Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **1**, January **2015**

A Group Key Distribution System Based on the Generalized Aryabhata Remainder Theorem for Enterprise Digital Rights Management

Yanjun Liu

Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education
School of Computer Science and Technology
Anhui University
No.111 Jiulong Rd., Hefei, 230601, China
Department of Computer Science and Information Engineering
Asia University
No.500, Lioufeng Rd., Wufeng, Taichung, 413, Taiwan
yjliu104@gmail.com

Chin-Chen Chang

Department of Information Engineering and Computer Science
Feng Chia University
No.100 Wenhwa Rd., Seatwen, Taichung, 407, Taiwan
Department of Computer Science and Information Engineering
Asia University
No.500, Lioufeng Rd., Wufeng, Taichung, 413, Taiwan
alan3c@gmail.com

Shih-Chang Chang

Department of Computer Science and Information Engineering
National Chung Cheng University
No.168, Sec.1, University Rd., Min-Hsiung Township, Chiayi, 621, Taiwan
chang.coby@gmail.com

Received May, 2013; revised July, 2014

Abstract. *Digital Rights Management (DRM) is a type of access-control technology that is used by diverse content providers to restrict the use of digital content. Enterprise Digital Rights Management (E-DRM) is an application of DRM techniques for managing the control of access to sensitive documents in an enterprise. In this paper, we propose a group key distribution system for E-DRM based on the concept of a secret-sharing mechanism and the generalized Aryabhata remainder theorem (GART). To the best of our knowledge, no previous publications related to E-DRM technologies have presented the group key distribution system that we propose. In our system, different groups of enterprise members within the enterprise obtain different digital content in parallel without collision by using their group's unique session key; all members within each group share a common session key to decrypt the same digital documents to acquire the correct content. We proved that our system achieves the goal of providing high security, which includes preventing inside and outside attacks and offering forward and backward secrecy.*

Keywords: Group key distribution, Enterprise digital rights management (E-DRM), Generalized aryabhata remainder theorem (GART), Secret sharing

1. **Introduction.** As a result of the development of digital-media technology and e-commerce, content providers around the world are delivering more and more digital content, such as electronic books, on-line digital music or video, financial reports, and product specifications. Therefore, protecting such digital content has become a major issue for the content providers. Digital rights management (DRM) can solve this problem to some extent. DRM [29, 30] is a type of access control technology that is used by diverse content providers to restrict access to digital content to authorized users. Thus, a user is prohibited from accessing the digital content if he or she has not been authorized by DRM. Because DRM provides the ability to control and manage rights to intellectual property in digital form, it is used extensively in numerous fields, especially in the entertainment industry (e.g., to protect digital copies of audio and video). Due to DRM's extensive use and its value to the owners of digital content, it has been addressed in many scientific publications [2, 9, 11, 14, 15, 26, 27, 28].

Applications of prevailing DRM techniques can be classified according to their functionalities. Enterprise digital rights management (E-DRM) is an application of DRM techniques for managing access to an enterprise's sensitive documents, such as MS Word and PDF documents, and its AutoCAD files, emails, and Intranet web pages. This differs from the functionality of controlling consumer media. In an enterprise, crucial and confidential information generally is stored in the corporate database. If unauthorized employees were to access these confidential files using their computers, mobile devices, and the Internet, they could acquire and reveal secret data, and the enterprise could incur tremendous economic losses. E-DRM can maintain the confidentiality of the company's data by ensuring that only legal employees have access to the secret files or information. A traditional E-DRM scheme consists of three parties, i.e., the content provider, the employee and the authorization center. The employee in the enterprise downloads a desired, encrypted digital document that is stored on the content provider through Internet. Meanwhile, the employee should send some useful messages to the authorization center to execute the authentication. Then, the authorization center distributes a key associated with the downloaded document. Ultimately, the employee uses the key to decrypt the document. If an enterprise adopts E-DRM, the following security robustness can be achieved. 1) Confidential digital documents cannot be revealed; 2) the right of accessing digital documents can be well managed. Since E-DRM has an extremely important role in protecting confidential data, many E-DRM mechanisms have been proposed [6, 8, 19].

Our main concern is in assisting enterprises in determining and specifying the office employees who should have the right to access the enterprise's confidential digital documents. Confidential digital documents in a database can be organized into several groups based on their relative importance, and all of the documents that belong to any specific group should be encrypted by the same key. Thus, any employee could download encrypted digital documents, but only employees who are authorized would have access to the correct key to decrypt the documents.

To determine the employees in the enterprise who should have access to the correct digital content, employees can be categorized into different groups according to their positions and responsibilities in the company. Different groups of employees would be authorized to access different groups of confidential digital documents, e.g., groups of employees in higher positions could access more sensitive documents than groups of lower-level employees. In this approach, each group of employees has the right to access the correct content of certain, specified documents, and the members of the group must have a unique key that allows them to access the correct content.

For example, in a computer company, it consists of three groups of employees, i.e., division managers, project managers, and clerks. Correspondingly, there are three groups of confidential information, ordered by their importance, with the group of division managers containing the most important information and the group of clerks containing the least important information. Each of the three groups is encrypted by a group key, i.e., K_1, K_2, and K_3, respectively. Hence, the group of division managers can obtain and use K_1 to access the information in group 1, the group of project managers can obtain and use K_2 to access the information in group 2, and the clerks can obtain and use K_3 to access the information in group 3. Of course, a person can simultaneously belong to multiple groups and has multiple group keys to view data or digital contents from different groups. For example, if a person belongs to the group of division managers and the group of project managers at the same time, he or she can obtain K_1 and K_2, and uses K_1 and K_2 to access the information in group 1 and 2, respectively.

Therefore, a secure E-DRM scheme must be utilized to ensure that only authorized group members can share a common session key with others in the same group. Simultaneously, different groups of enterprise members can use their groups' keys to obtain digital information they are authorized to see. To achieve the above requirements, we present a group key distribution system for E-DRM based on the concept of a secret-sharing mechanism [13, 23] and the generalized Aryabhata remainder theorem (GART) [7]. To the best of our knowledge, existing applications of E-DRM are all focusing on authentication problems. Unlike other E-DRM schemes, we are the first to concentrate on the issue of key distribution for E-DRM and propose a group key distribution system to apply to the E-DRM scheme. No previous publications related to E-DRM technologies have followed our approach. The main contributions of our proposed group key distribution system for E-DRM are listed below:

(1) A group key distribution system is applied to the E-DRM scheme for the first time.
(2) Within an enterprise, different groups of the members can obtain different digital information at the same time without interfering with each other by using their groups' unique session keys; all the members within a specific group share a common session key to decrypt the same digital documents to gain the correct content.

