PriExpress: Privacy-Preserving Express Delivery with Fine-Grained Attribute-Based Access Control

Tao Li
Arizona State University
Tempe, AZ, USA
tli@asu.edu

Rui Zhang
University of Delaware
Newark, DE, USA
ruizhang@udel.edu

Yanchao Zhang
Arizona State University
Tempe, AZ, USA
yczhang@asu.edu

***Abstract*—With the fast development of mobile Internet, e-commerce has been widely applied to the living of the masses. Because of the strong dependence of e-commerce, logistics industry has attracted much attention. However, when users get convenient service from the logistics industry, their privacy is compromised. Addresses, phone numbers and other private information on the parcel are accessible to anyone. Moreover, because users' logistics data is stored in plaintext in the companies' servers, it is vulnerable to the peep from staffs in the company and even the Hackers. This paper presents the first logistics system, PriExpress, which protects the users' privacy and ensures the efficient delivery of the parcel at the same time. To address the above problem, we improved attribute based encryption with a hidden access tree. Based on users' attributes, we enforce fine-grained access control on the logistic data. Our security and performance analysis shows that PriExpress is both secure and efficient.**

I. Introduction

E-commerce is gaining explosive popularity around the world and has promoted the fast development of express delivery services. For example, the parcels delivered by express delivery companies in China reached 14 billion in 2014, and the daily delivery volumes of UPS and FedEx in 2014 were over 18 million and 11 million, respectively.

Customers are enjoying the convenience of express delivery services unfortunately at the expense of personal privacy. In particular, customers' private information such as names, addresses, and phone numbers, are currently in plaintext on the parcels and thus exposed to anyone who can see the parcels. Besides, customer information is often stored in plaintext in the servers of express delivery companies which can analyze such data to pry into customers' interests and financial states in a particular time period. Moreover, data leakage may occur due to malicious employees and/or external hackers due to insecure information systems and poor business management. For example, two large express delivery companies in China were hacked, leading to the information disclosure of 14 million express delivery orders [1]. There have also been reports about online merchants selling private user information which is allegedly provided by the couriers of express delivery companies [1]. The disclosure of customers' private information may subject the victims to many attacks such as spams, crank calls, swindles, and even robbery [2].

This paper makes the first attempt to design a privacy-preserving express delivery system, called PriExpress, with three objectives. First, PriExpress is *correct* in the sense that any parcel can be correctly delivered to its destination. Second, PriExpress is *privacy-preserving* by providing privacy guarantees to parcel senders and receivers under a strong adversary model. In particular, the personal information of parcel senders and receivers is kept confidential from the express delivery company as well as all its involved couriers; and each parcel courier can only know the necessary information for fulfilling his duty. For example, assume that the parcel sender is in Chicago, while the parcel receiver is in San Diego. Any courier involved only needs to know where to deliver the parcel along the express delivery company's distribution chain from Chicago to San Diego. Finally, PriExpress is *efficient* such that it can be deployed on express users' mobile devices.

PriExpress fulfills the above objectives through the novel use of multiple cryptographic primitives. In PriExpress, each parcel sender submits to the express delivery company his express order encrypted under Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [3]. With CP-ABE, the parcel sender specifies an attribute-based access policy for enforcing fine-grained access control to his delivery order which contains sensitive personal information. Different employees of the express delivery company can only decrypt and see different portions of the express order in accordance with their respective private keys, each corresponding to a different set of attributes satisfying the access policy. For example, an intermediate courier can only know and only needs to know the next delivery station for the parcel, and other information such as the names and phone numbers of the parcel sender and receiver is unnecessary for his duty. Traditional CP-ABE [3] does not protect the user's access policy which is sent along with the ciphertext in plaintext, so the adversary can still infer significant user information from the access policy. In contrast, we propose a novel algorithm based on Private Set Intersection [4] to keep the access policy confidential while ensuring correct and efficient decryption by different entities with various access privileges. In addition, PriExpress adopts anonymous and non-interactive authentication primitives [5], [6] with four goals in mind. First, the express delivery company need not store any plaintext order or personal information of any express user. Second, the express deliver company cannot link different orders of the same express user together for profiling or tracing the user. Third, the express delivery company can ensure correct billing of the express user while withstanding DoS attacks from fake express users. Last, the parcel sender can authenticate the first courier, and the last courier can authenticate the parcel receiver.

PriExpress can be easily integrated into current express

delivery systems. For example, virtually all major express delivery companies offer apps for users to install on their mobile devices. These apps allow the users to track and receive the realtime status of their parcels, among many other convenient functionalities. A PriExpress user can then specify his access policy and send his encrypted order to the express delivery company. To deliver the parcel to the next transit point, an intermediate courier submits the current parcel location and also his access privileges. While receiving the delivery status of his parcel, the user verifies the courier's access privileges and allows the courier to decrypt the express order for knowing the next transit point if the verification succeeds. The entire process can be automated with minimal human effort.

The contributions of this paper are threefold. First, PriExpress is the first privacy-preserving express delivery system to the best of our knowledge. Second, we propose a novel policy-hiding CP-ABE algorithm which allows PriExpress users to have complete and fine-grained access control over their personal information. Finally, we thoroughly analyze the security and performance of PriExpress and confirm its high efficacy and efficiency by real experiments. Our results show that PriExpress can simultaneously ensure user privacy and correct delivery.

The rest of this paper is organized as follows. Section II introduces the crytographic primitives that PriExpress relies on. Section III introduces the system and threat models along with design goals. Section IV details the design of PriExpress. Section V discusses some practical issues of PriExpress. Sections VI and VII analyze PriExpress's security and performance, respectively. Section VIII evaluates our system through implementation. Section X concludes this paper.

II. Cryptographic Primitives

In this section, we briefly introduce some cryptographic primitives that PriExpress relies on.

