A Securely Transferable Ebooks using Public-Key Infrastructure

Myungsun Kim, Jongseong Kim, Jungyeon Lee, and Kwangjo Kim

International Research center for Information Security(IRIS)

Information and Communications Univ.(ICU), Korea

Abstract

This paper shows how Ebooks contents can be securely transferred to consumers in wireless environment using public key infrastructure (PKI). In addition, we show the proposed scheme to be secure. The final goal is to show that our scheme satisfies all secure requirements of digital contents in any environments.

I. Introduction

It is well known that the nature of digital world has given birth to a variety of issues. In spite of its dominant merits in terms of space, time, and efficiency from the light of media, the digital contents have enormous drawbacks such as vulnerability to various attacks e.g., illegal copy, modification, or deletion. There have been a number of schemes to cope with such disadvantages.

Most of prevalent protocols to protect digital contents are based on a secret-key based cryptosystems (SKCs). A typical example is digital rights management (DRM). It provides various functionalities as following: protection of digital contents, transaction non-repudiation, secure Ebook distribution, content authenticity, and market participant identification. DRM distributes a public identity paired with its corresponding secret key to each user. When transactions occur, users always use their own identity and secret key [1].

However, DRM has several critical weak points. The first is the key and identity management problem. The more the number of users, the more the number of identities and keys. Furthermore, random keys for decrypting Ebooks need to be distributed whenever users download Ebooks. The second is the problem of key distribution. After a user downloads Ebooks and his random key is exchanged with the receipt, he can transfer both his Ebooks and secret keys to anyone without restriction. The third is the mutual authentication problem. How can they trust each other? The last is the implementation complexity and the scalability problem. It is very difficult for DRM to manage and administrate a huge number of users by itself, especially in the authentication.

As the very alternative to resolve such problems, we take advantage of the public-key cryptosystem (PKC) under the assumption that the public key infrastructure (PKI) has already been established. In this paper, we show how PKC resolves the key management problem, and PKI addresses the complexity and scalability problem. In addition, the proposed scheme makes use of SKC to supplement weak points of PKC.

II. The Proposed Scheme

Firstly we have to define several ambiguous terminologies [2, 3].

Definition 3.1 (secure) *It is said to be secure if there exists a protocol meeting the following requirements:*

1. Confidentiality. keeping information secret

from all but those who are authorized to see it.

2. *Integrity. ensuring information has not been altered by unauthorized or unknown means.*
3. *Identity authentication. corroboration of the identity of an entity.*
4. *Non-repudiation. preventing the denial of previous commitments or actions.*

Definition 3.2 (securely transferable) *Information can be securely transferred through a protocol satisfying the four requirements.*

In addition, we need to define the notion of the discrete logarithm problem (DLP) that comes from the number theoretic problem. DLP is a basic tool for security proof of our scheme.

Definition 3.3 (discrete logarithm problem) *The discrete logarithm problem (DLP) is the following: given a prime p, a generator α of Z_p^*, and an element $\beta \in Z_p^*$, find the integer x, $0 \le x \le p-2$ such that $\alpha^x \equiv \beta \bmod p$*

1. The Proposed Protocol

Definition 3.4 *An Ebooks system $(A, P_{AB}, M_{CA}, X_{CD})$ consists of*

1. *An entity A with his public key e_A private key d_A and certificate $cert_A$.*
2. *A secure protocol P_{AB} between two entities A and B which is dependent on both a secret key K_A and a key pair (e_A, d_A) with an encryption mechanism $E()$ and decryption mechanism $D()$.*
3. *A certificate authority M_{CA} that issues a certificate $cert_A$.*
4. *A contents distributor X_{CD} that provides cipher-contents $c = E(e_A, m)$ of plain-contents m for payment.*

In particular, assume that certificates $cert_A$ of A and $cert_{X_{CD}}$ of X_{CD} have been securely given by a CA before a transaction occurs. In addition, for our scheme we require the following assumption

Assumption 3.1 There is an Ebooks system with the following assumptions

- Each user A never does reveal his private key and secret key.
- Payment can be delegated to a secure agent.
- Any decrypted contents cannot be transferred by software or hardware mechanism if a dedicated viewer is used.
- There is a single certificate authority.

