Information Processing Letters 147 (2019) 6–9

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A conditional access system with revocation for mobile pay-TV systems revisited

Alfredo Rial

University of Luxembourg, Luxembourg

ARTICLE INFO

Article history:
Received 1 February 2018
Received in revised form 25 January 2019
Accepted 24 February 2019
Available online 5 March 2019
Communicated by Luca Viganò

Keywords:
Attribute-based encryption
Broadcast encryption
Cryptography

ABSTRACT

In pay-TV, conditional access systems (CAS) are used by a rights issuer to guarantee that only authorized subscribers gain access to TV channels. A CAS scheme that applies attribute-based access control through attribute-based encryption (ABE) with revocation was proposed by Yeh and Huang (2013) [1]. Yeh and Huang extend an existing ABE scheme with a revocation mechanism. We show that the CAS by Yeh and Huang has security and efficiency problems. Particularly, we show that the revocation mechanism proposed by Yeh and Huang is vulnerable to collusion attacks.

1. Introduction

The CAS by Yeh and Huang [1] is based on the key-policy ABE scheme (KPABE) in [2]. Yeh and Huang extend this scheme with a revocation mechanism. In Section 2, we recall KPABE. In Section 3, we recall the CAS in [1]. We discuss its efficiency and security problems in Section 4.

2. Technical background

We refer to [2] for a description of bilinear maps and of access structures and access trees.

Key-policy attribute-based encryption. Let $\mathbb{S}$ be the set of all attributes. Let $\mathbb{Q}$ be the set of all access structures that are allowed over $\mathbb{S}$. We denote by $\mathbb{A} \vDash \mathbb{P}$ the fact that a set of attributes $\mathbb{A}$ satisfies the access structure $\mathbb{P}$. A key-policy attribute-based encryption (KPABE) scheme [2] consists of the algorithms ($\mathsf{ABESetup}$, $\mathsf{ABEKeyGen}$, ABEEnc, ABEDec). The key generation center ($\mathcal{KGC}$) executes $\mathsf{ABESetup}(1^k)$ to output public parameters *par* and master secret key *msk*. A user $\mathcal{U}$ with access structure $\mathbb{P} \in \mathbb{Q}$ queries $\mathcal{KGC}$, which runs $\mathsf{ABEKeyGen}(msk, \mathbb{P})$ and returns secret key $sk_{\mathbb{P}}$. On input message m and set of attributes $\mathbb{A} \in \mathbb{S}$, $\mathsf{ABEEnc}(par, m, \mathbb{A})$ computes ciphertext ct, which can only be decrypted by a user $\mathcal{U}$ that possesses an access structure $\mathbb{P}$ satisfied by $\mathbb{A}$ (ct implicitly contains $\mathbb{A}$). On input ciphertext ct and secret key $sk_{\mathbb{P}}$, ABEDec outputs message m if $\mathbb{A}$ satisfies $\mathbb{P}$.

3. Conditional access system by Yeh and Huang

The scheme proposed in [1] consists of an initialization phase, a subscriber registration phase, a rights management phase, and a subscriber revocation phase. (We note that the key stream and traffic encryption layers remain unchanged with respect to the DVB-H standard.) The scheme in [1] is based on the KPABE scheme in [2] and adds to it support for revocation. In the following description, we use boxes to highlight those elements that are added to the scheme in [2] in order to support revocation.

Initialization phase. The rights issuer (RI) executes the setup algorithm of the KPABE scheme to get the public parameters and the master secret key. Let $\mathbb{S}$ be the set of all

E-mail address: alfredo.rial@uni.lu.

https://doi.org/10.1016/j.ipl.2019.02.010

attributes and let 2^N be the maximum number of users. Algorithm $\mathsf{ABESetup}(1^k)$ works as follows.

1. Run $(p, \mathbb{G}, \mathbb{G}_t, e, g) \leftarrow \mathcal{G}(1^k)$.
2. Pick random $y \leftarrow \mathbb{Z}_p$.
3. For all $j \in \mathbb{S}$, pick random $t_j \leftarrow \mathbb{Z}_p$ and compute $T_j \leftarrow g^{t_j}$.
4. $\boxed{\text{Pick random } \tau \leftarrow \mathbb{Z}_p \text{ and compute } W \leftarrow g^{\tau}.}$
5. $\boxed{\text{Pick random } \Upsilon \leftarrow \mathbb{Z}_p.}$
6. $\boxed{\text{For } i = 1 \text{ to } N\text{, pick } c_i, d_i \leftarrow \mathbb{Z}_p \text{ and set } h_{i,0} \leftarrow g^{c_i} \text{ and } h_{i,1} \leftarrow g^{d_i}.}$
7. Set the parameters
   $$par \leftarrow (p, \mathbb{G}, \mathbb{G}_t, e, g, \{T_j\}_{j \in \mathbb{S}} \boxed{, W, \{h_{i,0}, h_{i,1}\}_{i=1}^{N}}).$$[1]
8. Set the master secret key
   $$msk \leftarrow (y, \{t_j\}_{j \in \mathbb{S}} \boxed{, \tau, \Upsilon, \{h_{i,0}, h_{i,1}\}_{i=1}^{N}}).$$[2]
9. Output the master secret key msk and the public parameters par.

