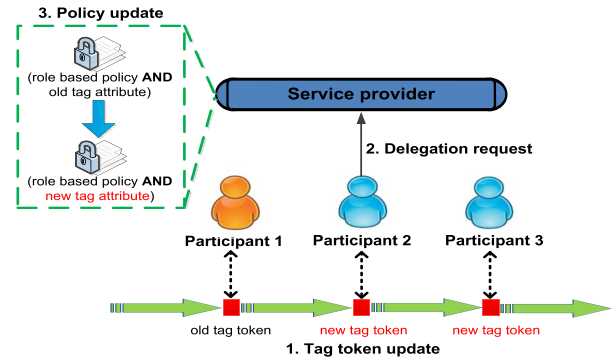


Fig. 4. An example of item-level revocation. Participant 2 updates the tag token stored in the attached tag and sends a delegation request to the service provider to enable it to update the tag attribute of the policy enforced encryption submitted by participant 1. After that, participant 3 will use the updated tag token to encrypt its data record and participant 1 is revoked.

policies of the policy enforced encryptions of its upstream participants. The update operation is lightweight and does not require decryption, thus preserving data privacy. Comparing with CP-ABE, Our mechanism allows the participant to complete revocation operation by itself, thus does not involving the key authority to issue credentials to any participants.

Figure 4 illustrates a usage scenario of our revocation mechanism, where participant 2 wants to revoke the data access privilege of the upstream participant 1 for a product. To do so, participant 2 directly updates the tag token stored in the attached tag. It then sends a delegation request to the service provider to enable it to update the tag attribute of the policy enforced encryption submitted by participant 1. After that, participant 1 is precluded from the product data record of participant 3 as participant 3 will use the updated tag token to encrypt its data record, while participant 3 can still access the product data record of participant 1. During the above process, the key authority is not involved for credential issuing.

Enforcement of Access Policy Updating: General symmetric encryption schemes do not support access policy updates without decryption. To solve this problem, we design a new encryption scheme called updatable encryption scheme to support this functionality. We use this scheme to replace the general symmetric encryption scheme. As a result, the credentials of tag attributes become the secret keys of the updatable encryption scheme.

Our updatable encryption scheme is inspired by the Proxy Re-Encryption [16], in which a proxy can update an encryption under an old key to another encryption under a new key without decryption. Such an approach, however, requires 1024-bit encryption keys to achieve standard security, while current passive tags only have 512-bit user memory. Furthermore, the security of the scheme is not proved in proper security model. Besides, recent proposals [22]–[26] only allow one-time update operation, while we desire multi-time update operations to support multiple revocation operations by different participants within the same supply chain.

We design a new updatable encryption scheme to overcome the above drawbacks. The tag token consists of a tag attribute and a secret key of the updatable encryption scheme. Our design ensures the length of the tag token to be suitable for

current passive tags. In a revocation operation, the participant sends a re-key to the service provider for it to update the access policies of policy enforced encryptions. Our design also ensures that the leakage of re-key does not break the data privacy and the policy enforced encryptions can be updated multiple times. We define two security models in supply chain setting to formalize malicious revoked participants and malicious service provider, respectively. We prove the security of our updatable encryption scheme in the two models.

B. Cryptographic Primitives: CP-ABE

The CP-ABE scheme [9] consists of four algorithms: (Setup, Encrypt, KeyGen, Decrypt).

- **Setup**(λ): On input a security parameter λ , the algorithm outputs a public key PK and a master key MSK .
- **Encrypt**(PK, RP, m): On input the public key PK , an access policy RP and a message m , the algorithm outputs an encryption CT .
- **KeyGen**(MSK, S): On input the master key MSK and a set S of attributes, the algorithm outputs a private key SK . An attribute can be any string.
- **Decrypt**(PK, CT, SK): On input the public key PK , an encryption CT and a private key SK , if the set S of attributes associated with SK satisfies the access policy RP associated with CT , the algorithm outputs a message m .

C. Updatable Encryption

Our updatable encryption scheme builds on bilinear map.

Bilinear Map: Bilinear map [21] is a mathematical tool to construct a rich types of cryptographic primitives. We present a few facts related to efficiently computable bilinear maps. Let G_1, G_2, G_T be groups of prime order p . Let g and h be generators of G_1 and G_2 respectively. Let e be a bilinear map: $G_1 \times G_2 \rightarrow G_T$. The bilinear map e has three properties: (1) for all $u \in G_1, v \in G_2$ and $x, y \in \mathbb{Z}_p$, $e(u^x, v^y) = e(u, v)^{xy}$; (2) $e(g, h) \neq 1$; (3) $(p, G_1, G_2, G_T, e, g, h)$ can be efficiently generated and the bilinear map $e: G_1 \times G_2 \rightarrow G_T$ is efficiently computable.

Our updatable encryption scheme consists of six algorithms: (USetup, UKeyGen, UEncrypt, UDecrypt, UKeyUpdate, EncUpdate). The first algorithm USetup outputs some parameters for other algorithms to use. The subsequent three algorithms (UKeyGen, UEncrypt, UDecrypt) consist of a general symmetric key encryption scheme. The last two algorithms (UKeyUpdate, EncUpdate) are designed for key updating and encryption updating respectively. The functionalities of the six algorithms are shown as follows:

- **USetup**(λ): On input a security parameter λ , the algorithm outputs a public parameter ugp and a private parameter usp .
- **UKeyGen**(usp): On input the private parameter usp , the algorithm outputs a secret key USK .
- **UEncrypt**(m, ugp, USK): On input a message m , the public parameter ugp and a secret key USK , the algorithm outputs an encryption C^{UP} .
- **UDecrypt**(C^{UP}, USK): On input an encryption C^{UP} and a secret key USK , the algorithm outputs a message m .

TABLE I
DETAILS OF UPDATABLE ENCRYPTION

- **USetup**(λ): Chooses a random number $x \in \mathbb{Z}_p$ and computes $X = e(g, h)^x$. Outputs $ugp = X$ and $usp = x$.
- **UKeyGen**(usp): Given $usp = x$, chooses a random number $a \in \mathbb{Z}_p$, computes g^a and $h^{\frac{x}{a}}$. Outputs $USK = (g^a, h^{\frac{x}{a}})$.
- **UEncrypt**(m, ugp, USK): Given $m, ugp = X$ and $USK = (g^a, h^{\frac{x}{a}})$, chooses a random number $s \in \mathbb{Z}_p$ and computes $C_1 = mX^s$ and $C_2 = (g^a)^s$. Outputs $C^{UP} = (C_1, C_2)$.
- **UDecrypt**(C^{UP}, USK): Given $C^{UP} = (C_1, C_2)$ and $USK = (g^a, h^{\frac{x}{a}})$, computes:

$$e(C_2, h^{\frac{x}{a}}) = e((g^a)^s, h^{\frac{x}{a}}) = e(g, h)^{xs} = X^s$$

Computes $m = \frac{C_1 X^s}{X^s}$ and outputs m .

- **UKeyUpdate**(USK): Given a USK , first updates USK as:

$$USK' = (g^{aa'}, h^{\frac{x}{aa'}}) \leftarrow USK = (g^a, h^{\frac{x}{a}})$$

where a' is chosen randomly from \mathbb{Z}_p . Sets $rk = a'$. Outputs USK' and rk .

