Hong and Sun *SpringerPlus (2016)5:131*
DOI 10.1186/s40064-016-1765-9

SpringerPlus
a SpringerOpen Journal

RESEARCH Open Access

High efficient key-insulated attribute based encryption scheme without bilinear pairing operations

Hanshu Hong and Zhixin Sun*

*Correspondence:
sunzx@njupt.edu.cn
Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Ministry Education, Nanjing University of Posts and Telecommunications, Nanjing, China

Abstract

Attribute based encryption (ABE) has been widely applied for secure data protection in various data sharing systems. However, the efficiency of existing ABE schemes is not high enough since running encrypt and decrypt algorithms need frequent bilinear pairing operations, which may occupy too much computing resources on terminal devices. What's more, since different users may share the same attributes in the system, a single user's private key exposure will threaten the security and confidentiality of the whole system. Therefore, to further decrease the computation cost in attribute based cryptosystem as well as provide secure protection when key exposure happens, in this paper, we firstly propose a high efficient key-insulated ABE algorithm without pairings. The key-insulated mechanism guarantees both forward security and backward security when key exposure or user revocation happens. Besides, during the running of algorithms in our scheme, users and attribute authority needn't run any bilinear pairing operations, which will increase the efficiency to a large extent. The high efficiency and security analysis indicate that our scheme is more appropriate for secure protection in data sharing systems.

Keywords: ABE, Key-insulated, Without pairings, High efficiency

Background

With the continuing increase of network information resources, users are confronted with urgent challenges such as how to make secure data sharing with others efficiently. To help users achieve secure and flexible data access control, Sahai and Waters (2005) proposed a new notion called attribute based encryption (ABE). In this cryptosystem, data receiver's access privileges are described by a certain number of attributes. A data receiver can get access to the ciphertext only if the attributes he owns match with the access control policy set by the data owner. Equipped with the advantages of providing secure data protection as well as flexible access control, ABE (Goyal et al. 2006; Lewko et al. 2010; Waters 2011) has become an effective tool for secure data sharing between users.

However, the efficiency of current ABE schemes is still not high enough compared to traditional public key cryptosystem. One important factor is that encryption and decryption in ABE need frequent bilinear pairing calculations. Researchers have proved that the computational complexity of bilinear pairing is much larger than that of other operations (exponential operation, multiplication, addition) in discrete group (Chen et al. 2007; Bertoni et al. 2005). In some special network systems such as wireless sensor networks (Yu et al. 2011) or body area networks (Hu and Zhang 2013; Tan et al. 2011), the computation capacity and energy resources of terminal devices are limited, frequent bilinear pairing operations may consume too much computing resources and lead to bottleneck or node failure during the process of data sharing. Consequently, to further enrich the application scenarios of ABE, it is essential to improve the efficiency by reducing the number of pairing operations. To the best of our knowledge, the elimination of pairing operation in attribute based cryptosystem is quite new in the research literature, which has not been solved yet.

Key exposure protection is another issue remains to be tackled in attribute based cryptosystem. Although many schemes have achieved forward and backward security in terms of attribute revocation (Hur 2013; Yu et al. 2010), however, the system is still at risk when key exposure happens. If the private key owned by a non-revoked user leaks, any user can use the private key to decrypt the corresponding ciphertext since the leaked private key is still a valid one. Consequently, all the potential threat calls for frequent key refreshing in attribute based cryptosystem. When key exposure happens, effective and efficient key updating mechanism should be implemented to keep the system from potential threat.

To better guarantee the security during the process of data sharing as well as minimize the total computation cost, in the paper, we do the following research:

We propose a high efficient key-insulated ABE scheme without pairings (KI-ABE-WP). In our scheme, each user's private key corresponds to an access structure. A user can decrypt the ciphertext only if the attributes used for encryption match with the access structure he owns. Besides, we divide the system lifetime into discrete time periods. When time period evolves, only part of the private key has to be updated and the system public parameters remain unchanged. This saves a lot of computation and transmission load when attribute revocation or key exposure happens. What's more, during the running of algorithms in our scheme, users and attribute authority (AA) needn't do any bilinear pairing operations, which will increase the total efficiency to a large extent compared to current ABE schemes. At last, our scheme is proved to be secure under CDH hardness assumption. The high efficiency due to the elimination of bilinear pairings makes our scheme more appropriate for secure data sharing in various network systems, especially those with limited computing capacity such as wireless sensor networks, mobile communication, etc.

The rest sections are arranged as follows:

In "Related works and preliminaries" section, we introduce the related works and essential mathematical preliminaries used to construct our scheme. The security model and concrete constructions of our scheme are proposed in "Models and assumptions" and "Constructions to our KI-ABE-WP" sections respectively. The security and performance analysis are given in "Security proof and performance analysis" section. At last, we conclude our paper and make prospects on future directions in "Conclusion" section.

