Toward Dynamic Attribute-Based Signcryption (Poster)

Keita Emura[1], Atsuko Miyaji[2], and Mohammad Shahriar Rahman[2]

[1] Center for Highly Dependable Embedded Systems Technology
[2] School of Information Science
Japan Advanced Institute of Science and Technology, 1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
{k-emura,miyaji,mohammad}@jaist.ac.jp

Abstract. This paper presents an ongoing work toward the proposal of the new concept of the attribute-based cryptosystem. In SCN2010, Gagné, Narayan, and Safavi-Naini proposed attribute-based signcryption (ABSC) with threshold structure. As in ciphertext-policy attribute-based encryption (CP-ABE), an encryptor can specify the access structure of decryptors, and as in attribute-based signature (ABS), each decryptor can verify the encryptor's attributes. On the contrary to the access structure of decryptors, the access structure of the encryptor needs to be fixed in the setup phase. In this paper, we investigate ABSC with dynamic property, called dynamic ABSC (DABSC), where access structures of encryptor can be updated flexibly without re-issuing secret keys of users.

1 Introduction

1.1 Attribute-Based Signcryption (ABSC)

Recently, Gagné, Narayan, and Safavi-Naini proposed ABSC with threshold structure [3] to achieve Cost(ABS & CP-ABE) $<$ Cost(ABS)+Cost(CP-ABE). That is, their ABSC scheme is efficient compared with encrypt-then-sign paradigm. As in Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [1], an encryptor can specify the access structure of decryptors, and as in Attribute-Based Signature (ABS) [4], each decryptor can verify the encryptor's attributes. Note that, in the Gagné et al. definition, a decryptor can verify the encryptor's attribute *explicitly*. This property is preferable for the following encrypted storage system usage.

1.2 Encrypted Storage System

Encrypted storage system is a well-known application of CP-ABE. To indicate the set of common attributes of decryptors (such as affiliation, post, and so on), CP-ABE schemes can achieve a fine-grained access control without increasing the number of keys. On the contrary to the decryptor's attributes, there is no way to verify the set of attributes of encryptor if CP-ABE is applied only. To check the source of storage files, attributes of encryptor is important information. By applying the Gagné et al. ABSC, both CP-ABE and ABS properties can be handled for encrypted storage system usage, simultaneously. So, a decryptor can check the encryptor's attribute explicitly. However, the threshold structure (which is supported by the Gagné et al. ABSC) is not suitable for encrypted storage system usage, although it is useful for fault tolerance usage. In addition, the access structure of the encryptor is specified only once, and it cannot be changed. More precisely, the threshold value is decided in the key generation phase, and it cannot be updated without re-issuing the new key.

2 Our Approach: Dynamic ABSC

In this paper, we investigate the new concept "ABSC with dynamic property", called Dynamic ABSC (DABSC), where access structures of encryptor can be changed without re-issuing secret keys of users. As an application of DABSC, we consider authenticated fine-grained storage systems.

For example, let a teaching assistant of a lecture "Applied Cryptography" would like to store an encrypted examination data for students (who take Applied Cryptography) only. In addition, students would like to check whether a stored file was made by a teaching assistant of Applied Cryptography. Then an encryptor makes a ciphertext part associated with attributes of a decryptor (Student $\wedge$ Applied Cryptography), and also makes a signature part associated with attributes of the encryptor (Teaching Assistant $\wedge$ Applied Cryptography).

The dynamic property is suitable for the following example. We assume that the encryptor (who is a teaching assistant of Applied Cryptography) becomes a teaching assistant of a lecture "Discrete Mathematics", and the encryptor has obtained the secret key for attributes Teaching Assistant and Applied Cryptography. If the dynamic property is not handled, then key generation center (KGC) needs to re-issue the secret key of both Applied Cryptography and Teaching Assistant for handling the updated access structure of encryptor. It is quite inefficient and impractical (See Table 1).

Table 1. Computational complexity of changing predicate

	KGC	User
Non-dynamic scheme	$O(N \cdot n_e)$	$O(n_e)$
Dynamic scheme	$O(n_e)$	None

N : the number of users
n_e : the maximum number of attributes having each user

Under the dynamic property, KGC has only to issue the secret key of Discrete Mathematics, and no computation that the encryptor is required. While the above example describes the case of small number of predicates, we believe that the dynamic property gives us a very efficient and practical solution when the number of predicates grows large. Actually, the number of attributes is polynomial-size (of the security parameter k, i.e., $O(poly(k))$) and the corresponding predicates can grow exponentially (i.e., $O(2^{poly(k)})$) in large systems.