A. Bilinear Map

Let $\mathbb{G}_0$ be a multiplicative cyclic group of prime order p with generator g. The bilinear map e is defined as : $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$, where $\mathbb{G}_1$ is the codomain of e. The bilinear map e has three properties:

- *Bilinearity*: for all $x, y \in \mathbb{G}_0$ and $m, n \in \mathbb{Z}_p$, $e(x^m, y^n) = e(x, y)^{mn}$.
- *Symmetry*: for all $x, y \in \mathbb{G}_0, e(x, y) = e(y, x)$.
- *Non-degeneracy*: $e(g, g) \neq 1$.

Definition 1. *The Decisional Diffie-Hellman (DDH) problem in group $\mathbb{G}_0$ of prime order p with generator g is defined as follows: on input $g, g^a, g^b, g^c = g^{ab} \in \mathbb{G}_0$, where $a, b, c \in \mathbb{Z}_p$, decide whether $c = ab$ or c is a random element.*

Definition 2. The Decisional Bilinear Diffie-Hellman (DBDH) problem in group $\mathbb{G}_0$ of prime order p with generator g is defined as follows: on input $g, g^a, g^b, g^c \in \mathbb{G}_0$ and $e(g, g)^z = e(g, g)^{abc} \in \mathbb{G}_1$, where $a, b, c \in \mathbb{Z}_p$, decide whether $z = abc$ or z is a random element.

The security of PriExpress relies on the hardness of solving the DDH or DBDH problem with polynomial-time algorithms, which is a common assumption because discrete logarithm problems are widely considered hard to solve in large number field. For example, it is hard to deduce a given g and g^a.

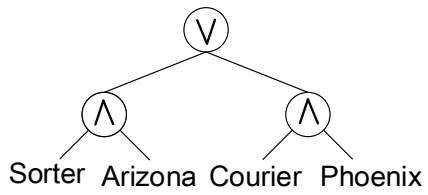


Fig. 1: An exemplary access tree

We then define Lagrange coefficient $\Delta_{m,S}$ as follow

$$\Delta_{m,S(x)} = \prod_{n \in S, n \neq m} \frac{x - n}{m - n},$$

where $m \in \mathbb{Z}_p$ and S is a set of elements in $\mathbb{Z}_p$. Lagrange coefficient will be used in the polynomial interpolation to decrypt the shared ciphertext. We also define a one-way hash function $H : \{0, 1\}^* \to \mathbb{G}_0$, which will be used as a random oracle to map the attribute value to a random element in $\mathbb{Z}_p$.

B. Ciphertext-Policy Attribute-Based Encryption(CP-ABE)

Because of the reasons stated in Section I, we employ CP-ABE as a basic component of access control. In CP-ABE [3], the ciphertext is associated with an access tree specified by the encrypter. Only if the decrypter's attributes satisfy the access tree, can he recover the plaintext.

1) *Access Tree T* : Many ABE schemes adopt access tree (e.g., Figure 1) as expressive encryption policy. Every non-leaf node x of the tree represents a threshold gate with a threshold value k_x, and each leaf node x represents an attribute $\mathsf{att}(x)$. Let $\mathsf{num}(x)$ be the number of children of node x. It follows that $0 < k_x <= \mathsf{num}(x)$. We denote by $\mathsf{parent}(x)$ and $\mathsf{children}(x)$ node x's parent node and the set of children, respectively. The children of a node x is numbered from 1 to $\mathsf{num}(x)$ from left to right and $\mathsf{index}(z)$ returns such number associated with z among $\mathsf{children}(x)$.

We define the *leaf node vector* L_T of a tree T is formed by the leaf nodes under inorder traversal. Then, the *attribute vector* A_T of a tree T is formed by the attributes represented by the nodes in L_T. In addition, the attribute vector A_u of a user is formed by all of his attributes in an arbitrary order.

We say that a leaf node is satisfied if a key contains the corresponding attribute the node represents, and a non-leaf node is satisfied if at least k_x of its children are satisfied. Recursively, if the root node of the tree is satisfied, we say the tree is satisfied. For example, in Figure 1, attributes set of a sorter in Arizona or a courier in Phoenix can satisfy the access tree.

2) *Definition*: We now describe some components of the CP-ABE scheme briefly:

$\mathsf{Setup} \to (PK, MK)$. The setup algorithm takes some implicit security parameters as input and outputs the global public key PK and the master key MK. The master key belongs to the key issuer and is kept secret.

TABLE I: Notation for Access Tree

k_x	threshold value of the node x
$\mathsf{num}(x)$	number of child nodes of x
$\mathsf{att}(x)$	attribute node x represents, if it is a leaf node
$\mathsf{index}(x)$	index of node x in the children of its parent
$\mathsf{parent}(x)$	node x's parent node
A_T	attribute vector of tree T

$\mathsf{Encrypt}(PK, M, T) \rightarrow CT$. The encryption algorithm takes as input the three parameters: the public key PK, a plaintext message M, and an access tree T and outputs a ciphertext CT. Only the user with the key satisfying the access tree T can decrypt CT.

$\mathsf{KeyGenerate}(PK, MK, S) \rightarrow SK$. The Key Generation algorithm takes as input the public key PK, the master key MK and an attribute set S and outputs a private key SK that contains the attributes in S. The private key is used by users to decrypt ciphertext.

$\mathsf{Decrypt}(PK, SK, CT) \rightarrow M$. The decryption algorithm takes as input the public key PK, a private key SK with corresponding attribute set S, and a ciphertext CT with the encrypter defined access tree T. It outputs the original plaintext message M if and only if the attribute set S satisfies the access tree T.

C. Paillier Cryptosystem

Our PriExpress also relies on the Paillier cryptosystem [7] that comprises the following components.