Related works and preliminaries

Related works

Existing literatures have achieved much progress in ABE with respect to fine-grained access control (Goyal et al. 2006; Waters 2011; Bethencourt et al. 2007; Goyal et al. 2008), user flexible revocation (Hur and Noh 2011; Yu et al. 2011) and attribute based signcryption (Wang and Huang 2011), etc. Meanwhile, ABE has been widely designed for providing data protection in various network systems such as personal health record system (Li and Yu 2013), body area networks (Hu and Zhang 2013; Tan et al. 2011), wireless sensor networks (Yu et al. 2011), cloud computing (Yang et al. 2012). However, these schemes may not be entirely realistic to be applied to some application scenarios thanks to the heavy computation cost from bilinear pairing operations. Take the proposed scheme in Xhafa et al. (2015) for instance, if the number of attributes involved in encryption is n, then the decryption will take $4n$ times of pairing operations, which will bring a heavy computation burden on terminal devices. Consequently, to further improve the efficiency and performance of ABE, the number of pairing operations should be reduced, even totally eliminated.

Besides efficiency, key exposure protection is another issue urgently to be solved in ABE. Many existing schemes have guaranteed forward and backward security when attribute revocation happens by introducing a proxy re-encryption server (Hur and Noh 2011; Yu et al. 2010). However, these schemes only focus on the key regeneration of the revoked users, but neglect the key updating for non-revoked users. If a non-revoked user's private key leaks, the confidentiality of the system will be threatened since the leaked private key is still a valid one. In fact, in attribute based cryptosystem, key refreshing is more important since either attribute revocation or private key exposure protection calls for frequent key-updating. Xu and Martin (2012) proposed an ABE scheme with secure key refreshing in, but their scheme has to regenerate the master key and public parameters in the system, this will bring about much more computation overheads when key updating happens. Key-insulation (Dodis et al. 2002) is a promising tool to guarantee forward and backward security as well as achieving high efficiency of key updating. In this mechanism, the lifetime of the system is divided into discrete periods. The public key remains unchanged throughout the lifetime, while temporary secret keys are updated periodically. Key-insulation mechanism can provide full security when user's private key exposure happens and it has been designed for effective key exposure protection in identity based cryptosystem (Zhu et al. 2014), certificateless cryptosystem (Chen et al. 2015), etc. The advantage of key-insulation mechanism can also be combined into attribute based cryptosystem and propose a key-insulated ABE scheme with efficient and secure key updating.

Hardness assumptions

(a) Discrete logarithm assumption (DL):

Given $X, P \in G$, it is computational infeasible to calculate the value of a ($a \in Z_q^*$) such that $X = aP$ with a non-negligible probability within probabilistic polynomial-time.

(b) Computational Diffie–Hellman assumption (CDH):
For $a, b \in Z_q^*$, given (p, ap, bp), it is computational infeasible to calculate the value of abp with a non-negligible probability within probabilistic polynomial-time.

Models and assumptions

A real world example of KI-ABE-WP

One typical application of our KI-ABE-WP is the mobile communication system, which can be illustrated in Fig. 1. It consists of six entitles: AA, key helper, base station, data centre, data sender and receiver. Base station and data centre are hardware architectures which are responsible for mobile communications and file storage. AA generates initial attribute private keys for each user in the system and the private key corresponds with an access structure. Data sender and receiver are the two sides of communication, data sender encrypts the file with a set of attributes, while a receiver can decrypt the ciphertext if the attributes used for encryption match with the access structure he owns. When system evolves into a new time period, users in the system update their private keys to the latest version with the assistance of key helper. Due to the elimination of bilinear pairing operations, the proposed KI-ABE-WP will relieve the terminal devices from heavy computation burden, thus improving the efficiency and performance of the whole system.

Formalized definition of the algorithms in KI-ABE-WP

Our KI-ABE-WP consist of five algorithms:

$Setup(1^\lambda)\{PK, MK\}$ This algorithm takes a security parameter λ as input and outputs the public parameter PK and master key MK. PK is shared by users while MK is kept private by AA.