Our proposed group key distribution system for E-DRM has the following properties:

(1) Since transfers of people from one location to another within an enterprise occur frequently, our system provides ease of use by enterprise members who are joining or leaving a group.
(2) Our system has high security that it can achieve key freshness, key confidentiality and key authentication.
(3) Our system can resist inside and outside attacks, and it provides forward and backward secrecy.

The rest of this paper is organized as follows. In Section 2, we briefly review related work in the literatures. In Section 3, we propose our group key distribution system for E-DRM on the basis of a secret-sharing mechanism and the GART. Section4 and 5 present security and performance analyses of the proposed system, respectively. Our conclusions are presented in Section 6.

2. **Related work.**

In this section, we briefly introduce some essential information that was important in the development of our system. We review the Asmuth-Bloom key safeguarding scheme [1] and the generalized Aryabhata remainder theorem (GART) [7], and these provide the basis for our group key distribution system for E-DRM that is presented in Section 3. The Asmuth-Bloom key safeguarding scheme, which was based upon the Chinese remainder theorem (CRT), is essentially a threshold scheme of secret sharing. As a variation of the Aryabhata remainder theorem (ART) [21], the generalized

Aryabhata remainder theorem is more flexible and more efficient than the traditional CRT. Therefore, the Asmuth-Bloom key safeguarding scheme and the GART can be incorporated to yield a new group key distribution system. We also review some of existing group key distribution mechanisms based on secret sharing in this section.

2.1. **The Asmuth-Bloom key safeguarding scheme.** In 1979, Shamir [23] and Blakley [4] presented the well-known threshold scheme respectively to reach the goal of secret sharing. The threshold scheme is usually called a (t, n) secret sharing scheme, and it is denoted as (t, n)-SS, where the parameter t is called the threshold value. In the threshold scheme, a mutually-trusted dealer divides a master key K into n shadows, and each user receives one shadow. The key K can be recovered from any t or more shadows; however, no information of K can be derived from fewer than t shadows. The threshold scheme has been researched extensively, and there are many related publications in the literatures [3, 17, 18, 22].

The Asmuth-Bloom key safeguarding scheme, proposed by Asmuth and Bloom in 1983 [1], is, in fact, a threshold scheme of secret sharing, and it is described below.

A set of integers $\{p, q_1 < q_2 < ... < q_n\}$, called the Asmuth-Bloom sequence, is selected conforming to the regulations below [1]:

(1) GCD(q_i, q_j)=1 for $i \neq j$
(2) GCD(p, q_i)=1 for all i
(3) $\prod_{i=1}^{r} q_i > p \prod_{i=1}^{r-1} q_{n-i+1}$

We assume that an integer a is the session key that satisfies $0 \leq a \leq p - 1$. Let $s = a + bp$, where b is an arbitrary integer that is subject to the condition $0 \leq s < \prod_{i=1}^{r} q_i$. Let $s_i = s(\text{mod} q_i)$ be the shadows of s for all $1 \leq i \leq n$. It was proved in [1] that, on the basis of the CRT and the intrinsic characteristics of the Asmuth-Bloom sequence, s can be recovered from any r or more shadows, but it cannot be recovered from fewer than r shadows. Once s is recovered, the session key a associated with s can be computed immediately.

2.2. **Generalized Aryabhata remainder theorem.** The generalized Aryabhata remainder theorem (GART) [7] evolved from the Aryabhata remainder theorem (ART). In GART, an extra modulus k is offered in the computing process. Let n positive integers $q_1, q_2, ..., q_n$ form a moduli set $\{q_1, q_2, ..., q_n\}$, where GCD(q_i, q_j)=1 for $i \neq j$. A number X can be represented as n-tuple $\{x_1, x_2, ..., x_n\}$, satisfying $\text{Max}\{x_i\}_{1 \leq i \leq n} < k < \text{Min}\{q_j\}_{1 \leq j \leq n}$, where $x_i = \lfloor X/q_i \rfloor (\text{mod} k)$ for $i = 1, 2, ..., n$. According to the GART, the number X can be transformed from its n-tuple representation $\{x_1, x_2, ..., x_n\}$ to its decimal representation by the following iterative approach:

```
Input: $(k, \{x_1, x_2, ..., x_n\}, \{q_1, q_2, ..., q_n\})$
Output: $X$
1. $Q_1 = q_1, X_1 = x_1 \cdot q_1$
2. for $i = 2$ to $n$ do
3. $Q_i = Q_{i-1} \cdot q_i$
4. $X_i = k \cdot Q_{i-1} \cdot ((\lceil (x_i \cdot q_i - X_{i-1})/k \rceil \cdot (Q_{i-1})^{-1}) \text{mod} q_i) + X_{i-1}$, where $(Q_{i-1})^{-1} \text{mod} q_i$
   is the multiplicative inverse of $Q_{i-1}$ modulo $q_i$.
5. end for
6. Return $X_n$
```

By this approach, X_n is the unique solution of X in Z_{kQ_n}, where $Q_n = \prod_{i=1}^{n} q_i$. Example 2.1 illustrates the computation process of the GART:

Example 2.1. Determine the number $X = \{x_1, x_2, x_3\} = \{3, 4, 5\}$ with the moduli set $\{q_1, q_2, q_3\} = \{11, 13, 17\}$ and $k = 6$ using GART.

According to the GART, the computing approach consists of the three steps shown below:

Step 1:

$$Q_1 = q_1 = 11, X_1 = x_1 \cdot q_1 = 3 \cdot 11 = 33.$$

Step 2:

$$Q_2 = Q_1 \cdot q_2 = 11 \cdot 13 = 143,$$

$$\begin{aligned} X_2 &= k \cdot Q_1 \cdot (((\lceil (x_2 \cdot q_2 - X_1)/k \rceil \cdot (Q_1)^{-1}) \bmod q_2) + X_1 \\ &= 6 \cdot 11 \cdot (((\lceil (4 \cdot 13 - 33)/6 \rceil \cdot 11^{-1}) \bmod 13) + 33 \\ &= 6 \cdot 11 \cdot 11 + 33 = 759. \end{aligned}$$

Step 3:

$$Q_3 = Q_2 \cdot q_3 = 143 \cdot 17 = 2431,$$

$$\begin{aligned} X_3 &= k \cdot Q_2 \cdot (((\lceil (x_3 \cdot q_3 - X_2)/k \rceil \cdot (Q_2)^{-1}) \bmod q_3) + X_2 \\ &= 6 \cdot 143 \cdot (((\lceil (5 \cdot 17 - 759)/6 \rceil \cdot 143^{-1}) \bmod 17) + 759 \\ &= 6 \cdot 143 \cdot 1 + 759 = 1617. \end{aligned}$$

The validation of solution X_3 can be conducted as follows:
$x_1 = \lfloor X_3/q_1 \rfloor \bmod k = \lfloor 1617/11 \rfloor \bmod 6 = 3$,
$x_2 = \lfloor X_3/q_2 \rfloor \bmod k = \lfloor 1617/13 \rfloor \bmod 6 = 4$,
and $x_3 = \lfloor X_3/q_3 \rfloor \bmod k = \lfloor 1617/17 \rfloor \bmod 6 = 5$.

