Theory and Applications of Outsider Anonymity in Broadcast Encryption

Dissertation Defense

Irippuge Deshan Milinda Perera

Graduate Center of CUNY
iperera@gradcenter.cuny.edu
Outline

① Summary of Contributions
② Outsider Anonymous Broadcast Encryption
③ Broadcast Steganography
④ Oblivious Group Storage
① Summary of Contributions
Anonymity in Broadcast Encryption

Broadcast Encryption (BE)
Outsider-Anonymous Broadcast Encryption (oABE)
Anonymous Broadcast Encryption (AnoBE)

1. Broadcast Steganography (BS)
2. Oblivious Group Storage (OGS)
② Outsider-Anonymous Broadcast Encryption
Broadcast Encryption

- Proposed by Fiat and Naor [FiNa93]
- Secure broadcast of messages to arbitrary subsets of users

- **Flavors:**
  - Private-Key / Public-Key
  - Stateful / Stateless
    - Are the users required to update keys?
$1^\lambda, N \xrightarrow{} \text{Setup} \xrightarrow{} \text{MSK, MPK}$
BE – Setting

$1^\lambda, N \xrightarrow{} \text{Setup} \xrightarrow{} \text{MSK, MPK}$

$\text{MSK, MPK} \xrightarrow{} \text{KeyGen} \xrightarrow{} \text{SK}$
BE – Setting

MPK

Encrypt

Message

Payload

Ciphertext
BE – Setting

Message → Encrypt → Payload

Ciphertext

Payload → Decrypt → Message or X

Ciphertext

MPK

SK

Payload
BE – Drawbacks

- Does not provide recipient anonymity!
- Set of receivers sent as a part of broadcast content
Anonymous Broadcast Encryption

- **Proposed by Barth et. al [BBW06]**
- **Goal:** Completely hide the identities of recipients
- **Idea:** Remove the header (H) altogether

The broadcast ciphertext

The broadcast ciphertext

- **Payload**
- **Body (B)**

The broadcast ciphertext

- **Header (H)**
AnoBE – Security Models

- **Purposes:**
  - Confidentiality of the messages
  - Privacy of the recipients w.r.t. any user

- **Models:**
  - AnoBE-IND-CPA (only corruption queries)
  - AnoBE-IND-CCA (corruption and decryption queries)
AnoBE – Drawbacks

- Ciphertext length is linear
  - In the number of receivers [KiSa12]

- In some applications, full anonymity is overkill
  - Content may give recipient set away
    » Example: Alice goes to jury duty
Outsider-Anonymous Broadcast Encryption

- **Goal:** Define a less stringent and efficient notion of anonymity for BE
- **Idea:** Hide the receiver IDs only from the outsiders

- Lies in between regular BE and AnoBE
- Trades some degree of anonymity for efficiency
- Allows constructions with sublinear ciphertext length

- Published in PKC 2012 [FaPe12]
  - Formal model + constructions + proofs of security
oABE – Security Models

✦ Purposes:
  ✦ Confidentiality of the messages
  ✦ Privacy of the recipients w.r.t. outsiders

✦ Models:
  ✦ AnoBE-IND-CPA (only corruption queries)
  ✦ AnoBE-IND-CCA (corruption and decryption queries)
oABE – Constructions

- Encrypt(S, m):
Encrypt(S, m):

1. Group users in S into S’, a set of disjoint subsets (using public-key Subset Cover Framework [DoFa02])
   » |S’| is sublinear in |S|
oABE – Constructions

• Encrypt(S, m):

1. Group users in S into S’, a set of disjoint subsets (using public-key Subset Cover Framework [DoFa02])
   » |S’| is sublinear in |S|

2. Generate a ciphertext $c_i$ for each $s_i$ in S’
   (using anonymous IBE)

\[ \begin{array}{ccc}
C_1 & \ldots & C_l \\
\end{array} \]
Encrypt(S, m):

1. Group users in $S$ into $S'$, a set of disjoint subsets (using public-key Subset Cover Framework [DoFa02])
   » $|S'|$ is sublinear in $|S|$
2. Generate a ciphertext $c_i$ for each $s_i$ in $S'$ (using anonymous IBE)
3. Attach a tag $t_i$ to each $c_i$ (for efficient decryption at the receivers)
Encrypt(S, m):

1. Group users in S into $S'$, a set of disjoint subsets (using public-key Subset Cover Framework [DoFa02])
   » $|S'|$ is sublinear in $|S|$
2. Generate a ciphertext $c_i$ for each $s_i$ in $S'$ (using anonymous IBE)
3. Attach a tag $t_i$ to each $c_i$ (for efficient decryption at the receivers)
4. Bundle all $(t_i, c_i)$ components using one-time signature
Comparison of AnoBe/oABE Constructions

| Scheme     | | PK | | sk | | c | | Security Model | | Anonymity |
|------------|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| BBW06      | O(N)         | O(1)            | O(N-r)          | Static A, RO    | Full            |
| LPQ12      | O(N)         | O(1)            | O(N-r)          | Adaptive A, Standard | Full            |
| FaPe12a    | O(N)         | O(log N)        | O(r log (N/r))  | Adaptive A, Standard | Outsider        |
| FaPe12b    | O(N log N)   | O(N)            | O(r)            | Adaptive A, Standard | Outsider        |

- **N**: total number of users in the system, **r**: number of revoked users in \( c \)
- Each is CCA secure and requires only one decryption attempt
③ Broadcast Steganography
Steganography