The elements $(y, \{t_j\}_{j \in \mathbb{S}})$ in the master secret key and the elements $(p, \mathbb{G}, \mathbb{G}_t, e, g, \{T_j\}_{j \in \mathbb{S}})$ in the public parameters are those of Goyal et al.'s KPABE scheme [2]. The remaining elements are added by Yeh and Huang in order to support revocation.

Subscriber registration phase. A subscriber possesses a set of attributes or characteristics, such as age and subscription category. Given those attributes, RI constructs an access structure $\mathbb{P}$. Then RI runs the key generation algorithm of the KPABE scheme on input $\mathbb{P}$ in order to compute a key that allows the subscriber to obtain all the right objects (RO) that match her age and subscription category. Algorithm $\mathsf{ABEKeyGen}$ also receives as input an n-bit revocation number $X_{n-1}X_{n-2}\cdots X_0$, which must be unique for each subscriber. $\mathsf{ABEKeyGen}(msk, \mathbb{P})$ works as follows.

1. $\boxed{\text{Pick random } \Psi \leftarrow \mathbb{Z}_p \text{ and compute } WD \leftarrow g^{(y-\Psi)/\tau}.}$
2. For each node x in the tree $\mathcal{T}$ that defines $\mathbb{P}$, choose a polynomial q_x of degree $d_x = k_x - 1$, where k_x is the threshold value of x. Polynomials q_x are chosen in a top-down manner, starting from the root node R. Set $q_R(0) = \Psi$. (In Goyal et al., $q_R(0) = y$. In Yeh and Huang, an additional element WD is included in the key, which depends on both y and Ψ. WD is the key element that is updated when a subscriber is revoked.) Choose other d_R points randomly to define q_R completely. For any other node x, set $q_x(0) \leftarrow q_{\mathsf{parent}(x)}(\mathsf{index}(x))$ and choose other d_x points randomly to define q_x completely.
3. Let Y be the set of leaf nodes in $\mathcal{T}$. Compute $\{D_j \leftarrow g^{q_j(0)/t_i}\}_{j \in Y}$.
4. $\boxed{\text{Choose a value } q_1(0).}$
5. $\boxed{\text{For } i \in [0, N-1],\ DR_i \leftarrow g^{(q_1(0)\Upsilon)/c_i} \text{ if } X_i = 0 \text{ else } DR_i \leftarrow g^{(q_1(0)\Upsilon)/d_i}.}$
6. Set the secret key as
   $$sk_{\mathbb{P}} \leftarrow (\{D_j\}_{j \in Y} \boxed{, WD, \{DR_i\}_{i=0}^{N-1}, q_1(0)^{-1}}).$$[3]

The elements $\{D_j\}_{j \in Y}$ are those of the secret key in the scheme by Goyal et al. [2]. The elements $(WD, \{DR_i\}_{i=0}^{N-1}, q_1(0)^{-1})$ are added by Yeh and Huang in order to support revocation.

Rights management phase. RI creates a right object (RO) that consists of the service encryption key (SEK), a set of attributes $\mathbb{A}$, and related parameters, such as a time stamp. RI encrypts the right object RO by running the encryption algorithm of the KPABE scheme on input $\mathbb{A}$. $\mathsf{ABEEnc}(par, RO, \mathbb{A})$ works as follows.

1. Pick random $r \leftarrow \mathbb{Z}_p$, compute $E' \leftarrow RO \cdot (g, g)^{yr}$ (here, y is taken from the master secret key) and, for all $j \in \mathbb{A}$, compute $E_j \leftarrow T_j^r$.
2. $\boxed{\text{Compute } g^r.}$
3. Set the ciphertext $ct \leftarrow (\mathbb{A}, \{E_j\}_{j \in \mathbb{A}}, E', \boxed{g^r})$.