- **Key generation.** An entity generates two large primes p and q and computes $N = pq$ and $\lambda = \mathrm{lcm}(p-1, q-1)$. It then selects a random $g \in \mathbb{Z}^*_{N^2}$ such that $\gcd(\mathsf{L}(g^\lambda \mod N^2), N) = 1$, where $\mathsf{L}(x) = (x-1)/N$. The Paillier public and private keys for the entity are $\langle N, g\rangle$ and λ, respectively.
- **Encryption.** Let $m \in \mathbb{Z}_N$ be a plaintext and $r \in \mathbb{Z}_N$ a random number. The ciphertext is computed as

$$\mathsf{E}(m, r) = g^m r^N \mod N^2 , \tag{1}$$

  where $\mathsf{E}(\cdot)$ denotes the Paillier encryption operation.
- **Decryption.** Given a ciphertext $c \in \mathbb{Z}_{N^2}$, the plaintext can be computed as

$$\mathsf{D}(c) = \frac{\mathsf{L}(c^\lambda \mod N^2)}{\mathsf{L}(g^\lambda \mod N^2)} \mod N , \tag{2}$$

  where $\mathsf{D}(\cdot)$ denotes the Paillier decryption operation.

The Paillier's cryptosystem has two very useful properties.

- **Homomorphic.** For any $m_1, m_2, r_1, r_2 \in \mathbb{Z}_N$, we have

$$\mathsf{E}(m_1, r_1)\mathsf{E}(m_2, r_2) = \mathsf{E}(m_1 + m_2, r_1 r_2) \mod N^2,$$
$$\mathsf{E}^{m_2}(m_1, r_1) = \mathsf{E}(m_1 m_2, r_1^{m_2}) \mod N^2 .$$

- **Self-blinding.**

$$\mathsf{E}(m_1, r_1) r_2^N \mod N^2 = \mathsf{E}(m_1, r_1 r_2) ,$$

  which implies that any ciphertext can be changed to another without knowing the plaintext.

Since the random number r in a ciphertext $\mathsf{E}(m, r)$ does not affect decryption or other homomorphic operations, we subsequently use $\mathsf{E}(m)$ instead of $\mathsf{E}(m, r)$ in the remaining paper.

D. Anonymous and Non-interactive Authentication

PriExpress also uses anonymous and non-interactive authentication [5], [6]. Anonymous and non-interactive authentication allows one party, say Alice, to prove to another party, say Bob, that some statement is true without revealing her identity to Bob.

Here we briefly introduce the two techniques used in this paper: non-interactive zero-knowledge proof (NIZK) and witness-indistinguishable (NIWI). We refer interested readers to [5] for more formal definitions. Given a computable ternary relation $\mathcal{R}$, we represent each tuple as (crs, n, w), where crs is a global reference string, n is a statement to be proved, and w is the witness. We also let $\mathcal{L}$ denote the language consisting of all the statements in $\mathcal{R}$. Assume that $\mathcal{R}$ consists of three polynomial time algorithms $(\mathcal{K}, \mathcal{P}, \mathcal{V})$, where $\mathcal{K}$ is crs generation algorithm, $\mathcal{P}$ and $\mathcal{V}$ are prover and verifier, respectively. $\mathcal{P}$ takes a tuple (crs, n, w) as input and outputs a proof π. When there is need to verify a published π, $\mathcal{V}(crs, n, w)$ will output 1 if the proof is acceptable and 0 otherwise.

The proof system $(\mathcal{K}, \mathcal{P}, \mathcal{V})$ needs satisfy two properties: completeness and soundness. The former means that if a statement is true, an honest verifier can always be convinced of this fact by an honest prover. The latter means that a cheating prover can convince an honest verifier that a false statement is true with negligible probability. To prevent tracing attack from adversaries, NIWI should guarantee that an adversary cannot differentiate between a true crs and a simulated crs. Finally, we rely on NIWI to guarantee that no additional information can be obtained by the verifier except the output generated from the statement. Due to space limitations, we do not describe the algorithms in detail and will use them directly at some place. We refer readers to [5], [6] for more details.

III. System and Adversary Models

A. System Model

As shown in Fig. 2, our system consists of four types of entities: *Attribute Authority*, *PriExpress Operator*, *Customer*, and *Courier*.

- *Attribute Authority (AA)* is a trusted entity administrated by the government or large third-party organization with abundant computation capability. It is in charge of issuing CP-ABE keys to couriers after verifying their identities and attributes.
- *PriExpress Operator* provides parcel delivery service to the customers. It accepts orders from customers and distributes orders to appropriate couriers for shipping and delivery. When a parcel arrives at a new location, PriExpress operator receives status update of the parcel and forwards it to the customer.

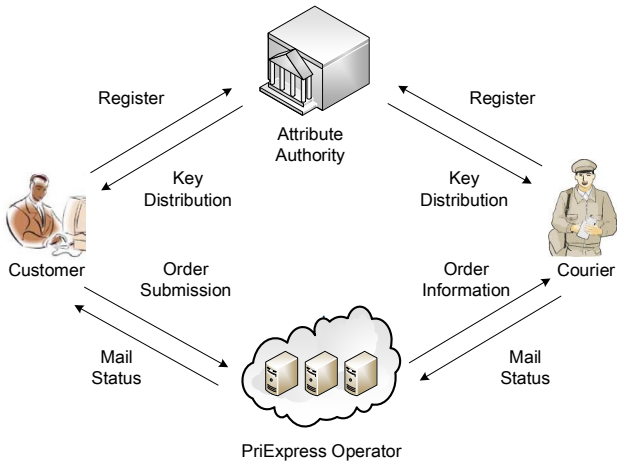


Fig. 2: PriExpress System Model

- *Customer*s are senders or receivers of parcels. To get keys and credentials, they should register with the authority first. Then they can use the keys and credentials to get service from the PriExpress operator and couriers after success of authentication.
- *Couriers* are the entities who actually moving parcels between senders and receivers. They are employed by PriExpress operator and have to register with AA to get corresponding CP-ABE keys according to their attribute sets. They are responsible to deliver the parcel in their corresponding scopes.

Sometimes, we use *user* to represent *Customer* and *Courier* when no confusion arise.

When a customer has a parcel to send, he first sends an encrypted order to PriExpress via designated app installed on this smartphone. The order consists of multiple pieces of information, including sender and receiver's names, addresses (e.g., country, state, city, street address, and zip code), phone numbers, etc. Each piece of information has different privacy requirement and should be accessed only by the courier with specific attributes. Then the customer goes to a local store of PriExpress and drops the parcel to the courier. Each intermediate courier decrypts the encrypted order and obtains the information he is designated to learn to deliver the parcel to next station. The sorter in next station extracts the order information from the server and decrypts the parts he can to continue the delivery. When the parcel is scanned or delivered in a station, the courier/sorter will send encrypted updates to the customer via PriExpress sever. The parcel finally arrives to the last courier and the receiver gets the parcel after being authenticated.

