Broadcast Encryption

Ahmed Obied
Department of Computer Science
University of Calgary

April 22, 2005

Abstract

Broadcast encryption is an interesting application of cryptography which allows one to broadcast a secret to a changing group of intended recipients in such a way that no one outside this group can view the secret. Interest in using broadcast encryption techniques has grown considerably in recent years and such techniques have been integrated in many applications and technologies such as virtual private networks, cable TV networks, mobile and wireless networks and many more. This report describes broadcast encryption in depth along with the related techniques, threats, protocols and applications.

Keywords: Information Security, Broadcast Encryption, Media Content Protection

1 Introduction

A new game is coming out next month. Its all over the papers. "Its about time ... this is the game you have been waiting for", "Its the only game which will beat all the other games", "Amazing 3D rendering engine, creative storyline and unbelievable AI". Millions of gamers around the world started counting down and the company which will release the game is getting concerned. How can they ensure that all the effort, money and time spent developing the game will pay off when the game is released. How can they ensure that only the people who are going to buy the original copy of the game can play it and no one else. In this scenario, there is a "secret" (the game's content) which only privileged users (the people who buy the original copy of the game) can view. To be able to broadcast the secret, one can use broadcast encryption techniques.

Broadcasting means transmitting information through a medium that is accessible to more than one receiver. Radio transmission for example uses air as a medium and anyone who has a radio receiver is able to listen to the broadcast. Broadcast communication most of time is one-way, that is, receivers can not send anything back to the broadcaster. Broadcast encryption is a way to broadcast information securely, that is to say, broadcasting a secret to a dynamically changing set of intended recipients in such a way that no one outside this set can recover the secret. It involves a broadcaster B (a.k.a the transmitter) and a set of N receivers where $N = \{ 1, 2, 3, ..., n \}$. If B wants to broadcast a secret s and only receivers in K where $K \subseteq N$ can view it, then every element $x \in K$ is given a key which can be used to recover s. If a receiver $y \in N$ and $y \notin K$ then y is considered an eavesdropper and it is computationally infeasible for y to recover s.

Symmetric encryption techniques use a shared key k to allow secure communications between members of a set N of users over an insecure channel. For instance, if a user $x \in N$ wants to send a message M to another user $y \in N$ over an insecure channel then x encrypts M using k, produces the ciphertext C and sends C to y. Upon receiving C, y decrypts it using k and recovers M. Symmetric encryption techniques have been proven to be successful when the set N is static, that is, no current users leave N and no new users join N. If the set N is dynamic, then it is obvious that symmetric encryption techniques will fail. For instance, if a user z leaves N then he will still be able to decrypt any ciphertext C which is encrypted using k because z simply knows k. In broadcast encryption, the basic principle is the same as in symmetric encryption, that is, a shared key is used by all members of N. However, some extra information is also given to each member so that a new shared key can be distributed securely and efficiently to those remaining in N after someone leaves [10].

2 Defining Broadcast Encryption

To fully understand the idea of broadcast encryption and the related issues, one must first describes the conventions that are going to be used in this report and gives an overview of a simple broadcast encryption scheme.

2.1 Conventions

Let Q denote a network with $N + 2$ nodes and N receivers (users). One node in Q which will be denoted as B is known as the broadcaster and another node in Q which will be denoted as K_s is known as the key server. The broadcaster B has a set S of secrets which can only be viewed by a set of privileged receivers R_p where $R_p \subseteq N$ and the key server K_s has a set K of symmetric keys. Any $k \in K$ can be used to encrypt a secret $s \in S$ using an encryption function E_k and decrypt s using a decryption function D_k. K_s is responsible for both generating the symmetric keys and associating them with the privileged receivers in R_p. The set of non-privileged receivers will be denoted as R_r where $R_r \subseteq N$ and $R_r \cap R_p = \emptyset$ along with $R_r \cup R_p = N$ must both be true. Any receiver $r \in R_p$ is said to be a privileged receiver and any receiver $r \in R_r$ is said to be a revoked receiver.

A broadcast message M is usually divided into two parts: a header (M_h) and a body (M_b). The header contains information that can be used to access the content (e.g. key material and user memberships) and the body contains the protected (encrypted) content. The header in a broadcast message is the most important part when one analyzes any broadcast encryption scheme.