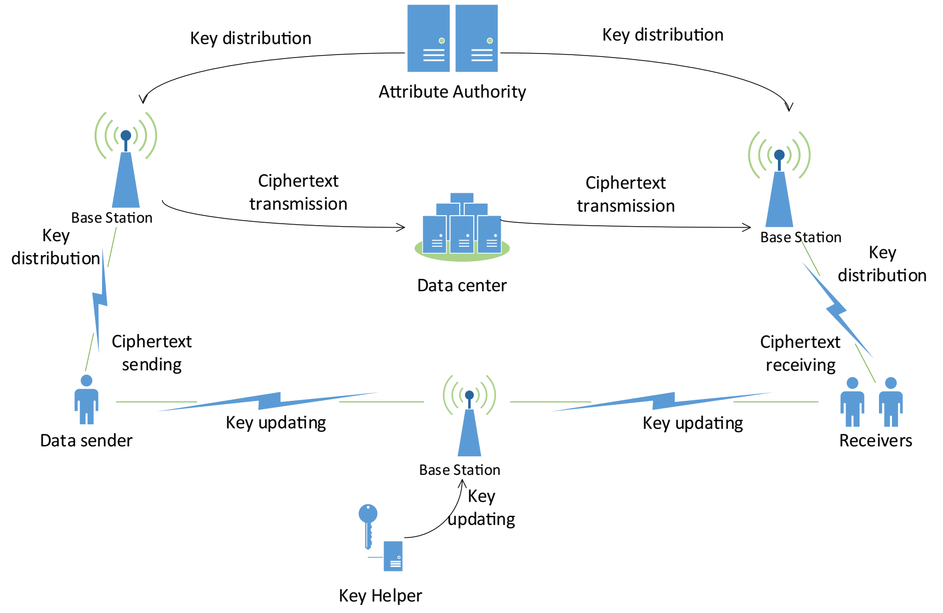

Fig. 1 Apply KI-ABE-WP for mobile communication system

Initial private key generation: $\{PK, MK, \gamma, TP_0\} \rightarrow \{TD_{\gamma, TP_0}\}$ This algorithm is operated by AA. It takes *PK*, *MK*, initial time period TP_0 and the user's access structure γ as input. The output of this algorithm is user's initial private key TD_{γ, TP_0}.

Keyupdating: $\{PK, MK, \gamma, TP_n\} \rightarrow \{TD_{\gamma, TP_n}\}$ This algorithm is an interaction between AA and user. On input *PK*, *MK*, γ and the current time period TP_n, AA outputs the key-updating component U_{γ, TP_n} and transfers it to users. User updates his temporal private key to the latest version using U_{γ, TP_n}.

Encrypt $\{PK, M, \{A_i\}\} \rightarrow \{CT\}$ This algorithm is operated by the data sender. It takes *PK*, a plaintext *M* and an attribute set $\{A_i\}$ as input and outputs the corresponding ciphertext *CT*.

Decrypt: $\{D_{i,\gamma}, CT\} \rightarrow \{M\}$ This algorithm is run by the data receiver. The algorithm takes as input the ciphertext *CT* and the receiver's temporal private key TD_{γ, TP_n}, it outputs the plaintext *M*.

Security model of KI-ABE-WP

Definition Our KI-ABE-WP scheme is secure under chosen ciphertext attacks if there exists an *Adversary* has non-negligible advantage in the following game played by a *Challenger* and an *Adversary*.

Phase 1 Setup *Challenger* runs *Setup* procedure to obtain the system parameters *PK* and master keys *MK*. It sends *PK* to *Adversary*.

Phase 2 Queries *Adversary* can make the following queries to *Challenger*.

Initial private key generation query *Challenger* can obtain user's initial private key D_{γ, TP_0} by running *Initial private key generation* algorithm and returns the result back to *Adversary*.

Temporal private key generation query *Challenger* can obtain user's temporal private key at the current time period and returns the result D_{γ, TP_n} back to *Adversary*.

Decrypt query *Adversary* can ask *Decrypt query* for ciphertext *CT*. *Challenger* runs *Decrypt* algorithm and returns the results to *Adversary*.

Phase 3 Challenge *Adversary* chooses two plaintexts M_0 and M_1 and a challenging access structure γ^* at current time period.

Challenger chooses $\sigma \in \{0, 1\}$ randomly and calculates $CT_\sigma = Encrypt(PK, M_\sigma, \{A_i\})$ and returns the result to *Adversary*.

Adversary outputs a value σ^* as a conjecture of σ.

During the whole process of the challenge game:

Adversary cannot ask *Challenger* for *Decrypt query* of M_0 and M_1.

Adversary cannot ask *Challenger* for *Temporal private key generation query* for the challenging structure γ^*.

If $\sigma^* = \sigma$ then *Adversary* wins the game.

We denote $Adv(A) = \left| Pr[\sigma^* = \sigma] - \frac{1}{2} \right|$ to be the *Adversary's* advantage in the above challenge game.

Constructions to our KI-ABE-WP

Concrete algorithms of KI-ABE-WP

Setup

Let G to be a cyclic addition group. Denote q and p to be the prime order and generator of G respectively. AA defines a global attribute set $\{A_i\}$ and picks $t_i \in Z_q^*$ for each attribute in $\{A_i\}$. Let $T_i = t_i p$ to be the public key of A_i. Picks $k_n \in Z_q^*$ for each time period TP_n in the system lifetime. Let $K_n = k_n p$. Chooses a secret number $y \in Z_q^*$ and calculates $Y = yp$. Define two hash functions $H_1{:}\{0,1\}^* \rightarrow Z_q^*, H_2{:}\{0,1\}^* \rightarrow \{0,1\}^m$, m is the size of plaintext. Define a Lagrange interpolation function $\Delta_{i,S(x)} = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$.