B. Threat Model

We assume that AA is trusted. In contrast, PriExpress Operator is Honest-but-Curious (HbC), which will faithfully follow the system operation but may be interested in learning customers' information such as addresses and profiles.

In addition, both the customer and courier are untrusted. They will find other user's information as much as they can with the intention such as selling information to make money or just curious abut others. They may collude with each other to enlarge their privileges.

C. Design Goals

We design PriExpress with the following goals in mind.

- *Correct.* The whole information on the parcel can and can only be decrypted by targeted users. Also, the express will be delivered *correctly* to the targeted user following the protocol. If couriers transfer the parcel to a wrong station, then couriers in next station can detect the mistakes and return the parcel or just continue the delivery to a correct next station.
- *Privacy-preserving*. The PriExpress Operator can only see the encrypted order information, and he cannot decrypt these information. Both the customer and courier are anonymous to the PriExpress Operator, so he cannot infer anything on the link between customer and courier. The courier can only decrypt the information part which is necessary for correct parcel delivery corresponding to his particular attributes. Customers can only decrypt information on parcel targeted to himself and cannot decrypt any information targeted to others. Both couriers and customers cannot enlarge their privileges by colluding with other users.

TABLE II: Notation for PriExpress

u	a user (either Customer or Courier)
$\mathbb{A}_u$	attribute set of user u
A_u	attribute vector of user u
o_j	the j's part of the order
T_j	access tree of o_j
CT_j	ciphertext of o_j under our ABE
$\mathsf{E}(m)$	ciphertext of m under *Paillier Cryptosystem*

IV. PriExpress System

In this section, we present the design of PriExpress, which consists of six phases: *setup*, *key and credential generation*, *encrypted order submission*, *payment and parcel drop-off*, and *intermediate parcel handling*.

A. Setup

The AA chooses a bilinear group $\mathbb{G}_0$ of prime order p with generator g and two random exponents $\alpha, \beta \in \mathbb{Z}_p$. Then the authority generates the public and master keys of the whole system as $PK = \langle \mathbb{G}_0, g, h = g^\beta, e(g,g)^\alpha \rangle$ and $MK = \langle \beta, g^\alpha \rangle$.

B. Key and Credential Generation

Customers and couriers both need register at the authority to obtain keys and credentials based on their attributes stored in the authority. First, in CP-ABE, the authority generates secret key for every user in the system to decrypt message. The key generation algorithm takes as input each user u's attributes set $\mathbb{A}_u$ and outputs a secret key corresponding to that set. Then it chooses a random element $r \in \mathbb{Z}_p$ and a random $r_i \in \mathbb{Z}_p$ for each attribute $a_i \in \mathbb{A}_u$. The user's secret key is computed as $SK_u = \langle D = g^{\frac{\alpha+r}{\beta}}, \forall a_i \in \mathbb{A}_u : D_i = g^r \cdot H(a_i)^{r_i}, D'_i = g^{r_i} \rangle$.

C. Encrypted Order Submission

Assume that customer u intends to submit order O to PriExpress. Without loss of generality, assume that the order O consists of m pieces of information, denoted by $O = \langle o_1, \ldots, o_m \rangle$. User u encrypts each o_j in the following steps.

First, customer u defines an access tree T_j and generates a corresponding trimmed access tree $\hat{T}_j$ by replacing every leaf node in T_j by special symbol "null".

Second, for each node $x \in T_j$, customer u constructs a polynomial f_x of degree $d_x = k_x - 1$ by specifying $d_x + 1$ points, where k_x is the threshold value of node x. There are two cases.

- Case 1: if node x is the root of T_j, then randomly pick $s_j \in Z_p$ and set $f_x(0) = s_j$ and chooses d_x remaining points randomly.
- Case 2: if node x is not the root of T_j, then set $f_x(0) = f_{\mathsf{parent}(x)}(\mathsf{index}(x))$ and chooses d_x points remaining randomly.

Third, let $A_j = (a_{j,1}, \ldots, a_{j,\lambda_j})$ be the corresponding corresponding attribute vector of access tree T_j, where λ_j is the number of leaf nodes in T_j. The ciphertext CT_j is computed as

$$\begin{aligned} CT_j = &\langle \widetilde{C} = o_j \cdot e(g,g)^{\alpha s}, C = h^s, \\ &\forall k \in [1, \lambda_j], C_k = g^{f_k(0)}, C'_k = H(a_{j,k})^{f_k(0)} \rangle, \end{aligned}$$

where we have abused the notation to let $f_k()$ denote the polynomial constructed for the node corresponding to attribute $a_{j,k}$.

Fourth, the customer constructs a polynomial $Q_j(z)$ with roots being the elements of A_j:

$$Q_j(z) = \prod_{k=1}^{\lambda_j} (z - H(a_{j,k})) = \sum_{k=0}^{\lambda_j} \chi_{j,k} z^k \ .$$

Fifth, the customer u computes $\mathsf{E}(\chi_{j,0}), \ldots, \mathsf{E}(\chi_{j,\lambda_j})$ using Palliar public key pk.

After repeating the above five steps for every o_j, the customer finally performs anonymous authentication with PriExpress and submits the following information to PriExpress.

- m ciphertexts $CT_1, \ldots, CT_m$.
- m trimmed access trees $\hat{T}_1, \ldots, \hat{T}_m$.
- Ciphertext of every coefficients $\{\mathsf{E}(\chi_{j,0}), \ldots, \mathsf{E}(\chi_{j,\lambda_j})\}_{j \in \{1,\ldots,m\}}$.
- His Palliar public key pk.

On receiving the encrypted order, PriExpress stores the order in its database and returns a two-dimension bar-code that the customer can print out and attach to the parcel.