Although CRT has been applied successfully in various applications, such as cryptography, access control, and information coding, GART surpasses CRT due to the following two features. First, while there can be as many applications of GART as there are of CRT, GART can be more flexible than CRT because GART possesses an additional modulus k. For instance, if it is necessary to alter a number X in CRT, the entire system of linear congruences $x_i = X \bmod q_i$ for $i = 1, 2, ..., n$ must be modified. Unlike CRT, given $x_i = \lfloor X/q_i \rfloor (\bmod k)$ for $i = 1, 2, ..., n$ in GART, it is only necessary to reconfigure the modulus k to avoid modification of the system of linear congruences to generate a new X. Second, GART reduces the computational complexity significantly [7] and, thus, is more efficient than CRT and its variations (e.g., generalized Chinese remainder theorem (GCRT)) [5, 16]. Because of the advantages that GART has over CRT, a novel group key distribution system for E-DRM can be obtained by combining the Asmuth-Bloom key safeguarding scheme and the GART.

2.3. **Review of group key distribution mechanisms.** In group communication, a group session key must be utilized by authorized group members to encrypt and authenticate the messages that they attempt to transmit with each other. A group key distribution mechanism [10, 12, 13, 20, 24, 25] can be established to generate and distribute such a group session key, in which a mutually-trusted server, usually called the key generation and distribution center (KGDC), is employed to choose a group session key and then transport it to every authorized group member secretly.

There are many group key distribution mechanisms based on (t, n)-SS. In 2010, Harn and Lin [13] proposed an authenticated group key distribution mechanism based on (t, n)-SS. First, the KGDC shares a secret, (x_i, y_i), with each authorized group member u_i in a secure channel, where $i = 1, 2, ..., t$. Meanwhile, each u_i must transmit a random challenge R_i to the KGDC. Then, the KGDC randomly chooses a group session key k and uses $(t+1)$ pairs, i.e., $(0, k)$ and $(x_i, y_i \oplus R_i)$, for $i = 1, 2, ..., t$, to generate an t^{th}-degree interpolating polynomial $f(x)$. The KGDC also calculates t extra pairs, P_i, for $i = 1, 2, ..., t$, on $f(x)$ and makes P_i publicly known to all authorized group members. Therefore, each group member

u_i can compute $f(x)$ by using the secret, $(x_i, y_i \oplus R_i)$, which is shared with the KGDC, and t public pairs, P_i, for $i = 1, 2, ..., t$, thereby immediately recovering the group session key k. Harn and Lin analyzed that this approach of generating and distributing the group key can achieve key confidentiality, such that the group key can only be reconstructed by authorized group members. In addition, the KGDC broadcasts a one-way hash function to all group members to ensure group key authentication.

Recently, Guo and Chang [12] pointed out that although Harn and Lin's mechanism can withstand outside and inside attacks, the strategy that a random challenge R_i must be transmitted from each authorized group member to the KGDC, would incur a lot of network traffic and increase communication cost. Moreover, Guo and Chang proposed a new authenticated group key distribution mechanism based on the concepts of Asmuth-Bloom's SS scheme and the generalized Chinese remainder theorem (GCRT) [18, 22]. They claimed that by using the GCRT, their protocol can avoid sending random challenges, thereby reducing communication cost while meeting the same security goals.

3. **Our proposed system.**

Inspired by the two group key distribution mechanism discussed in Subsection 2.3, we propose a group key distribution system for E-DRM, which is based on the GART and the Asmuth-Bloom key safeguarding scheme. Since it is the first time to apply a group key distribution mechanism to the E-DRM scheme, the difference between our proposed system and a traditional group key distribution mechanism is that our system can support the distribution of multiple group keys at the same time in an E-DRM scheme. Within an enterprise, different groups of enterprise members can obtain different digital information at the same time without interfering with each other by using their unique group session key; all of the members within each group share a common session key to decrypt the same digital information to obtain the correct content. A trusted KGDC takes charge of selecting the group session keys, provides one key for each group of enterprise members, and then broadcasts some public information for the corresponding enterprise members to recover a group key. The proposed system can achieve high security, and it is a practical and effective E-DRM mechanism.

Our proposed system consists of three phases: 1) initialization of the KGDC, 2) registration of enterprise members, and 3) the generation and distribution of group keys. Our proposed system is described in detail below.

3.1. **Initialization of the KGDC.**

We assume that there are n groups of enterprise members and that each group can be denoted as $U_i = \{u_{i,1}, u_{i,2}, ..., u_{i,t_i}\}$ for $1 \leq i \leq n$, where t_i is the number of members in U_i. The first subscript of member u refers to the group number, and the second subscript indicates the enterprise member's number in the group. The KGDC creates n Asmuth-Bloom sequences $\{p, q_{1,0} < q_{1,1} < q_{1,2} < ... < q_{1,t_1}\}$, $\{p, q_{2,0} < q_{2,1} < q_{2,2} < ... < q_{2,t_2}\}$, ..., and $\{p, q_{n,0} < q_{n,1} < q_{n,2} < ... < q_{n,t_n}\}$, i.e., one separate sequence for each group.

3.2. **Registration of Enterprise Members.**

An enterprise member should be authorized to register at the KGDC and ask for the group key distribution service. The registration phase for enterprise members consists of three steps, as shown in Figure 1.

Step 1. Each $u_{i,s}(1 \leq i \leq n, 1 \leq s \leq t_i)$ sends a registration request to the KGDC.

Step 2. The KGDC randomly chooses a unique integer $k_i(i = 1, 2, ..., n)$ subject to $k_i < \text{Min}\{q_{i,j}\}_{0 \leq j \leq t_i}$ for group U_i. Then, the KGDC selects n random sequences $\{y_{1,0}, y_{1,1}, y_{1,2}, ..., y_{1,t_1}\}$, $\{y_{2,0}, y_{2,1}, y_{2,2}, ..., y_{2,t_2}\}$, ..., and $\{y_{n,0}, y_{n,1}, y_{n,2}, ..., y_{n,t_n}\}$ that satisfy $\text{Max}\{y_{i,j}\}_{0 \leq j \leq t_i} < k_i < \text{Min}\{q_{i,j}\}_{0 \leq j \leq t_i}$. In addition, the KGDC computes $A_{i,s} = h_1(y_{i,s}, q_{i,s}, k_i)$ for $1 \leq i \leq n, 1 \leq s \leq t_i$, where h_1 is a collision-free, one-way hash function.