- **EncUpdate**(C^{UP}, rk): Given C^{UP} and rk , updates C^{UP} as:

$$C^{UP'} = (C_1, C_2^{rk}) \leftarrow C^{UP} = (C_1, C_2)$$

where $(C_1, C_2^{rk}) = (C_1, ((g^a)^s)^{a'}) = (C_1, (g^{aa'})^s)$. Outputs $C^{UP'}$.

- **UKeyUpdate**(USK): On input a secret key USK , the algorithm outputs a new secret key USK' and a re-key rk between the old/new key pair.
- **EncUpdate**(C^{UP}, rk): On input an encryption under USK and a re-key rk , the algorithm outputs a new encryption $C^{UP'}$ under USK' . This algorithm updates an encryption under a secret key to an encryption under a new secret key by using the re-key between the two secret keys.

Table I presents the details of our updatable encryption scheme. We explain our design in three steps. We first show how to use a secret key USK to encrypt/decrypt an encryption C^{UP} . We then describe how to update USK to a new key USK' as well as a re-key rk . Finally, we describe how to use rk to update C^{UP} encrypted by USK to a new encryption $C^{UP'}$ encrypted by USK' .

USK contains two elements $(g^a, h^{\frac{x}{a}})$. In the execution of **UEncrypt**(m, ugp, USK), $ugp = X = e(g, h)^x$ is used to generate a secret X^s , which is used to hide m in an encryption element C_1 . Also, the element g^a of USK is used to generate another encryption element C_2 . The final encryption C^{UP} consists of two encryption elements (C_1, C_2) .

In the execution of **UDecrypt**(C^{UP}, USK), $h^{\frac{x}{a}}$ is used with C_2 to reconstruct the secret X^s through bilinear map. C_1 is then divided by X^s to recover the message m .

In the execution of **UKeyUpdate**(USK), both the elements $(g^a, h^{\frac{x}{a}})$ of USK are refreshed by a random number $a' \in \mathbb{Z}_p$. The distribution of the new secret key USK' is identical to a secret key generated by **UKeyGen**(usp). The re-key rk between USK/USK' is a' .

In the execution of **EncUpdate**(C^{UP}, rk), the second element C_2 of C^{UP} is refreshed by rk . By doing so, we convert the encryption under USK into a new encryption under USK' . The update operation does not require any decryption.

D. Item-Level Access Control and Revocation

In the following, we describe how to integrate our updatable encryption scheme with CP-ABE to achieve item-level access

Algorithm 1 {role attributes, PK , product data, ugp , tag token (att_T, USK_T)}

- 1: Define RP according to the role attributes
 - 2: $CT = (P, C) \leftarrow \text{Encrypt}(PK, RP, \text{product data})$
 - 3: $C^{UP} \leftarrow \text{UEncrypt}(C, ugp, USK_T)$
 - 4: Return (att_T, C^{UP}, P)
-

control and revocation. For clarity, we term “a tagged product” as “a tag” in this subsection.

Initialization: The service provider, key authority and participants need to exchange some cryptographic parameters. The key authority generates $(PK, MSK) \leftarrow \text{Setup}(\lambda)$ for its managed role attributes. It publishes PK and the role attributes and keeps MSK secret. Also, the first participant in the supply chain generates $(ugp, usp) \leftarrow \text{USetup}(\lambda)$. The participant then delegates the service provider to publish ugp and keeps usp secret. On the other hand, each participant maintains a table to store the published PK , role attributes and ugp . Each participant then selects a subset of suitable role attributes and acquires the corresponding credential from the key authority.

Tag Preparation: When a tag T enters the supply chain, the first participant generates a random number as tag attribute att_T and a secret key $USK_T \leftarrow \text{UKeyGen}(usp)$ as the corresponding credential. The participant sets the tag token as (att_T, USK_T) and saves the tag token into T . To reduce the tag storage overhead, the tag attribute att_T is derived from the secret key USK_T by computing a hash function on it, so that att_T does not need to be explicitly stored in the tag.

When T flows through the supply chain, each participant can perform three operations: data submission, data retrieval and privilege revocation by using the tag token.

Data Submission: This operation allows a participant with tag token (att_T, USK_T) to submit a product data record (att_T, C^{UP}, P) to the service provider. att_T is the tag attribute used for index purpose and (C^{UP}, P) is a policy enforced encryption with the participant specified access policy P . The record generation process is presented in Algorithm 1.

The participant defines a role based policy RP and ABE-encrypts the product data bound to RP . The resulted ABE encryption CT contains a policy part P and an encrypted data part C . The participant then encrypts C by using the secret key USK_T contained in the tag token to get the policy enforced encryption (C^{UP}, P) . Finally, the participant submits a product data record (att_T, C^{UP}, P) to the service provider.

Our system uses general symmetric encryption scheme to increase the plaintext size supported by both updatable encryption scheme and CP-ABE. In particular, the updatable encryption scheme/CP-ABE only encrypts a symmetric key and uses the key to encrypt the plaintext. For simplicity, we do not explicitly indicate the usage of the general symmetric encryption scheme.

Data Retrieval: This operation allows a participant with tag token (att_T, USK_T) to retrieve and decrypt the product data records submitted by other participants. The participant uses the tag attribute att_T as an index to retrieve all the submitted product data records about the tag and tries to decrypt them.

Algorithm 2 {product data record (att_T, C^{UP}, P) , tag token (att_T, USK_T) , PK , credential of role attributes}

- 1: $C \leftarrow \text{UDecrypt}(C^{UP}, USK_T)$
 - 2: $CT \leftarrow (P, C)$
 - 3: $m \leftarrow \text{Decrypt}(PK, CT, \text{credential of role attributes})$
 - 4: Return m
-

Algorithm 3 {product data record (att_T, C^{UP}, P) , (att_T, att'_T, rk) }

- 1: $C^{UP'} \leftarrow \text{EncUpdate}(C^{UP}, rk)$
 - 2: $(att'_T, C^{UP'}, P) \leftarrow (att_T, C^{UP}, P)$
 - 3: Return $(att'_T, C^{UP'}, P)$
-

For a retrieved record (att_T, C^{UP}, P) , the decryption process is presented in Algorithm 2.

The participant decrypts C^{UP} by using the secret key USK_T . The decrypted C and P forms a valid ABE encryption CT . After that, the participant uses its credential of role attributes to decrypt the ABE encryption.

Revocation: This operation allows a participant with the tag token (att_T, USK_T) of a tag T to revoke the data access privilege of its upstream participants for T . The participant first generates a new secret key USK'_T and a re-key $rk: (USK'_T, rk) \leftarrow \text{UKeyUpdate}(USK_T)$. The participant updates the tag token to (att'_T, USK'_T) by overwriting the original tag token in T . Note that the tag attribute is a hash of the secret key. As the secret key is updated from USK_T to USK'_T , the tag attribute is also changed from att_T to att'_T . The participant then uses USK'_T to generate policy enforced encryption for its product data. By doing so, the participant precludes its upstream participants from the newly uploaded data.

In addition, the participant needs to delegate the service provider to update the submitted policy enforced encryptions about T so that the downstream participants can still decrypt them with the new tag token. To do so, the participant sends (att_T, att'_T, rk) to the service provider. The latter uses the old tag attribute att_T as an index to search all the submitted product data records about T . For each searched record (att_T, C^{UP}, P) , the updating process is described in Algorithm 3.

The service provider first updates C^{UP} as $C^{UP'}$ and then updates the record (att_T, C^{UP}, P) as $(att'_T, C^{UP'}, P)$. Note that the updated record now contains the new tag attribute att'_T so that all the downstream participants can use att'_T to index this record. After the update operation, all the downstream participants can still access these product data records using the new tag token.

