AN OFFLINE PEER-TO-PEER BROADCASTING SCHEME WITH ANONYMITY

Shinsaku Kiyomoto, Kazuhide Fukushima
KDDI R & D Laboratories Inc., Japan

Keith M. Martin
Information Security Group, Royal Holloway University of London, U.K.

Keywords: Peer-to-peer broadcasting, Anonymous broadcasting, Timed release encryption.

Abstract: Content broadcasting is an effective method of delivering content to a large number of users. Although IP-based broadcasting services are available on the Internet, their services require a broadcasting server with huge resources, which makes them unsuitable for personal broadcast applications. An obvious alternative is to deliver personal broadcasting services over a peer-to-peer network. Several products are now available to support live broadcasting in such networks. However, live broadcasting in a peer-to-peer network involves a heavy concentrated communication load and incurs network delays that result in users not necessarily viewing a live transmission simultaneously. In this paper we consider an alternative type of peer-to-peer broadcasting scheme, *offline broadcasting*, which provides a decentralized broadcasting service with anonymity. While offline broadcasting cannot be used for live broadcast streaming, it can be used to guarantee simultaneous viewing of predistributed content. We demonstrate that this scheme provides a practical alternative to existing techniques for broadcasting content that can be created in advance, and present security analysis of the scheme.

1 INTRODUCTION

1.1 Background

Content broadcasting services are used for simultaneously distributing information to a great number of users. A conventional broadcasting service requires a broadcasting server with huge computational resources and places a heavy burden on the network, which makes a broadcasting service costly.

We are seeing an increase in the significance of personal users as multimedia content providers. For example, *Consumer generated media* (CGM) (Blackshaw and Nazzaro, 2004) is a term that is increasingly used to describe multimedia content that is created by consumers and delivered over the Internet in relation to their own experience of products or services. We have also seen the rise of YouTube (YouTube, LLC, 2005) as a successful content distribution service for the likes of CGM. Notably, YouTube provides for *anonymous* content distribution, which may be one of the reasons for its popularity since virtual identities are a widely used in Internet communities. However, applications such as YouTube tend to adopt a centralized approach to content distribution, which does not suit all application environments.

The increasing importance of CGM raises the interesting question as to whether it is possible for personal users to advance from the open publication of multimedia content, which they can do already, to the provision of their own broadcasting services, which we term *personal broadcasting*. From experience of existing applications such as YouTube, the capability to support anonymous broadcasting would seem a desirable property of any personal broadcasting application.

1.2 Peer-to-Peer Broadcasting

Peer-to-peer networks provide efficient decentralized platforms for distributing multimedia content. Several network architectures have been proposed for peer-to-peer technology, for example *Content-Addressable Network* (CAN) (Ratnasamy et al., 2001), *Freenet* (Clarke et al., 2000), *Chord* (Stoica et al., 2001), and *Tapestry* (Zhao et al., 2004). It has

Kiyomoto S., Fukushima K. and M. Martin K. (2009).
AN OFFLINE PEER-TO-PEER BROADCASTING SCHEME WITH ANONYMITY.
In *Proceedings of the International Conference on Security and Cryptography*, pages 75-82
DOI: 10.5220/0002182100750082

also been shown that a peer-to-peer network can reduce the overall cost to content providers in comparison with a content distribution scheme based on a centralized server (Gkantsidis et al., 2006).

A number of products to support live broadcast services in peer-to-peer networks are now available. One example is *PeerCast* (PeerCast.org, 2002), which organizes user nodes into a hierarchical structure in order to enable efficient forwarding of streamed content. Several other peer-to-peer live broadcasting schemes use a similar approach (Liao et al., 2006; Luac et al., 2007). These schemes successfully use decentralization to reduce the heavy burden of providing a broadcasting service.

However, these live broadcasting schemes suffer from two problems. Firstly, at the time of a live broadcast the network experiences a concentrated communication load as the live content stream is propagated throughout the community of nodes. The network is further stressed if two or more nodes are simultaneously broadcasting. Secondly, although these schemes aim to be efficient, it is unavoidable that there are delays in delivery of the broadcast content, with some nodes receiving it before others.