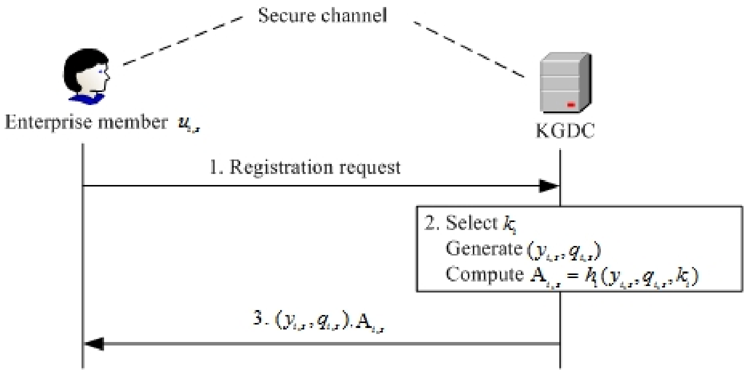
Secure channel
Enterprise member $u_{i,s}$
KGDC
1. Registration request
2. Select k_i
Generate $(y_{i,s}, q_{i,s})$
Compute $A_{i,s} = h_1(y_{i,s}, q_{i,s}, k_i)$
3. $(y_{i,s}, q_{i,s}), A_{i,s}$

FIGURE 1. Registration phase for enterprise members

Step 3. The KGDC transmits $\{(y_{i,s}, q_{i,s}), A_{i,s}\}$ to authorized member $u_{i,s}$ in a secure channel. A secret value pair, $(y_{i,s}, q_{i,s})$, is shared between the KGDC and each member $u_{i,s}$. Notice that $(y_{i,0}, q_{i,0})$ for $1 \leq i \leq n$ is not a secret value pair, and it is used as information that is broadcasted by the KGDC in the group key distribution phase. $A_{i,s} = h_1(y_{i,s}, q_{i,s}, k_i)$ is an authentication message for $u_{i,s}$ to determine which public information he or she will select to reconstruct K_i in the group key distribution process.

3.3. **Generation and distribution of group keys.** In this phase, the KGDC must generate a random group session key K_i for each group U_i , and then distribute K_i to the corresponding U_i in a secure and authenticated method. To achieve security, the KGDC broadcasts public information of all of the n group keys at one time to each authorized participant $u_{i,s}$. Then, each participant can compute only the group session key K_i shared with others in group U_i by using the public information and her or his secret value pair shared with the KGDC. The authentication character is accomplished by broadcasting separate authentication messages to all enterprise members involved. The KGDC conducts the process of group key generation and distribution in the following steps.

Step 1. The initiator transmits a requirement of group key distribution for all the n groups of registered participants U_1, U_2, ..., and U_n. Each $u_{i,s}$ for $1 \leq i \leq n, 1 \leq s \leq t_i$ is aware of the set U_i.

Step 2. The KGDC generates a number Y_i that satisfies $0 \leq Y_i < k_i \prod_{j=0}^{t_i} q_{i,j}$ for each group U_i by using the GART. Assuming $y_{i,j} = \lfloor Y_i/q_{i,j} \rfloor \text{mod} k_i$ for $0 \leq j \leq t_i$ are $t_i + 1$ shadows of Y_i, Y_i can be constructed by these $(t_i + 1)$ secret pairs $\{(y_{i,j}, q_{i,j})\}_{0 \leq j \leq t_i}$ and the parameter k_i. Especially, $y_{i,0}$ and $q_{i,0}$ are used in the first step of GART's iterative algorithm mentioned in Section 2.2 to compute Y_i. Then, the KGDC computes the group session key K_i for each U_i by the equation $K_i = Y_i - b_i p$, where b_i is an arbitrary integer subject to the condition $0 \leq K_i < p$. In addition, the KGDC encrypts the enterprise's confidential digital files that group U_i can access with K_i.

Step 3. The KGDC appends $(t_i - 1)$ additional integers $\{q_{i,t_i+1}, q_{i,t_i+2}, ..., q_{i,2t_i-1}\}$ into each initial Asmuth-Bloom sequence $\{p, q_{i,0} < q_{i,1} < q_{i,2} < ... < q_{i,t_i}\}_{1 \leq i \leq n}$ so that $\{p, q_{i,0} < q_{i,1} < q_{i,2} < ... < q_{i,t_i}\}_{1 \leq i \leq n}$ is now expanded as $\{p, q_{i,0} < q_{i,1} < ... < q_{i,t_i} < q_{i,t_i+1}, ..., q_{i,2t_i-1}\}_{1 \leq i \leq n}$, which is also an Asmuth-Bloom sequence. Then, the KGDC generates $(t_i - 1)$ extra pairs $(y_{i,j}, q_{i,j})$ for $t_i + 1 \leq j \leq 2t_i - 1$ for group U_i by computing $y_{i,j} = \lfloor Y_i/q_{i,j} \rfloor \text{mod} k_i$. It is easy to satisfy the Asmuth-Bloom secret sharing scheme characteristics that, with any $(t_i + 1)$ shadows $\{y_{i,j}\}_{0 \leq j \leq 2t_i-1}$ from $2t_i$ shadows, it is sufficient to recover the number Y_i. The KGDC also computes authentication messages

$V_i = h_2(K_i, k_i, b_i, p, (y_{i,0}, q_{i,0}), (y_{i,t_i+1}, q_{i,t_i+1}), ..., (y_{i,2t_i-1}, q_{i,2t_i-1}))$ for $1 \leq i \leq n$, where h_2 is a collision-free, one-way hash function.

Step 4. The KGDC broadcasts $p, \{k_r, b_r, (y_{r,0}, q_{r,0}), (y_{r,t_r+1}, q_{r,t_r+1}), ..., (y_{r,2t_r-1}, q_{r,2t_r-1})\}_{1 \leq r \leq n}$, and $\{V_r\}_{1 \leq r \leq n}$ to each $u_{i,s}$.