IV. SECURITY ANALYSIS

We first analyze the security properties of our updatable encryption scheme as the security of our system builds on these properties.

A. Security Analysis of Updatable Encryption Scheme

In the following, we prove that our updatable encryption scheme provides data privacy against revoked

participants as well as the service provider. We formally define the desired security properties based on interactive game, which consists of an adversary (a malicious revoked participant or the service provider) and a challenger (our updatable encryption scheme). The adversary interacts with the challenger to attack our scheme. We then show that our updatable encryption scheme satisfies these security properties.

The data privacy of an encryption scheme can be formalized as *IND-CPA* [27]. Comparing with the standard definition of *IND-CPA*, a revoked participant and the service provider also own additional knowledge about revoked secret key and re-keys, respectively. We thus extend *IND-CPA* to define two security properties: *Revoked Secret Key Resistant (RSKR)-IND-CPA* and *Re-Key Resistant (RKR)-IND-CPA*. We prove that our updatable encryption scheme satisfies the two security properties.

Definition 1 (RSKR-IND-CPA): An updatable encryption scheme satisfies *RSKR-IND-CPA* if all polynomial time adversaries have at most a negligible advantage to win the following 3-phase game.

Phase 1: The challenger runs the algorithms of the updatable encryption scheme to generate a public parameter UGP , a private parameter usp , a secret key USK . The challenger returns (UGP, USK) to the adversary. Upon receiving the pair, the adversary returns a polynomial bounded number q to the challenger.

Phase 2: The challenger runs the algorithms of the updatable encryption scheme to generate a sequence of secret keys $USK_1, USK_2, \dots, USK_q$ from USK . The challenger then allows the adversary to send a polynomial bounded number of queries. In each query, the adversary selects a number $i \in [1, q]$, a message m and sends (i, m) to the challenger. The challenger generates an encryption C_i^{UP} of m by USK_i and returns C_i^{UP} to the adversary.

Phase 3: The adversary selects a number $i^* \in [1, q]$, generates two equal-length messages m_0^*, m_1^* and returns a challenge (i^*, m_0^*, m_1^*) to the challenger. The challenger selects a message m_η^* from (m_0^*, m_1^*) by flipping a fair bit η . The challenger then generates an encryption $C_{i^*}^{UP}$ of m_η^* by USK_{i^*} and returns $C_{i^*}^{UP}$ to the adversary.

Finally, the adversary returns a guess η' . If $\eta = \eta'$, the adversary wins the game. We define the adversary's advantage in the above game as $|P[\eta = \eta'] - \frac{1}{2}|$.

Theorem 1: Our updatable encryption scheme satisfies *RSKR-IND-CPA* based on the Decisional Bilinear Diffie-Hellman (DBDH) assumption.

DBDH Assumption: Let $y_1, y_2, y_3, z \in Z_p$ be chosen at random. Let g and h be the generators of G_1 and G_2 respectively. The DBDH assumption [21] is that no probabilistic polynomial-time adversary \mathcal{B} can distinguish the tuple $(g, g^{y_1}, g^{y_3}, h, h^{y_1}, h^{y_2}, e(g, h)^{y_1 y_2 y_3})$ from the tuple $(g, g^{y_1}, g^{y_3}, h, h^{y_1}, h^{y_2}, e(g, h)^z)$ with more than a negligible advantage. The advantage of \mathcal{B} is:

$$\begin{aligned} & |P[\mathcal{B}(g, g^{y_1}, g^{y_3}, h, h^{y_1}, h^{y_2}, e(g, h)^{y_1 y_2 y_3})] \\ & = 0 - P[\mathcal{B}(g, g^{y_1}, g^{y_3}, h, h^{y_1}, h^{y_2}, e(g, h)^z)] = 0| \end{aligned}$$

where the probability is taken over the random choice of the generators g and h , the random choice of y_1, y_2, y_3, z in Z_p , and the random bits used by \mathcal{B} .

Proof: Suppose there exists a polynomial time adversary \mathcal{A} that can break *RSKR-IND-CPA* of our updatable encryption scheme with advantage ϵ . We can construct an adversary \mathcal{B} to break the DBDH assumption with advantage $\frac{\epsilon}{2}$ as follows:

The challenger of DBDH game selects three groups: G_1, G_2, G_T with an efficient bilinear map e and generators of G_1 and G_2 : g, h . The challenger flips a fair bit μ . If $\mu = 0$, the challenger generates $(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Z) = (g, g^{y_1}, g^{y_3}, h, h^{y_1}, h^{y_2}, e(g, h)^{y_1 y_2 y_3})$; otherwise it generates $(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Z) = (g, g^{y_1}, g^{y_3}, h, h^{y_1}, h^{y_2}, e(g, h)^z)$. The challenger then gives the generated $(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Z)$ to \mathcal{B} .

Setup: \mathcal{B} computes $X = e(Y_2, Y_6) = e(g, h)^{y_1 y_2}$. \mathcal{B} chooses a random number $\beta \in Z_p$ and computes $(Y_2)^\beta = g^{y_1 \beta}$ and $Y_5^{\frac{1}{\beta}} = h^{\frac{y_1}{\beta}} = h^{\frac{y_1 y_2}{y_2 \beta}}$. \mathcal{B} sets $UGP = X$ and $USK = (g^{y_1 \beta}, h^{\frac{y_1 y_2}{y_1 \beta}})$ and returns (UGP, USK) to \mathcal{A} . The distribution of the constructed (UGP, USK) is identical with the valid ones generated by the algorithms of updatable encryption scheme. After that, \mathcal{B} receives q from \mathcal{A} .

Query: \mathcal{B} maintains a list with size q . When receiving a query (i, m) from \mathcal{A} , \mathcal{B} checks if there exists a pair (i, β_i) in the list. If yes, \mathcal{B} chooses a random number $s \in Z_p$ and constructs an encryption C_i^{UP} :

$$C_i^{UP} = (C_{i1}, C_{i2}) = (mX^s, (g^{\beta_i})^s)$$

If no, \mathcal{B} chooses a random number $\beta_i \in Z_p$, adds (i, β_i) into the list and generates an encryption C_i^{UP} as above. \mathcal{B} then returns C_i^{UP} to \mathcal{A} . The distribution of the constructed encryption is identical to the valid ones generated by the algorithms of updatable encryption scheme with the secret key USK_i .

Challenge: \mathcal{A} submits a challenge (i^*, m_0^*, m_1^*) to \mathcal{B} . \mathcal{B} flips a fair bit η , retrieves (i^*, β_{i^*}) from its maintained list and constructs an encryption $C_{i^*}^{UP}$:

$$C_{i^*}^{UP} = (C_{i^*1}, C_{i^*2}) = (m_\eta^* Z, (Y_3)^{\beta_{i^*}})$$

If $\mu = 0$, then $Z = e(g, h)^{y_1 y_2 y_3}$. Let $s = y_3$, then we have $C_{i^*1} = m_\eta^* e(g, h)^{y_1 y_2 y_3} = m_\eta^* (e(g, h)^{y_1 y_2})^{y_3} = m_\eta^* X^s$ and $C_{i^*2} = (Y_3)^{\beta_{i^*}} = (g^{\beta_{i^*}})^{y_3} = (g^{\beta_{i^*}})^s$. The distribution of the constructed encryption is identical with the valid ones generated by the algorithms of updatable encryption scheme with the secret key USK_{i^*} . If $\mu = 1$, then $Z = e(g, h)^z$. We then have $C_{i^*1} = m_\eta^* e(g, h)^z$. Since z is random, C_{i^*1} is actually a random element of G_T from the view of \mathcal{A} and the constructed encryption reveals no information about m_η^* .

