Broadcast Encryption:
Recent progress and Open problem

Dan Boneh
Stanford University

(1 hour)
Public-key Broadcast Encryption for Stateless Receivers  

[Fiat-Naor 1993]

Encrypt to arbitrary subsets $S$:

$c \leftarrow E(pk, S, m)$

$S \subseteq \{1, \ldots, n\}$

Security goal (informal):

Full collusion resistance: secure even if all users in $S^c$ collude
Broadcast Encryption

Public-key BE system:

- \( \text{Setup}(n) \rightarrow \text{pub. key } pk, \text{ master sec. key } msk \)

- \( \text{KeyGen}(msk, j) \rightarrow d_j \) (private key for user \( j \))

- \( \text{Enc}(pk, S) \rightarrow ct, k \)
  
  \( k \) used to encrypt \( msg \) for users \( S \subseteq \{1, ..., n\} \)

- \( \text{Dec}(pk, d_j, S, ct) \): If \( j \in S \), output \( k \)

Broadcast contains \( ( [S], ct, E_{\text{SYM}}(k, msg) ) \)
Semantic security when users collude (static adversary)

**Def:** \( \text{Adv}[A] = | \text{Pr}[b' \text{ is correct}] - \frac{1}{2} | \)

**Security:** \( \forall \) poly-time A: \( \text{Adv}[A] \) is negligible
Broadcast systems are everywhere

File sharing in **encrypted file systems** (e.g. EFS):

- **ACL**
  - File
  - `d_{Bob}`
  - `d_{Ned}`

Encrypted mail system:

- **Recipients**
  - `d_{Bob}`
  - `d_{Ned}`

Social networks: privately send message to a group
Constructions

<table>
<thead>
<tr>
<th>ct</th>
<th>sk</th>
<th>pk</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(</td>
<td>S</td>
<td>)$</td>
</tr>
</tbody>
</table>

The trivial system:

Revocation schemes:

- $O(n-|S|)$
- $O(\log n)$
- $O(1)$

[NNL, HS, GST, LSW, DPP, ...]

Can we have $O(1)$ size ciphertext for all sets $S$??

The BGW system:

- $O(1)$
- $O(1)$
- $O(n)$

[B-Gentry-Waters’05]
The BGW system

Setup(n): \( g \leftarrow G, \ \alpha, \ \text{msk} \leftarrow Z_q, \ \text{def:} \ g_k = g(\alpha^k) \)

\[
\begin{align*}
\text{pk} &= (g, \ g_1, \ g_2, \ldots, \ g_n, \ g_{n+2}, \ldots, \ g_{2n}, \ v = g^{\text{msk}}) \in G^{2n+1}
\end{align*}
\]

KeyGen( msk, j) \rightarrow \ d_j = (g_j)^{\text{msk}} \in G

Enc(pk, S): \ t \leftarrow Z_q

\[
\begin{align*}
\text{ct} &= \left( g^t, \ (v \cdot \prod_{j \in S} g_{n+1-j})^t \right), \quad \text{key} = e(g_n, g_1)^t
\end{align*}
\]
Decryption

\textbf{Decrypt}(ct, S, d_u, pk): \quad ct = (c_0, c_1, c_2)

\[ e(g_u, c_1) / e(d_u \cdot \prod_{j \in S, j \neq u} g_{n+1-j+u}, c_0) = e(g_n, g_1)^t \]

- If \( \Delta S \) small, can easily update precomputation

\[ key \]

\[ \text{precompute} \]
Security

**Thm:** BGW is statically secure for \( n \) users in a bilinear group where \( n \text{-DDHE} \) assumption holds.

**n-DDHE:** for rand. \( g,h \leftarrow G, \alpha \leftarrow Z_q, R \leftarrow G_2 \) :

\[
\left[ h, g, g^\alpha, g^{(\alpha^2)}, \ldots, g^{(\alpha^n)}, g^{(\alpha^{n+2})}, \ldots, g^{(\alpha^{2n})}, e(g,h)^{(\alpha^{n+1})}\right] \approx_p \left[ h, g, g^\alpha, g^{(\alpha^2)}, \ldots, g^{(\alpha^n)}, g^{(\alpha^{n+2})}, \ldots, g^{(\alpha^{2n})}, R \right]
\]
Extensions, Variations, Improvements