1.3 Offline Broadcasting

In this paper, we propose an alternative approach to providing broadcasting services in a peer-to-peer network. We refer to it as *offline broadcasting* since the content is distributed ahead of its start time, but users cannot view the content until a designated time. While offline broadcasting cannot be used for live streaming, it does have the following properties:

1. nodes all have the capability to simultaneously access the content;
2. there is no communication congestion at the designated broadcast time;
3. broadcasters can create content anonymously.

Our offline broadcasting scheme does not require direct communication between the broadcasting server and users. By predistributing the content, the heaviest burden of the scheme is temporally and spatially distributed amongst the users.

One way to implement an offline broadcasting scheme would be to distribute encrypted content in advance and then make the decryption key available at the broadcast start time. However, if the broadcaster forwards this decryption the key at the start time then similar network delays to those for live broadcasting schemes will be experienced. Alternatively, if the decryption key is released by a central key server then the complex issue of managing content-specific keys at the server end needs to be addressed.

We propose the use of a *timed-release encryption scheme*, which only requires the periodic broadcasting of time information. In this way the offline broadcasting scheme requires no interaction between the broadcaster and users at the time of viewing. With respect to anonymity, broadcasters can choose whether or not to link their identity to the content.

1.4 Organization

The rest of the paper is organized as follows. In Section 2, we discuss timed-release encryption. In Section 3 we define the main components and in Section 4 we describe our offline broadcasting scheme. We conduct a basic security and efficiency analysis of our scheme in Section 5 and then conclude.

2 TIMED-RELEASE ENCRYPTION

Timed-release encryption (TRE) schemes are used to control the time at which confidential data is disclosed. TRE schemes are useful for applications such as electronic voting, sealed-bid auction, e-lotteries, and online games (Chalkias, 2007). There are two approaches to realizing a TRE scheme. The first uses *time-lock puzzles* (Mao, 2001) (Boneh and Naor, 2000) (Garay and Jakobsson, 2003) to require the receiver to perform non-parallelizable computations in order to recover confidential data. The second approach is based on the use of a *trusted time server*, which periodically publishes content independent "time information". Blake and Chan proposed a scalable public key TRE (Blake and Chan, 2005) scheme (which we will call *BCTRE*) that uses a timestamp server.

We base our offline broadcasting scheme on BCTRE, which we now briefly outline. Suppose that $\mathbb{G}_1$ and $\mathbb{G}_2$ are respectively additive and multiplicative cyclic groups of order q (q is a prime number). Let the bilinear pairing be a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:

1. *Bilinearity.* The following equations must be satisfied:
   $\hat{e}(aP,Q) = \hat{e}(P,Q)^a = \hat{e}(P,aQ)$,
   $\hat{e}(aP,bQ) = \hat{e}(P,Q)^{ab}$,
   $\hat{e}((a+b)P,Q) = \hat{e}(aP,Q)\hat{e}(bP,Q)$,
   $\hat{e}(P,(a+b)Q) = \hat{e}(P,aQ)\hat{e}(P,bQ)$,
   for all $a,b \in \mathbb{Z}_q^*$, which are taken from among n equally spaced points on an elliptic curve.

2. *Non-degeneracy.* $\hat{e}(P,Q) \neq 1$ is satisfied.
3. *Computability.* There exists an efficient algorithm for the computation $\hat{e}(P,Q)$.

We assume that the discrete logarithm (DL) problem over $\mathbb{G}_1$, and the computational Diffie-Hellman problem are hard. From the existence of bilinear pairing in the underlying group, the decisional Diffie-Hellman (DDH) problem can be solved. Thus, $\mathbb{G}_1$ is a *Gap Diffie-Hellman group*, which is found in supersingular elliptic curves or hyperelliptic curves over a finite field. The bilinear pairing is derived from the Weil or Tate pairing. Furthermore, the Bilinear Diffie-Hellman (BDH) problem is assumed to be hard. For more information, see (Boneh and Franklin, 2001).