As alluded above, since PKC has the computational complexity hurdle, an Ebooks system constructed only with PKC is not practical. Hence our scheme leverages a hybrid scheme incorporating SKP and a public-key based protocol.

The proposed protocol consists of three phases: contents download, payment, and contents recovery. In fact the contents download phase consists of two sub-phases: first is the mutual identity authentication phase, and second is the contents download phase. Generally, a certificate $cert_A$ contains a issuer name M_{GA} his public key $e_{M_{CA}}$, user's public key e_A digital signature algorithm S_A, and so forth (e.g., the modulus p and a generator α).

To avoid confusion, we denote $E_{K_A}(\cdot)$ as an encryption mechanism with As secret key K_A, $E(e_A, \cdot)$ as an encryption mechanism with As public key.

A mutual authentication between a user A and a contents distributor X_{CD} counts on each of certificate $cert_A$ and $cert_{X_{CD}}$, which can be summarized as the Table 1.

Table 1. The Contents Download Phase

Phase contents download	
1	$A \rightarrow X_{CD}$: generates a random number k, computes $r_a \equiv e_A^k \bmod p$, and sends $\langle cert_A, r_A, S_A(r_A), X_{CD} \rangle$ to X_{CD}.
2	$X_{CD} \rightarrow M_{CA}$: requests an identity verification of $cert_A$.
3	$X_{CD} \leftarrow M_{CA}$: returns the result of verification.
4	$A \leftarrow X_{CD}$: if valid, generates a random number ε, gets $r_{CD} \equiv e_{CD}^{\varepsilon} \bmod p$, using a hash function $h()$, computes $K_A = h(e_A \oplus r_A \oplus r_{CD})$. X_{CD} generates another nounce $r'_{CD} \equiv e_{CD}^{x} \bmod p$, sends $\langle cert_{X_{CD}}, r'_{CD}, S_{CD}(r'_{CD}), K_A, E_{K_A}(EBOC$

If a user A wants to read an Ebooks contents provided by X_{CD}, at first he has to request an authentication of X_{CD} with his certificate $cert_A$, nonce r_A computed by $r_A \equiv e_A^k \bmod p$ using a random number k, and $S_A(r_A)$. $S_A(r_A)$ can be used in the non-repudiation service

Then by Assumption 3.1 a certificate authority M_{CA} can return the result of verification to X_{CD} immediately.

According to the result, X_{CD} decides to accept A or not. If once accept, X_{CD} has to build a shared secret key, since it takes too much time for an Ebooks contents to be encrypted with the public key.

Thus X_{CD} generates a random number ε and obtains $r_{CD} \equiv e_{CD}^{\varepsilon} \bmod p$. X_{CD} computes a shared secret key $K_A = h(e_A \oplus r_A \oplus r_{CD})$ using a public hash function $h()$.

For non-repudiation requirement, X_{CD} generates another nonce $r'_{CD} \equiv e_{CD}^{x} \bmod p$ with a random number x, and sends

$\langle cert_{X_{CD}}, r'_{CD}, S_{CD}(r'_{CD}), K_A, E_{K_A}(EBOOK), \rangle$ to A. However, the user A cannot decrypt $E_{K_A}(EBOOK)$ with the secret key K_A because he does not know X_{CD}'s nonce r_{CD}.

Even though we won't describe the payment phase because it is trivial, it is easy to expect that only if A submits the receipt to X_{CD}, he can receive the nonce r_{CD} which is a fraction of the secret key K_A.

When a user A wants to decipher an encrypted Ebooks contents, he must carry the next phase, where a user A has already gotten the receipt Y for his payment.

