When ABE Meets RSS

Yu Chen, Hyun Sung Kim, Jianbin Hu, Zhong Chen

When ABE Meets RSS

Yu Chen[1]*, Hyun Sung Kim[2], Jianbin Hu[1], and Zhong Chen[1]

[1] School of Electronics Engineering and Computer Science, Peking University, China
Key Laboratory of High Confidence Software Technologies, Ministry of Education
{chenyu, hujb, chen}@infosec.pku.edu.cn
[2] Department of Computer Engineering, Kyungil University, Kyungbuk, Korea
kim@kiu.ac.kr

Abstract. RSS (Really Simple Syndication) has been introduced as a response to the need of efficient information distribution and subscription. However, the current RSS standard does not provide privacy preservation and confidentiality protection. Independently, the emergence of ABE (Attribute-based Encryption) provides us a brand new cryptographic primitive for access control. This paper sets out to examine an unexplored area to date - how to apply ABE to RSS.

1 Introduction

As a widely used technique for information dissemination, RSS has been around for more than a decade. Starting from 1997 until 2007, RSS was keeping on popping with new versions. However, revisions are made to add some new features and make it backward-compatible, not for providing security mechanisms.

Most researches about RSS only focus on its applications. Cold [1] suggested that RSS can be used to enhance the efficiency of student research. Glotzbach *et al.* [2] discussed how to apply RSS to online education area. The security of RSS is neglected except in [3], Gregorio used Greasemonkey (a Mozilla Firefox extension) to encrypt and decrypt RSS feeds symmetrically. However, Gregorio's approach has many limitations and only works with Mozilla Firefox browser.

RSS works well when the feeds contents are public, but fails when the contents are only intended to a particular set of subscribers. The reason is that RSS lacks of basic security mechanism. Orthogonal to the development of RSS, ABE [4] [5] [6] is a fast emerging research area in cryptography. Messages are encrypted using a set of attributes describing the intended receivers. Only the principals processing appropriate attributes can recover the ciphertext. ABE allows encryption to inextricably bind expressive and enforceable access policy to data. It has enormous potentials for providing data security in distributed environment. Recently, Baden *et al.* [7] combined ABE and traditional public key cryptography to provide the functionality of existing online social networks with additional privacy benefits. RSS system may also be an example of such beneficiary. So given the emergence of ABE as a serious contender to access control

* This author is supported by China Scholarship Council (CSC).

mechanism based on conventional cryptographic techniques, it is important and timely to examine the suitability of ABE for securing RSS.

Our Contribution. First, we investigate the overlooked security issues of RSS. Then we explain the necessities of introducing security mechanism into RSS and outline the security targets RSS should achieve. After comparing with symmetric cryptography, public key cryptography and identity-based broadcast encryption, we conclude that ABE matches RSS best. We propose ABE-RSS, which can be viewed as an security enhanced version of RSS. By integrating ABE into RSS, ABE-RSS provides an excellent privacy preservation and confidentiality protection for RSS.

2 RSS and its Security Issues

2.1 RSS Overview

RSS feed is an XML-based document which allows online information sources to be published once and viewed by many programs. An RSS feed includes full or summarized text, plus metadata such as publishing dates and authorship.

RSS reader is a program that helps users to view and manage their feeds. The RSS reader checks the user's subscribed feeds regularly for new items, fetches the updates it finds, and interprets the RSS feeds into human readable format. RSS reader can be browser-based (Google Reader) or desktop-based (FeedDemon).

RSS model consists of four entities: publisher, RSS server, RSS reader and subscriber. Generally speaking, publisher posts news on RSS server. RSS server automatically generate the RSS feeds of the news subsequently. Subscriber registers his interesting RSS feeds in RSS reader, RSS reader gathers the RSS feeds and shows them to subscriber.

2.2 Security and Privacy

RSS standard only focuses on the feed format. It provides no security mechanism itself, everyone can fetch and read it. When a publisher wants to publish some information via RSS feed only intended to particular set of subscribers, RSS fails to meet this demand. We describe the following two scenarios to illustrate why security and privacy are important for RSS.

Paid Resources. From [1] [2] we learn that many education institutions are trying to enhance the online education with RSS technique. By the advantages of RSS, the online education systems work well when the education resources are free to everyone, but when the online education charge, the RSS feeds shouldn't be readable to those who haven't paid for them. There are also many other examples. Now AppleMusic and YahooMusic provide RSS feeds containing the latest popular music on their sites. However, if the music needs to be paid in the future, the music can not be published via raw RSS feeds anymore.