2.2 Basic Broadcast Encryption scheme

A basic broadcast encryption scheme consists of 4 algorithms [3] which can be performed in polynomial time. These algorithms are as follows:
REGISTRATION: This algorithm is used to register new receivers that can view some secret $s \in S$. In particular, whenever a new receiver $r \in R_r$ wants to join R_p, then K_s removes r from R_r and adds it to R_p. If a receiver $r \in R_p$ wants to leave R_m, then K_s removes r from R_p and adds it to R_r.
KEY_GENERATION: This algorithm is used to generate symmetric keys from K and associate them with privileged receivers. Generating strong session keys is important to make it computationally infeasible for any revoked receiver $r \in R_r$ to compromise any key.
ENCRYPTION: This algorithm is used to encrypt some secret $s \in S$ with some key $k \in K$.
DECRYPTION: This algorithm is used to decrypt some secret $s \in S$ with some key $k \in K$.
A basic working model for broadcast encryption can be described as follows: K_s associates each privileged receiver in U_p with a key from K. If B wants to broadcast a secret $s \in S$ then B first communicates with K_s to get the keys associated with each privileged receiver, encrypts s once for each privileged receiver and finally broadcasts all encrypted messages. It is assumed in this model that there is a "secure" key server which uses a protocol to authenticate the broadcaster, and the receivers (e.g. has a list of privileged and revoked receivers). Furthermore, it is assumed that a communication medium exists between the nodes in Q and there is a secure channel between k_s and B.

The Simple broadcast encryption scheme that was described is inefficient indeed. The aim of all intelligent broadcast encryption schemes is to take into consideration

many other factors and improve the performance of the scheme. Examples for such improvements would be to reduce the processing time at both the broadcaster and at the receivers, to reduce the storage size at the receivers, and to reduce the broadcast message size. Efficient and more reliable broadcast encryption schemes are described in detail in next sections.

3 Security threats

To analyze and prove the security of a cryptosystem, a model is needed to define the capabilities of the attacker [10]. In classical cryptography, many different type of attacker capabilities have been defined. For instance, known text attacks, chosen ciphertext attacks, chosen plaintext attacks, ciphertext only attacks, etc. These attacks are still applicable in broadcast encryption along with new type of attacks that are going to be defined in this section.

3.1 Collusion

Collusion (a.k.a cooperation) means that revoked receivers collude to break the security of a broadcast encryption scheme. Some broadcast encryption schemes are highly secure against collusion and they can resist even if all revoked receivers in R_r collude. Other broadcast encryption schemes can also resist against collusion but up to a fixed number m of cooperative revoked receivers.

3.2 Piracy

Piracy is a major concern which violates copyright protection techniques. Piracy is related to collusion where one or more revoked receivers in R_r can cooperate to break the encryption scheme used to protect some media. Piracy and its countermeasure which is known as "Traitor Tracing" are described in more detail in the next sections.

3.3 Backward secrecy

Backward secrecy means that a privileged receiver cannot use the information it has to recover material that was broadcast before it was added [10]. Imagine a situation when the broadcaster B wants to broadcast secrets $s_1, s_2, ..., s_k$ from S where such secrets are related, that is, one can think of it as a TV show which has k minutes and each s_i corresponds to 1 minute in the show. If a revoked receiver $r \in R_r$ eavesdrops and records secrets $s_1, s_2, ..., s_{k-1}$ and right before s_k is broadcast r registers, that is, r gets removed from R_r and added to R_p. If B now broadcasts the ciphertext of s_k, and r uses his knowledge of recovering s_k to recover $s_1, s_2, ..., s_{k-1}$ and fails then backward secrecy is maintained.

3.4 Forward secrecy

Forward secrecy means that when a privileged receiver is removed from R_p then it must not be able to continue viewing protected content of the broadcast. Imagine a situation similar to the one described in the above, that is, a broadcaster B wants to broadcast secrets $s_1, s_2, ..., s_k$ from S where such secrets are related. If a privileged receiver $r \in R_p$ receives $s_1, s_2, ..., s_{k-1}$ and right before s_k is broadcast r leaves R_p, that is, r gets removed from R_p and added to R_r. If B now broadcasts the ciphertext of s_k, and r uses his knowledge of recovering $s_1, s_2, ..., s_{k-1}$ to recover s_k and fails then forward secrecy is maintained.

4 Broadcast Encryption Schemes

Several broadcast encryption schemes and key management techniques were proposed to broadcast a secret to a changing group of intended recipients in such a way that no one outside this group can view the secret. A broadcasting scheme involves encrypting a message so that more than one privileged receiver can decrypt it. To achieve that, privileged receiver are grouped together either dynamically or statically according to the scheme being used. The performance of any broadcast encryption scheme depends on how the privileged receivers are grouped. In the dynamic case, one way to achiever this is to build a key graph, that is, a set of encryption keys ordered in a graph [7]. Privileged receivers are added to this graph, and the keys are distributed. Two different broadcast encryption techniques are described in this section. One of these schemes is a stateful scheme and the other one is a stateless scheme.

4.1 Stateful schemes

A stateful broadcast encryption scheme requires that all the receivers have to be able to update the stored keys, usually when receivers are added or removed from the privileged receiver set R_p. This implies that any receiver $r \in R_p$ must be connected all the time to the broadcast network Q in order not to lose any key update message that might be sent. An example of a stateful broadcast scheme is the Logical Key Hierarchy scheme which is described in the following section.

4.1.1 Logical Key Hierarchy (LKH)

The Logical Key Hierarchy scheme was proposed by [17]. The basic idea of the LKH scheme is to build a key graph that contains a set of encryption keys and associate each privileged receiver in R_p with the leaf nodes in the graph. Whenever a receiver r joins R_p or leaves R_p, the keys in the graph are updated in such a way that can maintain both forward and backward secrecy.