The BCTRE scheme uses two cryptographic one-way hash functions: $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$. The scheme consists of the following five steps.

- **Server's Key Generation.** A time server of a trusted third party randomly picks G as a generator of $\mathbb{G}_1$ and a private key $s \in \mathbb{Z}_q^*$. The server then computes the public key sG and the public key and G are made publicly available.
- **User's Key Generation.** Each user picks a secret key $a_i \in \mathbb{Z}_q^*$ and computes the corresponding public key (a_iG, a_isG).
- **Time Information Generation.** At a time $T \in \{0,1\}^*$, the time server publishes a signature $sH_1(T)$ for the time T. Note that every user can verify its validity by checking $\hat{e}(sG, H_1(T)) = \hat{e}(G, sH_1(T)))$.
- **Encryption.** Given a message M and a receiver's public key (a_iG, a_isG), a sender randomly picks $r = \mathbb{Z}_q^*$ and computes rG and ra_isG. The sender then computes $\rho = \hat{e}(ra_isG, H_1(T))$, where T is the moment in time when the ciphertext should be decrypted. Finally, the sender computes the ciphertext $CT = (rG, \sigma = M \oplus H_2(\rho))$.
- **Decryption.** Given a ciphertext CT, a receiver's private key a_i and a time datum $sH_1(T)$ from the server, the receiver computes $\rho' = \hat{e}(rG, sH_1(T))^{a_i}$, and then the receiver recovers M by computing $\sigma \oplus H_2(\rho')$.

The BCTRE scheme is intended for content distribution to one user. In Section 3.3 we adapt it so that we can apply it to a broadcasting scheme.

3 COMPONENTS OF PROPOSED SCHEME

In this section, we discuss the security requirements and components of our offline broadcasting scheme.

3.1 Entities

Four entities are involved in our offline broadcasting scheme.

- **Broadcast Stations.** Broadcast stations broadcast their content to users according to a time schedule. The stations upload a program schedule detailing their broadcast content to a program-listing server to spread information about their offerings.
- **Users.** Users receive broadcasting services from broadcast stations with the assistance of a time server.
- **Time Server.** A time server distributes valid time information to users. The time server is assumed to be trusted.
- **Program Listing Server.** A program listing server manages programs of broadcast content. Users obtain information on future broadcast content from the program-listing server. The program-listing server is assumed to be trusted.

Note that in many of the CGM applications for this type of scheme the set of broadcast stations and the set of users will be the same.

3.2 Security Requirements

The main security requirements for an offline broadcasting scheme are summarized as follows.

- **Timed Release Capability.** Access to broadcast content must be infeasible until the designated time. While users may be able to obtain encrypted content in advance, they should not be able to view the content until the intended time, at which point they should be able to view the content at the same time.
- **No Content Plagiarism.** Content must not be plagiarized before its scheduled viewing time. This is a particularly important issue for an offline broadcasting service. The content is distributed in advance, and an attacker can potentially redistribute the encrypted content as their own original content. Thus, information about the ownership of the intellectual property must be protected until the designated time for viewing. Details of this type of attack will be discussed in Section 5.

- **No Masquerading.** Masquerading as a legitimate broadcast station must be impossible. An attacker may create content while pretending to be a broadcast station. It must be impossible for attackers to place the appearance of responsibility for illegal content on other broadcast stations.