The system public parameters are $\{G, q, p, A_i, T_i, K_n, Y, H_1, H_2\}$ and the system master keys are $\{t_i, y, k_n\}$.

Initial private key generation

AA randomly chooses a polynomial q_x for each node x in the user's access tree γ. Denote d_x to be the degree of q_x and thr_x to be the threshold value node. Let $d_x = thr_x - 1$. For the root node AA sets $q_{root}(0) = y$. For any other node (except for root node) in the access tree, let $q_x(0) = q_{parent(x)}^{index(x)}$. The initial attribute private key at time period TP_0 for access structure γ can be denoted by $TD_{\gamma,TP_0} = \{q_x(0) + t_i + k_0 \cdot H_1(T_i, TP_0),\ i \in \gamma\}$.

Key updating

When time period evolves from TP_n to TP_{n+1}, AA calculates the updated key component $U_{\gamma,TP_{n+1}} = (k_{n+1} \cdot H_1(T_i, TP_{n+1}) - k_n \cdot H_1(T_i, TP_n),\ i \in \gamma)$ and transfers it to user. User calculates $TD_{\gamma,TP_{n+1}} = TD_{\gamma,TP_n} + U_{\gamma,TP_{n+1}}$ as the temporal private key at time period TP_{n+1}.

Encrypt

At time period TP_n, for a plaintext M,data sender picks a random number $s \in Z_q^*$ and calculates:

$$\begin{aligned} &C_1 = sp, \quad C_2 = sT_i \\ &C_3 = sK_n, \quad C_4 = H_2(sY) \oplus M \end{aligned} \tag{1}$$

Then data sender sends $CT = \{C_1, C_2, C_3, C_4\}$ to data receiver.

Decrypt

Upon receiving CT, receiver calculates:

$$M = H_2\left(\sum_{i \in \gamma} \left(TD_{\gamma,TP_n} \cdot C_1 - C_2 - C_3 \cdot H_1(T_i, TP_n) \right) \right) \oplus C_4 \tag{2}$$

Correctness proof

If x is a leaf node, the calculation process is as follows:

$$\begin{aligned} DecryptNode\big(x, TD_{\gamma,TP_n}, C_1, C_2, C_3\big) &= TD_{\gamma,TP_n} \cdot C_1 - C_2 - C_3 \cdot H_1(T_i, TP_n) \\ &= (q_x(0) + t_i + k_n \cdot H_1(T_i, TP_n)) \cdot sp \\ &\quad - sT_i - sK_n \cdot H_1(T_i, TP_n) = q_x(0) \cdot sp + t_i \cdot sp \\ &\quad + k_n \cdot H_1(T_i, TP_0) \cdot sp - sT_i - sK_n \cdot H_1(T_i, TP_n) \\ &= q_x(0) \cdot sp \end{aligned} \tag{3}$$

All the value calculated from $DecryptNode(x, TD_{\gamma,TP_n}, C_1, C_2, C_3)$ will be stored as F_z. For any $F_z \neq 0$,the algorithm calculates F_{root} (the value of root node) using Lagrange interpolation method.

If x is a non-leaf node, z is the child node of x, then the algorithm calculates the value of $DecryptNode(x, TD_{\gamma,TP_n}, C_1, C_2, C_3)$ as follows:

Let $i = index(z)$, $S_{x'} = \{index(z){:}z \in S_x\}$

$$\begin{aligned} F_x &= \sum_{z \in S_x} F_z^{\Delta_{i,S_{x'}(0)}} \\ &= \sum_{z \in S_x} sp \cdot q_z(0)^{\Delta_{i,S_{x'}(0)}} \\ &= \sum_{z \in S_x} sp \cdot q_{parent}(z)^{(index(z))^{\Delta_{i,S_{x'}(0)}}} \\ &= \sum_{z \in S_x} sp \cdot q_z(x)^{\Delta_{i,S_{x'}(0)}} \\ &= q_x(0) \cdot sp \end{aligned} \tag{4}$$

Since the value of $DecryptNode(x, TD_{i,TP_n}, C_1, C_2, C_3) = q_x(0) \cdot sp$, whether x is a leaf node or non-leaf node, consequently, the value of root node F_{root} and the plaintext M can be calculated by:

$$\begin{aligned} F_{root} &= q_{root}(0) \cdot sp = syp = sY \\ M &= H_2(F_{root}) \oplus C_4 = H_2(sY) \oplus C_4 \\ &= H_2(sY)H_2(sY)M = M \end{aligned} \tag{5}$$