The elements $(\mathbb{A}, \{E_j\}_{j \in \mathbb{A}}, E')$ form the ciphertext in the scheme by Goyal et al. [2]. The value g^r is included by Yeh and Huang [1] in order to allow decrypting the message along with the new value WD in the secret key.

RI broadcasts the ciphertext to the subscribers. A subscriber decrypts the ciphertext by using the decryption algorithm $\mathsf{ABEDec}(par, ct, sk_{\mathbb{P}})$. If the policy $\mathbb{P}$ associated with $sk_{\mathbb{P}}$ is satisfied by the set of attributes $\mathbb{A}$ in ct, ABEDec first computes $A = e(g, g)^{r\Psi}$ and finally outputs $RO = E'/(e(WD, W^r)A)$. The decryption algorithm in Yeh and Huang follows that of Goyal et al. The only difference is the last equation $RO = E'/(e(WD, W^r)A)$, which includes the additional pairing operation $e(WD, W^r)$.

Subscriber revocation phase. The following revocation mechanism is proposed by Yeh and Huang [1]. To revoke the subscriber with revocation number $X_{n-1}X_{n-2}\cdots X_0$, RI and the subscribers execute a protocol by means of which RI updates the value y in the master secret key, while the non-revoked subscribers update the value WD in their respective secret keys. First, RI executes the following steps.

1. Pick $y' \leftarrow \mathbb{Z}_p$, compute $\Delta_y \leftarrow y' - y$ and $W' \leftarrow g^{\Delta_y/\tau}$.[4]

[1] We note that the scheme by Goyal et al. [2] includes an element $e(g, g)^y$ in the public parameters. This element is not included in the public parameters in Yeh and Huang [1] because the rights issuer is the only party that needs to compute ciphertexts and the rights issuer has knowledge of the master secret key, which includes y.

[2] In Yeh and Huang [1], Υ is included neither in msk nor in par.

[3] We note that the value $q_1(0)$ is unnecessary. One can simply compute $DR_i \leftarrow g^{\Upsilon/c_i}$, if $X_i = 0$, or $DR_i \leftarrow g^{\Upsilon/d_i}$, if $X_i = 1$.

[4] We note that [1] says "generate the new public parameter $W' \leftarrow g^{\Delta_y/\tau}$ to replace W". This is a mistake. If W' replaces W in the public parameters and in the computation of ciphertexts, decryption does not work anymore. Furthermore, W' is a value that should not be revealed to the revoked subscriber. Consequently, W' should not become part of the public parameters because otherwise the revoked subscriber can learn it.

2. Pick random $r \leftarrow \mathbb{Z}_p$, compute $RE' \leftarrow W'e(g,g)^{\Upsilon r}$ and, for $i = 0$ to $N-1$, compute $RE_i \leftarrow h^r_{i,\bar{X}_i}$, where $\bar{X}_i \leftarrow 1 - X_i$.
3. Set a ciphertext $ct \leftarrow (\bar{X}_{n-1}\bar{X}_{n-2}\cdots\bar{X}_0, RE', \{RE_i\}_{i=0}^{N-1})$.

RI broadcasts the ciphertext ct to the subscribers. As can be seen, only the revoked subscriber is unable to decrypt the ciphertext. The reason is that the revoked subscriber has a secret key for the revocation number $X_{n-1}X_{n-2}\cdots X_0$ and none of the bits in $X_{n-1}X_{n-2}\cdots X_0$ matches a bit in $\bar{X}_{n-1}\bar{X}_{n-2}\cdots\bar{X}_0$. However, because revocation numbers are unique for each subscribers, for the non-revoked subscribers there is at least one bit that matches. A non-revoked subscriber decrypts the ciphertext as follows. First, find a bit X_i such that $X_i = \bar{X}_i$ and take the element DR_i of the secret key. Then compute $W' \leftarrow RE'/e(DR_i, q_1(0)^{-1}RE_i)$. After retrieving W', a non-revoked subscriber computes $WD' \leftarrow WD \cdot W' = g^{(y'-\Psi)/\tau}$ to update her secret key.

4. Discussion of the CAS proposed by Yeh and Huang

Collusion-resistance. Yeh and Huang define collusion-resistance as follows: “A subscriber cannot cooperate with other subscribers to promote his own privileges.” This basically means that two or more colluding subscribers that hold secret keys for the KPABE scheme should not be able to decrypt any ciphertext that any of them is not able to decrypt on their own. I.e., the colluding subscribers are not able to combine their secret keys in such a way that they are able to decrypt ciphertexts that none of them can decrypt individually. The KPABE scheme in [2] is collusion-resistant.

