1 Broadcast Systems

Broadcasting refers to simultaneous transmission of a message to many receivers/users. Examples of broadcast systems include FM radio, DirectTV, streaming audio/video, shared file system and GPS. Some of the concerns faced by broadcast systems are:

- **Bandwidth**: The transmission of the broadcast signal should consume as less bandwidth as possible, which has become a scarce and costly resource nowadays.

- **Denial-of-Service**: The broadcast system should take measures to minimize the possibility of Denial-of-Service or D.O.S. attacks i.e. a group of non-subscribers should not be able to block or manipulate the broadcast signal for a subscriber.

- **Confidentiality**: The broadcast system should ensure that only the subscribers can extract the message from the broadcast signal. The unsubscription of some users should not affect message security and the remaining subscribers.

Broadcast encryption takes care of the confidentiality concern – schemes have been given which ensure message security even when there are multiple data streams, each having a different set of subscribers.

2 Broadcast Encryption

2.1 A Simple Broadcast Encryption Scheme

Suppose there are $n$ subscribers and $S$ be the set of subscribers to which a message needs to be broadcasted securely. A simple way to do this is to assign a $PK$-$SK$ pair to each of the $n$ subscribers. The transmitter then uses some public key encryption technique to encrypt the message using the $PK$’s of the subscribers in $S$ to get $|S|$ different cipher-texts. These cipher-texts are broadcasted and only the subscribers in $S$ are able to retrieve the message from the broadcast signal.
This scheme proves to be inefficient as significant amount of bandwidth is used up by the broadcast signal – if $BW$ is the average bandwidth needed to broadcast a cipher-text, the bandwidth needed to transmit the broadcast signal for $S$ under this scheme is approximately $|S| \cdot BW$. So we desire a broadcast encryption scheme which gives only one cipher-text for a given message and $S$ thereby conserving bandwidth.

2.2 Algorithms for Broadcast Encryption

A typical broadcast encryption scheme consists of the following algorithms:

- $\text{Setup}(\lambda, n)$: Takes as input the number of subscribers $n$ to output the public parameters $PP$ and secret keys $K_1, K_2, \ldots, K_n$ for the subscribers.

- $\text{Encrypt}(S, PP, M)$: Takes as input a message $M$, the set of subscribers $S$ to which $M$ needs to be broadcasted and $PP$ to output the cipher-text $C$.

- $\text{Decrypt}(S, i, K_i, C)$: Takes as input the set of subscribers $S$, the subscriber id $i$, $K_i$ and $C$ to output message $M$ if $i \in S$ and nothing ($\bot$) if $i \notin S$.

The $\text{Decrypt}$ algorithm is run by all the subscribers. The correctness of the scheme lies in the fact that retrieval of $M$ from $C$ should be possible only when the subscriber is in $S$.

2.3 Security Model for Broadcast Encryption

A broadcast encryption scheme is said to be secure if given any $S$, the subscribers not in $S$ as well as the non-subscribers are not able to extract the message from its cipher-text, meant for the subscribers in $S$, even through collusion. Formally, the security can be defined using the following game between a challenger $A$ and an attacker $B$:

- $\text{Setup}$: $A$ runs $\text{Setup}(\lambda, n)$ to generate the public parameters which it passes onto $B$.

- $\text{Query Phase 1}$: $B$ adaptively queries about the secret keys of subscribers $i_1, i_2, \ldots, i_l$ and $A$ responds with the keys $K_{i_1}, K_{i_2}, \ldots, K_{i_l}$.

- $\text{Challenge}$: $B$ decides on a set $S^* \subseteq \{1, 2, \ldots, n\} \setminus \{i_1, i_2, \ldots, i_l\}$ of subscribers it wants to attack. It chooses a pair of distinct messages $(M_0, M_1)$ and gives it to $A$ along with $S^*$. $A$ chooses a random $b \in \{0, 1\}$ and runs $\text{Encrypt}(S^*, PP, M_b)$ to obtain the cipher-text $C^*$ which it gives to $B$.

- $\text{Query Phase 2}$: $B$ continues to adaptively query about the secret keys of other subscribers $i_{l+1}, i_{l+2}, \ldots, i_{l+m}$, who are not in $S^*$, and gets the keys $K_{i_{l+1}}, K_{i_{l+2}}, \ldots, K_{i_{l+m}}$. 

12-2
• **Guess:** \( \mathcal{B} \) guesses \( b' \in \{0, 1\} \) for \( b \) and wins the game if \( b = b' \).

The broadcast encryption scheme is secure against CPA if for all attacks

\[
Pr(b = b') = \frac{1}{2} + \epsilon(\lambda)
\]

where \( \epsilon(\lambda) \) is a negligible function in \( \lambda \).

A weaker notion of security called ‘static security against CPA’, can be defined where the set \( S^* \) to be attacked is known to the challenger before Query Phase 1 starts. Then the following game is played between the attacker \( \mathcal{A} \) and challenger \( \mathcal{B} \):

• **Init:** \( \mathcal{B} \) declares the set \( S^* \subseteq \{1, 2, \ldots, n\} \) of subscribers it wants to attack.

• **Setup:** \( \mathcal{A} \) runs \( \text{Setup}(\lambda, n) \) to generate the public parameters which it passes onto \( \mathcal{B} \).