Personal Blog. One of the most common usage of RSS is blog subscription. Blog is a widely used service through Internet. Many people treat blog as a platform to publish personal photos and diaries. As soon as a user submits a new post, blog service provider will generate and publish the corresponding RSS feed for it. This facilitates people viewing blogs by subscribing the corresponding RSS feeds. But when a blog entry is not public to everyone, for private concern, the blog service providers simply choose not to generate the corresponding RSS feed. However, this naive approach impedes the real intended readers from viewing the blogs by RSS feeds.

The above use cases show that for the lack of privacy and confidentiality protection, the current RSS standard is not appropriate for disseminating personal or confidential information. Users have to rely on the host trusted sites to protect their information security. Although the use of trusted sites allow for a relatively straightforward solution, there are some downsides to the approach:

1. User is unable to define his own access control policy on data but has to rely upon the less flexible mechanisms provided by the site. For instance, Alice writes a diary on facebook only intended to her female friend. If facebook does not support such kind of policy, Alice will fail to realize her desirable access control over the diary.

2. Different sites have different security mechanisms, which hinders users from obtaining information conveniently. e.g. Sarah is Alice's female friend, but she cannot read Alice's diary until she has registered as a legitimate user of facebook and passed the identity authentication. This will bring back the terrible user experiences.

3. Both the host sites must be trusted and remain secure. With the increasing number of attacks and intrusions, maintaining the security of any particular host is becoming increasingly difficult.

In order to meet the privacy and security demands as well as maintaining the convenience and openness of RSS, we expect RSS admits:

1. RSS feeds are still available to everyone, but only the subscribers with appropriate rights can read it.

2. Publishers can determine the intended readers by exerting describable and expressive access control policies over RSS feeds.

3. The contents of RSS feeds are transparent to the RSS server, which means RSS server just serves as a storage service provider.

4. Subscribers can delegate their rights further to RSS readers, thus enables RSS readers to aggregate the RSS feeds in a secure and automatic manner.

3 ABE and Its Properties

3.1 ABE Overview

After Sahai and Waters [4] introduced the concept of ABE, Goyal *et al.* [6] proposed key-policy ABE in which ciphertext is associated with a set of attributes

and a user's private key can be associated with any monotonic tree-access structure over attributes. Subsequently, Bethencourt *et al.* [5] proposed ciphertext-policy ABE in which user's private keys are specified by a set of attributes and a principal encrypting data can specify a policy over these attributes indicating which users are able to decrypt.

3.2 Properties of ABE

We now identify the properties of ABE that distinguish it from traditional cryptographic techniques as follows.

1. Provide a natural mapping for target broadcast encryption.
2. Enable user to define any semantic access control policy over data.
3. Public keys are computable from self-describing attributes.
4. Allow user to further delegate his rights by issuing delegation private key.

It is clear that ABE happens to have the exact attractive properties which naturally meet the security demands of RSS.

4 Secure RSS with ABE

In this section, we propose ABE-RSS, in which ABE is used to secure RSS. Before presenting the concrete scheme, we first state that why ABE matches RSS best.

1. Why not symmetric cryptography?
   - The complexity of key management grows rapidly when the number of the subscribers increases.
   - When the number of the intended receivers is n, publisher has to encrypt one RSS feed n times.
   - It is impossible to exert any semantic access control policy over RSS feed.
2. Why not traditional public-key cryptography?
   - It requires the underlying PKI (Public Key Infrastructure) to be available.
   - Same as symmetric approach, publisher has to separately encrypt one RSS feed for every intended subscriber. Besides, public keys have to be authenticated before use.
3. Why not identity-based broadcast encryption [8] [9]?
   - The efficiency of identity-based broadcast encryption is dependent on the size of the authorized user set and the total number of users in the system.
   - Publisher must know the identities of all the intended subscribers each time a new RSS feed needs to be encrypted, which is inefficient and less flexible.

While using ABE, publisher only needs to describe the intended subscribers in terms of descriptive attributes and encrypt the RSS feed once. Additionally, ABE allows further delegation of private keys, which is useful in real use. We conclude that ABE is more efficient, flexible and suitable than other cryptographic techniques for RSS.

The choice of concrete ABE algorithm. There are two types of ABE algorithm, key-policy ABE [6] and ciphertext-policy ABE [5]. In key-policy ABE, user exerts no control over who can access to the ciphertext. If he wants to change the access control policy over the ciphertext in the future, he has to re-issue the private keys. The drawback is that private key generation and distribution are always time consuming and complex. In ciphertext-policy ABE, user's private key is associated with a set of attributes, ciphertext is associated with some access structure. When user wants to modify access control policy over the ciphertext, he just needs to re-encrypt the data using new access control policy without the need of re-issuing private keys. So we choose ciphertext-policy ABE as the underlying cryptographic primitive in order to ease the private key management and enable publisher to exert the access control policy over the RSS feeds. The following part of this section presents ABE-RSS, in which ciphertext-policy ABE is used to secure for RSS. We remark that ABE can also be used to enhance the security for Atom [10] in the same way, which is another syndication standard.