Key graph structure

The graph used in the LKH scheme is a directed acyclic graph which forms a tree. The tree does not have to be a binary tree and to increase the performance, it is

recommended to balance the tree. In the LKH scheme, the key server K_s generates random keys from K and assign them to each node in the graph. Each receiver in R_p is associated with a leaf key node in the key graph. At first the set R_p starts out as empty and the graph only contains one node which is the root node. Nodes are added to the graph whenever a revoked receiver joins and nodes are removed from the graph whenever a privileged receiver revokes. Any privileged receiver must know its own key and the keys in the path from its key node up to and including the root node.

Adding a revoked receiver to R_p

When a revoked receiver $r \in R_r$ wants to leave R_r and join R_p, then the following is done:

1. K_s authenticates r via some authentication protocol and ensures that r is allowed to join.
2. K_s removes r from R_r and adds it to R_p.
3. K_s generates a new leaf key node and assign it to r along with the keys from the path where r's key node is located and up to and including the root node.
4. To maintain backward secrecy and prevent r from decrypting previous broadcast, all the key nodes from r's key node location and up to and including the root node are regenerated and sent via a rekeying message (more on rekeying is described later) to the current privileged receivers.

Removing a privileged receiver from R_p

When a privileged receiver $r \in R_p$ wants to leave R_p and join R_r, then the following is done:

1. K_s removes r from R_p and adds it to R_r.
2. K_s removes the leaf key node from the key graph.
3. To maintain forward secrecy and prevent r from decrypting future broadcast, all the key nodes from r's old key node location and up to the root and including the root node are regenerated and sent via a rekeying message to the current privileged receivers.

Rekeying

Whenever a receiver joins or leaves the set of privileged receiver R_p, some keys are regenerated to maintain backward and forward secrecy. The process on when K_s sends new keys to some privileged receiver is known as "rekeying". In [17], three different strategies that differ in how to construct and send the rekey message have been proposed. These strategies are described below:

- **User-oriented rekeying:** In this strategy, K_s encrypts all the new keys needed by a privileged receiver with a key held by the receiver and sends it to that receiver.

- **Key-oriented rekeying:** In this strategy, K_s encrypts each new key needed by privileged receivers individually and using each of its children keys. The rekey message is then distributed to the user set of that child (where the user set of a key is the set of users that share that key) [17].
- **Group-oriented rekeying:** In this strategy, K_s encrypts all the new keys with the child nodes' keys and broadcast the rekey message to the entire group.

Encryption

Encryption is done by encrypting any broadcast secret $s \in S$ with the key of the root node. If the key of the root node is k_r and the broadcaster B wants to broadcast a secret $s \in S$ then B broadcasts $E_{K_r}(s) = C$.

Decryption

Decryption is done by decryption any broadcast secret $s \in S$ with the key of the root node. If the key of the root node is K_r and the broadcast message was $C = E_{k_r}(s)$ then a privileged receiver can recover s by applying $D_{k_r}(C) = D_{k_r}(E_{k_r}(s)) = s$.

Broadcast message

It was mentioned before that any broadcast message M has a header M_h and a message body M_b. In the LKH scheme, a broadcast message looks as follows:

$null$	$E_{k_r}(s)$

The header in this scheme is contains no information and the body contains the ciphertext of the broadcast message.

Analysis and Complexity

One might wonder what is the point of keeping all the other keys if everything is encrypted with the root node key. Basically the other keys are used in the rekeying procedure to protect the key of the root node whenever it is regenerated and redistributed. Since all the privileged receivers know the key of the root node then it is quite efficient in terms of encryption and decryption. The only drawback in terms of performance issues of this scheme is the storage space of the keys. If the height of the key graph is h then a receiver must know all the keys from its assigned key node up to and including the root node. Below is a table which shows the complexity of the LKH scheme in terms of big-O notation [17]:

Storage space	**Message size**	**Processing time**	**No. Decryption**
$O(h)$	$O(1)$	$O(1)$	$O(1)$

Security issues

The security of the LKH scheme can be broken if an adversary can decrypt any broadcast message. There are some security issues that must be considered and taken into account to avoid opening up a security hole. These issues are described below:

- The key server K_s must generate new random keys each time a receiver joins R_p or leaves R_p to maintain backward and forward secrecy.
- Generating strong random keys from K which can be done by using strong cryptographic pseudorandom number generators will make it extremely difficult and sometimes impossible for any adversary from R_r to decrypt any broadcast message without knowing the key.
- The use of public key digital signatures schemes and message digests (e.g. using RSA and SHA-1) can prevent against masquerading attacks when sending rekeying messages, that is, if an adversary masquerades as the server and sends unauthorized rekey messages then the attacker will not succeed.