• **Query Phase 1:** \( \mathcal{B} \) adaptively queries about the secret keys of subscribers \( i_1, i_2, \ldots, i_t \), who are not in \( S^* \), and \( \mathcal{A} \) responds with the keys \( K_{i_1}, K_{i_2}, \ldots, K_{i_t} \).

• **Challenge:** \( \mathcal{B} \) chooses a pair of distinct messages \( (M_0, M_1) \) and gives it to \( \mathcal{A} \). \( \mathcal{A} \) chooses a random \( b \in \{0, 1\} \) and runs \( \text{Encrypt}(S^*, PP, M_b) \) to obtain the cipher-text \( C^* \) which it gives to \( \mathcal{B} \).

• **Query Phase 2:** \( \mathcal{B} \) continues to adaptively query about the secret keys of other subscribers \( i_{t+1}, i_{t+2}, \ldots, i_{t+m} \), who are not in \( S^* \), and gets the keys \( K_{i_{t+1}}, K_{i_{t+2}}, \ldots, K_{i_{t+m}} \).

• **Guess:** \( \mathcal{B} \) guesses \( b' \in \{0, 1\} \) for \( b \) and wins the game if \( b = b' \).

The broadcast encryption scheme has ‘static’ security against CPA if for all attacks

\[
Pr(b = b') = \frac{1}{2} + \epsilon(\lambda)
\]

where \( \epsilon(\lambda) \) is a negligible function in \( \lambda \).

### 3 Boneh-Gentry-Waters (BGW) Scheme (’05)

Broadcast encryption was first formally discussed in [FN93] and a \( t \)-user collusion resistant scheme was proposed in the same. The scheme suffers from one main drawback – the cipher-text size is of the order \( O(t(\log t)^2 \log n) \), which increases with increasing \( t \) and \( n \).

The BGW scheme proposed in [BGW05] offers reduced cipher-text size, depending only on security parameter \( \lambda \). Moreover it proves to be fully collusion resistant, which is definitely secure than any \( t \)-user collusion resistant scheme.
3.1 Algorithms for BGW Scheme

We present a simplified version of the BGW scheme consisting of the following algorithms:

- **Setup**($\lambda, n$): Fix a prime number $p$ of order $\lambda$. Choose a bilinear group $G$ with generator $g$ and target group $G_T$, both of order $p$. Pick $\alpha \in \mathbb{Z}_p$, random $r_1, r_2, \ldots, r_n \in \mathbb{Z}_p$ and $u_1, u_2, \ldots, u_n \in G$. Consider a function $F(S) = \prod_{i \in S} u_i$ where $S \subseteq \{1, 2, \ldots, n\}$. Define public parameters as $PP = (e(g, g)^\alpha, u_1, u_2, \ldots, u_n)$ and the secret key for subscriber $i$ as $K_i = (K_{i,1}, K_{i,2}, u_{i,1}^{r_i}, \ldots, u_{i,1}^{r_{i-1}}, u_{i,1}^{r_{i+1}}, \ldots, u_{n,1}^{r_i})$ with $K_{i,1} = g^\alpha u_{i,1}^{r_i}, K_{i,2} = g^{r_i}$.

- **Encrypt**($S, PP, M$): Pick a random $s \in \mathbb{Z}_p$. Define the cipher-text for message $M \in G$ as $C = (C_1, C_2, C_3)$ where $C_1 = Me(g, g)^{as}, C_2 = g^a, C_3 = F(S)^a$, $S$ being the set of subscribers to which $M$ is communicated.

- **Decrypt**($S, i, K_i, C$): If $i \in S$, $M$ can be retrieved from $C$ as follows:
  
  Dividing $e(K_{i,2}, C_3) = e(g, \prod_{j \in S \setminus \{i\}} u_j)^{r_i}$ by $e(\prod_{j \in S \setminus \{i\}} u_j, C_2) = e(g, \prod_{j \in S \setminus \{i\}} u_j)^{r_i}$ gives $e(g, u_i)^{r_i}$, which can be divided from $e(K_{i,1}, C_2) = e(g, g)^{as} e(g, u_i)^{r_i}$ to get the blinding factor for the message, $e(g, g)^{as}$. The blinding factor can now be eliminated from $C_1 = Me(g, g)^{as}$ to get $M$.

One may note that $r_i$’s cannot be a part of $PP$ as then $e(g, g)^{r_i} = e(C_2, u_i^{r_i})$ can be computed for all $i$ and the blinding factor $e(g, g)^{as}$ can be determined by all subscribers.

3.2 Security of BGW Scheme

We describe a new hard problem called the $q$-BDHE assumption:

Consider a prime-order bilinear group $G$ with generator $g$ and target group $G_T$. Let $h \in G$ and $a, q \in \mathbb{Z}_p$. Choose $b \in \{0, 1\}$ randomly and define $T = e(g, h)^a q^b$ if $b = 0$, else make $T$ equal to some random element in $G_T$. Then given the groups $G, G_T$ and the values $g, h, g^a, \ldots, g^{aq}, g^{aq+2}, \ldots, g^{aq+b}, T$, it is hard to determine $b$.

It can be shown that the BGW scheme with $n$ subscribers possesses ‘static’ security if the $n$-BDH problem is hard. The proof of security will be given in the next scribe-note.

References