Guess: \mathcal{A} submits a guess η' of η . If $\eta' = \eta$, \mathcal{B} will output $\mu' = 0$ to indicate that it was given a valid DBDH-tuple; otherwise it will output $\mu' = 1$ to indicate that it was given a random 4-tuple.

In the case where $\mu = 1$, \mathcal{A} gains no information about η . Therefore, we have $P[\eta' \neq \eta | \mu = 1] = \frac{1}{2}$. Since \mathcal{B} guesses $\mu' = 1$ when $\eta' \neq \eta$, we have $P[\mu' = \mu | \mu = 1] = \frac{1}{2}$.

In the case where $\mu = 0$, \mathcal{A} sees an encryption of m_η^* . The advantage of \mathcal{A} is ϵ . Therefore, we have $P[\eta' = \eta | \mu =$

$0] = \frac{1}{2} + \epsilon$. Since \mathcal{B} guesses $\mu' = 0$ when $\eta' = \eta$, we have $P[\mu' = \mu | \mu = 0] = \frac{1}{2} + \epsilon$.

Combing the above analysis, the overall advantage of \mathcal{B} in the DBDH game is:

$$\begin{aligned} & P[\mu = 0]P[\mu' = \mu | \mu = 0] + P[\mu = 1]P[\mu' = \mu | \mu = 1] - \frac{1}{2} \\ &= \frac{1}{2}P[\mu' = \mu | \mu = 0] + \frac{1}{2}P[\mu' = \mu | \mu = 1] - \frac{1}{2} \\ &= \frac{1}{2}P[\eta' = \eta | \mu = 0] + \frac{1}{2}P[\eta' \neq \eta | \mu = 1] - \frac{1}{2} \\ &= \frac{1}{2}(\frac{1}{2} + \epsilon) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} = \frac{\epsilon}{2}. \quad \blacksquare \end{aligned}$$

Definition 2 (RKR-IND-CPA): An updatable encryption scheme satisfies *RKR-IND-CPA* if all polynomial time adversaries have at most a negligible advantage to win the following 3-phase game.

Phase 1: The challenger runs the algorithms of the updatable encryption scheme to generate a public parameter UGP , a private parameter usp , a secret key USK . The challenger returns UGP to the adversary. Upon receiving it, the adversary returns a polynomial bounded number q to the challenger.

Phase 2: The challenger runs the algorithms of the updatable encryption scheme to generate a sequence of secret keys $USK_1, USK_2, \dots, USK_q$ started from USK and record the corresponding re-keys rk_1, rk_2, \dots, rk_q . The challenger then allows the adversary to send a polynomial bounded number of queries. In each query, the adversary selects to send j where $j \in [1, q]$, or (i, m) where $i \in [0, q]$ and m is a message. Upon receiving j , the challenger returns rk_j . Upon receiving (i, m) , if $i \in [1, q]$, the challenger generates an encryption C_i^{UP} of m by USK_i and returns C_i^{UP} ; if $i = 0$, the challenger generates an encryption C_0^{UP} of m by USK and returns C_0^{UP} .

Phase 3: The adversary generates two equal-length messages m_0^*, m_1^* and returns a challenge (m_0^*, m_1^*) to the challenger. The challenger selects one of the two messages by flipping a fair bit η , generates an encryption C^{UP} of m_η^* by USK and returns C^{UP} to the adversary.

Finally, the adversary returns a guess η' . If $\eta = \eta'$, the adversary wins the game. We define the adversary's advantage in the above game as $|P[\eta = \eta'] - \frac{1}{2}|$.

Theorem 2: Our updatable encryption scheme satisfies *RKR-IND-CPA* based on the DBDH assumption.

Proof: Suppose there exists a polynomial time adversary \mathcal{A} that can break *RKR-IND-CPA* of our updatable encryption scheme with advantage ϵ . We build an adversary \mathcal{B} to break the DBDH assumption with advantage $\frac{\epsilon}{2}$. In the following, we just describe the difference with the proof of Theorem 1.

Similar with the proof of Theorem 1, the challenger of DBDH game gives \mathcal{B} proper $(Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Z)$ based on flipping result of a fair bit μ .

In the setup phase, \mathcal{B} computes $X = e(Y_2, Y_6) = e(g, h)^{y_1 y_2}$. \mathcal{B} chooses a random number $\beta \in Z_p$ and computes g^β . \mathcal{B} sets $UGP = X$ and $USK = (g^\beta, h^{\frac{y_1 y_2}{\beta}})$. Note that \mathcal{B} does not have enough knowledge to compute $h^{\frac{y_1 y_2}{\beta}}$. However, as \mathcal{B} does not need to show USK to \mathcal{A} , \mathcal{A} will not

be aware of this event. \mathcal{B} then returns UGP to \mathcal{A} . After that, \mathcal{B} receives q from \mathcal{A} .

In the query phase, \mathcal{B} chooses a sequence of random numbers $\alpha_1, \alpha_2, \dots, \alpha_q \in Z_p$. The distribution of the constructed sequence is identical with the sequence of re-keys rk_1, rk_2, \dots, rk_q generated by the algorithms of the updatable encryption scheme. \mathcal{B} then informs \mathcal{A} to send queries. When receiving j , \mathcal{B} returns α_j . When receiving (i, m) , \mathcal{B} chooses a random number $s \in Z_p$ and constructs an encryption C_i^{UP} :

$$\begin{aligned} C_i^{UP} &= (C_{i1}, C_{i2}) = (mX^s, (g^{\beta \alpha_1 \alpha_2 \dots \alpha_i})^s) \text{ if } i \in [1, q] \\ C_i^{UP} &= (C_{i1}, C_{i2}) = (mX^s, (g^\beta)^s) \text{ if } i = 0 \end{aligned}$$

\mathcal{B} then returns C_i^{UP} to \mathcal{A} .

In the challenge phase, \mathcal{B} constructs an encryption C^{UP} :

$$C^{UP} = (C_1, C_2) = (m_\eta^* Z, (Y_3)^\beta)$$

If $\mu = 0$, then $Z = e(g, h)^{y_1 y_2 y_3}$. Let $s = y_3$, then we have $C_1 = m_\eta^* e(g, h)^{y_1 y_2 y_3} = m_\eta^* (e(g, h)^{y_1 y_2})^{y_3} = m_\eta^* X^s$ and $C_2 = (Y_3)^\beta = (g^\beta)^{y_3} = (g^\beta)^s$. The distribution of the constructed encryption is identical to the valid ones generated by the algorithms of updatable encryption scheme with the secret key USK . If $\eta = 1$, then $Z = e(g, h)^z$. We then have $C_1 = m_\eta^* e(g, h)^z$. Since z is random, C_1 is actually a random element of G_T from the view of \mathcal{A} and the constructed encryption reveals no information about m_η^* .