That is, as an expectation, the opportunity of updating predicates also increases in such large systems. Under the dynamic property, even if the current predicate is updated, users do not have to be involved the updating procedure. This is the most benefit point of our proposal.

3 System Operations of DABSC

In the following, values are subscripted by e for encryptors, and values are subscripted by d for decryptors. Let $\mathbb{A}_e = (att_1, att_2, \ldots, att_{n_e})$ be the universe of possible attributes of encryptors, $\mathbb{A}_d = (att_1, att_2, \ldots, att_{n_d})$ be the universe of possible attributes of decryptors, and Υ_e (resp. Υ_d) be a claim-predicate over $\mathbb{A}_e$ (resp. $\mathbb{A}_d$) of encryptors (resp. decryptors). We say that an attribute set $\Gamma_e \subseteq \mathbb{A}_e$ (resp. $\Gamma_d \subseteq \mathbb{A}_d$) satisfies a claim-predicate Υ_e (resp. Υ_d) if $\Upsilon_e(\Gamma_e) = 1$ (resp. $\Upsilon_d(\Gamma_d) = 1$). In the following definition, an encryptor can select an access structure of decryptor Υ_d for each signcryption ciphertext (which follows the ciphertext-policy property of ABE). On the contrary, an access structure of encryptor Υ_e is a publicly opened. This means that legitimated encryptor who have attributes satisfying Υ_e can make a signcryption ciphertext.

Next, we modify the definitions of the Gagné et al. ABSC [3] to handle the dynamic property.

Definition 1 (Dynamic Attribute-Based Signcryption (DABSC)).

Setup: *This algorithm takes as inputs a security parameter $k \in \mathbb{N}$, and returns public parameters $params$ and a master key msk.*

sExtract: *This algorithm takes as inputs $params$, msk, and a set of attributes of an encryptor $\Gamma_e \subseteq \mathbb{A}_e$, and returns signing keys $\{sk_{e,i}\}_{att_i \in \Gamma_e}$.*

uExtract: *This algorithm takes as inputs $params$, msk, and a set of attributes of an decryptor $\Gamma_d \subseteq \mathbb{A}_d$, and returns decryption keys $\{sk_{d,i}\}_{att_i \in \Gamma_d}$.*

BuildPredicate: *This algorithm takes as inputs $params$, msk, and the ℓ-th access tree T_ℓ, and returns the public value of ℓ-th access tree Υ_e^ℓ.*

Signcrypt: *This algorithm takes as inputs $params$, Υ_e^ℓ, $\{sk_{e,i}\}_{att_i \in \Gamma_e}$, where $\Upsilon_e^\ell(\Gamma_e) = 1$, an access structure Υ_d, and a plaintext M, and returns a ciphertext C on M. We assume that Γ_e and Υ_d are included into C.*

Unsigncrypt: *This algorithm takes as inputs $params$, Υ_e^ℓ, $\{sk_{d,i}\}_{att_i \in \Gamma_d}$, where $\Upsilon_d(\Gamma_d) = 1$, and C, and verifies whether the encryptor's attributes satisfy Υ_e^ℓ or not, along with Γ_e and Υ_e^ℓ. If not, then output $\perp$, and M otherwise.*

The above algorithms follow the correctness requirement: for all $(params, msk) \leftarrow \mathsf{Setup}(1^k)$, $\{sk_{e,i}\}_{att_i \in \Gamma_e} \leftarrow \mathsf{sExtract}(params, msk, \Gamma_e)$, $\{sk_{d,i}\}_{att_i \in \Gamma_d} \leftarrow \mathsf{uExtract}(params, msk, \Gamma_d)$, $\Upsilon_e^\ell \leftarrow \mathsf{BuildPredicate}(params, msk, T_\ell)$, and $C \leftarrow \mathsf{Signcrypt}(params, \Upsilon_e^\ell, \{sk_{e,i}\}_{att_i \in \Gamma_e}, \Upsilon_d, M)$ with $\Upsilon_e^\ell(\Gamma_e) = 1$, $M \leftarrow \mathsf{Unsigncrypt}(params, \Upsilon_e^\ell, \{sk_{d,i}\}_{att_i \in \Gamma_d}, C)$ holds when $\Upsilon_d(\Gamma_d) = 1$.

