

# Broadcast Encryption

\*\*\*

## Algorithmic Research Ltd.



Copyright 1995 Algorithmic Research, Ltd.  
Product names are trademarks of respective companies.

compress **1**

# Broadcast Encryption

- We present several schemes that allow a center to broadcast a secret to any subset of privileged users out of a universe of size  $n$ , such that coalitions of  $k$  users, not in the privileged set, cannot learn the secret.

# Broadcast Encryption

- A scheme is called **k-resilient** if it is resilient to any subset of size  $k$  out of the universe  $U$ .
- We also deal with random coalitions : A scheme is called **(k,p)-random-resilient** if with probability at least  $1-p$  the scheme is resilient to a random set of size  $k$ .

# Broadcast Encryption

- The resources we attempt to optimize are:
- The number of transmissions sent by the center in order to create a common secret for the privileged subset of users.
- The number of keys associated with each user. Since the user may be weak, e.g. smartcard, this should be minimized.
- The computation effort of each user.

## Broadcast Encryption

- Tradeoff between the two resources : **user's memory** and **transmission length**.
- The transmission length must be at least  **$n$**  (the total number of users).

## The Basic Scheme

- For every subset  $B$  of at most  $k$  users, define a key  $K_B$  and give the key to every user  $x$  not in  $B$ .
- The common key shared by the privileged set  $T$  is the XOR of all keys  $K_B$  for subsets  $B$  in  $U-T$ .
- Every coalition  $S$  of at most  $k$  users in  $U-T$  will all be missing the key  $K_S \Rightarrow$  unable to compute the common key of  $T$ .

## The Basic Scheme

- In the 1-resilient version of the basic scheme every user is required to store  $n$  keys.
- This can be reduced to  $\log n$  keys per user if the keys are pseudo-randomly generated from a common seed.
- Let  $f$  be a pseudo-random generator (the length of  $f(s)$  is twice the length of  $s$ )

## The Basic Scheme

- Associate the  $n$  users with the leaves of a balanced binary tree.
- The root is labeled with a common seed  $s$ .
- Other vertices are labeled as follows:
- Apply  $f$  to the root's label and take the left half of  $f(s)$  to be the label of the root of the left sub-tree while the right half is the label of the right sub-tree.



# Broadcast Encryption

- How to convert 1-resilient schemes to  $k$ -resilient schemes:
- Consider a family of functions:
- $f_1, \dots, f_l: U \rightarrow \{1, \dots, m\}$  with the following property: For every subset  $S$  of  $U$  ( $S$  of size  $k$ ) there exists some  $1 \leq i \leq l$  such that for all  $x, y$  in  $S$  :  $f_i(x)$  and  $f_i(y)$  are different. (perfect hash function)

## k-Resilient Schemes

- Such a family can be used to obtain a  $k$ -resilient scheme from a 1-resilient scheme.
- For every  $i=1,\dots,l$  and  $j=1,\dots,m$  use an independent 1-resilient scheme  $R(i,j)$  :
- Every user  $x$  in  $U$  receives the keys associated with the schemes  $R(i,f_i(x))$  for all  $i=1,\dots,l$ .
- In order to send a secret message  $M$  to a subset  $T$  of  $U$ , the center chooses random strings  $M_1,\dots,M_l$  such that their XOR is  $M$ .

## **k-Resilient Schemes**

- The center broadcasts for all  $i=1,\dots,l$  and  $j=1,\dots,m$  the message  $M_i$  to the privileged subset  $\{x \text{ in } T: f_i(x)=j\}$  using the scheme  $R(i,j)$ .
- Every user  $x$  in  $T$  can obtain all messages  $M_1,\dots,M_l$  and by Xoring them get  $M$ .
- Due to the perfect hash property of  $f_1,\dots,f_l$ , every subset  $S$  (of size  $k$ ) will be missing some  $M_i$ .

## k-Resilient Schemes

- The length of the transmission is  $km$  times the length of the transmission needed in the 1-resilient scheme.
- The number of keys, each user should store, is  $k w$ , where  $w$  is the number of keys needed in the 1-resilient scheme.

## k-Resilient Schemes

- If we set appropriate values for  $m$  and  $l$  we obtain:
- A  $k$ -resilient scheme that requires each user to store  $O(wk \log n)$  keys and the center to broadcast  $O(k^3 \log n)$  messages.
- A  $(k, p)$ -random-resilient scheme that requires each user to store  $O(w \log(1/p))$  keys and the center to broadcast  $O(k^2 \log(1/p))$  messages.

## **k-Resilient Schemes**

- **We have more efficient schemes:**
- A  $k$ -resilient scheme that requires each user to store  $O(w k \log k \log n)$  keys and the center to broadcast  $O(k^2 \log^2 k \log n)$  messages.
- A  $(k,p)$ -random-resilient scheme that requires each user to store  $O(w \log k \log(1/p))$  keys and the center to broadcast  $O(k \log^2 k \log(1/p))$  messages.

# Traitor Tracing Schemes

- Cryptographic schemes that help trace the source of leaks when sensitive proprietary data is made available to a large set of users. This is particularly important for pay-TV, where the data should be accessible only to authorized users.
- The schemes combine easily with and complement the Broadcast Encryption schemes of Fiat and Naor.

# Components of Piracy Protection

- Fighting piracy generally has the following components:
- Identify that piracy is occurring and prevent the transmission of information to pirate users, while harming no legitimate users.
- Take legal measures against the source of such piracy, supply legal evidence of the pirate identity.



# Performance Parameters

- Any solution must be considered in the light of certain performance parameters:
- What are the memory and computation requirements per authorized user ?
- What are the memory and computation requirements for the broadcast center ?
- What is the communication overhead ?

# Traitor Tracing Schemes

- A traitor tracing message consists of many pairs of: enabling block, cipher block.
- The cipher block is the symmetric encryption of the actual data, under some secret random key  $S$ .
- The enabling block allows authorized users to obtain  $S$ . It consists of encrypted values under some keys.
- Every user can compute  $S$  by decrypting those values for which he has the keys and then computing the actual key from these values. The computation on the user end is the XOR of all values he has been able to decrypt.

## Objectives of the Scheme

- Traitors may conspire and give unauthorized users a subset of their keys so that the unauthorized users will be able to compute the actual key.
- The goal of the designer of the system is to assign keys to users such that when a pirate decoder is captured it should be possible to detect at least one traitor, subject to the limitation that there are at most  $k$  traitors.

## Example of a Traitor Tracing Scheme

- Suppose there is a requirement to create a scheme for up to 1,000,000,000 users of a digital video channel, so that if there are at most 1000 traitors, the probability for false identification is  $< 2^{-10}$ .
- Allocating 5% of the compressed MPEG II digital video channel to the traitor tracing scheme allows to change keys every minute (a new enabling block every minute).