Step 5. On receiving the messages broadcasted by the KGDC, each authorized participant $u_{i,s}$ computes $h_1(y_{i,s}, q_{i,s}, k_1)$, $h_1(y_{i,s}, q_{i,s}, k_2)$, ..., $h_1(y_{i,s}, q_{i,s}, k_n)$. Since only the value of $h_1(y_{i,s}, q_{i,s}, k_i)$ can be identical to $A_{i,s}$, $u_{i,s}$ must extract $(k_i, b_i, (y_{i,0}, q_{i,0}), (y_{i,t_i+1}, q_{i,t_i+1}), ..., (y_{i,2t_i-1}, q_{i,2t_i-1}))$ from the message $\{k_r, b_r, (y_{r,0}, q_{r,0}), (y_{r,t_r+1}, q_{r,t_r+1}), ..., (y_{r,2t_r-1}, q_{r,2t_r-1})\}_{1 \leq r \leq n}$ and neglect other useless information in this message. This approach is to simplify the recovery of K_i for $u_{i,s}$. According to the GART, $u_{i,s}$ uses the secret value pair $(y_{i,s}, q_{i,s})$ shared with the KGDC, t_i additional public pairs $(y_{i,0}, q_{i,0})$, $(y_{i,t_i+1}, q_{i,t_i+1})$, ..., $(y_{i,2t_i-1}, q_{i,2t_i-1})$), and k_i to compute an integer $Y'_{i,s}$. Notice that, as was the case for the computation procedure of Y_i by the KGDC previously, it is necessary to use $(y_{i,0}, q_{i,0})$ in Step 1 of GART's iterative algorithm to compute $Y'_{i,s}$. Therefore, there is a nice relationship between $Y'_{i,s}$ and Y_i such that $Y_i = Y'_{i,s} - k_i d_{i,s}$, where $d_{i,s}$ is an integer that satisfies $0 \leq d_{i,s} < \prod_{j=0}^{t_i} q_{i,j}$. Thus, $u_{i,s}$ can uniquely recover group session key K_i of group U_i by computing $K_i = Y_i - b_i p = Y'_{i,s} - k_i d_{i,s} - b_i p$. Then, $u_{i,s}$ computes $h_2(K_i, k_i, b_i, p, (y_{i,0}, q_{i,0}), (y_{i,t_i+1}, q_{i,t_i+1}), ..., (y_{i,2t_i-1}, q_{i,2t_i-1}))$ and verifies whether the hash value is equal to V_i. If the two values are equal, $u_{i,s}$ authenticates that K_i was transmitted by the KGDC.

Step 6. All of the members in group U_i utilize K_i to decrypt the confidential digital documents that are encrypted by K_i to get the correct content.

Figure 2 illustrates the group key generation and distribution process.

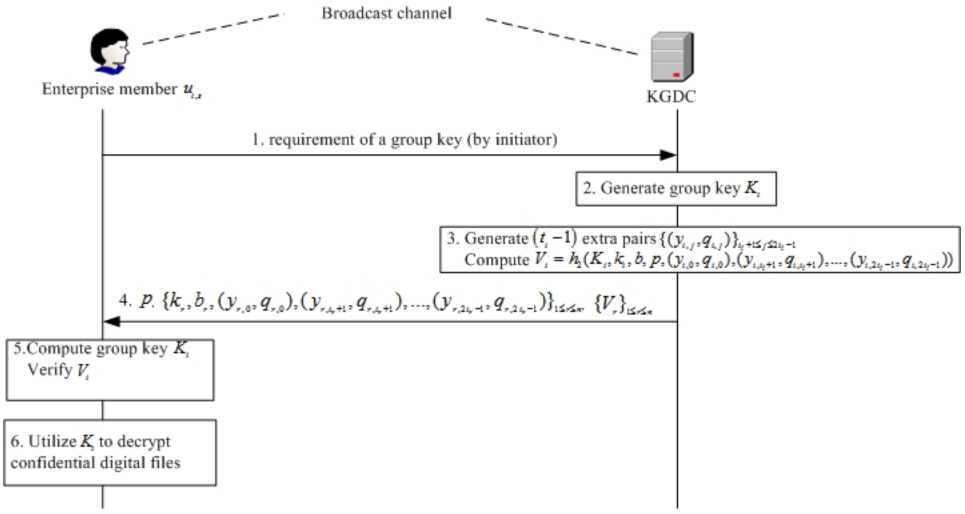
Broadcast channel
Enterprise member $u_{i,s}$
KGDC
1. requirement of a group key (by initiator)
2. Generate group key K_i
3. Generate $(t_i - 1)$ extra pairs $\{(y_{i,j}, q_{i,j})\}_{t_i+1 \leq j \leq 2t_i-1}$
Compute $V_i = h_2(K_i, k_i, b_i, p, (y_{i,0}, q_{i,0}), (y_{i,t_i+1}, q_{i,t_i+1}), ..., (y_{i,2t_i-1}, q_{i,2t_i-1}))$
4. $p, \{k_r, b_r, (y_{r,0}, q_{r,0}), (y_{r,t_r+1}, q_{r,t_r+1}), ..., (y_{r,2t_r-1}, q_{r,2t_r-1})\}_{1 \leq r \leq n}, \{V_r\}_{1 \leq r \leq n}$
5.Compute group key K_i
Verify V_i
6. Utilize K_i to decrypt confidential digital files

FIGURE 2. Group key generation and distribution process

4. Security analysis.

In this section, we prove that our proposed group key distribution system for E-DRM meets the following security requirements:

- Achieve key freshness
- Achieve key confidentiality
- Accomplish key authentication
- Prevent outside attack
- Prevent inside attack
- Provide forward secrecy

- Provide backward secrecy

Now, we analyze these security features in detail.

4.1. **Key freshness.** Key freshness means that a group key must be updated to a different, new key that is independent of the previous key whenever an authorized group member sends a new request to the KGDC or when there is any change in the group members. Thus, even if a previous group key is compromised, it is unable to derive any useful information about the future group key. The process for updating the group key under the two situations in our proposed system is described below.

4.1.1. *No change in the group members.* If an enterprise member $u_{i,s}$ in group U_i sends a new request for group key distribution in order to obtain the content of confidential digital documents in the database, the KGDC generates a new group session key for U_i. First, the KGDC selects another unique random integer $k_i^{'}$ that satisfies $\mathrm{Max}\{y_{i,j}\}_{0\leq j\leq t_i} < k_i^{'} < \mathrm{Min}\{q_{i,j}\}_{0\leq j\leq t_i}$ without updating any existing secret value pair $\{(y_{i,j}, q_{i,j})\}_{0\leq j\leq t_i}$ shared with the KGDC. The KGDC also sends the new $A_{i,s} = h_1(y_{i,s}, q_{i,s}, k_i^{'})$ to each member $u_{i,s}$ in U_i. Then, the KGDC uses the GART to generate the new number $Y_i^{'}$ by using $k_i^{'}$ and (t_i+1) secret pairs $\{(y_{i,j}, q_{i,j})\}_{0\leq j\leq t_i}$. The KGDC also chooses $b_i^{'}$ instead of b_i to compute a new group key $K_i^{'}$, subject to $0 \leq K_i^{'} < p$. Then, the KGDC broadcasts $k_i^{'}$, $b_i^{'}$, the new extra (t_i-1) pairs $(y_{i,j}, q_{i,j})$ for $t_i+1 \leq j \leq 2t_i-1$, where $y_{i,j} = \lfloor Y_i^{'}/q_{i,j}\rfloor \mathrm{mod} k_i^{'}$, and the corresponding modified authentication message is $V_i^{'}$. Finally, each member in group U_i can reconstruct $K_i^{'}$ by using her or his original secret value pair $(y_{i,s}, q_{i,s})$, the original public pair $(y_{i,0}, q_{i,0})$, the new (t_i-1) extra broadcasted pairs $(y_{i,j}, q_{i,j})_{t_i+1\leq j\leq 2t_i-1}$, and $k_i^{'}$.