Next, we define indistinguishability against adaptive chosen-ciphertext attack property under selective attribute model (S-IND-DABSC-CCA2), existential unforgeability against chosen-message attack in the selective attribute model

Table 2. DABSC Experiments

S-IND-DABSC-CCA2
$Adv_{\mathcal{A}}^{\text{S-IND-DABSC-CCA2}}(k) =$ $\vert \Pr\big[(\Upsilon_d^*, T_0, State) \leftarrow \mathcal{A}(k);\ (params, msk) \leftarrow \mathsf{Setup}(1^k);$ Set $\mathcal{O} := \{\mathsf{sExtract}(params, msk, \cdot), \mathsf{uExtract}(params, msk, \cdot),$ $\mathsf{Unsigncrypt}(params, \cdot, \cdot), \mathsf{BuildPredicate}(params, msk, \cdot)\};$ $(M_0^*, M_1^*, \Gamma_e^*, State) \leftarrow \mathcal{A}^{\mathcal{O}}(params, State);\ b \xleftarrow{\$} \{0,1\};$ $C^* \leftarrow \mathsf{Signcrypt}(params, \Upsilon_e^{\ell}, \{sk_{e,i}\}_{att_i \in \Gamma_e^*}, \Upsilon_d^*, M_b^*);\ b' \leftarrow \mathcal{A}^{\mathcal{O}}(C^*, State);\ b = b'\big] - \frac{1}{2}\vert$
S-EUF-DABSC-CMA
$Adv_{\mathcal{A}}^{\text{S-EUF-DABSC-CMA}}(k) =$ $\Pr\big[(\Upsilon_e^*, T_0, State) \leftarrow \mathcal{A}(k);\ (params, msk) \leftarrow \mathsf{Setup}(1^k);$ Set $\mathcal{O} := \{\mathsf{sExtract}(params, msk, \cdot), \mathsf{uExtract}(params, msk, \cdot),$ $\mathsf{BuildPredicate}(params, msk, \cdot), \mathsf{Signcrypt}(params, \cdot, \cdot, \cdot)\};\ (C^*, \Gamma_d^*) \leftarrow \mathcal{A}^{\mathcal{O}}(params, State);$ $\mathsf{Unsigncrypt}(params, \Upsilon_e^*, \{sk_{d,i}\}_{att_i \in \Gamma_d^*}, C^*) = M^* \neq \perp;$ (For Γ_e where $\Upsilon_e^*(\Gamma_e) = 1$, $\mathcal{A}$ did not query either $(M, \Gamma_e, \Upsilon_d^*)$ to the $\mathsf{Signcrypt}$ oracle or Γ_e to the $\mathsf{sExtract}$ oracle, where $\Upsilon_e^*(\Gamma_e) = 1) \vee (\Upsilon_e^*(\Gamma_e^*) \neq 1)\big]$

(S-EUF-DABSC-CMA). In the following, T (and the initial access tree T_0 also) must follow the condition that leaves of trees are appeared in $\mathbb{A}_e$. S-IND-DABSC-CCA2 guarantees that no PPT adversary $\mathcal{A}$ (which is essentially the same as the CCA adversary of CP-ABE [1]) can guess whether the actual plaintext is M_0^* or M_1^*, namely, no plaintext information is revealed from the ciphertext. Note that S-IND-DABSC-CCA2 captures collusion resistance (i.e., $\mathcal{A}$ is allowed to issue Γ_d and Γ_d' to the $\mathsf{uExtract}$ oracle such that $\Upsilon_d^*(\Gamma_d) \neq 1$, $\Upsilon_d^*(\Gamma_d') \neq 1$, $\Gamma_d \cup \Gamma_d' = \Gamma_d^*$, and $\Upsilon_d^*(\Gamma_d^*) = 1$) as in the conventional CP-ABE definition.

Definition 2 (S-IND-DABSC-CCA2). *A DABSC scheme is said to be S-IND-DABSC-CCA2 secure if the advantage $Adv_{\mathcal{A}}^{\text{S-IND-DABSC-CCA2}}(k)$ is negligible for any PPT adversary $\mathcal{A}$ in the S-IND-DABSC-CCA2 experiment (defined in Table 2).*

Note that we require $\Upsilon_e^{\ell}(\Gamma_e^*) = 1$, where Υ_e^{ℓ} is the public predicate in the challenge phase. In addition, for (C, Γ_d) which is an input of the unsigncryption oracle $\mathsf{Unsigncrypt}$, if $C = C^*$ and $\Upsilon_d^*(\Gamma_d) = 1$, then the oracle returns $\perp$. Otherwise, it returns the result of $\mathsf{Unsigncrypt}(params, \Upsilon_e^i, \{sk_{d,i}\}_{att_i \in \Gamma_d}, C)$, where Υ_e^i is the current predicate when $\mathcal{A}$ issues the unsigncryption query.