Security proof and performance analysis

Security proof

Theorem *If the proposed KI-ABE-WP can be broken by an Adversary in the random oracle model, then a Simulator can be constructed to break the CDH hardness assumption in group G successfully with a non-negligible advantage.*

Proof In the challenge game, if there exists an *Adversary* can break our KI-ABE-WP with an advantage (t, ε) in the random oracle model, then there exists a *Simulator* which can break the CDH assumption with an advantage of ε' which satisfies:

$$\begin{aligned} t' &\leq t + \big(n\big(q_p + q_i + q_t + 2q_d + 3\big) + 4\big) \cdot t_{sm} + n(2q_i + 2q_t + 2q_d + 2) \cdot t_a \\ \varepsilon' &\geq \frac{\varepsilon}{e(q_d + 1)} \cdot \left(\frac{q_i \cdot q_{H_1}}{2^l}\right)^{q_t} \cdot \left(\frac{q_p}{2^l}\right)^{1+q_i} \end{aligned} \tag{6}$$

In the lemma (6), $q_p, q_{H_1}, q_i, q_t, q_d$ are the maximum numbers of *Public key generation query, H_1 query, Initial private key generation query, Temporal private key generation query* and *Decrypt query* respectively. Denote t_{sm} and t_a to be the time consumption for running a scalar multiplication operation and an addition operation respectively.

The process of the challenge game is as follows:

Phase 1 *Setup*:

Challenger sets the parameters as follows:

Defines a global attribute set $\{A_i\}$. Let G be a cyclic addition group with prime order q. The generator of group G is denoted by p. Defines two hash functions: H_1:$\{0,1\}^* \to Z_q^*$, H_2:$\{0,1\}^* \to \{0,1\}^m$, m is the size of plaintext. Randomly picks $a, b \in Z_q^*$, sets $X = bp$, $Y = ap$.

The aim of *Simulator* is to calculate the value of *abp* according to the process of the challenge game. *Simulator* plays the role of *Challenger* and runs *Adversary* as a sub-program.

Phase 2 *Queries*:

The proof skills used in our scheme resembles the method which has been proposed in Coron (2000). Without loss of generality, supposing that *Adversary* will make *Public key generation query* for an attribute A_i before making *Initial private key generation query*, *Temporal private key generation query* and *Decrypt query* to *Simulator*.

Then *Adversary* makes the following queries to *Simulator*:

Public key generation query: *Simulator* maintains a list $L_p\{A_i, \gamma, c, t_i, T_i\}$. When *Adversary* asks a *Public key generation query* for A_i in the γ. *Simulator* responds as follows:

Checks if A_i has already existed in the list L_p. If so, *Simulator* returns the result of T_i to *Adversary*. If not, *Simulator* picks a biased coin $c \in \{0,1\}^l$ at random and sets $Pr[c = 0] = \theta$ while $Pr[c = 1] = 1 - \theta$. When $c = 0$, *Simulator* chooses $t_i \in Z_q^*$ and sets $T_i = t_i p$. Otherwise let $T_i = t_i X$. *Simulator* adds the tuple $\{A_i, \gamma, c, t_i, T_i\}$ into L_p and sends T_i to *Adversary*.

H_1 query: *Simulator* maintains a list $L_{H_1}\{A_i, \gamma, T_i, TP_n, H_1(T_i, TP_n)\}$. When *Adversary* asks a *H_1 query* for A_i, *Simulator* responds as follows:

Checks if A_i has already existed in L_{H_1}. If so, *Simulator* sends the result back to *Adversary*. If not, *Simulator* calculates the value of $H_1(T_i, TP_n)$ and adds it into the L_{H_1}.

Initial private key generation query: *Simulator* maintains a list $L_i\{A_i, \gamma, TD_{\gamma,TP_0}\}$. When *Adversary* asks a *Initial private key generation query* for γ, *Simulator* responds as follows:

Checks if γ exists in the list $L_p\{A_i, \gamma, c, t_i, T_i\}$. If not, *Simulator* aborts the challenge game and outputs failure. We denote this incident by E_1.

Otherwise, *Simulator* randomly chooses a polynomial q_x for each node x in the user's access tree γ. Denote d_x to be the degree of q_x and thr_x to be the threshold value node. Let $d_x = thr_x - 1$. For any other node (except for root node) in the access tree, let $q_x(0) = q_{parent(x)}^{index(x)}$. *Simulator* chooses $k_0 \in Z_q^*$ and sets initial private key $TD_{\gamma,TP_0} = \{q_x(0) + t_i + k_0 \cdot H_1(T_i, TP_0),\ i \in \gamma\}$. Then simulator adds the tuple into L_i and sends TD_{γ,TP_0} to *Adversary*.