4.1.2. *Change in the group members.* Since employees frequently change positions in an enterprise, it is essential for the KGDC to issue a new group key when adding or removing any group member. Our system allows these changes to be made conveniently when enterprise members join or leave a group.

If a new member $u_{i,s}$ wants to join a group U_i in the enterprise, he or she must first register at the KGDC. The KGDC adds a new $q_{i,s}$ into the initial Asmuth-Bloom sequences $\{p, q_{i,0} < q_{i,1} < ... < q_{i,t_i} < q_{i,t_i+1}, ..., q_{i,2t_i-1}\}$ and selects new $k_i^{'}$ and $y_{i,s}$ that are subject to $\mathrm{Max}\{y_{i,j}\}_{0\leq j\leq 2t_i} < k_i^{'} < \mathrm{Min}\{q_{i,j}\}_{0\leq j\leq 2t_i}$. Then, the KGDC shares $\{(y_{i,s}, q_{i,s}), A_{i,s} = h_1(y_{i,s}, q_{i,s}, k_i^{'})\}$ with $u_{i,s}$ secretly and uses $k_i^{'}$ to update $A_{i,j}$ for any other member $u_{i,j}(1 \leq j \leq t, j \neq i)$ in group U_i. During the group key generation and distribution phase, the KGDC creates a new $K_i^{'}$ through $k_i^{'}$ and (t_i+2) secret pairs $\{(y_{i,j}, q_{i,j})\}_{0\leq j\leq t_i+1}$ according to the GART. The KGDC computes the additional t_i pairs $(y_{i,j}, q_{i,j})$ by using $y_{i,j} = \lfloor Y_i/q_{i,j}\rfloor \mathrm{mod} k_i^{'}$ and then broadcasts the additional t_i pairs, $k_i^{'}$, and the corresponding $V_i^{'}$. Every member in group U_i can recover $K_i^{'}$ by using her or his secret value pair $(y_{i,s}, q_{i,s})$, the original public pair $(y_{i,0}, q_{i,0})$, the new t_i extra broadcasted pairs $(y_{i,j}, q_{i,j})$, and $k_i^{'}$.

If a member $u_{i,s}$ wants to leave group U_i, the KGDC immediately makes $u_{i,s}$ an unauthorized user; thus, the secret value pair $(y_{i,s}, q_{i,s})$ hared between the KGDC and $u_{i,s}$ becomes invalid. Afterwards, the KGDC picks a new random integer $k_i^{'}$ and sends the updated $A_{i,j}$, which uses $k_i^{'}$ to replace k_i as one parameter, to members who remain in group U_i. Finally, the KGDC simply repeats the group key generation and distribution process with members who remain in group U_i to transport a new group key $K_i^{'}$.

4.2. **Key confidentiality.** Key confidentiality is a secure feature that ensures that the group key can only be reconstructed by authorized group members. Our system achieves key confidentiality due to its use of the Asmuth-Bloom secret sharing scheme and the GART. The KGDC generates a group key K_i for group U_i via $(t_i + 1)$ secret pair $\{(y_{i,j}, q_{i,j})\}_{0\leq j\leq t_i}$ and k_i, where t_i is the number of members in U_i. Then, the KGDC computes $t_i - 1$ extra pairs $(y_{i,j}, q_{i,j})$ for $t_i + 1 \leq j \leq 2t_i - 1$ and broadcasts k_i and t_i pairs $(y_{i,0}, q_{i,0})$, $(y_{i,t_i+1}, q_{i,t_i+1})$, $(y_{i,t_i+2}, q_{i,t_i+2})$, ..., $(y_{i,2t_i-1}, q_{i,2t_i-1})$. Including the secret value pair $(y_{i,s}, q_{i,s})$, each authorized member $u_{i,s}$ in U_i has $(t_i + 1)$ pairs and can recover the group key K_i by using the $(t_i + 1)$ pairs and k_i. However, any unauthorized user is unable to gain any useful information about K_i since he or she only knows k_i and t_i pubic pairs. Thus, according to the Asmuth-Bloom secret sharing scheme, an illegal user cannot recover the group key if fewer than (t_i+1) shadows are available among $2t_i$ shadows in our system. The key confidentiality characteristic of our system is information theoretically secure because it does not depend on any unverified assumptions.

4.3. **Key authentication.** Key authentication is to ensure that the group key that each authorized group member $u_{i,s}$ in group U_i receives is transmitted by the KGDC rather than by an attacker. This property is achieved by broadcasting authentication messages $\{V_r\}_{1\leq r\leq n}$ in the process of group key distribution. V_i is a collision-free, one-way hash function of broadcasted messages and the group key K_i, which can only be created by k_i and every authorized group member's secret pair shared with the KGDC, so that any outside or inside adversary is unable to forge the group key K_i. If an outside adversary impersonates the KGDC to distribute a group key, he or she cannot generate the correct key because there is not a secret value pair shared between the outsider and the KGDC. Also, an inside adversary cannot forge the group key since he or she does not obtain the other group members' secret value pairs that are shared with the KGDC. Furthermore, any replay of V_i can be recognized, because k_i is a random integer that is one building element of K_i.

4.4. **Outside attack.** There are three types of attacks, i.e., outside attacks, inside attacks, and collusion attack. We proved that our proposed system can prevent these types of attack successfully.

An outside adversary attempts to reconstruct the group key of any group, which is kept secret between the KGDC and the authorized group members. In an outside attack, the outside adversary wants to obtain a group key, so he or she usually impersonates a group member to send a request for group key distribution. Our system can withstand this attack based on the key confidentiality that was discussed in Section 4.2. According to key confidentiality, any unauthorized user is unable to recover group key K_i of U_i since he or she only knows k_i and t_i public pairs from the broadcast channel but fails to share a secret value pair with the KGDC in a secure channel during the registration phase. Also, if a group key is compromised, the outside attacker may attempt to share the compromised key with other authorized members in the group. However, the outside attacker's plan will fail because the group key K_i is a function of a random integer k_i, thus it can be updated to ensure the goal of key freshness.

4.5. **Inside attack.** An inside adversary who has obtained the group key of a specific group legally, wants to recover other group's key and the secret value pair of any other authorized member shared with the KGDC in the enterprise. Because an inside attacker is authorized to know the group key of the group to which he or she belongs, we need to prevent the inside attacker from obtaining other groups' keys and the secret value pairs shared between any other authorized members and the KGDC. Since the KGDC sent the secret value pair to each member by a secure manner in the registration phase, the inside adversary cannot know the other members' secret value pairs that were shared with the KGDC. Furthermore, the inside adversary cannot compute other groups' keys from broadcasted messages without the secret value pairs of members of other groups.