However, we show that the KPABE with revocation scheme proposed by Yeh and Huang in [1] is vulnerable to collusion attacks. In the revocation method in [1], the rights issuer RI produces key update material that is broadcast to all the non-revoked subscribers. The scheme in [1] effectively prevents the revoked subscriber from retrieving key update material. However, it does not ensure that a revoked subscriber and a non-revoked one cannot collude to decrypt ciphertexts that they are not able to decrypt on their own.

Let us show this through an example with two subscribers S_1 and S_2 and three subscriber attributes or categories c_1, c_2 and c_3. S_1 (resp. S_2) possesses a secret key that allows her to obtain right objects RO for category c_1 (resp. c_2). Collusion-resistance of the KPABE scheme in [2] ensures that, if S_1 and S_2 collude, they are able to obtain right objects for categories c_1 and c_2 because they are able to do that without colluding, but they are not able to obtain right objects for category c_3. At some point, subscriber S_2 is revoked by the rights issuer. After revocation, S_1 updates her key and is still able to obtain right objects for category c_1. S_2 is not able to update her key and is unable to decrypt anything. However, if S_1 and S_2 collude, S_1 can give S_2 the key update material, and then they can obtain again right objects for category c_2. This violates collusion-resistance because, after S_2 is revoked, neither S_1 nor S_2 are able to obtain right objects for category c_2 on their own. The figure below describes the attack in the subscriber revocation phase.

- RI broadcasts the ciphertext $ct \leftarrow (\bar{X}_{n-1}\bar{X}_{n-2}\cdots\bar{X}_0, RE', \{RE_i\}_{i=0}^{N-1})$.
- A non-revoked subscriber S_1 computes $W' \leftarrow RE'/e(DR_i, q_1(0)^{-1}RE_i)$ by using DR_i for i such that $X_i = \bar{X}_i$.
- S_1 sends W' to a revoked subscriber S_2 with whom S_1 colludes.
- S_2 computes $WD' \leftarrow WD \cdot W' = g^{(y'-\Psi)/\tau}$ to update her secret key. Thanks to that, S_2 can decrypt as if S_2 was not revoked.

To illustrate that this is a serious security flaw, let us point out that a trivial (albeit inefficient) revocation method does not suffer from this problem. Consider a trivial revocation method where, when a subscriber is revoked, the rights issuer runs again the setup algorithm of the KPABE scheme to compute new public parameters and a new master secret key, and uses this master secret key to compute new secret keys for all the non-revoked subscribers. As can be seen, with this revocation method, after S_2 is revoked, S_1 and S_2 are not able to obtain right objects for category c_2. The reason is that the old secret key of S_2 is now useless because the parameters of the KPABE scheme have changed. In conclusion, the CAS proposed by Yeh and Huang is not collusion-resistant.

Non-repudiation. Yeh and Huang define non-repudiation as follows: “To ensure video content validity and quality, the video server should not deny that the video contents are delivered from it.” Non-repudiation basically implies that the integrity and origin of received data can be verified and that the sender of the data cannot deny being the originator.

Yeh and Huang claim that their CAS provides non-repudiation. Their argument basically states that, because RI is the only party that knows the values (y, Υ) in the master secret key, RI is the only party able to compute valid ciphertexts that encrypt right objects and valid ciphertexts that encrypt key update material. Consequently, Yeh and Huang conclude that, if a subscriber possesses a valid ciphertext, then RI is not able to deny that he computed and sent that ciphertext.

As can be seen, this argument is not valid. First, an adversary eavesdropping the network can get a ciphertext sent by RI, and forward it to subscribers at a later stage. Moreover, the adversary can modify the encrypted message before forwarding the ciphertext. For example, if the adversary wishes to modify a ciphertext $ct \leftarrow (\mathbb{A}, \{E_j\}_{j\in\mathbb{A}}, E', g^r)$ (where $E' \leftarrow RO \cdot (g,g)^{yr}$) that encrypts RO to a ciphertext that encrypts $RO \cdot M'$, the adversary simply replaces E' by $E' \cdot M$. An adversarial subscriber can also modify the message encrypted in a ciphertext received from RI by using the same technique. Therefore, RI is able to deny that he computed a given valid ciphertext. In conclusion, the CAS of Yeh and Huang does not provide non-repudiation. The figure below describes the attack in the rights management phase.

- A malicious RI broadcasts a ciphertext $ct \leftarrow (\mathbb{A}, \{E_j\}_{j\in\mathbb{A}}, E', \boxed{g^r})$, where $E' \leftarrow RO \cdot (g, g)^{yr}$ is computed on input an incorrect RO.
- An honest subscriber S decrypts the ciphertext and obtains the incorrect RO. S accuses RI of sending a ciphertext ct that encrypts an incorrect RO.
- RI claims ct was modified by an adversary that set $E' \leftarrow E' \cdot M$ for some M.
- S is unable to prove that RI indeed sent ct.