4.2 Stateless schemes

A stateless broadcast encryption scheme means that all receivers cannot update their keys between sessions, that is, the initialization step is performed only once and after the initialization step all receivers have the information that is needed to decrypt future broadcast message. An example on where a stateless broadcast encryption scheme is used is pay TV shows. When a new user subscribes to a a pay TV show, then the user receives a smartcard which contains the decryption key and can be used for decrypting the broadcast. An example of a stateful broadcast scheme is the Complete Subtree. Before one describes the scheme, some definitions need to be described first.

- *Cover:* A family C of nonempty subsets of N whose union contains the given set N (and which contains no duplicated subsets) is called a cover (or covering) of N.
- *Steiner tree:* The Steiner tree of some subset of the nodes (vertices) of a graph G is a minimum-weight connected subgraph of G that includes all the nodes. The result is always a tree.
- *Key indistinguishability:* [13] introduced the term key indistinguishability which means that any secret key $k_i \in K$ and which corresponds to subset Si remains pseudorandom to the adversary even if the adversary learns all the secret information belonging to all the users outside of Si ($i = 0, ..., l$ where l is the maximum number of subsets of N).

4.2.1 Complete Subtree (CS)

The Complete Subtree scheme was proposed in [13]. The basic idea of such scheme is to form a binary tree and assign each receiver in N a leaf node in the binary tree. All receivers are then divided into subsets and a steiner tree is formed to highlight the revoked receivers and exclude them when forming the subsets. Each subset S_i contains a number of privileged receivers and gets assigned a key k_i and A subset cover C is

then formed to keep track of the current subsets (in this scheme and for simplicity purposes, it is assumed that the key server K_s can broadcast secret messages from S).

Key graph structure

The graph used in the CS scheme is a directed acyclic graph which forms a tree. The tree in the CS scheme must be a binary tree and for simplicity purposes, the number of receivers in N must be powers of 2. All receivers in N gets assigned to the leaf nodes of the binary tree.

Initialization step

Before the initialization step starts, it is assumed that the state of the key graph is known, that is, all the privileged and revoked receivers must be known before any subset is formed or any key is assigned . In the initialization step, a set of all subtrees of the key graph is maintained and a Steiner tree representing the revoked receivers is formed. The subtree that are hanging out of the Steiner tree forms valid subsets and the leaf nodes within each subtree are assigned to the corresponding privileged user that was assigned to the leaf earlier. The key server K_s generates random keys from K and assign them to each subset in the cover C. Each privileged user is given secret information via some secure channel (e.g. using smartcards or embedding the secret information into a chip of some receiver when its manufactured, etc). The secret information in the CS scheme contains the keys for the subsets the receiver belongs to. In other words, all the subsets from the leaf to the root node.

Encryption

In the CS scheme, privileged receivers are enveloped by a subset cover to allow them only to decrypt a broadcast message. Encryption works is as follows:

1. The key server K_s wants to broadcast a secret $s \in S$.
2. K_s first forms the Stenier tree graph and generates the subset cover C by adding all the subsets that are at distance one from the Steiner tree graph. Such subsets do not belong to the Stenier tree and thus all receivers of those subsets must be privileged.
3. K_s generates a random session key $k \in K$. K_s uses the encryption function E and k to produce the ciphertext $C = E_k(s)$.
4. For each subset in the cover C, K_s uses the encryption function E and the key K_i associated with subset S_i in C to produce ciphertexts $C_1 = E_{k_1}(k), C_2 = E_{k_2}(k), ..., C_i = E_{k_i}(k)$.
5. K_s adds the ciphertexts C_i to the header of the broadcast message and adds C to the message body M.
6. Finally K_s broadcasts M.

Decryption

The way decryption works is as follows:

1. A receiver r receives M.
2. r searches the header of M (M_h) to find the subset S_i it belongs to.
3. If r cannot find the subset it belongs to then the decryption process fails. Otherwise, r uses the secret information that was given to it in the initialization step to recover k_i and takes C_i from the M_h. r applies the decryption function D to recover the session key $D_{k_i}(C_i) = D_{k_i}(E_{k_i}(k)) = k$. r then takes the ciphertext C from the message body M_b and uses the session key k along with the decryption function D to recover the secret $D_k(C) = D_k(E_k(s)) = s$.

Broadcast message

A broadcast message in the CS scheme looks as follows:

$C_1 = E_{k_1}(k), C_2 = E_{k_2}(k), ..., C_i = E_{k_i}(k)$	$E_k(s)$

Complexity

The complexity of the Complete Subtree scheme can be summarized as follows:

Storage space	**Message size**	**Processing time**	**No. Decryption**
$O(log(N))$	$O(\|R_r\|log(\frac{N}{\|R_r\|}))$	$O(log(log(N)))$	$O(1)$

Security issues

The security of the CS scheme can be broken if an adversary can decrypt any broadcast message. There are some security issues that must be considered and taken into account to avoid opening up a security hole. These issues are described below:

- The key indistinguishability property must be fulfilled, that is, an adversary from the revoked set R_r should not be able to distinguish a random key from any subset key k_i. If the key indistinguishability property is not fulfilled then an adversary can compromise the subset keys and decrypt any broadcast messages.
- To maintain backward and forward secrecy, the Steiner tree graph and the minimal subset cover C must be reformed whenever a new receiver is added or removed.
- The encryption functions used to encrypt any secret $s \in S$ and the session key $k \in K$ must be chosen carefully. Choosing a well known secure encryption functions that stood against a number of attacks for a long period of time can help prevent against chosen ciphertext attacks and chosen plaintext attacks.
- In the initialization step, the secret information used to deduce the subset key k_i must be transmitted to the user (or receiver) via a secure channel. If an adversary can intercept the communication and deduce the secret information, then the security of the algorithm is temporarily compromised until the affected receivers have been revoked [7].

5 Piracy and Traitor Tracing

One of the security threats known as piracy was described briefly in previous sections. Piracy, which violates copyright and ownership laws, is a major concern in today's highly technical life and interest for effective countermeasure techniques to protect against such violation in active broadcast scenarios has grown considerably in recent years. In most broadcast scenario, a data provider supplies keys to privileged users. However, privileged users may collude in order to make these keys publicly available to anyone (e.g. publishing the keys on the Internet). In such scenario, these privileged users are called pirates (also called traitors). Pirates make a business of breaking the security safeguards of the conditional system and sell devices that allow unauthorized users to view the content illegally [4]. Countermeasure schemes known as traitor tracing constitute a very useful tool against piracy in the context of digital content broadcast. Such schemes allow one to trace the source of piracy, disconnect the unauthorized users from further transmittal of information and supply legal evidence of the pirate identity.

5.1 A Public Key Traitor Tracing Scheme

An efficient public key traitor tracing scheme was proposed in [2] and provided a simple and efficient solution to the traitor tracing problem. The main idea is to have one public encryption key and many private decryption keys. If some digital content is encrypted using the public key and distributed through a broadcast channel, then each legitimate user can decrypt using its own private key. Furthermore, if a coalition of users collude to create a new decryption key then there is an efficient algorithm to trace the new key to its creators [2].

The scheme proposed in [2] is known to be deterministic, that is, the scheme can catch all of the pirates who contributed to the attack and innocent users are never accused as long as the number of colluders (pirates) is at or below the collusion threshold.

Components

The public key traitor tracing scheme consists of four components. A basic and general description of the four components is given below. In depth description of the four components is given afterwords.

KEY_GENERATION: This algorithm takes two parameters as input: the number of private keys l to generate and a security parameter z (random seed). The algorithm outputs the public encryption key e and the private keys $k_1, k_2, ..., k_l$. Any private key k_i where $i = 1, ..., l$ can decrypt any ciphertext encrypted with the public key e.

ENCRYPTION: This algorithm takes two parameters as input: the public key e and a secret s. The algorithm outputs a ciphertext C.

DECRYPTION: This algorithm takes as input two parameters: the ciphertext C and any of the private keys k_i where $i = 1, ..., l$. The algorithm outputs the secret s.

TRACING: If a pirate compromises m decryption keys $k_1, k_2, ..., K_m$ then the the pirate can use the m keys to create a decryption decoder D. The encryption scheme is said to be "m-resilient" if there is a tracing algorithm that can determine at least one of the k_i's in the pirates possession [2].

Representations

The public key traitor tracing scheme relies on the representation problem [2], that is, when $y = \prod_{i=1}^{2m} {h_i}^{\gamma_i}$ then $(\gamma_1, \gamma_2, ..., \gamma_{2m})$ is a "representation" of y with respect to the base $h_1, h_2, ..., h_{2m}$. If $\bar{k_1}, \bar{k_2}, ..., \bar{k_n}$ are representations of y with respect to the same base then so is any "convex combination" of the representations: $\bar{d} = \sum_{i=1}^{n} \psi_i \bar{k_i}$ where $\psi_1, \psi_2, ..., \psi_n$ are scalars such that $\sum_{i=1}^{n} \psi_i = 1$.

The Encryption algorithm

Assumptions

- l represents the number of private decryption keys and m represents the maximal coalition size. Hence without loss of generality, assume that $l \geq 2m + 2$ and assume that there is a large prime $q > \max(l, 2m)$.
- Let G_q be a group of prime order q.
- The public key traitor tracing scheme makes use of a linear space tracing code Γ which is a collection of l codewords over Z_q^{2m}. To fully understand how the set Γ is defined, consider the matrix A which is an $(l - 2m)$ x l matrix:

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & l \\ 1^2 & 2^2 & 3^2 & \dots & l^2 \\ 1^3 & 2^3 & 3^3 & \dots & l^3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1^{l-2m-1} & 2^{l-2m-1} & 3^{l-2m-1} & \dots & l^{l-2m-1} \end{pmatrix} \mod q$$