Temporalprivatekeygenerationquery:*Simulator* maintains a list $L_t\left\{A_i, \gamma, TP_n, TD_{\gamma,TP_n}\right\}$. When *Adversary* asks a *Temporal private key generation query* for γ, *Simulator* responds as follows:

Checks γ in the list $L_i\{A_i, \gamma, TD_{\gamma,TP_0}\}$. If γ does not exist in L_i, *Simulator* aborts the challenge game and outputs failure. We denote this incident by E_2.

Checks $L_{H_1}\{A_i, \gamma, T_i, TP_n, H_1(T_i, TP_n)\}$.. If the tuple $\{\gamma, A_i\}$ does not exist in L_{H_1}, *Simulator* aborts the challenge game and outputs failure. We denote this incident by E_3.

Otherwise, *Simulator* randomly chooses $k_n \in Z_q^*$ and calculates $TD_{\gamma,TP_n} = TD_{\gamma,TP_0} + (k_n H_1(T_i, TP_n) - k_0 \cdot H_1(T_i, TP_0))$. *Simulator* sends TD_{γ,TP_n} to *Adversary* and adds the tuple into $L_t\{A_i, \gamma, TP_n, TD_{\gamma,TP_n}\}$.

Decrypt query: *Simulator* maintains a list $L_D\{A_i, \gamma, CT = \{C_1, C_2, C_3, C_4\}, M\}$. When *Adversary* asks a *Decrypt query* for $\{A_i, \gamma, CT = \{C_1, C_2, C_3, C_4\}\}$, *Simulator* responds as follows:

Checks $\{A_i, \gamma\}$ in the list $L_p\{A_i, \gamma, c, t_i, T_i\}$. If $c = 1$, *Simulator* aborts the game and outputs failure. We denote this incident by E_4.

Otherwise, *Simulator* recovers TD_{γ,TP_n} from $L_t\{A_i, \gamma, TP_n, TD_{\gamma,TP_n}\}$ and calculates $M = C_4 \oplus H_2\left(\sum_{i\in\gamma} TD_{\gamma,TP_n} \cdot C_1 - C_2 - C_3 \cdot H_1(T_i, TP_n)\right)$. *Simulator* sends the result to *Adversary* and adds the tuple into $L_D\{A_i, \gamma, CT = \{C_1, C_2, C_3, C_4\}, M\}$.

Phase 3 *Challenge*:

Adversary outputs two plaintext M_0 and M_1 with a challenging access structure γ^* at the current time period TP_n.

Simulator checks if γ^* exists in the list $L_p\{A_i, \gamma, c, t_i, T_i\}$. If not, *Simulator* aborts the challenge game and outputs failure. We denote this incident by E_5. If γ^* exists in the list $L_p\{A_i, \gamma, c, t_i, T_i\}$ and $c = 0$, *Simulator* aborts the challenge game and outputs failure. We denote this incident by E_6.

Simulator runs Temporal private key generation query for γ^* and calculates $TD_{\gamma^*,TP_n} = (q_x(0) + t_i + k_n \cdot H_1(T_i, TP_n))^*$. Then, *Simulator* picks $\sigma \in \{0, 1\}$, $s \in Z_q^*$ and calculates:

$$\begin{aligned}
&C_{1,\sigma} = s \cdot X, \quad C_{2,\sigma} = s \cdot t_i X \\
&C_{3,\sigma} = s \cdot k_n X \\
&C_{4,\sigma} = H_2\left(\sum_{i\in\gamma} \left(TD_{\gamma^*,TP_n} \cdot C_{1,\sigma} - C_{2,\sigma} - C_{3,\sigma} \cdot H_1(T_i, TP_n)\right)\right) \oplus M_\sigma
\end{aligned} \tag{7}$$

Simulator sends $CT_\sigma = \{C_{1,\sigma}, C_{2,\sigma}, C_{3,\sigma}, C_{4,\sigma}\}$ to *Adversary.*

Adversary outputs M_σ^* as a guess of M_σ. If $M_\sigma^* = M_\sigma$ and *Adversary* wins the game, *Simulator* outputs $abp = s^{-1} \cdot \sum_{i\in\gamma} \left(TD_{\gamma^*,TP_n} \cdot C_{1,\sigma} - C_{2,\sigma} - C_{3,\sigma} \cdot H_1(T_i, TP_n)\right)$ as the solution to CDH assumption in group G.

Then we will analyse the time complexity of *Simulator* in breaking CDH assumption in group G.

From the description, assuming the average number of attributes involved is "n", for each request of *Public key generation query*, *Initial Private key generation query*, *Temporal private key generation query* and *Decrypt query*, *Simulator* has to run n times of multiplication operation, n times of multiplication operation and $2n$ times of addition operation, n times of multiplication operation and $2n$ times of addition operation, $2n$ times of multiplication operation and $2n$ times of addition operation respectively.

During the *Challenge* phase, *Simulator* has to run $(3n + 4)$ times of multiplication operation and $2n$ times of addition operation.