3.3 Timed-Release Broadcast Encryption Scheme

We now modify the BCTRE scheme to allow it to be used for broadcast encryption. We refer to the modification as *Broadcasting-BCTRE (B-BCTRE)*. The notation is the same as for BCTRE (see Section 2.2). The scheme is as follows:

- **Server's Key Generation.** This step is the same as for BCTRE.
- **Broadcast Station's Key Generation.** Each broadcast station selects a secret key $b_i \in \mathbb{Z}_q^*$ and computes a public key b_iG that is made publicly available.
- **Time Information Generation.** At a time $T \in \{0,1\}^*$, the time server publishes a signature $sH_0(T)$ and $sH_1(T)$ for the time T. Every user can verify its validity by checking $\hat{e}(sG, H_0(T)) = \hat{e}(G, sH_0(T)))$ and $\hat{e}(sG, H_1(T)) = \hat{e}(G, sH_1(T)))$. H_0 is a cryptographic one-way hash function: $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$.
- **Encryption.** A broadcast station randomly picks a content key seed $KS = \{0,1\}^k$ and $r = \mathbb{Z}_q^*$, where k is a security parameter and satisfies $k \leq n$. Then the station computes rG. Next, the station computes $\rho = \hat{e}(rsG, H_0(T))\hat{e}(b_isG, H_1(T))$, where T is the target time for decryption of the ciphertext. Finally, the station computes the ciphertext $CT = (rG, \sigma = KS \oplus H_2(\rho))$.
- **Decryption.** Given the ciphertext CT, the broadcast station's public key b_iG, and a time datum $sH_0(T)$ and $sH_1(T)$ from the server, the receiver computes $\rho' = \hat{e}(rG, sH_0(T))\hat{e}(b_iG, sH_1(T))$, and then recovers KS by computing $\sigma \oplus H_2(\rho')$. Note that only a valid public key of the broadcast station can recover a valid KS, therefore, the decryption process implicitly checks the authenticity of content distributors.

3.4 Content Encryption

Multimedia content typically involves huge amounts of data and the software that displays it must decrypt the content in real time. Thus, the content itself should be encrypted using relatively efficient symmetric techniques based on either a block cipher or stream cipher. The content can be played with decrypting the encrypted content. In general, a two-layered structure is used by distribution services to encrypt multimedia content (Open Mobile Alliance Ltd, 2008). First, content is encrypted with a content key CK using a symmetric key encryption algorithm, and then CK is wrapped using another key that is unrelated to the content key.

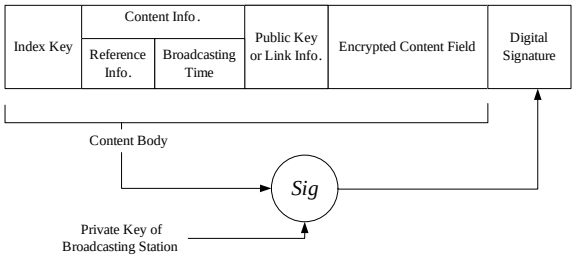

Figure 1: Data Format of Distributed Content.

We denote the encryption and decryption algorithms for the B-BCTRE scheme by $\mathcal{TE}$ and $\mathcal{TD}$ respectively. The B-BCTRE scheme is used for encrypting the content key seed KS. Symmetric key encryption and decryption are denoted by $\mathcal{SE}$ and $\mathcal{SD}$. We also define a key derivation function $\mathcal{KDF}(KS)$ that outputs the content key from the seed. The functions are described as follows:

$$\mathcal{TE} : CT \leftarrow \mathcal{TE}(KS, Pu_t, Pr_b, T),$$
$$\mathcal{TD} : KS \leftarrow \mathcal{TD}(CT, Pu_b, TI),$$
$$\mathcal{KDF} : CK \leftarrow \mathcal{KDF}(KS),$$
$$\mathcal{SE} : EMC \leftarrow \mathcal{SE}(CK, MC),$$
$$\mathcal{SD} : MC \leftarrow \mathcal{SD}(CK, EMC),$$

where MC and EMC are multimedia content and encrypted multimedia content respectively. The symbols Pu_b and Pr_b denotes a public key b_iG and private key b_i of the broadcast station. The public key of the time server sG, designated time T, and time information $sH_1(T)$ from the time server are denoted by Pu_t, T, and TI. We assume that the symmetric key encryption scheme $\mathcal{SE}$, $\mathcal{SD}$, and the key distribution function $\mathcal{KDF}$ are secure.