4.6. **Forward secrecy.** Forward secrecy guarantees that it is unable to discover any previous group key by compromising the group's current key. In our proposed system, if a new member $u_{i,s}$ joins group U_i in the enterprise, the KGDC immediately updates the group key K_i to $K_i^{'}$ by selecting a new, random integer $k_i^{'}$ and a new secret value pair shared with $u_{i,s}$; then the KGDC uses $K_i^{'}$ to generate new additional t_i public pairs. The new member $u_{i,s}$ cannot obtain the previous group key K_i by combining the current $K_i^{'}$, the broadcasted $k_i^{'}$, and extra t_i public pairs.

4.7. **Backward secrecy.** Backward secrecy guarantees that any current group key cannot be determined from a previous key that has been compromised. According to our proposed system, if a member $u_{i,s}$ leaves group U_i in the enterprise, the KGDC first deletes the secret value pair that $u_{i,s}$ shared with the KGDC, updates the group key K_i to $K_i^{'}$ by selecting a new random integer $k_i^{'}$, and uses $K_i^{'}$ to generate new, additional $(t_i - 2)$ public pairs. Thus, even if $u_{i,s}$ knows the previous group key K_i, he or she cannot get the current group key $K_i^{'}$ without a secret value pair.

Table 1 shows the functionality comparison of our scheme and other related schemes proposed in [6, 8, 19].

Table 1. Functionality comparison of our scheme and other related schemes

Schemes	F1	F2	F3	F4	F5	F6	F7	F8
[6]	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
[19]	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
[8]	Yes	Yes	Yes	Yes	No	Yes	Yes	No
Ours	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

F1: mutual authentication; F2: confidentiality; F3: withstanding man in the middle attack; F4: withstanding replay attack; F5: withstanding insider attack; F6: withstanding outsider attack; F7: providing forward secrecy; F8: providing backward secrecy.

5. **Performance analysis.** In this section, we give performance analysis of our proposed group key distribution system for E-DRM in terms of computation and communication overhead.

5.1. **Computation overhead analysis.** Our system consists of three phases: 1) initialization of the KGDC, 2) registration of enterprise members, and 3) the generation and distribution of group keys. Since Phase 1 just initializes Asmuth-Bloom sequences for each group and Phase 2 just shares secret value pairs between authorized members and the KGDC, we only need to analyze computation overhead of Phase 3. In Phase 3, the group key generation in step 2 and the group key recovery in step 5 dominate the computation overhead. Consequently, let us analyze the computation overhead of the two steps in Phase 3, respectively.

In Step 2, the most time-consumed process is to determine a number Y_i for each group U_i based on (t_i+1) secret pairs $\{(y_{i,j}, q_{i,j})\}_{0\leq j\leq t_i}$ and k_i by using the GART. Since Y_i can be obtained after t_i rounds from the GART, the symbol Y_{i_c} is used to stand for the value of Y_i in the c^{th} round, where $c = 1$ to t_i. In each round, Y_{i_c} is computed as

$$Y_{i_c} = k_i \cdot Q_{i,c-1} \cdot (((\lceil (y_{i,c} \cdot q_{i,c} - Y_{i_c-1})/k_i \rceil \cdot (Q_{i,c-1})^{-1}) \text{mod} q_{i,c}) + Y_{i_c-1} \qquad (1)$$

where $Q_{i,0} = q_{i,0}$, $Y_{i_0} = y_{i,0} \cdot q_{i,0}$, and $Q_{i,c} = Q_{i,c-1} \cdot q_{i,c}$.

Now we analyze the time complexity of computing Y_i via the GART. Assume that k_i, $y_{i,c}$, and $q_{i,c}$ are all assigned b digits. The expression $k_i \cdot Q_{i,c-1} \cdot ((Q_{i,c-1})^{-1} \text{mod} q_{i,c})$ can be pre-computed. Hence, there are two multiplications, one subtraction, one division, one addition, and one modular operation in Equation (1). Because 1) the multiplication/division of two integers, each of which has b digits, requires b^2 bit operations, 2) the addition/subtraction of two integers, each of which has b digits, requires b bit operations, and 3) a modular operation with b bits requires b^2 bit operations, $(n-1) \times (2b^2 + b + b^2 + b + b^2)$ bit operations must be performed to obtain Y_i. As a result, the time complexity of computing Y_i is $O(nb^2)$. Since the KGDC can generate Y_i for $i = 1$ to n simultaneously, it is implied that the computation overhead of Step 2 is $O(nb^2)$.

Similar to Step 2, the GART is used to recover each group's key in step 5. Therefore, the computation overhead of Step 5 is also $O(nb^2)$. This indicates that the computation overhead of our proposed group key distribution system for E-DRM is $O(nb^2)$, while the computation overhead of Guo and Chang's mechanism is $O(n^2b^2)$ [12].

5.2. **Communication overhead analysis.** The communication overhead of our proposed group key distribution system for E-DRM is measured by the number of messages transmitted between each enterprise member and the KGDC from three aspects, i.e., 1) in the distribution of group keys; 2) when a member joins a group; and 3) when a member leaves a group. The transmitted messages include messages broadcasted by the KGDC, messages sent by each member and messages received by each member. Tables 2-4 compare the communication overhead of our proposed system, Harn and Lin's mechanism [13], and Guo and Chang's mechanism [12]. In Table 2, the communication overhead of our system seems as large as other two mechanisms. However, the other two mechanisms only can distribute multiple group keys one by one. On the contrary, our system can support the distribution of multiple group keys simultaneously in an E-DRM scheme. Tables 3 and 4 indicate that when a member joins or leaves a group, our system only modify messages transmitted in this group without changing existing messages of other groups in the E-DRM.

Table 2. Comparisons of communication overhead – Distribution of group keys

Mechanisms	KGDC broadcasts	messages sent by each member	messages received by each member
[13]	$n(t+1)$	1	$2nt$
[12]	$n(t+2)$	0	$n(t+2)$
Our system	$t_1 + t_2 + ... + t_n + 3n + 1$	0	$t_1 + t_2 + ... + t_n + 3n + 1$

Here, t denotes the number of each group members; n denotes the number of groups; t_i for $i = 1, 2, ..., n$ denotes the number of members in Group i.