Finally, the guess phase is similar with the proof of Theorem 1. The overall advantage of \mathcal{B} in the DBDH game is thus $\frac{\epsilon}{2}$. \blacksquare

B. Security Analysis of Our System

Data Privacy: In our system, participants follow double encryption paradigm to encrypt and submit their product data records (policy enforced encryptions). As a result, the service provider has to decrypt two layers of encryptions to access the underlying product data. Considering a tagged product flows through the supply chain; the *RKR-IND-CPA* property of updatable encryption scheme prevents the service provider to decrypt the outside layers of submitted policy enforced encryptions, and the security property of CP-ABE scheme prevents the service provider to decrypt the inside layers of submitted policy enforced encryptions. Double encryption paradigm also resists collusion attack between the service provider and the key authority, since the successful decryption requires tag attribute credentials.

Item-Level Access Control: Recall that our access policy is defined over both role attributes AND tag attribute. The usage of AND tag attribute part first restricts that the potential authorized participants must have processed the tag. The usage of role attributes part further restricts a desired subset of actual authorized partners from the potential authorized participants (who have processed the tag). The key authority is also excluded from accessing the product data records as it does not manage the credentials of tag attributes.

Item-Level Revocation: We need to prove that a revoked participant with old tag attribute credentials cannot decrypt any encryption associated with a new tag attribute. Considering a tagged product flows through the supply chain, the

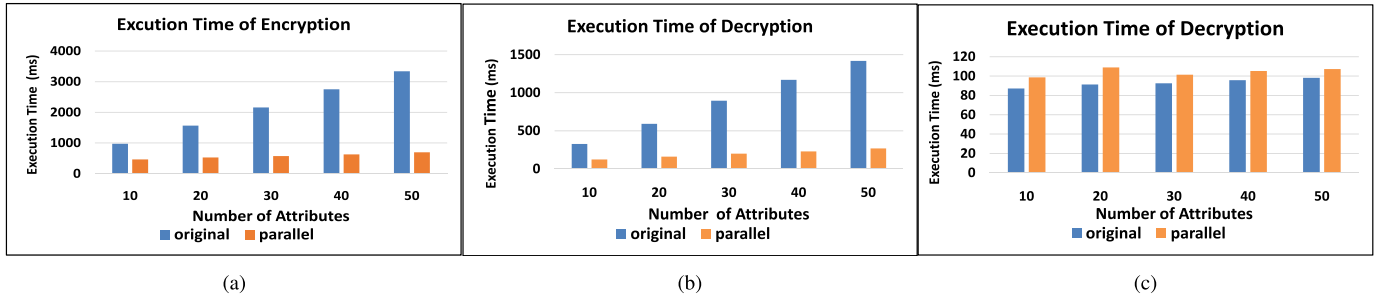


Fig. 5. Performance comparison between parallel encryption/decryption operations and encryption/decryption operations. (a) Computation overhead (ms) of encryption operation. (b) Computation overhead (ms) of decryption operation with all AND policy. (c) Computation overhead (ms) of decryption operation with all OR policy.

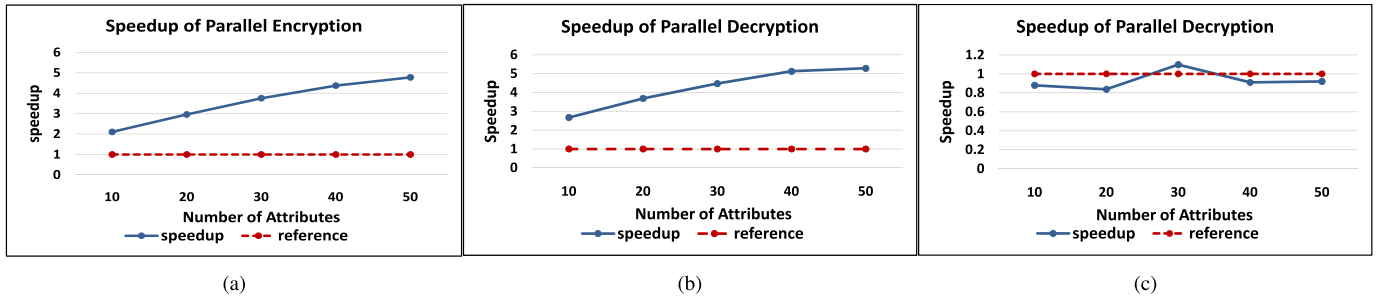


Fig. 6. Speedup ratio between parallel encryption/decryption operations and encryption/decryption operations. (a) Speedup ratio between parallel encryption operation and encryption operation. (b) Speedup ratio between parallel decryption operation and decryption operation with all AND policy. (c) Speedup ratio between parallel decryption operation and decryption operation with all OR policy.

RSKR-IND-CPA property of updatable encryption scheme prevents each revoked participant with its old tag attribute credential to decrypt any submitted policy enforced encryption associated with a new tag attribute.

V. PERFORMANCE EVALUATION

We evaluate our system in four steps. We first investigate the performance of encrypting/decrypting product data for a tagged product at the participant side. We then compare the performance of our system against CP-ABE in achieving item-level access control and item-level privilege revocation. We next study the overall performance of our system through large-scale experiments. Finally, we evaluate whether our system can meet the resource requirements of commodity C1G2 RFID system. We use a java implementation [20] of CP-ABE and implement our updatable encryption scheme based on the Pairing Based Cryptography (PBC) library [17]–[19].

Recall that our system consists of three types of entities: supply chain participant, service provider and key authority. Each entity has its own server to execute cryptographic operations. Besides, each supply chain participant owns readers to collect tag tokens, which are then sent to its server for further processing. The servers of the three types of entities are connected through network for communication. In our experiment, we adopt high-end workstations to conduct the computation tasks. The workstations are equipped with 16-core AMD Opteron Processor 6320 and 16GB RAM running Ubuntu 13.10.

A. Basic Evaluation

In our system, a participant needs to frequently encrypt its product data and decrypt other participants' product data.

Therefore, we evaluate the basic overhead of encrypting/decrypting a policy enforced encryption. As current computation tasks are often conducted on multi-core PCs or even PC clusters, we investigate how our tasks can be paralleled by multi-cores. We compare the parallel version using multi-cores and the original version using single-core. Figure 5 shows the overhead of encryption/decryption with 108 KB data.

Figure 5(a) shows the encryption overhead as a function of the number of attributes contained in the associated policy. Our findings show that the overhead of the original version grows quickly with the number of attributes and use multi-cores can greatly increase the speed (as shown in Figure 6(a)). The reason is that the CP-ABE encryption operation consists of multiple independent sub tasks, which can be assigned to multi-cores for parallel execution. We also find that the overhead of both the versions have no strict linear relation with the number of attributes. The reason is that besides the CP-ABE encryption operation, the policy enforced encryption implementation also contains many other execution steps (i.e., data read/write-back and AES encryption), which cannot be paralleled and are not affected by the number of attributes.

On the other hand, the decryption overhead depends on the type of policies associated with the encryptions. Specifically, the policy of a policy enforced encryption regulates a minimum set of role attributes that a credential of role attributes needs to be satisfied for successful decryption. Due to the design of CP-ABE, the decryption overhead is affected by the number of attributes. Figures 5(b)-(c) show two computational extreme cases: all AND policy (where all attributes need to be satisfied) and all OR policy (where at least one attribute needs to be satisfied) respectively.

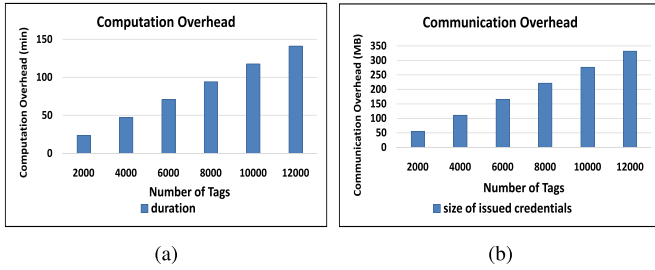


Fig. 7. Overhead of CP-ABE in achieving item-level data access control. (a) Computation overhead (min) as a function of tag number. (b) Communication overhead (MB) as a function of tag number.