Denote t_{sm}, t_a to be the time consumption of scalar multiplication operation and addition operation in group G respectively. From what has been discussed above, the total time complexity of *Simulator* t' satisfies:

$$t' \leq t + \big(n\big(q_p + q_i + q_t + 2q_d + 3\big) + 4\big) \cdot t_{sm} + n(2q_i + 2q_t + 2q_d + 2) \cdot t_a \tag{8}$$

Next we will discuss the advantage of *Simulator* in breaking the CDH assumption.

During the process of the challenge game, the responses of *Initial Private key generation query*, *Temporal private key generation query* and *Decrypt query* return to *Adversary* are valid and indistinguishable if $E1$, $E2$, $E3$, $E4$ do not happen. Furthermore, if *Adversary* succeeds in distinguishing M_σ and $E5$, $E6$ do not happen, then *Simulator* is capable of breaking the CDH assumption.

Next we will calculate the probability of the incidents discussed above.

According to the process of queries phase, the probability of $E4$ and $E6$ not occurring can be denoted by lemma (9):

$$Pr\left|\overline{E4} \cap \overline{E6}\right| = \theta^{q_d} \cdot (1 - \theta) \tag{9}$$

The value of $Pr\left|\overline{E4} \cap \overline{E6}\right|$ is maximized in lemma (10) when $\theta = \frac{q_d}{1+q_d}$.

$$Pr_{max}\left|\overline{E4} \cap \overline{E6}\right| = \frac{e^{-1}}{1 + q_d} \tag{10}$$

Since the responses of *Public key generation query* act as random oracle model, consequently, the probability of $E1$ and $E5$ not occurring can be denoted by lemma (11):

$$Pr\left|\overline{E1} \cap \overline{E5}\right| = \left(\frac{q_p}{2^l}\right)^{1+q_i} \tag{11}$$

Likewisely, the probability of $E2$ and $E3$ not occurring can be denoted by lemma (12):

$$Pr\left|\overline{E1} \cap \overline{E5}\right| = \left(\frac{q_i \cdot q_{H_1}}{2^l}\right)^{q_t} \tag{12}$$

Taking all the probabilities of the above incidents into account, it can be figured out that if *Adversary* successfully attacks our scheme with an advantage ε, then a *Simulator* can break the CDH assumption in group G with an advantage of ε' which satisfies:

$$\varepsilon' \geq \frac{\varepsilon}{e(q_d + 1)} \cdot \left(\frac{q_i \cdot q_{H_1}}{2^l}\right)^{q_t} \cdot \left(\frac{q_p}{2^l}\right)^{1+q_i} \tag{13}$$

Secure and efficient key updating

Our scheme achieves both secure and efficient key updating. According to the *Keyupdating* algorithm, the updated key component for attribute A_i at time period TP_{n+1} is calculated as $U_{\gamma,TP_{n+1}} = (k_{n+1} \cdot H_1(T_i, TP_{n+1}) - k_n \cdot H_1(T_i, TP_n),\ i \in \gamma)$. Since a user cannot obtain the value of k_{n+1}, k_n, it is computational infeasible for him to calculate the value of $U_{\gamma,TP_{n+1}}$ and update his private keys. Without loss of generality, when a user's private key TD_{γ,TP_n} was leaked during the time period TP_n, the system still maintains safe after TP_n since all the private keys have been securely updated.

With respect to the computation cost, key updating for a single attribute at a discrete period only needs one multiplication operation, one addition operation and 1 H_1

Table 1 Computation cost of algorithms in our scheme

Algorithms	Setup	Initial private key generation	Encrypt	Decrypt	Key updating
Multiplication	n + 1	n	n + 3	2n	n
Addition	0	2n	1	2n	n
Hash 1	0	n	0	0	n
Hash 2	0	0	1	1	0

operation. Besides, the system parameters remain unchanged throughout different time periods and this will reduce more the communication overheads.

Performance analysis

In this paper, assuming that the number of attributes involved in encryption is n, according to the algorithms discussed above, the *Encrypt* algorithm will take $(n + 3)$ times of multiplication operations, one addition operation and one H_2 operation, while the *Decrypt* algorithm will take *2n* times of multiplication operations, $2n$ times of addition operations and 1 H_2 operation. Detailed computation costs of each algorithm is listed in Table 1.

From Table 1, it can be seen that the total computation efficiency is much higher in our scheme compared to current ABE schemes since bilinear pairing operations have been totally eliminated.

Conclusion

In this paper, we combine the advantage of key-insulation mechanism with ABE and propose a high efficient key-insulated ABE algorithm without pairings (KI-ABE-WP). During the running of algorithms in our scheme, users and AA needn't run any bilinear pairing operations. The high efficiency and proved security make our scheme more appropriate for data sharing in network systems, especially those with limited computing capacity such as wireless sensor networks, mobile communication system, etc.