Next, we define S-EUF-DABSC-CMA. In the definition of S-EUF-DABSC-CMA, we consider two types adversaries. S-EUF-DABSC-CMA guarantees that no (type 1) adversary $\mathcal{A}$ can make a forged ciphertext which is correctly decrypted (i.e., the $\mathsf{Unsigncrypt}$ algorithm outputs $M \neq \perp$) even though $\mathcal{A}$ did not issue either Γ_e to the $\mathsf{sExtract}$ oracle such that $\Upsilon_e^*(\Gamma_e) = 1$ or $(M, \Gamma_e, \Upsilon_d^*)$ to the $\mathsf{Signcrypt}$ oracle such that $\Upsilon_e^*(\Gamma_e) = 1$, and no (type 2) $\mathcal{A}$ (who can obtain all $\{sk_{e,i}\}_{att_i \in \Gamma_e}$) can make a forged ciphertext which is correctly decrypted even though $\Upsilon_e^*(\Gamma_e) \neq 1$. Type 1 adversary (which is the same as the unforgeability adversary of ABS [4]) captures collusion resistance (i.e., $\mathcal{A}$ is allowed to issue Γ_e and Γ_e' to the $\mathsf{sExtract}$ oracle such that $\Upsilon_e^*(\Gamma_e) \neq 1$, $\Upsilon_e^*(\Gamma_e') \neq 1$, $\Gamma_e \cup \Gamma_e' = \Gamma_e^*$, and $\Upsilon_e^*(\Gamma_e^*) = 1$). Type 2 adversary captures that the Unsigncrypt algorithm does not accept the ciphertext made by Γ_e such that $\Upsilon_e^*(\Gamma_e) \neq 1$ with overwhelming probability.

Definition 3 (S-EUF-DABSC-CMA). *A DABSC scheme is said to be wS-EUF-DABSC-CMA secure if the advantage* $Adv_{\mathcal{A}}^{S\text{-}EUF\text{-}DABSC\text{-}CMA}(k)$ *is negligible for any PPT adversary* $\mathcal{A}$ *in the S-EUF-DABSC-CMA experiment (defined in Table 2).*

Note that, let $\Upsilon_e^* \leftarrow \mathsf{BuildPredicate}(params, msk, T_e^*)$ be the predicate when $\mathcal{A}$ outputs the forged ciphertext.

4 Conclusion and Toward the Concrete Construction of DABSC Scheme

This paper has presented an ongoing work toward the new concept, called DABSC. We define the system operations and the security requirements of DABSC. Toward the concrete construction of DABSC scheme, our methodology is described as follows. The dynamic property is achieved by applying a *bottom-up approach* construction [2], where first all secret values (assigned with leaves) are chosen, and then each parents secret is computed from bottom up. It seems that DABSC can be implemented based on appropriate CP-ABE and ABS with the bottom-up approach. It is particularly worth noting that the bottom-up approach construction itself does not require the random oracle, although the eventual dynamic ABGS [2] requires the random oracle. That is, DABSC secure in the standard model is expected.

It might be the case that the actual complexity of Signcrypt/Unsigncrypt algorithms in the dynamic scheme is worse as compared to the non-dynamic schemes since certain dynamic-property-related values may have to be included in the ciphertext. As for small system the dynamic property may not be very effective. It remains to be seen how large the number of attributes should be set as threshold between the dynamic and non-dynamic schemes.

References

1. Cheung, L., Newport, C.C.: Provably secure ciphertext policy ABE. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) ACM Conference on Computer and Communications Security. pp. 456–465. ACM (2007)
2. Emura, K., Miyaji, A., Omote, K.: A dynamic attribute-based group signature scheme and its application in an anonymous survey for the collection of attribute statistics. IPSJ Journal 50(9), 1968–1983 (2009)
3. Gagné, M., Narayan, S., Safavi-Naini, R.: Threshold attribute-based signcryption. In: Garay, J.A., Prisco, R.D. (eds.) SCN. Lecture Notes in Computer Science, vol. 6280, pp. 154–171. Springer (2010)
4. Li, J., Au, M.H., Susilo, W., Xie, D., Ren, K.: Attribute-based signature and its applications. In: Feng, D., Basin, D.A., Liu, P. (eds.) ASIACCS. pp. 13–16 (2010)