In the all AND case, the overhead of the original version grows quickly with the size of the policy (as shown in Figure 5(b)). The reason is that the minimum set is all the role attributes in the policy. Instead, using multi-cores can greatly increase the speed (as shown in Figure 6(b)). The reason is that processing each of the in-set role attributes raises several sub tasks and all these tasks are independent, which facilitates the parallel execution on multiple cores. Also, the overhead of both the versions have no strict linear relation with the number of attributes, and the reason is the same with the encryption case.

In the all OR case, the decryption overhead grows slowly with the size of the policy (as shown in Figure 5(c)). The reason is that the minimum set is any one of the role attributes in the policy. Besides, using multi-cores cannot increase the speed (as shown in Figure 6(c)). The reason is that processing one role attribute raises few sub tasks and the multiple cores cannot be fully utilized. Also, we find that the performance of the parallel version is even slower than the original version. Such a performance degradation is due to thread pool initialization and thread schedule raised in the recursive parallel framework.

B. Comparison of Our System With CP-ABE

We compare the performance of our system against CP-ABE in achieving item-level access control, item-level privilege revocation and data privacy. All the experiments fully utilize the 16 cores of our PC to improve the running speed.

Access Control Overhead: CP-ABE requires the key authority to issue CP-ABE credentials for each of the participants in item-level. The storage and computation overhead of a CP-ABE credential is proportional to its associated attributes. For simplicity, we assume that each participant is described by 20 role attributes. A CP-ABE credential thus is associated with 21 attributes (20 role attributes and 1 tag attribute).

Figure 7(a) plots the computation overhead of the key authority to compute credentials of tagged products for one participant. The computation overhead grows linearly with the number of products. When 12000 tagged products flow through the supply chain, the key authority has to cost more than 2 hours to compute the credentials for each of the participants. Suppose the supply chain consists of 15 participants, the total computation delay can be as high as 30 hours.

Figure 7(b) plots the communication overhead of the key authority to distribute the credentials of tagged products for one participant. Similar to the computation overhead, the

TABLE II
REVOCATION COMPARISON IN HANDLING 6000 TAGGED PRODUCTS

Scheme	Computation overhead	Communication overhead
CP-ABE	140 mins	340 MB
Our system	27 mins	6000×168 bits≈0.96 MB

traffic volume grows linearly with the number of products. The key authority needs to distribute 340 MB credentials for each participant when 12000 tagged products flow through the supply chain. Moreover, credentials are actually secret keys for decryption purposes. These credentials thus need to be distributed through secure channels, which incurs further overhead for channel maintenance. As a large scale supply chain needs to handle millions of products, it poses heavy burdens on the key authority and renders such an approach inapplicable.

Instead, our system avoids fussy credential issuing tasks and leverages tags to locally distribute tag attribute credentials. Participants can immediately read the tag attribute credential of a product from the attached tag. By doing so, our system totally avoids the computation overhead and the communication overhead between the key authority and the participants, greatly improving product handling efficiency.

Revocation Overhead: In a revocation operation, CP-ABE relies on the key authority to issue two types of credentials for each of the unrevoked participants in item-level. For multiple tagged products and for a unrevoked participant, the computation overhead of the key authority doubles the computation overhead shown in Figure 7(a). The unrevoked participant needs to acquire an old credential and an updated credential for each tagged product to access its product data record. For the same reason, the communication overhead for credential distribution of the key authority also doubles the communication overhead shown in Figure 7(b).

Instead, our system allows the participant who conducts the revocation operation to locally update the tag tokens of tagged products and delegate the service provider to update policy enforced encryptions for them. Both of the update operations are computation efficient. The participant also needs to send a short 168-bit re-key for each tagged product to the service provider for it to update the encryptions, which is communication efficient. Table II compares the computation overhead and communication overhead of CP-ABE and our system to handle 6000 tagged products. According to the experiment results, our system dramatically reduces both the computation overhead and the communication overhead.

Data Privacy: From the above comparison, we can see that CP-ABE relies on a key authority to manage all the credentials of attributes, introducing vulnerability to product data records. If the authority is compromised or the credentials are accidentally exposed, the product data of the whole supply chain can be decrypted. As a result, all the supply chain participants have to trust the key authority to securely manage their credentials. Instead, our system relies on tags to distribute the credentials of tag attributes, which precludes the key authority from accessing the product data of the supply chain.

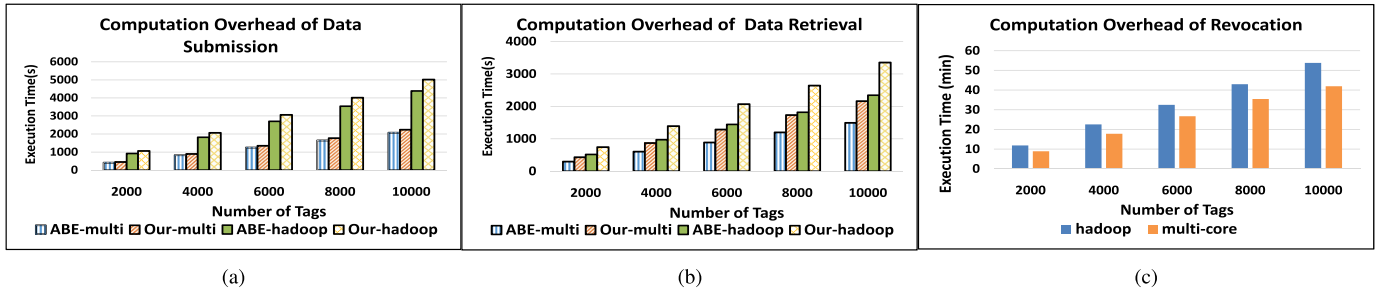


Fig. 8. Computation evaluation in the multi-cores single PC and the cluster of PCs with hadoop. (a) Computation overhead (s) of data submission as a function of tag number. (b) Computation overhead (s) of data retrieval as a function of tag number. (c) Computation overhead (min) of data privilege revocation as a function of tag number.

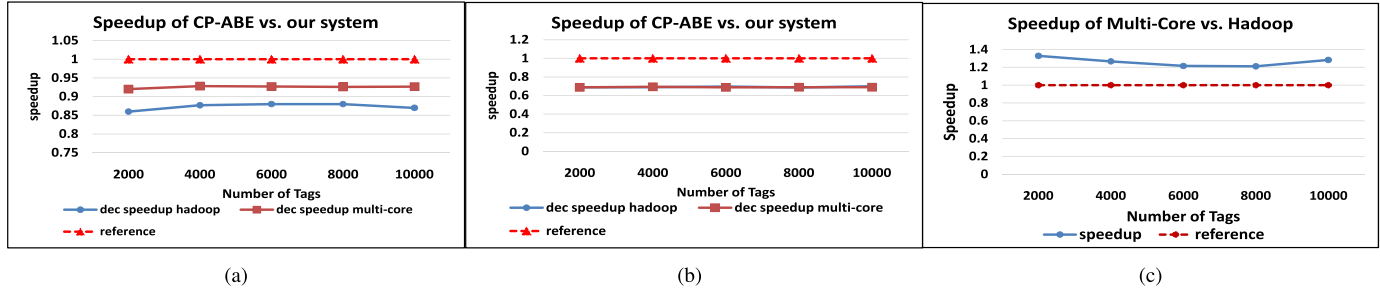


Fig. 9. Speedup evaluation in the multi-cores single PC and the cluster of PCs with hadoop. (a) Speedup of data submission between CP-ABE and our system as a function of tag number. (b) Speedup of data retrieval between CP-ABE and our system as a function of tag number. (c) Speedup of privilege revocation between multi-cores and hadoop as a function of tag number.