3.5 Data Format of Distributed Content

The data format of the distributed content is shown in Figure 1. Content data consists of the body of the content and its digital signature. The body of content consists of four data blocks as follows:

- *Index Key:* This information is an index key for searching for the content on a peer-to-peer system.
- *Content Info.:* The content information field includes reference information, such as index keys for the previous and following content and the

broadcasting time (start time for the content and its expected end time). The designated time of the content is specified in the content information field as the start time of the content. If the content is a part of a program that is divided into sections, the user obtains the next section by referring to the content information field before the start time.

- *Public Key or Link Info.:* This data is a public key certificate of the broadcasting station or its link information. The user obtains the public key from the content information field or a server that stores the public key certificate of the broadcast station, using the link information. The user first verifies the public key, using its certificate, and then uses the public key for decryption at the designated time.
- *Encrypted Content Field:* This data consists of all the encrypted data, namely the encrypted content key seed using B-BCTRE and the encrypted content using the symmetric key encryption algorithm.

In order to detect modification of any information that must be opened before the designated time, such as the index key and content information, the broadcast station may compute the digital signature $S = b_i H_1(CB)$ using the appropriate private key b_i, where CB is a bit string of the content body. This signature can be verified by using the public key of the broadcast station $b_i G$ as $\hat{e}(G,S) = \hat{e}(b_i G, H_1(CB))$. The signature must be computed using the same private key of the broadcast station as the one used for timed-release encryption of the content.

4 OFFLINE BROADCASTING SCHEME

We now present our offline broadcasting scheme. We will also explain how to realize an anonymous broadcasting service.

4.1 System Architecture

Figure 2 provides an overview of the system architecture to support our offline broadcasting scheme. We assume that the scheme is constructed on an existing peer-to-peer network architecture such as CAN, Chord, Freenet or Tapestry (see references). Such peer-to-peer platforms normally provide the following functionality:

- Discovery of Content: The peer-to-peer system provides a search function for each node. This function outputs addresses of nodes where content is stored. An index key for the desired content is used to search for the content. A scalable search mechanism is designed into existing peer-to-peer systems.
- Storage for Content: Each node provides storage for content. This function also provides a registration mechanism for content items in order to provide information to the peer-to-peer system. Nodes register the content in their storage with the peer-to-peer system.
- Content Distribution: This provides a protocol for distributing content between nodes. A node sends content that is requested by another node. Content is sometimes distributed using indirect communication between nodes, in which case content is sent to a requesting node via one or more intervening nodes. Nodes along the communication chain store the requested content automatically, if their storage has sufficient space.

In addition to the above general functionality, in our system each node has additional security functionality as follows:

- Content Management: Stored content can be verified, ensuring that the signature is valid. This functionality also checks content information and will remove any content whose designated time has expired.
- Decryption: This function receives time information and decrypts content on its scheduled time for viewing. This function is assumed to decrypt the content to protected memory in order to avoid the potential for making an illegal copy of the content.

Broadcast stations join the peer-to-peer system as nodes and can both create and encrypt content. The encryption function uses the B-BCTRE scheme described in Section 3.3. Furthermore, for the offline broadcasting service, we add two servers to the peer-to-peer system as follows. These servers are common services for all broadcast stations and users.

- Time Server: The time server generates time information periodically, and sends time data to other nodes by broadcasting/multicasting.
- Program Listing Server: The program listing server publishes a program list of broadcast services. Broadcast stations register information about their content, such as a summary, index key, and the designated viewing time of their content items.