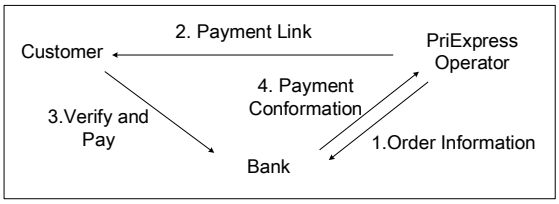


Fig. 3: Payment protocol to avoid being traced.

D. Payment and Parcel Drop-off

After submitting the encrypted order, the customer proceeds to pay and drop off the parcel at a PriExpress local store. To allow local PriExpress staff to calculate the rate, the customer can allow the staff to decrypt the destination zip code by properly setting the access tree. The customer can either pay cash to avoid being tracked by his bank account or execute a simple payment protocol shown in Fig. 3. In particular, PriExpress can send the price information to a designated bank, which returns a payment link. PriExpress can forward the link to the customer. After the customer makes the payment, the bank returns a payment conformation to PriExpress.

E. Intermediate Parcel Handling

We now illustrate how the parcel is handled by an intermediate courier c with attribute set $\mathbb{A}_c = \{a_1, \ldots, a_\delta\}$, where δ is the number of c's attributes.

First, the courier scans the two-dimension barcode on the parcel using his smartphone. The PriExpress app installed on his phone then sends an information request for the order along with his credential.

Second, on receiving the request, the PriExpress verifies the credential of the courier. If succeed, PriExpress sends the the detail of order $\{CT_j\}_{j \in \{1,\ldots,m\}}$, the corresponding trimmed access trees $\{\hat{T}_j\}_{j \in \{1,\ldots,m\}}$, the ciphertexts of all the coefficients $(\mathsf{E}(\chi_{j,0}), \ldots, \mathsf{E}(\chi_{j,\lambda_j}))_{j \in \{1,\ldots,m\}}$, and the customer's Palliar public key pk to the courier.

Third, for each set of encrypted coefficients $(\mathsf{E}(\chi_{j,0}), \ldots, \mathsf{E}(\chi_{j,\lambda_j}))$, $j \in [1, m]$, the courier does the following for each of his attributes $a_k \in A_c$

1. Evaluate the polynomial $Q_j(H(a_k))$ using pk and the homomorphic properties of the Paillier's cryptosystem
2. Choose a random element $r_{j,k} \in \mathbb{Z}_p$ and computes $\mathsf{E}(r_{j,k} \cdot Q_j(H(a_k)) + H(a_k))$ using the homomorphic properties of the Paillier's cryptosystem.

As a result, for each polynomial $Q_j(z), j \in [1, m]$, the courier obtains a vector $V_j = (\mathsf{E}(r_{j,1} \cdot Q_j(H(a_1)) + H(a_1)), \ldots, \mathsf{E}(r_{j,\delta} \cdot Q_j(H(a_\delta)) + H(a_\delta)))$. He then sends $V_1, \ldots, V_m$ to the customer u via PriExpress.

For each received $V_j, j \in [1, m]$, the customer u uses his Paillier private key sk to decrypt it and obtain the plaintext $P_j = [p_{j,1}, \ldots, p_{j,\delta}]$, where $p_{j,k} = r_{j,k} \cdot Q_j(H(a_k)) + H(a_k)$. It is easy to see that $p_{j,k} = H(a_k)$ if $a_k \in A_j$.

The customer then constructs a vector $Y_j = (y_{j,1}, \ldots, y_{j,\lambda_j})$, where

$$y_{j,k} = \begin{cases} l & \text{if there exists } l \in [1,\delta] \text{ s.t. } H(a_{j,k}) = p_{j,l}. \\ \mathsf{null} & \text{otherwise,} \end{cases}$$

for all $k \in [1, \lambda_j]$. In other words, y_i indicates that for attribute $a_{j,i} \in A_j$ whether courier c has some matching attribute $a_k \in A_c$. The customer can then determine based on Y_j whether the courier's attribute set A_c satisfies the access tree T_j. If so, the customer sends Y_j to the courier via PriExpress server. Otherwise, the customer notifies PriExpress about the failure and terminates the protocol.

On receiving each Y_j, the courier proceeds to decrypt each ciphertext CT_j using a recursive algorithm $\mathsf{DecryptNode}(CT_j, SK, x)$, where x is a node in T_j corresponding to ciphertext CT_j and SK is courier's CP-ABE decryption key. $\mathsf{DecryptNode}(CT_j, SK, x)$ is defined according to the following three cases.

- Case 1: If x is a leaf node corresponding to attribute a_k and $y_{j,k} \neq \mathsf{null}$. Then

$$\begin{aligned} \mathsf{DecryptNode}(CT_j, SK, x) &= \frac{e(D_k, C_x)}{e(D'_k, C'_x)} \\ &= \frac{e(g^r \cdot H(a_k)^{r_k}, g^{f_x(0)})}{e(g^{r_k}, H(a_k)^{f_x(0)})} \\ &= e(g,g)^{r f_x(0)} \end{aligned}$$

- Case 2: If x is a leaf node corresponding to attribute a_k and $y_{j,k} = \mathsf{null}$, then $\mathsf{DecryptNode}(CT_j, SK, x) = \perp$.