Adaptive security: \[GW’10, \text{ PPSS’12, …} \]
- Adversary can adaptively select what keys to request

Identity-based: \[SF’07, D’07, GW’10, \ldots \]
- Smaller public key size: \(|pk| = O(\text{ maximal } |S|)\)
  \[\Rightarrow \text{ Set of all users can be } \{0, 1, 2, 3, \ldots, 2^{256}\} \]

Chosen ciphertext secure: \[BGW’05, \text{ PPSS’12, …}\]

Trace & revoke: \[BW’06\]
**k-linear maps: a recent breakthrough**
S. Garg, C. Gentry, S. Halevi

**Properties:** (informal)

- Representation of \( g \in G \) is \( O(k) \) bits
- Better than \( k \)-linear map: **gradation**

\[
\begin{align*}
e_1 &: G \times G \rightarrow G_2 \\
e_2 &: G \times G_2 \rightarrow G_3 \\
& \quad \vdots \\
e_k &: G \times G_k \rightarrow G_{k+1} \\
\end{align*}
\]

For our purposes:

\[
\begin{align*}
e_k &: G \times \cdots \times G \rightarrow G_k \\
e &: G_k \times G_k \rightarrow G_{2k}
\end{align*}
\]
BGW using (log n)-linear map

Recall: **BGW Setup**\( (n): \) \( g \leftarrow G, \ \alpha, \ \text{msk} \leftarrow Z_q. \) \( \text{pk}: \)

\[
g, \ g^\alpha, g^{(\alpha^2)}, ..., g^{(\alpha^n)}, \ g^{(\alpha^{n+2})}, ..., g^{(\alpha^{2n})}, \ v = g^\text{msk}
\]

Suppose: \( \text{e}_k: G \times \cdots \times G \rightarrow G^k \); \( e: G^k \times G^k \rightarrow G_{2^k} \)

Set \( \text{pk} \) as: \( (\text{#users} \approx 2^{k-1}) \)

\[
g, \ g^\alpha, g^{(\alpha^2)}, g^{(\alpha^4)}..., g^{(\alpha^{2^{2k}})}, g^{(\alpha^{2^{2k+1}})}, \ v = g^\text{msk}
\]

Using \( \text{e}_k: \) can build all needed elements in \( \text{pk} \)

but for rand. \( h \in G \) cannot build \( e(g,...,g,h)^{(\alpha^{2^{2k-1}})} \in G_{2^k} \)
BGW using \((\log n)\)-linear map

|            | \(|ct|\)    | \(|sk|\)    | \(|pk|\)    |
|------------|-------------|-------------|-------------|
| **Bilinear BGW:** [B-Gentry-Waters’05] | \(O(1)\)    | \(O(1)\)    | \(O(n)\)    |
| **(log n)-linear BGW:** | \(O(\log n)\) | \(O(\log n)\) | \(O(\log^2 n)\) |

Open questions:

- Same parameters without \(k\)-linear maps ??
- \(O(1)\) size \(ct\) from standard lattice assumptions (LWE) ??
Distributed Broadcast Encryption?
Distributed Broadcast Encryption?

(users generate keys for themselves)

Facebook

Alice

pk_a

sk_a

Bob

pk_b

sk_b

Charlie

pk_c

sk_c

David

pk_d

sk_d

Ethan

pk_e

sk_e

Sender

post

[ [S], ct, E_{SYM}(k,msg) ]
Distributed Broadcast Encryption?

