Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency [☆]

Kwangsu Lee [a], Seung Geol Choi [b], Dong Hoon Lee [c,*], Jong Hwan Park [d], Moti Yung [e]

[a] Sejong University, Seoul, Republic of Korea
[b] United States Naval Academy, MD, USA
[c] Korea University, Seoul, Republic of Korea
[d] Sangmyung University, Seoul, Republic of Korea
[e] Snapchat Inc. and Columbia University, New York, USA

ARTICLE INFO

Article history:
Received 24 September 2015
Received in revised form 21 November 2016
Accepted 23 December 2016
Available online xxxx
Communicated by G. Persiano

Keywords:
Public-key encryption
Attribute-based encryption
Predicate encryption
Self-updatable encryption
Revocation
Key evolving systems
Cloud storage

ABSTRACT

Revocation and key evolving paradigms are central issues in cryptography, and in PKI in particular. A novel concern related to these areas was raised in the recent work of Sahai, Seyalioglu, and Waters (CRYPTO 2012) who noticed that revoking past keys should at times (e.g., the scenario of cloud storage) be accompanied by revocation of past ciphertexts (to prevent unread ciphertexts from being read by revoked users). They introduced revocable-storage attribute-based encryption (RS-ABE) as a good access control mechanism for cloud storage. RS-ABE protects against the revoked users not only the future data by supporting key-revocation but also the past data by supporting ciphertext-update, through which a ciphertext at time T can be updated to a new ciphertext at time $T+1$ *using only the public key*. Motivated by this pioneering work, we ask whether it is possible to have a modular approach, which includes a primitive for time managed ciphertext update as a primitive. We call encryption which supports this primitive a "self-updatable encryption" (SUE). We then suggest a modular cryptosystems design methodology based on three sub-components: a primary encryption scheme, a key-revocation mechanism, and a time-evolution mechanism which controls the ciphertext self-updating via an SUE method, coordinated with the revocation (when needed). Our goal in this is to allow the self-updating ciphertext component to take part in the design of new and improved cryptosystems and protocols in a flexible fashion. Specifically, we achieve the following results:

We first introduce a new cryptographic primitive called *self-updatable encryption (SUE)*, realizing a time-evolution mechanism. In SUE, a ciphertext and a private key are associated with time. A user can decrypt a ciphertext if its time is earlier than that of his private key. Additionally, *anyone (e.g., a cloud server) can update the ciphertext* to a ciphertext with a newer time. We also construct an SUE scheme and prove its full security under static assumptions. Following our modular approach, we present a new RS-ABE scheme with shorter ciphertexts than that of Sahai et al. and prove its security. The length efficiency is mainly due to our SUE scheme and the underlying modularity. We apply our approach to predicate encryption (PE) supporting attribute-hiding property, and obtain a revocable-storage PE (RS-PE) scheme that is selectively-secure. We further demonstrate that SUE is

[☆] A preliminary version of this article [25] appeared in ASIACRYPT 2013.

[*] Corresponding author.

E-mail addresses: kwangsu@sejong.ac.kr (K. Lee), choi@usna.edu (S.G. Choi), donghlee@korea.ac.kr (D.H. Lee), jhpark@smu.ac.kr (J.H. Park), moti@cs.columbia.edu (M. Yung).

http://dx.doi.org/10.1016/j.tcs.2016.12.027

of independent interest, by showing it can be used for timed-release encryption (and its applications), and for augmenting key-insulated encryption with forward-secure storage.

1. Introduction

Cloud data storage has many advantages: A virtually unlimited amount of space can be flexibly allocated with very low costs, and storage management, including back-up and recovery, has never been easier. More importantly, it provides great accessibility: users in any geographic location can access their data through the Internet. However, when an organization is to store *privacy-sensitive data*, existing cloud services do not seem to provide a good security guarantee yet (since the area is in its infancy). In particular, access control is one of the greatest concerns, that is, the sensitive data items have to be protected from any illegal access, whether it comes from outsiders or even from insiders without proper access rights.

One possible approach for this problem is to use attribute-based encryption (ABE) that provides cryptographically enhanced access control functionality in encrypted data [43,19,28]. In ABE, each user in the system is issued a private key from an authority that reflects their attributes (or credentials), and each ciphertext specifies access to itself as a boolean formula over a set of attributes. A user will be able to decrypt a ciphertext if the attributes associated with their private key satisfy the boolean formula associated with the ciphertext. To deal with the change of user's credentials that takes place over time, revocable ABE (R-ABE) [3] has been suggested, in which a user's private key can be revoked. In R-ABE, a key generation authority uses broadcast encryption to allow legitimate users to update their keys. Therefore, a revoked user cannot learn any partial information about the messages encrypted when the ciphertext is created after the time of revocation (or after the user's credential has expired).

As pointed out by Sahai, Seyalioglu, and Waters [42], R-ABE alone does not suffice in managing dynamic credentials for cloud storage. In fact, R-ABE cannot prevent *a revoked user from accessing ciphertexts that were created before the revocation*, since the old private key of the revoked user is enough to decrypt these ciphertexts. To overcome this, they introduced a novel revocable-storage ABE (RS-ABE) which solves this issue by supporting not only the revocation functionality but also the ciphertext update functionality such that a ciphertext at any arbitrary time T can be updated to a new ciphertext at time $T+1$ by any party *just using the public key* (in particular, by the cloud servers).

Key-revocation and key evolution are general sub-area in cryptosystems design, and ciphertext-update is a new concern which may be useful elsewhere. So, in this paper, we ask natural questions:

> Can we achieve key-revocation and ciphertext-update in other encryption schemes? Can we use ciphertext-update as an underlying primitive by itself?

We note that, in contrast to our questions, the methodology that Sahai et al. [42] used to achieve ciphertext-update is customized to the context of ABE. In particular, they first added ciphertext-delegation to ABE, and then, they *represented time as a set of attributes*, and by doing so they reduced ciphertext-update to ciphertext-delegation.

1.1. Our results

We address the questions by taking a modular approach, that is, by actually constructing a cryptographic component realizing each of the two functionalities: key revocation and ciphertext update. In particular, our design approach is as follows:

- The overall system has three components: a primary encryption scheme (i.e., ABE or some other encryption scheme), a key-revocation mechanism, and a time-evolution mechanism.
- We combine the components by putting the key-revocation mechanism in the center and connecting it with the other two. This is because the revoked users need to be taken into account both in the decryption of the primary scheme and in the time-evolution of ciphertexts.

There are a few potential benefits to this approach. First, we may be able to achieve key-revocation and time-evolution mechanisms, *independently of the primary encryption scheme*. Secondly, each mechanism may be of independent interest and be used in other interesting scenarios. Thirdly, looking at each mechanism alone may open the door to various optimizations and flexibilities of implementations.

Time-evolution mechanism: self-updatable encryption. We first formulate a new cryptographic primitive called *self-updatable encryption (SUE)*, realizing a time-evolution mechanism. In SUE, a ciphertext and a private key are associated with time T_c and T_k respectively. A user who has a private key with time T_k can decrypt the ciphertext with time T_c if $T_c \leq T_k$. Additionally, *anyone can update the ciphertext* with time T_c to a new ciphertext with new time T'_c such that $T_c < T'_c$. We construct an SUE scheme in composite order bilinear groups. In our SUE scheme, a ciphertext consists of $O(\log T_{max})$