- Case 3: If x is not a leaf node, the algorithm proceeds as follows: For every node v that is a child of x, it calls $\mathsf{DecryptNode}(CT_j, SK, v)$ and stores the output as F_v. Let S_x be an arbitrary k_x-sized set of $\mathsf{childern}(x)$ such that $F_v \neq \perp$ for all $v \in S_x$. If no such set exists then the node x was not satisfied and the function returns $\perp$. Otherwise, let $S'_x = \{z | \mathsf{index}(z) \in S_x\}$.

$$\begin{aligned} F_x &= \prod_{v \in S_x} F_v^{\triangle_{\mathsf{index}(v), S'_{x_j}(0)}} \\ &= \prod_{v \in S_x} \left(e(g,g)^{r \cdot f_z(0)}\right)^{\triangle_{\mathsf{index}(v), S'_x(0)}} \\ &= \prod_{v \in S_x} \left(e(g,g)^{r \cdot f_{\mathsf{parent}(v)}(\mathsf{index}(v))}\right)^{\triangle_{\mathsf{index}(v), S'_x(0)}} \\ &= \prod_{v \in S_x} e(g,g)^{r \cdot f_{x_j}(\mathsf{index}(v)) \cdot \triangle_{\mathsf{index}(v), S'_x(0)}} \\ &= e(g,g)^{r \cdot f_x(0)} \end{aligned}$$

The courier recursively calls DecryptNode, starting from the root node r of the tree T_j. If the tree is satisfied, he can compute $DecryptNode(CT_j, SK, \mathsf{r}) = e(g,g)^{rs}$. Then he can decrypt by computing

$$\frac{\widetilde{C}}{\left(\frac{e(C,D)}{e(g,g)^{rs}}\right)} = \frac{o_j \cdot e(g,g)^{\alpha s}}{\left(\frac{e(h^s, g^{(\alpha+r)/\beta})}{e(g,g)^{rs}}\right)} = o_j$$

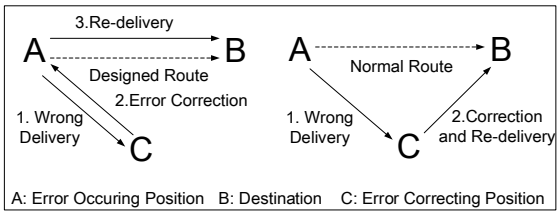


Fig. 4: Two scenarios of wrong delivery

After decrypting every o_j that he can decrypt, he delivers the parcel to the next location based on the information he learned.

V. Discussion

We now discuss some practical issues in PriExpress along with possible solutions.

A. Sender to Courier and Courier to Receiver

We now discuss how the sender securely gives its parcel to courier after successfully submitting order to PriExpress operator and how the receiver gets parcel from the couriers. The sender can goes to stores of the PriExpress Operator to give the parcel to the courier, or the courier can talk to the sender via APP in the phone to decide place for the parcel pick-up. The sender can use anonymous proof to authenticate the courier with the help of PriExpress Operator. Similarly, the courier can authenticate the receiver with the help of PriExpress Operator.

B. Wrong Delivery

In practice, parcels are sorted and delivered manually by couriers, so it is difficulty to avoid occasional sorting error in PriExpress. For example, a parcel at location A intended for location B could be falsely delivered to location C. The sender of the parcel can designate in the access tree the couriers at which places can decrypt the order when the package is on the way to destination. That is, the sender can designate the route the package can go through. Also, he can only designate the origin and end of the package. We now discuss when error happens in the above two scenarios. As illustrated in Figure 4(left), when the route is designated, an error delivery will be detected in position C just after position A, because couriers out of the route cannot decrypt the order. As a result, the couriers in C need to return the package to location A and couriers in A will sort the package again. When the route is not designated (figure 4(right)), couriers in the wrong position C just need to continue the delivery to destination. Actually, in this scenario, the wrong delivery may not to be detected.

C. Collecting Parcel for Others

Customer sometimes cannot collect parcel by himself and has to ask friends for help. As a result, we must make our friends successfully authenticated by the courier. In our system, we just need to give our anonymous credential to friends. Anonymous credentials are generated by ourself based on the real credential issued by authority and can only be used once. When the order is finished, the corresponding credential will

expire, so the friends cannot use it again. Of course, customer must keep the credential well before the parcel is successfully received, or the parcel may be collected by impostor.

VI. Security Analysis

A. Attributes and Identity Privacy

Attributes are the important privacy in our system, because most of customers are identifiable with their attributes. We assume *Paillier Cryptosystem* is semantically secure. All the attributes of customer are processed under *Paillier Cryptosystem* before sending them to PriExpress server which can only see ciphertext of coefficients $(\mathsf{E}(\chi_{j,0}), \ldots, \mathsf{E}(\chi_{j,\lambda_j}))_{j\in\{1,\ldots,m\}}$. When encrypting these attributes using *Paillier Cryptosystem*, the customer will choose a random r, so PriExpress server will get different ciphertexts on the same coefficient each time. As a result, the PriExpress operator cannot launch tracing or exhaustive search even the attribute space is small. The courier will get ciphertext of coefficients and he can do some matching computation on the ciphertext to get $\mathsf{E}(r \cdot Q_j(H(a_i)) + H(a_i))$. However, the courier cannot decrypt the matching result and can only wait for the customer for answers. Only when the courier's credentials of attributes are successfully verified by the customer and attributes satisfy the access tree will the customer return matching result. With random r, the customer only knows the intersection of attributes with the access tree. Note that, the couriers is anonymous but verifiable to the PriExpress severs, so PriExpress severs cannot infer identity of customer or corresponding courier. From the analysis above, both the PriExpress server and courier cannot get extra information about attributes in the access policy.

In addition to the protection on attributes, PriExpress can also prevent attackers from forging credentials to launch impersonation attack. Like [6], legal provers can generate anonymous credentials himself based on the credential issued by authority. Each anonymous credential can only be used one time, so attackers cannot impersonate others using existing credentials. Also, attackers cannot create acceptable anonymous credential himself because he does not get legal credential from the authority. As a result, couriers cannot learn more attributes in the access tree via the intersection $A_j \bigcap A_c$ by forging their attributes. Similarly, PriExpress server cannot create courier accounts with fake credentials to pass customer's verification.

B. Logistic Data Confidentiality against Collusion Attack

Customers encrypt their orders before submitting them to customer server. Couriers with appropriate attributes and keys can only decrypt a small part in the ciphertext. Couriers may collude with each other and exchange their attributes and keys to enlarge they privilege. For example, couriers c_1, c_2 and c_3 with attribute sets $\mathbb{A}c_1$, $\mathbb{A}c_2$ and $\mathbb{A}c_3 = \mathbb{A}c_1 \bigcup \mathbb{A}c_2$, respectively. Courier c_1 and c_2 may want to collude to obtain the secret key of c_3. To do so, c_1 and c_2 must recover $e(g,g)^{\alpha s}$ to decrypt the ciphertext. To get $e(g,g)^{\alpha s}$, they must pair C in the ciphertext with D in the secret key. However, keys from different couriers are randomized by different r, so the customer' ciphertext cannot be decrypted even they collude. In addition, when the PriExpress severs colludes with a legal courier, their combined information cannot be enlarged as well. The PriExpress sever can only get the intersection as the courier which can only be linked to a one-time anonymous customer name.