Our future research should focus on the ABS (attribute based signature) without pairing operations, which provides secure data authentication with higher efficiency than current ABS schemes.

Authors' contributions

Dr. Hanshu Hong: Carried out the KI-ABE-WP studies, participated in the design of scheme and drafted the manuscript. Dr. Zhixin Sun: Participated in the performance analysis of the scheme. Both authors read and approved the final manuscript.

Authors' information

Dr. Zhixin Sun is the dean of Internet of Things institute, Nanjing University of Posts and Telecommunications. He has published more than 50 literatures on journals worldwide. His research area includes information security, computer networks, computer science, etc. Dr. Hanshu Hong is a Ph.D. candidate in Nanjing University of Posts and Telecommunications. His research area includes information security, cryptology.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (61373135). The authors thank the sponsors for their support and the reviewers for helpful comments.

Competing interests

Both authors declare that they have no competing financial interests.

Received: 23 November 2015 Accepted: 12 February 2016
Published online: 19 February 2016

References

Bertoni G, Chen L, Fragneto P, Harrison K, Pelosi G (2005) Computing tate pairing on smartcards. In: Proceedings of the workshop on cryptographic hardware and embedded systems (CHES '05)

Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute based encryption. In: Proceedings of the IEEE symposium on security and privacy, Washington, DC, pp 321–334

Chen L, Cheng Z, Smart NP (2007) Identity-based key agreement protocols from pairings. Int J Inf Secur 6(4):213–241

Chen Y, Xu W, Xiong H (2015) Strongly secure certificateless key-insulated signature secure in the standard model. Ann Telecommun 70(9):395–405

Coron J-S (2000) On the exact security of full domain hash. In: Advances in cryptology (CRYPTO '2000). LNCS no. 1880. Springer, Berlin, pp 229–235

Dodis Y, Katz J, Xu S, Yung M (2002) Key-insulated public-key cryptosystems. In: Knudsen LR (ed) Proceedings of the Eurocrypt 2002. LNCS no. 2332. Springer, Berlin, pp 65–82

Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute based encryption for fine-grained access control of encrypted data. In: ACM conference on computer and communications security, pp 89–98

Goyal V, Jain A, Pandey O, Sahai A (2008) Bounded ciphertext policy attribute based encryption. In: Proceedings of the 35th international colloquium, Reykjavik, Iceland, pp 579–591

Hu C, Zhang N (2013) Body area network security: a fuzzy attribute-based signcryption scheme. IEEE J Sel Areas Commun 31(9):37–46

Hur J (2013) Improving security and efficiency in attribute-based data sharing. IEEE Trans Knowl Data Eng 25(10):2271–2282

Hur J, Noh DK (2011) Attribute-based access control with efficient revocation in data outsourcing systems. IEEE Trans Parallel Distrib Syst 22(7):1214–1221

Lewko A, Okamoto T, Sahai A, Takashima K, Waters B (2010) Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Advances in cryptology—EUROCRYPT 2010. Springer, Berlin, pp 62–91

Li M, Yu S (2013) Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans Parallel Distrib Syst 24(1):131–143

Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Proceedings of the international conference in EUROCRYPT 2005, Aarhus, Denmark, pp 457–473

Tan Y-L, Goi B-M, Komiya R (2011) A study of attribute-based encryption for body sensor networks. Inform Eng Inf Sci 251:238–247

Wang C, Huang J (2011) Attribute based signcryption with ciphertext policy and claim predicate mechanism. In: Seventh international conference in CIS, pp 905–909

Waters B (2011) Ciphertext policy attribute based encryption: an expressive, efficient, and provably secure realization. In: Proceedings of the international conference in PKC 2011, Taormina, Italy, pp 53–70

Xhafa F, Feng J, Zhang Y (2015) Privacy-aware attribute-based PHR sharing with user accountability in cloud computing. J Supercomput 71(5):1607–1619

Xu Z, Martin KM (2012) Dynamic user revocation and key refreshing for attribute-based encryption in cloud storage. In: IEEE 11th international conference on trust, security and privacy in computing and communications

Yang K, Jia X, Kui R (2012) DAC-MACS: effective data access control for multiauthority cloud storage systems. IEEE Trans Inf Forensics Secur 7(2):743–754

Yu S, Wang C, Ren K (2010) Attribute based data sharing with attribute revocation. In: Proceedings of the 5th symposium on information, computer and communications security (ACM), pp 261–270

Yu S, Ren K, Lou W (2011) FDAC: toward fine-grained distributed data access control in wireless sensor networks. IEEE Trans Parallel Distrib Syst 22(4):673–686

Zhu G, Xiong H, Wang R (2014) An improvement of an identity-based key-insulated signcryption. In: Proceedings of the international conference on computer science and information technology, vol 255 of the series advances in intelligent systems and computing, pp 97–104