Table 2. The Contents Recovery Phase

Phase contents recovery	
1	$A \rightarrow X_{CD}$: generates a random number k', computes $r'_A \equiv e_A^{k'} \bmod p$ and gives $\langle cert_A, r'_A, S_A(r'_A), X_{CD}, Y \rangle$.
2	X_{CD} checks the receipt Y. If valid, computes $t = E(e_A, r'_A \oplus r_{CD})$.
3	$A \leftarrow X_{CD}$: sends $\langle t, S_{CD}(t) \rangle$.
4	A decrypts $D(d_A, t)$ with his private key d_A and computes $r_{CD} = r'_A \oplus r'_A \oplus r_{CD}$ and $K_A = h(e_A \oplus r_A \oplus r_{CD})$ decrypts the contents $D_{K_A}(E_{K_A}(EBOOK))$ using the secret key K_A

The contents recovery step shown Table 2, is intuitively clear. The user A exchanges his receipt Y with X_{CD} a fraction of key K_A, r_{CD}. For security, r_{CD} was transferred in the form of 3 of Table 2, and A can recover r_{CD} deciphering $r_{CD} = r'_A \oplus D(d_A, t)$

Finally, A can obtain a plain-contents

Ebooks decrypting $D_{K_A}(E_{K_A}(EBOOK))$ with $K_A = h(e_A \oplus r_A \oplus r_{CD})$.

2. The Security Proof

Lemma 3.1 *The proposed protocol satisfies the confidentiality requirement.*

Suppose that a secret key $K_A = h(e_A \oplus r_A \oplus r_{CD})$ is compromised, an unauthorized user $\hat{A}$ knows both e_A and $h()$. On inputs e_A and $h()$, computing K_A is his knowledge of r_A and r_{CD}, which is equivalent to solving DLP. ■

Lemma 3.2 *The proposed protocol satisfies the integrity requirement.*

Data integrity is simply to guarantee it impossible that information has been altered by unauthorized or unknown means. Assume that an adversary $\hat{A}$ can modify $E_{K_A}(EBOOK)$. Then $\hat{A}$ gets the values computed by each party A and X_{CD} as the components of the secret key. From r_A and r_{CD}, $\hat{A}$ can get the secret key K_A. However, it is contradictory to the intractability of DLP. ■

Lemma 3.3 The proposed protocol satisfies the identity authentication requirement.

Assume that an adversary $\hat{A}$ impersonates A. Then $\hat{A}$ knows e_A, r_A and $h()$. Given these, gaining the private key $d_A = \log_a e_A$ and $k = \log_a r_A$ is equivalent to solving DLP in prime-order group. ■

Lemma 3.4 *The proposed protocol satisfies the non-repudiation requirement.*

This requirement says that it is not possible to deny previous commitments and actions. Suppose that a user A denies receiving $D_{K_A}(E_{K_A}(EBOOK))$. But when A carried the contents recovery phase, he sent $\langle cert_A, r'_A, S_A(r'_A), X_{CD}, Y \rangle$ to X_{CD}. $S_A(r'_A)$ is a undeniable evidence of the commitment of the protocol. The reason is that $S_A(r'_A)$ can be generated only by A with his private key d_A. As for the X_{CD}, we can similarly prove that the non-repudiation is supported. ■

Corollary 3.1 *The proposed protocol does securely transfer a cipher-contents Ebooks within the Ebooks system* $(A, P_{AB}, M_{CA}, X_{CD})$

It is evident that the Ebooks system $(A, P_{AB}, M_{CA}, X_{CD})$ with the proposed protocol is secure on the basis of lemmas **3.1**, **3.2**, **3.3**, **3.4**. ■

III. Conclusion

In this paper, we introduce the new protocol for the secure transferability in Ebooks systems. The distinguishing features of our proposed protocol are that it uses PKI and incorporates the symmetric cryptosystem with asymmetric cryptosystem. The proposed protocol turns out to provide an Ebooks system with secure transferability. In this paper, we focus on the security particularly for Ebooks system. However, the proposed protocol can be applied in other fields.

References

[1] http://www.historyebook.org
[2] B. Schneier, APPLIED CRYPTOGRAPHY, John Wiley & Sons, 1996.
[3] A. J. Menezes, P. C.van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.