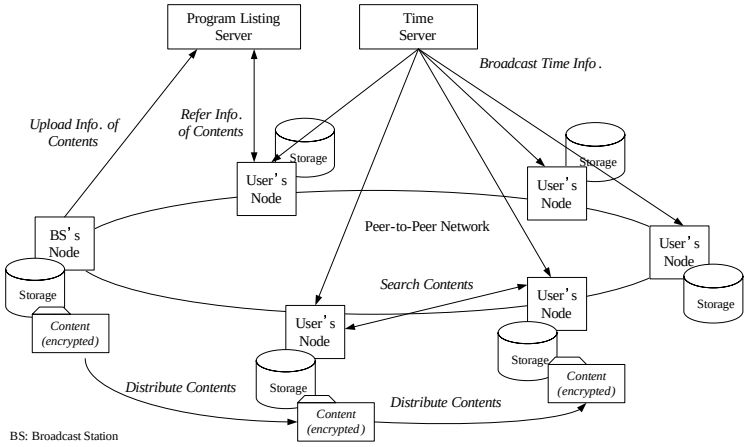

Figure 2: System Overview.

4.2 Procedure for Content Broadcasting

The procedure for content broadcasting in our offline broadcasting scheme is described as follows:

1. The broadcast station encrypts its content and registers it on the peer-to-peer network. The broadcast station uploads information about the content to the program listing server, which could include a summary of the content, an index key for the content, the designated time, and a public key certificate for the broadcast station.
2. A user checks a program list on the program listing server and searches for interesting content, then downloads that content from the nearest node that has it. The content is distributed to users by such actions.
3. When the user receives the content, the user verifies it as follows. If the content includes a digital signature from the broadcast station, the user obtains the public key of the station and verifies the signature. If the signature is invalid, the user destroys the content. Otherwise, the user stores the content in his own storage for the peer-to-peer system.
4. A time server periodically broadcasts content-independent time information. When the user receives a time datum corresponding to the designated time of the content, he can decrypt and view the content. If the content item indicates that another follows, the user searches for the next content item and downloads it while watching the current content. The user views the subsequent content item after finishing the current item, by decrypting the subsequent item with the next time datum.
5. A content item whose designated time has passed is removed from storage. In this way, content that the user has finished viewing is automatically deleted from storage.

4.3 Anonymous Broadcasting Service

Our scheme can be extended to an anonymous broadcasting service by making the broadcast station's public key anonymous. When the broadcast station wants an anonymous broadcasting service, the station generates a public-private key pair for each item of content and stores the public key in the content information field. If the service requires that only authorized broadcast stations can distribute their content to users, a certificate authority may provide a certificate of the public key that does not include the identifier of the station, but does include membership information. Users verify the public key using the membership certificate.

In a peer-to-peer network, it cannot be determined whether content is created by a node or copied from other nodes. Furthermore, users can verify the signature and decrypt the content using the public key and time information, however a user cannot determine the owner of the public key. Thus, the scheme can be used to provides an anonymous broadcasting service.

Note that a broadcast station can always prove that the station is a valid creator of the content because the broadcast station has the private key related to the public key. If the content from the station is plagiarized, the broadcast station can provide evidence confirming its role as creator by demonstrating the use of the corresponding private key to a trusted third party.

Our offline broadcasting scheme thus achieves

anonymity of broadcast stations, since it is not known who broadcast a particular content. If required, a broadcaster can repeatedly associate the same pseudonym with broadcast content in order to build up a reputation but without revealing their identity, just as in the case of YouTube. In addition, it is worth noting that users also receive a degree of anonymity since a node cannot know whether another node decrypts and views specified content, even if the node has the content.

5 ANALYSIS

5.1 Security Analysis

We now comment on the security of the proposed offline broadcasting scheme. We first examine the security of B-BCTRE, then the three main security requirements of Section 3.2, and finally consider some other security issues.