VII. Performance Analysis

In this section, we analyze complexity of each component of PriExpress.

Setup. During system setup, a global public key and a master key are generated. This is a one-time process and only involves limited numbers of exponentiations and bilinear maps computation, so the total complexity is $O(1)$.

Key generation. When generating the secret key for user u in the system, for every attribute a_i in $\mathbb{A}_u$, the system computes D_i, so the overall complexity is $O(|\mathbb{A}_u|)$.

Encrypted order submission. We assume every access tree T_j has t node and l leaf node. For every node x, we should find k_x points to determine the polynomial $f(x)$. Assuming average threshold value to be k, our CP-ABE encryption complexity is $O(kt)$. In addition, attribute in every leaf node in the tree is encrypted with Paillier Cryptosystem, so the complexity is $O(l)$. For every order O, the over all encryption complexity is $O(mkl)$.

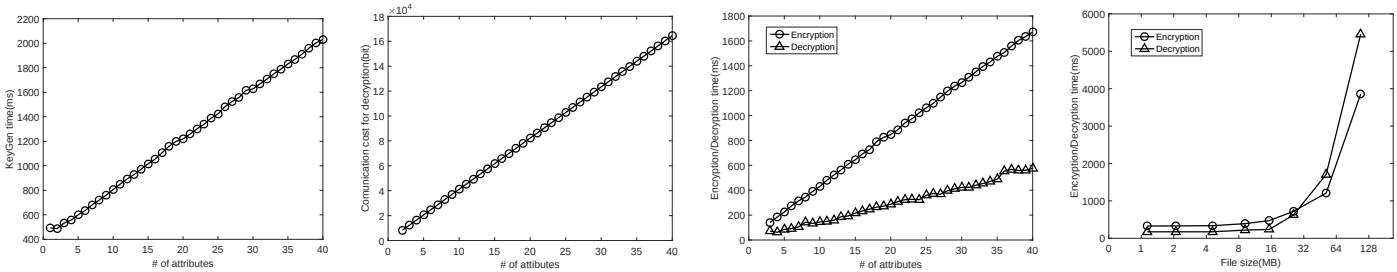
Intermediate parcel handling. For every attribute $a_i \in A_c$, the courier computes $\mathsf{E}(r \cdot Q_j(H(a_i)) + H(a_i))$. To compute $\mathsf{E}(r \cdot Q_j(H(a_i)) + H(a_i))$ based on the homomorphic propriety of *Paillier Cryptosystem*, it involves $|A_j| + 1$ exponentiations and $|A_j| + 1$ multiplication, so the complexity is $O(|A_c||A_j|)$. In addition, DecryptNode in our CP-ABE is a recursive algorithm, and it is executed exactly once at every node of the tree T_j, therefore the computation complexity of this process is $O(t)$. As a result, the overall decrypting complexity is $O(|A_c||A_j| + t)$.

VIII. Experiment Results

In this section, we give the experiment results of PriExpress. We implement PriExpress client app on Nexus 7 (2013) WiFi with android 4.4.4 and AA and PriExpress server on a Lenovo desktop with 3.20 GHz CPU, 4 GB RAM, and Windows 8 64-bit Professional. Our implementation is based on Java Pairing-Based Cryptography Library(JPBC) [8].

Similar to [9] [10], Fig. 5 shows the computation or communication overhead of the core algorithms consisting of Setup, Key Generation, Encryption and Decryption under various conditions.

Figure 5(a) shows the key generation time with different number of attributes. The key generation time increases with number of attributes linearly. Figure 5(b) plots the communication cost for courier to decrypt each part of the ciphertext. The communication cost increases linearly with the number of attributes of couriers. Figure 5(c) shows the encryption and decryption time with different number of attributes, where the file size is to fixed 4kB which is sufficient to store the logistics order. Both encryption and decryption time increase with number of attributes in access tree linearly. To pursue integrity, we test our scheme under different sizes of file, though the order size in real scenario is very small. Figure 5(d) shows the encryption and decryption computation cost with different size of file, where the number of attributes is fixed to 10.



(a) Key generation overhead with different number of attributes

(b) Communication overhead for decryption with different number of attributes

(c) Computation overhead for encryption and decryption with different number of attributes

(d) Computation overhead for encryption and decryption with different file size

Fig. 5: Computation and communication overhead of PriExpress system

IX. Related Work

Attribute-based Encryption(ABE) has attract much attention of researchers with increasing applications in networking and distributed systems, e.g., [11], [12]. Generally, access control is enforced based on the decrypter's attributes in ABE. ABE is more flexible than traditional encryption scheme, for the sender can specify himself the people with appropriate attributes have the privilege to see the message. Also, he does not need to know the receivers.

As mentioned in Section I, Sahai and Water first proposed the concept of Attribute Based Encryption (Fuzzy-IBE or threshold ABE) in [13]. They use a threshold based access control method that a user can decrypt the message only if he owns at least k attributes specified by the encrypter. To prevent collusion attack, each user is associated with a random polynomial such that multiple users cannot enlarge their privilege by combining their attributes. However, threshold based policy is not expressive enough.

Key policy based method can achieve more fine-grained access control in comparison with threshold policy. In KP-ABE, ciphertexts are associated with some attributes and an access structure is also built in each private key. However, the encrypter can only specify the attributes in the ciphertext and the access control structure is stored in the private key issued by the trust server. The first key-policy scheme (KP-ABE) was introduced in [14] by Goyal *et al.*, which has become increasingly popular [15]–[18]. A novel ciphertext policy scheme (CP-ABE) was presented in [3] and was later improved by Water in [19]. In CP-ABE, access structure is stored in ciphertext and secret key is associate with some attributes. Users can decrypt the message only if attributes in their key satisfy the access tree in the ciphertext. The CP-ABE scheme addresses the problem of KP-ABE that data owner has to trust the key issuer. In addition, the message sender just needs to re-encrypt the message if he wants to change the access policy. After that, many schemes were proposed based on the CP-ABE scheme [14], [20]–[23].