  Let $b_1, b_2, ..., b_{2m}$ be a basis of the linear space of vectors satisfying $A\bar{x} = 0 \mod q$. The $2m$ vectors in the basis can be viewed as the columns of an l x $2m$ matrix B. Γ is therefore defined as the set of rows of the matrix B which can be written as $\Gamma = \alpha^{(1)}, \alpha^{(2)}, ..., \alpha^{(l)}$. Furthermore, Γ is considered fixed and publicly known.
- Assume that the pirate decoder D contains at least one representation of y, that is, a pirate has m keys $\bar{k_1}, \bar{k_2}, ..., \bar{k_m}$ and built D to make use of these keys. Furthermore, assume that by examining the pirate decoder, it is possible to obtain one of these representation $\bar{k}$ which must lie in the linear span of the representation $\bar{k_1}, \bar{k_2}, ..., \bar{k_m}$.

KEY_GENERATION: In the key generation algorithm, the following steps are performed:

1. Pick an integer $g \in G_q$ which is a generator (primitive root) of G_q.
2. For $i = 1, ..., 2m$ do the following:
   - Randomly choose $r_i \in Z_q$.
   - Compute $h_i = g^{r_i}$.
3. Compute $y = \prod_{i=1}^{2m} {h_i}^{\gamma_i}$ for random $\gamma_1, \gamma_2, ..., \gamma_{2m} \in Z_q$.
4. Generate the public key $e = < y, h_1, h_2, ..., h_{2m} >$

5. To Generate the private keys, one must first note that a private key is an element $k_i \in Z_q$ such that $k_i \cdot \alpha^{(i)} = \bar{k_i}$ is a representation of y with respect to the base $h_1, h_2, ..., h_{2m}$. Hence to generate the ith key from the codeword $\alpha^{(i)} = (\alpha_1, \alpha_2, ..., \alpha_{2m})$, one computes:

$$k_i = \frac{\sum_{i=1}^{2m} r_i \gamma_i}{\sum_{i=1}^{2m} r_i \alpha_i} \mod q$$

ENCRYPTION: The public encryption key $e = < y, h_1, h_2, ..., h_{2m} >$ and hence to encrypting any secret $s \in G_q$, the following is done:

1. Randomly choose an element $a \in Z_q$.
2. Compute the ciphertext $C = < s \cdot y^a, h_1^a, h_2^a, ..., h_{2m}^a >$

DECRYPTION: To decrypt a ciphertext $C = < s \cdot y^a, h_1^a, h_2^a, ..., h_{2m}^a >$ with any decryption key k_i and recover s, the following is computed:

$$\begin{aligned}
\frac{s \cdot y^a}{(\prod_{i=1}^{2m} (h_i^a)^{\alpha(i)})^{k_i}} &= \frac{s \cdot y^a}{(\prod_{i=1}^{2m} (h_i^a)^{\alpha(i)})^{\frac{\sum_{i=1}^{2m} r_i \gamma_i}{\sum_{i=1}^{2m} r_i \alpha_i}}} = \frac{s \cdot y^a}{(\prod_{i=1}^{2m} (g^{a r_i})^{\alpha(i)})^{\frac{\sum_{i=1}^{2m} r_i \gamma_i}{\sum_{i=1}^{2m} r_i \alpha_i}}} \\
&= \frac{s \cdot y^a}{(\prod_{i=1}^{2m} g^{r_i \alpha(i)})^{a \frac{\sum_{i=1}^{2m} r_i \gamma_i}{\sum_{i=1}^{2m} r_i \alpha_i}}} = \frac{s \cdot y^a}{(g^{\sum_{i=1}^{2m} r_i \alpha(i)})^{a \frac{\sum_{i=1}^{2m} r_i \gamma_i}{\sum_{i=1}^{2m} r_i \alpha_i}}} \\
&= \frac{s \cdot y^a}{(g)^{a \sum_{i=1}^{2m} r_i \gamma_i}} = \frac{s \cdot y^a}{(g^{\sum_{i=1}^{2m} r_i \gamma(i)})^a} = \frac{s \cdot y^a}{(\prod_{i=1}^{2m} (g^{r_i})^{\gamma_i})^a} \\
&= \frac{s \cdot y^a}{(\prod_{i=1}^{2m} {h_i}^{\gamma_i})^a} = \frac{s \cdot y^a}{y^a} = s
\end{aligned}$$

TRACING: The main goal of the tracing algorithm is that given $\bar{k} \in Z_q^{2m}$ which is formed by taking a linear combination of at most m vectors in Γ, one can determine the unique set of vectors in Γ that were used to construct $\bar{k}$. The vectors in Γ form the rows of the matrix B (as shown before). Therefore, there must exist a vector $\bar{w} \in F_q^l$ of Hamming weight at most k such that $\bar{w} \cdot B = \bar{k}$. To recover $\bar{w}$ given $\bar{k}$, one can do the following:

1. Search for a vector $\bar{v} \in F_q^l$ that satisfies $\bar{v} \cdot B = \bar{k}$. $\bar{v} - \bar{w}$ is in the linear span of the rows of the matrix A (the matrix was described previously) since $(\bar{v} - \bar{w}) \cdot B = 0$. Thus, there exists a unique polynomial $p \in F_q[x]$ of degree at most $l - 2m - 1$ such that $\bar{v} - \bar{w} = < p(1), p(2), ..., p(l) >$.
2. $\bar{v} - \bar{w} = < p(1), p(2), ..., p(l) >$ equals $\bar{v}$ in all but m components since $\bar{w}$ is a Hamming weight of at most m.
3. Find p given $\bar{v} \in Z_q^l$ using Berlekamp's algorithm [16] which works as follows:
   - Let f be a polynomial of degree at most k such that $f(i) = 0$ for all $i = 1, 2, ..., l$ for which $p(i) \neq v_i$ where v_i is the ith component of v.
   - $p(i)f(i) = f(i)v_i$ for all $i = 1, 2, ..., l$. Since the polynomial pf has degree at most $l - m - 1$ then there are l equations for each $i = 1, 2, ..., l$ in l variables.

- Let h and f be a solution where h is a polynomial of degree at most $l-m-1$ and f is the polynomial as described before (of degree at most k). Hence, $h(i) = f(i)v_i = f(i)p(i)$ whenever $p(i) = v_i$ (e.g. at $l-k$ points). It follows that $p = \frac{h}{f}$.

4. Once p is found then p gives the vector $\bar{v} - \bar{w}$ from which one can recover $\bar{w}$.

Security issues

The security of the public key traitor tracing scheme depends on the hardness of the discrete logarithm problem (DLP), that is, if one can efficiently solve the DLP, then the scheme is broken.

6 Content Protection

An important application of broadcast encryption is content protection of recordable media such as DVDs. The scenario described at the beginning of this report can make one wonder of how such problem is solved. After describing the idea of broadcast encryption and the related issues, one can now understand how such techniques can be used and integrated to solve the problem. Companies that produce games release their games on DVDs which must only be read and played by the console on which the game was programmed for. Piracy causes the effort, time, and money spent to develop these games to vanish in thin air because privates simply buy the original copy of the game (the DVD), make extra copies of the it and either distribute it for free or sell it for a cheaper price.

Many cryptographically based content protection technologies exist these days and what follows is a brief description of a cracked content protection system and an in depth description of a promising system which uses broadcast encryption techniques.

6.1 DVD Content Scrambling System (CSS)

The main goal of the Content Scrambling System is to allow a DVD Player to authenticate itself to the DVD. The system consists of two entities: the DVD player, and the DVD disk. Whenever a DVD player wants to authenticate itself to the DVD then the CSS algorithm tries to find out which disk key the disk has its title keys encrypted with, then decrypts the title key. For each title key, the system decrypts the associated ".VOB" file that contains the video and audio data.

The Content Scrambling System is considered cryptographically weak. The keys used in the system are 5 bytes long and cipher being used in the algorithm allows one to calculate the key easily from any known plaintext/ciphertext pair. The CSS algorithm was broken in 1999 by a Norwegian programmer [12]. The programmer discovered that one licensee of a DVD technology (Xing technologies which is a subsidiary of RealNetwork) failed to follow licensing rules and encrypt their decryption keys. With this unencrypted key in hand, the programmer was able to reverse-engineer the XingDVD player and implement the CSS algorithm cracker code in a program called DeCSS.

Because the unlock key for the CSS encrypting system is only 5 bytes long, the programmer was able to guess roughly 170 other keys. Breaking DVDs encryption was considered extremely difficult at first but once the first key was found, the rest fell with ease, since the programmer was able to use the original valid key as a launch point to find even more valid decryption keys. The cracker claimed that his intent was never to pirate DVD content but rather just to find means to play DVDs on his Linux machine, for which a DVD player did not yet exist.

One can quickly observe that if broadcast encryption techniques had been used instead, this attack would not be as severe at that time, since the new DVDs would switch encryption keys and the attack would no longer be effective.

6.2 Content Protection for Recordable Media (CPRM)

The Content Protection for Recordable Media technology was jointly developed by 4C (Intel, IBM, Matsushita and Toshiba) for enforcing copy protection and prevention on personal computers. The CPRM technology uses broadcast encryption and traitor tracing techniques to protect content protection from simple replication, trace and revoke non-compliant devices, and survive multiple "hacking events". To accomplish this feat, 4C generates a 1MB media key block, which calculates a cryptographic key with a one-way key algorithm and writes the block to the media at the time of manufacture. The key is then later used in a non-handshaking broadcast to a central management server. The types of physical media supported by the CPRM technology include, but are not limited to, recordable DVD media and Flash memory.