ABE-RSS. Figure 1 depicts the ABE-RSS model. In ABE-RSS, publisher enforces access control policy on RSS feed by encrypting it with ABE. We refer to the encrypted RSS feed as ABE-RSS feed, which has the same open nature as RSS feed, i.e. available to everyone by making a HTTP request. Note that in ABE-RSS model, RSS server only serves as a storage service provider. The contents of the feeds are totally transparent to the RSS server. We give a list of related notations in Table 1, then describe the main operations in ABE-RSS.

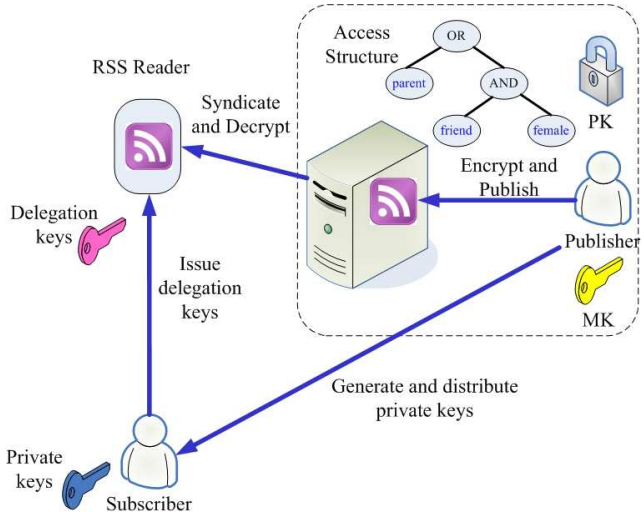

Fig. 1. ABE-RSS model

Term	Definition
PK	System public parameters
SK	Master secret
ASK	ABE user's private key
DSK	Delegation private key
$u.key$	Key created by u
$\mathbb{A}$	Access structure
ABESetup	Generate the public parameters PK and a master secret SK
ABEKeyGen(SK, R)	Generate private key ASK associated with attributes set R
ABEEncrypt$(PK, \mathbb{A}, M)$	Encrypt message M with public parameters PK and access structure $\mathbb{A}$
ABEDecrypt(PK, ASK, C)	Decrypt ciphertext C with public parameters PK and private key SK
ABEDelegate$(ASK, \widetilde{R})$	Further generate a delegation private key DSK of $\widetilde{R}$ taking ASK as the master secret, where $\widetilde{R} \subseteq R$
Publisher	The entity who publish new items in RSS format
Subscriber	The entity who subscribe RSS feeds
RSS Reader	The software syndicate and interpret RSS feed for subscribers
Key Authority	The trusted authority who set up the ABE system and act as Private Key Generator

Table 1. Notation used in this paper.

DefineAttributes. Publisher acts as a key authority, invokes algorithm ABESetup and defines a set of attributes as $\{P_1, \ldots, P_n\}$. The resulting PK is public to everyone, while SK is only known to the publisher.

AssignAttributes. In order to entitle a subscriber, the publisher first assigns a set of attributes R to describe the subscriber, then runs ABEKeyGen to generate a private key of R and sends it to the subscriber.

Encrypt. Publisher run ABEEncrypt taking PK and $\mathbb{A}$ as inputs, encrypt <item> element into <EncryptedData> element. The encrypted data implicitly contains the access structure $\mathbb{A}$. Note that RSS feed is XML-based, we can perform encryption and decryption according to XML-Encryption specifications [11].

Delegate. ABEDelegate provides subscribers a refined manner to protect their private keys and achieve a faster decryption. Each subscriber acts as a local key authority and issues a delegation key DSK to his RSS reader using his private key ASK serving as the master key. Then the RSS reader can use DSK to decrypt ciphertext but unable to deduce subscriber's ASK from DSK. Note that $\widetilde{R}$ is a subset of R, so DSK is always shorter than ASK, which means that RSS reader decrypting the ABE-RSS feeds using DSK is faster than the subscriber himself decrypting the ABE-RSS feeds using ASK.

Decrypt. On fetching an ABE-RSS feed, RSS reader runs the ABEDecrypt which takes the public parameter PK and the delegation private key DSK of $\widetilde{R}$ as

inputs to decrypt the <EncryptedData> element. RSS reader will be able to recover the <item> element if and only if $\widetilde{R}$ satisfies the access structure $\mathbb{A}$.