6. **Conclusions.** In this paper, we proposed a group key distribution system for E-DRM based on the concept of the secret sharing mechanism and the generalized Aryabhata remainder theorem (GART). Our system ensures that only authorized group members can share a common session key with others in the same group and that different groups of enterprise members can use their particular group session keys to obtain the corresponding digital information simultaneously. We proved that our system achieves key freshness, key

Table 3. Comparisons of communication overhead – Member joining

Mechanisms	KGDC broadcasts	messages sent by each member	messages received by each member
[13]	$t + 2$	1	$2t + 2$
[12]	$t + 3$	0	$t + 3$
Our system	$t + 3$	0	$t + 3$

Table 4. Comparisons of communication overhead – Member leaving

Mechanisms	KGDC broadcasts	messages sent by each member	messages received by each member
[13]	t	1	$2t - 2$
[12]	$t + 1$	0	$t + 1$
Our system	$t + 1$	0	$t + 1$

confidentiality, and key authentication. We also determined that the proposed system can withstand outside and inside attacks as well as provide forward and backward secrecy.

Acknowledgment. This research was supported in part by the National Nature Science Foundation of China (grant number: 61202228) and the College Natural Science Key Project of Anhui Province of China (grant number: KJ2012A008).

REFERENCES

[1] C. Asmuth, and J. Bloom, A modular approach to key safeguarding, *IEEE Trans. on Information Theory*, vol. 29, no. 2, pp. 208-210, 1983.

[2] M. Barhoush, and J. W. Atwood, Requirements for enforcing digital rights management in multicast content distribution, *Telecommunication Systems*, vol. 45, no. 1, pp. 3-20, 2010.

[3] S. Berkovits, How to broadcast a secret, *Advances in Cryptology X EUROCRYPT 91*, LNCS547, pp. 535-541, 1991.

[4] G. R. Blakley, Safeguarding cryptographic keys, *Proc. of the AFIPS on National Computer Conference* , vol. 48, pp. 313-317, 1979.

[5] C. C. Chang, and H. C. Lee, A new generalized group-oriented cryptoscheme without trusted centers, *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, pp. 725-729, 1993.

[6] C. C. Chang, J. H. Yang, and D. W. Wang, An efficient and reliable E-DRM scheme for mobile environments, *Expert Systems with Applications*, vol. 37, no.9, pp. 6176-6181, 2010.

[7] C. C. Chang, J. S. Yeh, and J. H. Yang, Generalized Aryabhata remainder theorem, *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 4, pp. 1865-1871, 2010.

[8] C. L. Chen, A secure and traceable E-DRM system based on mobile device, *Expert Systems with Applications*, vol. 35, no. 3, pp. 878-886, 2008.

[9] C. L. Chen, An "all-in-one" mobile DRM system design, *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 3, pp. 897-911, 2010.

[10] M. Eltoweissy, M. H. Heydari, L. Morales, and I. H. Sudborough, Combinatorial optimization of group key management, *Journal of Network and Systems Management*, vol. 12, no. 1, pp. 33-50, 2004.

[11] S. V. Ghita, V. V. Patriciu, and I. Bica, A new DRM architecture based on mobilel code and white-box encryption, *Proc. of the 9th International Conference on Communications (COMM)*, pp. 303-306, 2012.

[12] C. Guo, and C. C. Chang, An authenticated group key distribution protocol based on the generalized Chinese remainder theorem *International Journal of Communication Systems*, vol. 27, no. 1, pp. 126-134, 2014.

[13] L. Harn, and C. L. Lin, Authenticated group key transfer protocol based on secret sharing, *IEEE Trans. on Computers*, vol. 59, no. 6, pp. 842-846, 2010.

[14] S. O. Hwang, How viable is digital rights management, *Computer*, vol. 42, no. 4, pp. 28-34, 2009.

[15] H. Kim, Y. Lee, and Y. Park, A robust and flexible digital rights management system for home networks, *Journal of Systems and Software*, vol. 83, no. 12, pp. 2431-2440, 2010.

[16] Y. P. Lai, and C. C. Chang, Parallel computational algorithms for generalized Chinese remainder theorem, *Computers & Electrical Engineering*, vol. 29, no. 8, pp. 801-811, 2003.

[17] C. Laih, J. Lee, and L. Harn, A new threshold scheme and its application in designing the conference key distribution cryptosystem, *Information Processing Letters*, vol. 32, pp. 95-99, 1989.

[18] C. H. Li, and J. Pieprzyk, Conference key agreement from secret sharing, *Proc. of the 4th Australasian Conference Information Security and Privacy (ACISP '99)*, pp. 64-76, 1999.

[19] C. C. Lin, S. C. Wu, P. H. Chiang, and C. C. Chen, Enterprise-oriented digital rights management mechanism: eDRM, *Proc. of International Conference on Availability, Reliability and Security*, pp. 923-928, 2009.

[20] A. Perrig, D. Song, and J. D. Tygar, Elk, a new protocol for efficient large group key distribution, *Proc. of the IEEE Symposium on Security and Privacy*, pp. 247-262, 2001.

[21] T. R. N. Rao, and C. H. Yang, Aryabhata remainder theorem: relevance to public-key crypto-algorithms, *Circuits, Systems, and Signal Processing*, vol. 25, no. 1, pp. 1-15, 2006.

[22] G. Saze, Generation of key predistribution schemes using secret sharing schemes, *Discrete Applied Mathematics*, vol. 128, pp. 239-249, 2003.

[23] A. Shamir, How to share a secret, *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[24] A. T. Sherman, and D. A. McGrew, Key establishment in large dynamic groups using one-way function trees, *IEEE Trans. on Software Engineering*, vol. 29, no. 5, pp. 444-458, 2003.

[25] M. Steiner, G. Tsudik, and M. Waidner, Diffie-Hellman key distribution extended to group communication, *Proc. of the 3th ACM Conference on Computer and Communication Security (CCS'96)*, pp. 31-37, 1996.

[26] J. Wang, and X. Fu, Digital rights management (DRM) in the mobile P2P environment, *Proc. of the 6th International Conference on Wireless Communications Networking and Mobile Computing*, pp. 1-4, 2010.

[27] C. T. Yen, H. T. Liaw, N. W. Lo, T. C. Liu, and J. Stu, Transparent digital rights management system with superdistribution, *Proc. of International Conference on Broadband, Wireless Computing, Communication and Applications*, pp. 435-440, 2010.

[28] W. Zeng, and K. Liu, Sensitivity analysis of loss of corporate efficiency and productivity associated with enterprise DRM technology, *Proc. of International Conference on Availability, Reliability and Security (ARES)*, pp. 445-453, 2012.

[29] Adobe Lifecycle Document Security, *http://www.adobe.com/products/server/Securityserver/pdfs/docsecurityserver_ ds.pdf*, 2009.

[30] Microsoft Windows Right Management Services System, *http://www.microsoft.com/downloads/en/details.aspx?FamilyID=5794538F-E572-4542-A5BD-901B2720F068*, 2006.