C. Evaluation of Our System

A supply chain often processes a large amount of tagged products. To evaluate the performance of our system in this setting, we evaluate the overhead of our system through large-scale experiments. We test our experiments on two computing platforms: a single PC and a cluster of servers. The first platform is exactly our 16-core PC; while the second platform is a cluster of three servers scheduled by hadoop infrastructure. Both the platforms use 12 CPU cores.

We evaluate the three key operations of our system, namely data submission, data retrieval and privilege revocation. We generate random product data following normal distribution. Figures 8(a)-(c) show the computation overhead incurred by the three operations when processing a large amount of tagged products. We vary the number of products from 2000 to 10000.

In the data submission, we measure the time to encrypt policy enforced encryptions for all the products. The time can be as fast as about 2246 seconds (37 minutes) for 10000 products. In the data retrieval, we measure the time to decrypt policy enforced encryptions for all the products. The time can be as fast as about 2164 seconds (36 minutes) for 10000 products. For comparison purpose, we also measure the time to encrypt/decrypt CP-ABE encryptions for all the products in data submission/retrieval in Figures 8(a)-(b). From the two figures, we can see that CP-ABE performs better than our system. We list the performance speedup of CP-ABE vs. our system in data submission/retrieval in Figures 9(a)-(b). The reason is that our design of policy enforced encryption composes of an inner layer of CP-ABE encryption and an out layer of updatable encryption, which incurs additional encryption/decryption overhead.

In the privilege revocation, we measure the time to update the policy enforced encryptions for all the products. The time can be as fast as about 41 minutes for 10000 products.

We also notice that the three key operations need not to be frequently executed by supply chain participants. Generally, each participant only needs to execute the three operations once for tagged products. When a participant receives the products, it needs to execute one data submission operation for them. If the participant wants to retrieve product data records for them, it needs to execute one data retrieval operation. Once decrypted, the participant only needs to locally store the decryption result in plaintext for future usage (such as queries). To enhance data privacy, the participant can use lightweight symmetric cipher (e.g., AES) to re-encrypt the decryption result before local storage. The participant can further decide whether to execute privilege revocation operation for the received products; and if the decision is yes, it only needs to sent re-keys to the service provider and delegates the latter to execute the update operation. Thus, we believe the computation overhead is acceptable for real applications.

In the evaluation of the three operations, the multi-core platform always outperforms the hadoop platform as shown in Figures 9(a)-(c). This is because the multi-core platform assigns higher rate of computation resource to complete the three operations, while the hadoop platform incurs higher coordination overhead, e.g., task submission, data storage, etc.

D. Implementation on Commodity C1G2 RFID Systems

We describe our implementation on commodity C1G2 RFID systems. Our system requires each tag to carry a tag token, which consists of a tag attribute and a secret key of the updatable encryption scheme. We select type f elliptic curve

TABLE III
COMMUNICATION TIME OF TAG TOKEN TRANSFER

Types of passive tags	Writing time	Reading time
ALN-9640	509 ms	107 ms
AD-224	501 ms	100 ms

of PBC library to implement the updatable encryption scheme and the secret key is 496 bits long. As the tag attribute is a hash of the secret key, it does not need to be explicitly stored in the tag. As a result, a tag token is only 496 bits long.

An RFID reader can write/read tag tokens to/from an RFID tag with the C1G2 communication primitives. In particular, we use the Write command to write tag token into RFID tags, which allows the reader to write a 16-bit data block per operation. To transfer more data, the reader needs to divide the large trunk of data into several 16-bit blocks and write them via multiple Write operations. The Read command on the other hand supports the bulk data collection, which allows the reader to collect up to 512 bits per Read operation.

We focus on the communication overhead between the Alien ALR 9900+ commodity RFID reader and two different types of widely used passive tags—ALN-9640 and AD-224 tags (both with 512-bit user memory)—in tag token transfer. As the tag tokens are transferred using the C1G2 Write/Read primitives, the performance is largely dictated by the throughput of the commodity RFID system. Table III shows the communication overhead involved in the 496-bit tag token transfer. According to the experiment results, it requires more time to write tag tokens into tags, because as mentioned the Write command only allows the reader to write a 16-bit data block per Write operation. To transfer the 496-bit tag token, the reader needs to first divide the tag token into several blocks and transfer them separately which takes longer time. In comparison, the Read operation takes less time since it only requires one Read operation to collect the whole 496-bit tag token.

VI. RELATED WORK

Besides the original CP-ABE scheme [9], there are also many other constructions of CP-ABE [28]–[32]. We briefly discuss the suitability of these schemes for our system. Many new CP-ABE schemes [29]–[32] extend the primary CP-ABE [9] from the aspects of security strength, flexibility and efficiency. Our system can immediately inherit these advantages by replacing the primary CP-ABE with these new CP-ABE schemes. Recently, Lewko and Waters [28] propose a distributed version of CP-ABE (DCP-ABE) called DCP-ABE. In their scheme, multiple authorities could claim their own attributes and manage the corresponding credentials. On the other hand, one could jointly use the credentials across them for decryption purpose. This property enables DCP-ABE a good candidate to support item-level access control. However, DCP-ABE cannot be solely used to achieve item-level privilege revocation as the access policies of its encryptions cannot be updated without decryption.

Various applications in the RFID-enabled supply chain have been proposed [6]–[8], [11], [12], which heavily rely on

a central service provider to coordinate supply chain data. Li and Ding propose to store tag data in a central database and share the data among different supply chain partners. Zanetti *et al.* [7] design a clone tag detection system, where a central server collects tracing data for tagged products when they move in the supply chain. Blass *et al.* [8] and Elkhyaoui *et al.* [11] design several tag path authentication techniques, which rely on a central manager to issue path information for participants. Kerschbaum and Sorniotti [12] propose to establish trust relations leveraging RFID tags which requires a key authority to distribute secret keys. Different from those works, our system provides an item-level data access control mechanism that defines and enforces access policies based on participant attributes and tag credential.

Another line of RFID data sharing approach is to leverage the tag as a data transmitter [33]–[36]. Kuerschner *et al.* [33] discuss the advantages of data on tag manner in the context of RFID-enabled supply chain. Pearson [34] propose to store signatures in tags so that customers can authenticate the originality of the tagged products. Juels *et al.* [35] explore the mobility of tags and use them to distribute keys on a unidirectional channel through an RFID enabled supply chain. In [36], the authors design a solution to protect privacy in RFID-enabled batch recall. In their solution, product data are continuously added into the tag memory when tags flow through the supply chain and several cryptographic primitives are used to ensure RFID data privacy. Unfortunately, the usage of these works are limited by the storage constraint of RFID tags. As a tag can only store limited data, such a data on tag paradigm is not flexible enough to support various applications of RFID supply chain.