The trivial system is distributed, but $|ct| = O(|S|)$

Goal: $|ct| = \text{sub-linear}(|S|)$

Sender $\rightarrow$ post $\rightarrow$ [ [S], ct, $E_{\text{SYM}}(k,\text{msg})$ ]
An approach: n-way DH [J’00, BS’03, GGH’12]

Def: an n-way DH scheme is a pair of det. algorithms (F, G)

\[ F: R \rightarrow Y, \quad G: R \times Y^{n-1} \rightarrow K \]

Correctness: \( \forall r_1, \ldots, r_n: G( r_i, F(r_1), \ldots, F(r_i), \ldots, F(r_n) ) = K(r_1, \ldots, r_n) \)

Security: given \( F(r_1), \ldots, F(r_n) \): \( K(r_1, \ldots, r_n) \approx_p \text{ uniform}(K) \)
n-way DH: example [J’00, BS’03, GGH’12]

Example (Joux’00): $e_{n-1}: G \times \cdots \times G \rightarrow G_{n-1}$

$F(r) := g^r$ ; shared key = $e_{n-1}(g, \ldots, g)^{r_1 r_2 \cdots r_n}$

$G(r_1, g^{r_2}, \ldots, g^{r_n}) := e(g^{r_2}, \ldots, g^{r_n})^{r_1}$
n-way DH $\Rightarrow$ distrib. BE

KeyGen(i):

\[
\text{sk}_i \leftarrow \mathbb{R}, \quad \text{pk}_i = F(\text{sk}_i) = g^{\text{sk}_i}
\]

Enc(S, \{pk_i\}_{i \in S}):

choose \( r \leftarrow \mathbb{R} \)

output \( ct = F(r) = g^r, \quad \text{key} = G_{|S|+1}(r, \{pk_i\}_{i \in S}) \)

Problem: bit-size of \( g^r \) is \( O(n) \)

Is there a distributed BE where \(|ct|\) is sub-linear(|S|)?
Private broadcast encryption
Privacy violations in many implementations:

- Windows EFS.
- BCC in S/MIME: Outlook, Thunderbird, …
- BCC in PGP

Reason: CT header reveals recipient set.

Solution: broadcast ciphertext should conceal recipient set $S$ (but not its size)

⇒ private broadcast encryption
Private Broadcast Encryption \[\text{[BBW’ 06]}\]

**Definition:** Alg. A \(\epsilon\)-breaks BE sem. sec. if \(\Pr[b=b'] > \frac{1}{2} + \epsilon\)
Constructions [BBW’04, LPQ’12]

- The trivial system (with anon. pub-key enc.)
  ciphertext size: $|S| \times (\text{sec. param.})$, $O(|S|)$ decryption time

- Best known construction: [BBW’04]
  ciphertext size: $|S| \times (\text{sec. param.})$, $O(\log|S|)$ dec. time

Open: private BE of ct. size sub-linear$(|S|) \times (\text{sec. param.}) + |S|$

- Fazio-Perera’12: NNL-like system, but only outsider privacy
Summary so far ...

Many open problems in broadcast encryption:

• \(O(\log n)\) size ciphertext & secret keys from LWE?

• \(O(\log n)\) size ct, sk, and pub-key w/o k-linear maps?

• Sub-linear (fully) private broadcast encryption?
  note: (linear) private BE \(\Rightarrow\) traitor tracing \[\text{[BSW'05]}\]

• Distributed BE with sub-linear ciphertext?
Part II: Traitor Tracing
What if secret key $K_i$ is exposed?
- Goal: Trace pirate decoder $D$ to key $K_u$
  Then kill user $u$ (or revoke his key)
Tracing Traitors

- **Setup** (n): outputs private keys $K_1, \ldots, K_n$ public-key $PK$ and tracer key $TK$.

  User $i$ gets private key $K_i$

- **Encrypt** ($PK$, $M$) $\rightarrow$ ciphertext $C$

- **Decrypt** ($K_i$, $C$) $\rightarrow$ message $M$

- **Trace** $^D$ ($TK$) $\rightarrow$ $i \in \{1,\ldots,n\}$
  - Outputs index of at least one key used to build $D$
  - $D$ is a stateless black-box pirate decoder.
Why black-box tracing? [BF’ 99]

D: may contain unrecognized keys, is obfuscated, or tamper resistant.