Direct revocation vs indirect revocation. There are two revocation methods for attribute-based encryption: direct [3] and indirect [4]. In an indirect revocation method, the key generation center, i.e., the trusted party that keeps the master secret key and computes keys for users, computes and publishes key update material so that only non-revoked users can update their keys. The advantage of indirect revocation is that the senders, i.e., the parties that compute ciphertexts, do not need to know the revocation list. The disadvantage of the indirect revocation method is that all the non-revoked users need to update their keys.

In a direct revocation method, senders compute ciphertexts that revoked users are not able to decrypt. In order to do that, senders need to know the revocation list, but key updates are not required. In many applications, senders do not know the revocation list, and thus only indirect revocation can be applied. However, in CAS, the only sender is the rights issuer, and the rights issuer knows the revocation list. Therefore, in CAS, direct revocation methods can be applied and then key updates are not needed.

The CAS proposed by Yeh and Huang uses an indirect revocation method where non-revoked users need to update their keys. We note that direct revocation methods avoid the need of updating keys. For instance, in a CAS that uses identity-based broadcast encryption [5] or attribute-based broadcast encryption [3], ciphertexts that encrypt the right object are computed on input the list of subscribers. When a subscriber is revoked, the rights issuer simply removes the subscriber from the list. Thanks to that, revoked subscribers are not able to decrypt the ciphertext and obtain the RO.

User-based vs attribute-based access control. Attribute-based access control allows a party to describe an access control policy that defines the attributes that a user must possess in order to be granted access to a service or resource. In settings where the party aiming at controlling access to the service does not know the identities or the attributes of the users that may attempt to gain access to it, attribute-based encryption is adequate for implementing attribute-based access control.

However, in a CAS for pay-TV, RI usually knows the identities and the attributes of all the parties that may want to access TV broadcasts because those parties must subscribe and pay a fee. (Although it would be possible to design a CAS scheme that tries to provide user anonymity by using anonymous payment methods and anonymous communication networks, this is not done in practice for efficiency and usability reasons.) In the CAS by Yeh and Huang, RI learns the identities and the attributes of the subscribers during the subscriber registration phase. The identity is learnt to assign a unique revocation number. The attributes are learnt to compute the secret key for the subscriber.

Therefore, because RI knows the identities and the attributes of the subscribers, RI can decide whether a subscriber is entitled to obtain a given RO by simply checking himself whether the subscriber's attributes fulfill the access control policy for that RO. This allows RI to replace attribute-based encryption by an encryption scheme for user-based access control, which can be instantiated more efficiently. In user-based access control, the party that controls access to a service simply makes a list of all the users that must be granted access. In CAS, for each right object (RO), RI can make a list of the identities of the subscribers that are entitled to obtain that RO.

In a CAS that uses identity-based broadcast encryption (IBBE), for each RO, RI makes a list of subscribers that are allowed to obtain it. Remarkably, the attribute-based access control functionality is enhanced, because, while in the CAS proposed by Yeh and Huang the class of access control policies that can be applied is restricted to those supported by the KPABE scheme used as building block, now RI can apply any access control policy on the subscriber's attributes.

In comparison to attribute-based access control, user-based access control allows more efficient implementations. In fact, IBBE can be realized more efficiently than ABE. Therefore, for the sake of efficiency, encryption schemes for user-based access control are preferable whenever they can be used, i.e., when the party in charge of controlling access to a service knows the identities and the attributes of all the users of the service.

References

[1] L.-Y. Yeh, J.-L. Huang, A conditional access system with efficient key distribution and revocation for mobile pay-TV systems, ACM Trans. Multimed. Comput. Commun. Appl. (TOMCCAP) 9 (3) (2013) 18.

[2] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proceedings of the 13th ACM Conference on Computer and Communications Security, ACM, 2006, pp. 89–98.

[3] N. Attrapadung, H. Imai, Conjunctive broadcast and attribute-based encryption, in: Pairing-Based Cryptography–Pairing 2009, Springer, 2009, pp. 248–265.

[4] A. Boldyreva, V. Goyal, V. Kumar, Identity-based encryption with efficient revocation, in: Proceedings of the 15th ACM Conference on Computer and Communications Security, 2008, pp. 417–426.

[5] C. Delerablée, Identity-based broadcast encryption with constant size ciphertexts and private keys, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2007, pp. 200–215.