VII. CONCLUSION

We present a scalable industry data access control system for RFID-enabled supply chain. Our new system addresses three main limitations of CP-ABE when used in the supply chain, including scalability bottlenecks in item-level data access control, compromising vulnerability at the key authority, and credential issuing overhead in item-level revocation. Our system provides two mechanisms to support scalable data access control and privilege revocation in item-level. The item-level data access control mechanism enforces access control based on role attributes and tag attributes and the item-level revocation mechanism enables a participant to revoke the data access privileges about a product of its upstream participants. Experiment results based on the large-scale simulation and the real implementation on commodity RFID systems demonstrate the scalability and efficiency of our system. For the future work, we plan to investigate security approaches to hide the access policies.

REFERENCES

- [1] *Data Sharing in the Pharmaceutical Supply Chain: A Series of Case Studies*, accessed on 2009. [Online]. Available: http://www.hcsupplychainresearch.org/WP/IBM/_whitepaper.pdf
- [2] G. M. Gaukler and R. W. Seifert, "Applications of RFID in supply chains," in *Trends in Supply Chain Design and Management*. London, U.K.: Springer-Verlag, 2007, pp. 29–48.
- [3] *GT Nexus is Now an Infor company*, accessed on 2015. [Online]. Available: <http://www.gtnexus.com/>

- [4] *Epedigree*—Wikipedia, the Free Encyclopedia, accessed on 2008. [Online]. Available: <http://en.wikipedia.org/wiki/Epedigree>
- [5] *Pharma Logistics: Can RFID Heal Supply Chain Security?* accessed on 2009. [Online]. Available: <http://www.inboundlogistics.com/cms/article/pharma-logistics-can-rfid-heal-supply-chain-security/>
- [6] Y. Li and X. Ding, "Protecting RFID communications in supply chains," in *Proc. 2nd ACM ASIACCS*, 2007, pp. 234–241.
- [7] D. Zanetti, S. Capkun, and A. Juels, "Tailing RFID tags for clone detection," in *Proc. NDSS*, 2013, pp. 1–17.
- [8] E.-O. Blass, K. Elkhyaoui, and R. Molva, "Tracker: Security and privacy for RFID-based supply chains," in *Proc. NDSS*, 2011, pp. 1–20.
- [9] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 321–334.
- [10] *Securing RFID Data for the Supply Chain*. [Online]. Available: <http://www.verisign.com/static/028573.pdf>
- [11] K. Elkhyaoui, E.-O. Blass, and R. Molva, "CHECKER: On-site checking in RFID-based supply chains," in *Proc. 5th ACM WiSec*, 2012, pp. 173–184.
- [12] F. Kerschbaum and A. Sorniotti, "RFID-based supply chain partner authentication and key agreement," in *Proc. 2nd ACM WiSec*, 2009, pp. 41–50.
- [13] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: An online social network with user-defined privacy," in *Proc. ACM SIGCOMM*, 2009, pp. 135–146.
- [14] J. A. Akinyele, M. W. Pagano, M. D. Green, C. U. Lehmann, Z. N. J. Peterson, and A. D. Rubin, "Securing electronic medical records using attribute-based encryption on mobile devices," in *Proc. ACM CCS SPSM*, 2011, pp. 75–86.
- [15] N. Santos, R. Rodrigues, K. P. Gummadi, and S. Saroiu, "Policy-sealed data: A new abstraction for building trusted cloud services," in *Proc. 21st USENIX Conf. Secur. Symp.*, 2012, p. 10.
- [16] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. Int. Conf. EUROCRYPT*, 1998, pp. 127–144.
- [17] *jPBC: Java Pairing Based Cryptography*. [Online]. Available: <http://gas.dia.unisa.it/projects/jpbc>
- [18] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proc. 16th IEEE ISCC*, 2011, pp. 850–855.
- [19] B. Lynn. *The PBC Library*, accessed on 2012. [Online]. Available: <http://crypto.stanford>
- [20] *CPABE*, accessed on 2013. [Online]. Available: <http://junwei-wang.github.io/cpabe/>
- [21] D. Boneh and X. Boyen, "Efficient selective identity-based encryption without random oracles," *J. Cryptol.*, vol. 24, no. 4, pp. 659–693, 2011.
- [22] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," in *Proc. NDSS*, 2005.
- [23] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, 2006.
- [24] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Proc. 5th ACNS*, 2007, pp. 288–306.
- [25] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption," in *Proc. 11th Int. Workshop Pract. Theory PKC*, 2008, pp. 360–379.
- [26] T. Ishiki, M. H. Nguyen, and K. Tanaka, "Proxy re-encryption in a stronger security model extended from CT-RSA2012," in *Proc. CT-RSA*, 2013, pp. 277–292.
- [27] O. Goldreich, *Foundations of Cryptography: Part 2*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [28] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. 30th Annu. Int. Conf. EUROCRYPT*, 2011, pp. 568–588.
- [29] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM CCS*, 2007, pp. 195–203.
- [30] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. 14th Int. Conf. Pract. Theory PKC*, 2011, pp. 53–70.
- [31] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Proc. 30th CRYPTO*, 2010, pp. 191–208.
- [32] A. Lewko and B. Waters, "Unbounded HIBE and attribute-based encryption," in *Proc. 30th Annu. Int. Conf. EUROCRYPT*, 2011, pp. 547–567.
- [33] C. Kuerschner, F. Thiesse, and E. Fleisch, "An analysis of data-on-tag concepts in manufacturing," in *Proc. MMS*, 2008, pp. 122–135.
- [34] J. Pearson, "Securing the pharmaceutical supply chain with RFID and public-key infrastructure (PKI) technologies," Texas Instrum., Dallas, TX, USA, White Paper RFIDPH01, Jun. 2005.
- [35] A. Juels, R. Pappu, and B. Parno, "Unidirectional key distribution across time and space with applications to RFID security," in *Proc. 17th Conf. Secur. Symp.*, 2008, pp. 75–90.
- [36] L. W. F. Chaves and F. Kerschbaum, "Industrial privacy in RFID-based batch recalls," in *Proc. 12th Enterprise Distrib. Object Comput. Conf. Workshops*, 2008, pp. 192–198.



Saiyu Qi received the B.S. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2008, and the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2014.

He is currently an Assistant Professor with the School of Cyber Engineering, Xidian University, China. His research interests include applied cryptography, cloud security, distributed systems, and pervasive computing.



Yuanqing Zheng received the B.S. degree in electrical engineering and the M.E. degree in communication and information systems from Beijing Normal University, Beijing, China, in 2007 and 2010, respectively, and the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, in 2014.

He is currently an Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University. His research interest includes human centered computing, mobile and wireless computing, and RFID systems. He is a member of IEEE and ACM.



Mo Li received the B.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 2004, and the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2009.

He is currently an Assistant Professor with the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include wireless sensor networking, pervasive computing, and mobile and wireless computing.

Dr. Li is a member of the Association for Computing Machinery (ACM). He was a recipient of the Association for Computing Machinery Hong Kong Chapter Prof. Francis Chin Research Award in 2009 and the Hong Kong ICT Award Best Innovation and Research Grand Award in 2007.



Yunhao Liu received the B.S. degree in automation from Tsinghua University, China, in 1995, and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University, USA, in 2003 and 2004, respectively. He is currently a Cheung Kong Professor and the Dean of the School of Software with Tsinghua University. He is a Fellow of IEEE and ACM. His research interests include RFID and sensor network, the Internet, and cloud computing.



Jinli Qiu received the B.S. degree in computer science from the National University of Defense Technology, China, in 2007, and the M.E. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2014.