The CPRM technology requires specially designed copying software which bypasses the operating system and communicates directly with the hard drive. If a user wants to write for instance an MP3 file to the hard drive, then the hard drive and the software authenticate to each other. Once authentication succeeds then the hard drive sends the software a key k_1 that is stored in a non-standard place on the drive and which is unique to that hard drive, and a counter c that is also stored somewhere on drive. A central management server on the Internet or a user-level application generates a random key k_{r1} based on the current value of the counter, encrypts the MP3 file with k_{r1} and writes the encrypted form of the MP3 file as an ordinary file on the hard drive. The value of c and K_{r1} are then hashed into a key that is used to encrypt k_{r1}. The encrypted form of k_{r1} is also written as a second ordinary file on the hard drive.

Once the user wants to play the MP3 file then an MP3 player designed to deal with the encrypted forms of the MP3 files will read k_1 and c from the hard drive, hash them and use the resulted key to decrypt the k_{r1}. k_{r1} is then used by the MP3 player to decrypt the encrypted form of the MP3 file and play it.

If a user wants to move (as opposed to copy) the MP3 file to another hard drive, the software will check to determine if this is permissible. File permissions can either be embedded in the MP3 file or the software will query another computer (the central management server) over the Internet to get the correct permissions of the file. If moving an MP3 file is allowed, the software will encrypt the MP3 file for the new hard drive (only allowing it to be stored in a copy-protected medium), increment the counter c in the old hard drive, use the new value of c to generate a new random key k_{r2}, hash

k_2 and c to create a key to encrypt k_{r2} and finally overwrite the old key file containing the encrypted form of k_{r1} with the encrypted form of k_{r2}.

If a user copies the encrypted object to another hard drive without going through the approved procedure, its key will not be in the key file on the new hard drive. Therefore, the reader cannot play it. If the user copies both of them to another hard drive, the key-file will not be decryptable since its key depends on the hard drive specific keying information and the current value of c. If the user makes a backup copy of his entire disk, "moves" the encrypted song onto another hard drive, then scrubs and restores the entire original disk, the restored key file will not be decryptable, since the counter c would have changed.

7 Conclusion

Broadcast encryption is an interesting application of cryptography and an exciting field with many directions to take it in. Interest in using techniques offered by such field has grown considerably in recent years and such techniques have been integrated in many applications and technologies such as virtual private networks, cable TV networks, mobile and wireless networks and many more. This report described broadcast encryption in depth along with the related techniques, threats, protocols and applications.

References

[1] Berkovits, S. How to broadcast a secret. In *Advances in Cryptology: EUROCRYPT '91* (1992), pp. 536–541.

[2] Boneh, D., and Franklin, M. An efficient public key traitor tracing scheme. In *In Proceedings Crypto '99, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag* (1999), pp. 338–353.

[3] Dodis, Y., and Fazio, N. Public key broadcast encryption for stateless receivers. In *ACM Workshop on Digital Rights Management* (2002).

[4] Dodis, Y., Fazio, N., Kiayias, A., and Yung, M. Scalable public-key tracing and revoking. In *PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing* (2003), ACM Press, pp. 190–199.

[5] Fiat, A., and Naor, M. Broadcast encryption. In *Advances in Cryptology: CRYPTO '93* (1994), pp. 480–491.

[6] Halevy, D., and Shamir, A. On dynamic subset difference revocation scheme. In *Lecture Notes in Computer Science* (2004), pp. 453:316–322.

[7] Hesselius, T., and Savela, T. A java framework for broadcast encryption. Master's thesis, Linkopings Universitet, 2004.

[8] hua Chu, H., Qiao, L., Nahrstedt, K., Wang, H., and Jain, R. A secure multicast protocol with copyright protection. *SIGCOMM Comput. Commun. Rev. 32*, 2 (2002), 42–60.

[9] Jin, H., Lotspiech, J., and Nusser, S. Traitor tracing for prerecorded and recordable media. In *DRM '04: Proceedings of the 4th ACM workshop on Digital rights management* (2004), ACM Press, pp. 83–90.

[10] Kreitz, G. Optimization of broadcast encryption schemes. Master's thesis, Royal Institute of Technology, 2005.

[11] Lawrence, C., Tseung, N., and Yu, C. The implementation of guaranteed, reliable, secure broadcast networks. In *CSC '90: Proceedings of the 1990 ACM annual conference on Cooperation* (1990), ACM Press, pp. 259–265.

[12] Lotspiech, J., Nusser, S., and Pestoni, F. Broadcast encryption's bright future. In *IEEE Computer* (2002), vol. 35, pp. 57–63.

[13] Naor, D., Naor, M., and Lotspiech, J. Revokation and tracing schemes for stateless receivers. In *Advances in Cryptology: CRYPTO '01, volume 2139 of Lecture Notes in Computer Science* (2001), pp. 41–62.

[14] Shamir, A. How to share a secret.

[15] Shi, C., and Bhargava, B. A fast mpeg video encryption algorithm. In *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia* (1998), ACM Press, pp. 81–88.

[16] Welch, L., and Berlekamp, E. Error correction of algebraic block codes.

[17] Wong, C. K., Gouda, M., and Lam, S. Secure group communications using key graphs. In *SIGCOMM '98: Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication* (1998), ACM Press, pp. 68–79.