Security of B-BCTRE. Since B-BCTRE is only a slight modification of BCTRE, the security proof of B-BCTRE is very similar to that of BCTRE. A sketch of the security proof is as follows. As described in (Blake and Chan, 2005) for BCTRE, we can rewrite any $sH_0(T_j) = w_j sG$ for some unknown w_i. The problem of an adversary decrypting KS without $sH_0(T_j)$ becomes that of finding $\hat{e}(G,G)^{rsw_j}$ from sG, w_jG, rG, which is equivalent to the bilinear Diffie-Hellman problem. This problem is the same as the original BCTRE scheme. Hence, as long as the bilinear Diffie-Hellman is difficult, the adversary cannot decrypt KS before its designated time unless they collude with the time server. Secondly, we consider the problem of an adversary generating ciphertext of the broadcast station i without the private key b_i. This problem is also equivalent to the bilinear Diffie-Hellman problem of computing $\hat{e}(G,G)^{b_i sw_l}$ from sG, w_lG, and b_iG, where $sH_1(T_l) = w_l sG$. Thus, the adversary cannot alter a ciphertext of the broadcast station under the assumption that the bilinear Diffie-Hellman problem is difficult.

Timed Release Capability. No user can decrypt a content item without a time datum $sH_1(T)$, because it is impossible to obtain a private key s within the limits of feasible computational complexity.

No Content Plagiarism. A potential scenario for content plagiarism is as follows. An attacker obtains an encrypted content item created by a valid broadcast station. Next, the attacker separates the ownership information from the encrypted content and the B-BCTRE ciphertext, and then creates an altered content item using the encrypted content and the B-BCTRE ciphertext. Finally, the attacker distributes the altered content as their created content, before the designated time for the original content. If we used the basic BCTRE scheme without modification, this attack would be successful. However, our scheme requires both the public keys for the content creator and for the time datum specifying when the content is to be decrypted. The attacker cannot decrypt anything until the designated time for the content, and cannot create altered content that is decryptable using the attacker's public key. Therefore, this security requirement is met. After the designated time, we should protect the content not by using cryptographic techniques, but by using techniques that will be discussed later.

No Masquerading. To masquerade as a valid broadcast station, a private key b_i is needed to compute a valid B-BCTRE ciphertext and signature. From the assumptions of the DL and DH problems, it is infeasible to obtain b_i without unrealistic computational cost. Thus, the proposed scheme is secure against a masquerading attack.

Denial of Service. An attacker may try to obstruct the offline broadcasting service by altering content information such as reference information and start time, and then distributing altered content. However, this attack is infeasible because any modification of the information can be detected by verifying the signature on the content body. Content with an invalid signature will be removed from the peer-to-peer system. Another possible avenue for a denial of service attempt is for an attacker to destroy content items stored on their own node. This attack is also not effective because users can obtain the content items from other nodes. Also, an attacker may try to generate a huge volume of dummy content items and distribute them in order to saturate the storage capacity of other nodes. This is a general issue in peer-to-peer content distribution systems, however each node can check and remove non-valid content from its own storage, and it would be infeasible to saturate the storage of all nodes.

Content Protection on a Peer. After decryption of a content item, an attacker has full access to the content. The attacker may copy this illegally or use it as their own content. This problem is a general issue for digital rights management systems for multimedia content. A cryptographic technique cannot protect against such actions. However, commercial DRM services solve this problem using a number of techniques. If content decryption is executed in memory that the attacker cannot access then the content is securely protected. Therefore, we should prevent access to plaintext content by an attacker. Furthermore, we must ensure that content whose designated time is past is automatically removed from the peer-

to-peer system in a way that no attacker can circumvent. Hence, secure implementation of both the content management and decryption function is very important.

5.2 Efficiency Analysis

In existing peer-to-peer live broadcasting schemes, nodes receive and transfer streamed content in real time, thus experiencing time delays as the content is relayed. However, when using the B-BCTRE scheme, all users can simultaneously access content when the time server releases the relevant time datum, which is content-indpenendent. Further, all communication effort takes place before the start time. Hence, a broadband network is not mandatory and the communication load is spread over time. In addition, the scheme provides secure content distribution, unlike many of the existing peer-to-peer broadcast schemes which require content security to be bolted on at additional cost.