All we know:

\[ \Pr[ M \leftarrow^R G, \ C \leftarrow^R \text{Encrypt}(PK, M) : \ D(C)=M ] > 1-\varepsilon \]
Formally: Secure TT systems

(1) Semantically secure, and  
(2) Traceable:

\[ \text{Adversary wins if:} \quad (1) \quad \Pr[D(C)=M] > 1-\varepsilon, \quad \text{and} \]
\[ (2) \quad i \notin S \]
Brute Force System

• **Setup** (n): Generate n PKE pairs \((PK_i, K_i)\)
  Output private keys \(K_1, ..., K_n\)
  \(PK \leftarrow (PK_1, ..., PK_n)\), \(TK \leftarrow PK\).

• **Encrypt** (PK, M): \(C \leftarrow (E_{PK_1}(M), ..., E_{PK_n}(M))\)

• Tracing: next slide.
Trace\textsuperscript{D}(PK): [BF99, NNL00, KY02]

For \( i = 1, \ldots, n+1 \) define for \( M \leftarrow^R G \):

\[
p_i := \Pr[ D( E_{PK_1}(\perp), \ldots, E_{PK_{i-1}}(\perp), E_{PK_i}(M), \ldots, E_{PK_n}(M) ) = M ]
\]

Then: \( p_1 > 1-\varepsilon \); \( p_{n+1} \approx 0 \)

\[
1-\varepsilon = |p_{n+1} - p_1| = \sum_{i=1}^{n} |p_{i+1} - p_i| \leq \sum_{i=1}^{n} |p_{i+1} - p_i|
\]

\[\Rightarrow \text{ exists } i \in \{1, \ldots, n\} \text{ s.t. } |p_{i+1} - p_i| \geq (1-\varepsilon)/n\]

\[\Rightarrow \text{ user } i \text{ must be one of the pirates.}\]
Security Theorem

Tracing algorithm estimates: \[ |\hat{p}_i - p_i| < (1-\epsilon)/4n \]

\[ \Rightarrow \text{Need } O(n^2) \text{ samples per } p_i. \quad (D - \text{stateless}) \]

\[ \Rightarrow \text{Cubic time tracing} \]

(can be improved to quadratic in $|S|$)

\textbf{Thm:}

underlying PKE system is semantically secure

\[ \Rightarrow \]

no eff. adv wins tracing game with non-neg adv.
Abstracting the Idea [BSW’ 06]

Properties needed:

- For $i = 1, \ldots, n+1$ can encrypt M so:

  - Users cannot decrypt
  - Users can decrypt

- Without $K_i$ adversary cannot distinguish:
  \[ \text{Enc}(i, PK, M) \quad \text{from} \quad \text{Enc}(i+1, PK, M) \]
Private Linear Broadcast Enc (PLBE)

– **Setup(n):** outputs private keys \( K_1, ..., K_n \) and public-key PK

– **Encrypt( u, PK, M):**
  - encrypt M for users \( \{u, u+1, ..., n\} \)
  - output ciphertext CT.

– **Decrypt(CT, j, K_j, PK):** If \( j \geq u \), output M

• **Trace-Encrypt(PK,M) := Encrypt( 1, PK, M)**
Security definition

Message hiding: given all private keys:

\[ \text{Encrypt}(n+1, M, PK) \approx_p \text{Encrypt}(n+1, \bot, PK) \]

Index hiding: for \( u = 1, \ldots, n \):

\[ b \leftarrow \{0,1\} \]

\[ C_* \leftarrow \text{Enc}(u+b, PK, m) \]

\[ b' \in \{0,1\} \]
Known Results

Thm: Secure PLBE $\Rightarrow$ Secure TT
Same size CT and priv-keys
(black-box and publicly traceable)

Best pairing-based PLBE system: [BSW’06]
CT-size = $O(\sqrt{n})$ ; priv-key size = $O(1)$
enc-time = $O(\sqrt{n})$ ; dec-time = $O(1)$

Improvements using k-linear maps?