In a centralized authority design, all of the authorities must be globally trustworthy throughout the system lifetime. In [15] and [24], a multi-authority system is presented in which each authority knows only a part of any user's attributes, which are not enough to figure out the user's identity. Decentralizing the central authority into multiple ones based on CP-ABE was studied in [25] and [26]. In [25], a LSSS matrix is used as the access structure, but the scheme only allows transformation from AND and OR gates to the LSSS matrix, which limits its applicability. Many attribute based encryption that support multiple authorities have been proposed in [27]–[30]. For ease of deployment, we do not consider multi-authority scheme in this paper.

Similar to this paper, various works on ABE support hidden access policy. [31] uses inner product predicate encryption(PE) to realize policy hiding, but it requires all formulas be written in CNF or DNF forms, which may result in superpolynomial blowup in size for certain formulas. In addition, predicate encryption is less expressive than tree based access structure [32], [33]. Other ABE schemes [34], [35] supporting policy hiding also did not use structure expressive as ours'. Note that access tree in our paper is able to express arbitrary general policy, which makes the access control in our protocol flexible and scalable.

X. Conclusion

This paper presented the design and evaluation of PriExpress, the first logistics system which protects the users' privacy and ensures the efficient delivery of the parcel at the same time by exploring improved attribute-based encryption with a hidden access tree. Based on users' attributes, PriExpress enforces fine-grained access control on the logistic data. Our security and performance analysis shows that PriExpress is both secure and efficient.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive comments and helpful advice. This work was supported in part by the US National Science Foundation under grants CNS-1514381, CNS-1421999, CNS-1320906, CNS-1422301, and CNS-1514014.

References

[1] "http://www.nbd.com.cn/articles/2012-11-07/693154.html."

[2] "http://roll.sohu.com/20140220/n395362756.shtml."

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE S&P'07*, May 2007.

[4] M. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *EUROCRYPT'04*, May 2004.

[5] J. Groth and A. Sahai, "Efficient non-interactive proof systems for bilinear groups," in *EUROCRYPT'08*, Istanbul, Turkey, Apr. 2008.

[6] L. Guo, C. Zhang, J. Sun, and Y. Fang, "Paas: A privacy-preserving attribute-based authentication system for ehealth networks," in *IEEE ICDCS'12*, Macau, China, Jun. 2012.

[7] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT'99*, Prague, Czech Republic, May 1999.

[8] "Treasury moves to the cloud," http://gas.dia.unisa.it/projects/jpbc/#.VyHkhvkrIQ8.

[9] J. Sun, R. Zhang, and Y. Zhang, "Privacy-preserving spatiotemporal matching," in *INFOCOM'13*, Turin, Italy, Apr. 2013.

[10] ——, "Privacy-preserving spatiotemporal matching for secure device-to-device communications," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, Mar. 2016.

[11] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[12] W. Sun, S. Yu, W. Lou, Y. Hou, and H. Li, "Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," in *IEEE INFOCOM'14*, Apr. 2014.

[13] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *EUROCRYPT'05*, May 2005.

[14] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *ICALP'06*, Venice, Italy, Jul. 2008.

[15] M. Chase, "Multi-authority attribute based encryption," in *TCC'07*, Feb. 2007.

[16] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *ACM CCS'07*, Alexandria, VA, Oct. 2007.

[17] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. De Panafieu, and C. Ràfols, "Attribute-based encryption schemes with constant-size ciphertexts," *Theoretical Computer Science*, vol. 422, pp. 15–38, 2012.

[18] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2150–2162, 2012.

[19] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *International Workshop on Public Key Cryptography*. Springer.

[20] L. Cheung and C. Newport, "Provably secure ciphertext policy abe," in *ACM CCS'07*, Alexandria, VA, Oct. 2007.

[21] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *ISPEC'09*, Hangzhou, China, Apr. 2009.

[22] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *WISA'09*, Busan, Korea, Aug. 2009.

[23] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," in *ISPEC'09*, Xi'an, China, Apr. 2009.

[24] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *ACM CCS'09*, Chicago, IL, Nov. 2009.

[25] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *EUROCRYPT'11*, May 2011.

[26] S. Müller, S. Katzenbeisser, and C. Eckert, "On multi-authority ciphertext-policy attribute-based encryption," *Bulletin of the Korean Mathematical Society*, vol. 46, no. 4, pp. 803–819, 2009.

[27] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," *Information Sciences*, vol. 180, no. 13, pp. 2618–2632, 2010.

[28] V. Božović, D. Socek, R. Steinwandt, and V. I. Villányi, "Multi-authority attribute-based encryption with honest-but-curious central authority," *International Journal of Computer Mathematics*, vol. 89, no. 3, pp. 268–283, 2012.

[29] F. Li, Y. Rahulamathavan, M. Rajarajan, and R. C.-W. Phan, "Low complexity multi-authority attribute based encryption scheme for mobile cloud computing," in *SOSE'13*, San Francisco, CA, March 2013.

[30] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "Dac-macs: Effective data access control for multiauthority cloud storage systems," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1790–1801, 2013.

[31] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *EUROCRYPT'10*, May 2010.

[32] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *EUROCRYPT'08*, Istanbul, Turkey, Apr. 2008.

[33] J. Lai, R. Deng, and Y. Li, "Fully secure cipertext-policy hiding cp-abe," in *ISPEC'11*, Guangzhou, China, May 2011.

[34] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," in *ISC'09*, Sep. 2009.

[35] S. Yu, K. Ren, and W. Lou, "Attribute-based content distribution with hidden policy," in *IEEE NPSec'08*, Orlando, FL, Oct. 2008.