In comparison to live broadcast, users do have to store content ahead of time. At the designated start time, the only cost is that user nodes must decrypt the content. The decryption of the content key consists of two pairing computations, one multiplication, one hash operation, and one exclusive-or operation. The pairing computation on a 1.66 GHz Core2 perform is expected top take around 14.5 msec (Hankerson et al., 2008) and the transaction time of other operations is negligible. The bulk content decryption uses efficient standard symmetric techniques, which makes the scheme applicable for PC-based nodes.

6 CONCLUSIONS

In this paper, we have described the design of an offline broadcasting scheme for peer-to-peer networks that can provide anonymity for the broadcaster. In this scheme, personal users can broadcast multimedia content by delivering it to peers ahead of its start time. Peers can then view the content without accessing the network, thus distributing the burden of service to peer nodes. We have demonstrated that this scheme provides a practical alternative to existing techniques for broadcasting content that can be created in advance.

REFERENCES

Blackshaw, P. and Nazzaro, M. (2004). Consumer-generated media (CGM). In *Intelliseek White Paper*.

Blake, I. F. and Chan, A. C.-F. (2005). Scalable, server-passive, user-anonymous timed release cryptography. In *Proc. of IEEE International conference on Distributed Computing Systems*, pages 504–513. IEEE.

Boneh, D. and Franklin, M. (2001). Identity-based encryption from the weil pairing. In *Proc. of CRYPTO 2001, LNCS*, volume 2139, pages 213–229. Springer.

Boneh, D. and Naor, M. (2000). Timed commitments and applications. In *Proc. of CRYPTO 2000, LNCS*, volume 1880, pages 213–229. Springer.

Chalkias, K. (2007). Timed-release encryption (TRE). In *short presentation in ECRYPT PhD Summer School, Emerging Topics in Cryptographic Design and Cryptanalysis*.

Clarke, I., Sandberg, O., Wiley, B., and Hong, T. (2000). Freenet: A distributed anonymous information strage and retrieval system. In *Proc. of ICSI Workshop on Design Issues in Anonymity and Unobservability*.

Garay, J. and Jakobsson, M. (2003). Timed release of standard digital signatures. In *Proc. of FC 2002, LNCS*, volume 2357, pages 168–182. Springer.

Gkantsidis, C., Miller, J., and Rodriguez, P. (2006). Anatomy of a P2P content distribution system with network coding. In *Proc. of 5th International Workshop on Peer-to-Peer Systems (IPTPS 2006)*.

Hankerson, D., Menezes, A., and Scott, M. (2008). Software implementation of pairings. In *http://www.math.uwaterloo.ca/ãjmeneze/publications/pairings_software.pdf*.

Liao, X., Jin, H., Liu, Y., Ni, L. M., and Deng, D. (2006). Anysee: peer-to-peer live streaming. In *Proc. of INFOCOM 2006*, pages 1–10.

Luac, M.-T., Nienac, H., Wub, J.-C., Pengac, K.-J., Huangade, P., Yaoac, J. J., Laif, C.-C., and Chenacd, H. H. (2007). A scalable peer-to-peer IPTV system. In *Proc. of CCNC 2007*, pages 313–317.

Mao, W. (2001). Timed-release cryptography. In *Proc. of SAC 2001, LNCS*, volume 2259, pages 342–357. Springer.

Open Mobile Alliance Ltd (2008). Oma digital rights management v2.1.

PeerCast.org (2002). Peercast P2P broadcasting. In *http://www.peercast.org/*.

Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S. (2001). A scalable content-addressable network. In *Proc. of SIGCOMM'01*, pages 161–172. ACM.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup services for internet applications. In *Proc. of SIGCOMM'01*, pages 149–160. ACM.

YouTube, LLC (2005). YouTube - broadcast yourself, http://www.youtube.com/.

Zhao, B. Y., Huang, L., Stribling, J., and S. C. Rhea, A. D. J. (2004). Tapestry: A resilient global-scale overlay for service deployment. In *IEEE Journal on Selected Areas in Communications*, volume 22, No.1, pages 41–53. IEEE.