Hybrid Encryption. Compared to symmetric encryption, ABE is less efficient. As the size of RSS feed increases, it is wise to adopting hybrid encryption. Concretely, first use a symmetric key to encrypt the <item> element, then use ABE to encrypt the symmetric key. The encrypted element and the encrypted symmetric key are include in the <EncryptedData> as child elements.

5 Performance

The performance of ABE-RSS is evaluated on a desktop computer running GNU/Linux with Intel(R) Pentium(R) 4 CPU 3.00GHz and 1GB RAM. Hybrid encryption technique is adopted: using AES to encrypt the RSS feeds and using ciphertext-based ABE [5] to encapsulate the symmetric key. We use cpabe [12] library and their dependencies (pbc, gmp, glib, openssl) to implement ciphertext-policy ABE and AES.

ABESetup takes approximately 0.14 seconds. ABEKeyGen times average 0.16 seconds with one attribute and 0.06 seconds for each additional attributes. We assume that five attributes is enough to describe most access-control policy. Table 2 shows the times of encrypt/decrypt a 256 bits AES key when the number of attributes is from one to five. The value is the average time of 50 different experiments. The speed of AES-256 is around 0.03 millisecond for 1KB data. According to the results of Internet measurement, more than 80% of the RSS feeds are relatively small at less than 10KB. So the encryption and decryption times of AES-256 are negligible. Besides, thanks to the hybrid encryption technique, the ABE-RSS feed size is almost the same as the original size. The experiment results show that ABE-RSS scheme is feasible and efficient in practice.

Number of attributes	1	2	3	4	5
ABEEncrypt	0.17s	0.21s	0.28s	0.34s	0.43s
ABEDecrypt	0.09s	0.12s	0.15s	0.18s	0.20s

Table 2. Desktop Performance

6 Further Discussion

ABE-RSS focuses on the confidential and privacy issues of RSS, while neglects the validation issue. Next, we further discuss two unexplored problems.

Fake RSS feeds. Due to the open nature of RSS, attackers can easily generate fake RSS feeds. Even in ABE-RSS, attackers can still generate fake ABE-RSS feeds because the public key is available to everyone. So it is necessary for feeds subscriber to validate that RSS feeds really come from the right RSS publishers.

Malicious RSS feeds. Attackers may inject malicious scripts into RSS feeds, such as XSS (Cross-site script) [13]. RSS Reader should filter out the malicious scripts before interpretive execution.

A approach to eliminate the fake or malicious RSS feeds is applying digital signature to the RSS. Publisher signs the RSS feeds with his private key before publish them. Equivalently, on fetching new RSS feeds, the RSS readers first verify the contained signatures using the publisher's public key. If the verification passes, parse the feed, else reject the feed. In this way, RSS subscribers can make sure that the RSS feeds come from the trusted and real information source.

7 Conclusions

RSS brings unexperienced convenience and efficiency for information distribution, but it does not have any security mechanism. We propose ABE-RSS, an security enhanced version of RSS, in which ABE is used to secure RSS feeds. Experimental results demonstrate that ABE-RSS is efficient and feasible.

The study about ABE and RSS technique are still on the way, further research will unearth other potential advantages of attribute-based techniques and identify the associated practical and implementation issues.

References

1. Cold, S.J.: Using really simple syndication (rss) to enhance student research. SIGITE Newsl. **3** (2006) 6–9
2. Glotzbach, R.J., Mohler, J.L., Radwan, J.E.: Rss as a course information delivery method. ACM SIGGRAPH 2007 educators program (2007) 16–21
3. Gregorio, J.: Secure rss syndication. `http://www.xml.com/pub/a/2005/07/13/secure-rss.html` (2005)
4. Sahai, A., Waters, B.: Fuzzy identity based encryption. EUROCRYPT (2004) 457–473
5. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. IEEE Security and Privacy (2007) 321–334
6. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. ACM CCS'06 (2006) 89–98
7. Baden, R., Bender, A., Spring, N., Bhattacharjee, B., Starin, D.: Persona: an online social network with user-defined privacy. ACM SIGCOMM 2009 conference on Data communication (2009) 135–146
8. Halevy, D., Shamir, A.: The lsd broadcast encryption scheme. In: Advances in Cryptology - Crypto. (2002) 47–60
9. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. **3621** (2005) 258–275
10. IETF: The atom publishing protocol. `http://www.atomenabled.org/developers/protocol/atom-protocol-spec.php`.
11. Imamura, T., Dillaway, B., Simon, E.: XML Encryption Syntax and Processing. (December 2002) `http://www.w3.org/TR/xmlenc-core/`.
12. Advanced crypto software collection: `http://acsc.csl.sri.com/cpabe`
13. Grossman, J.: The origins of cross-site scripting (xss). (2008)