- Proposed by Simmons [Sim83]
- **Example:** Alice and Bob are in jail
- Hides the very *existence* of the communication in addition to the meaning of the communication

Ciphertext

Stegotext
Alice is an Activist
Alice is an Activist – without Crypto
Alice is an Activist – without Crypto
Alice is an Activist – without Crypto
Alice is an Activist – without Crypto

Take that down!
Alice is an Activist – with Encryption
Alice is an Activist – with Encryption
Alice is an Activist – with Encryption
Alice is an Activist – with Encryption
Alice is an Activist – with Encryption

Take that down!
Alice is an Activist – with Steganography
Alice is an Activist – with Steganography
Alice is an Activist – with Steganography
Alice is an Activist – with Steganography

Oh cute!
Alice is an Activist – with Steganography

Oh cute!

Take that down!
Alice is an Activist – with **Broadcast** Steganography [FNP14]
Alice is an Activist – with **Broadcast Steganography** [FNP14]
Alice is an Activist – with **Broadcast Steganography** [FNP14]
Alice is an Activist – with Public-Key Broadcast Steganography [FNP14]
Broadcast Steganography

- **Goal:** Extend steganography to multi-recipient setting
- **Idea:** Use outsider anonymity to allow dynamic revocations

- Published in CT-RSA 2014 [FNP14]
  - Formal model + constructions + proofs of security
BS – Setting

$1^\lambda, N \rightarrow \text{Setup} \rightarrow \text{MSK, MPK}$
BS – Setting

$1^\lambda, N \rightarrow \text{Setup} \rightarrow \text{MSK, MPK}$

$\text{MSK, MPK} \rightarrow \text{KeyGen} \rightarrow \text{SK}$
BS – Setting

MPK → Message → History → Encode → Stegotext
BS – Setting

MPK

Message

History

Encode

Stegotext

MPK

SK

Stegotext

Decode

Message or X
BS – Security Models

✦ Purpose:
  ✦ Indistinguishability of the communication

✦ Models:
  ✦ BS-IND-CHA (only corruption queries)
  ✦ BS-IND-CCA (also decoding queries)
  ✦ BS-IND-PDR-CCA (more restricted decoding queries)
Toward BS Constructions

- Encrypt-then-Embed Paradigm [HLvA02, BaCa05]
Toward BS Constructions

- Encrypt-then-Embed Paradigm [HLvA02, BaCa05]

  BE ciphertexts have structure!

- BE ciphertexts have structure!
Toward BS Constructions

- Encrypt-then-Embed Paradigm [HLvA02, BaCa05]

- BE ciphertexts have structure!

- oABE already removes header ...
  ... but, its payload still has structure!
oABE with Pseudorandom Ciphertexts (oABE$)$

How to make oABE ciphertexts pseudorandom?

- pseudorandom group elements
- one-time signature
- AIBE ciphertexts
How to make oABE ciphertexts pseudorandom?

1. Replace the underlying AIBE with AIBE$ [AgBo09]
How to make oABE ciphertexts pseudorandom?

1. Replace the underlying AIBE with AIBE$ [AgBo09]
2. Apply an entropy smoothing hash to group elements
How to make oABE ciphertexts pseudorandom?

1. Replace the underlying AIBE with AIBE$ [AgBo09]
2. Apply an entropy smoothing hash to group elements
3. Replace one-time signature with a MAC (implemented via PRF)
How to make oABE ciphertexts pseudorandom?

1. Replace the underlying AIBE with AIBE\$ [AgBo09]
2. Apply an entropy smoothing hash to group elements
3. Replace one-time signature with a MAC (implemented via PRF)

Problem: How to embed the MAC key in c_i’s and still obtain CCA security?
How to make oABE ciphertexts pseudorandom?

1. Replace the underlying AIBE with AIBE$ [AgBo09]
2. Apply an entropy smoothing hash to group elements
3. Replace one-time signature with a MAC (implemented via PRF)

Problem: How to embed the MAC key in $c_i$’s and still obtain CCA security?

Solution: Construct an encapsulation mechanism [DoKa05, BoKa05] with pseudorandom commitments
BS Constructions

<table>
<thead>
<tr>
<th>Scheme</th>
<th></th>
<th>PK</th>
<th></th>
<th>sk</th>
<th></th>
<th>s</th>
<th>Security Model</th>
<th>Channel Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS-CHA</td>
<td>O(N)</td>
<td>O(log N)</td>
<td>O(r log (N/r))</td>
<td>Adaptive, Standard</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BS-PDR-CCA</td>
<td>O(N)</td>
<td>O(log N)</td>
<td>O(r log (N/r))</td>
<td>Adaptive, Standard</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BS-CCA</td>
<td>O(N)</td>
<td>O(log N)</td>
<td>O(r log (N/r))</td>
<td>Adaptive, Standard</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **N**: total number of users in the system, **r**: number of revoked users in s
- **Type 1 Channel**: A stateful probabilistic oracle whose state may depend on past outputs
- **Type 2 Channel**: An efficiently computable randomized function whose distribution does not depend on past outputs
④ Oblivious Group Storage
Cloud Storage

- Export all data to a cloud and access on-demand
- Pay-per-use, much cheaper than in-house infrastructure
- High availability, low latency
Security Issues with Cloud Storage

- Dishonest server can exploit stored data
- Encryption preserves data secrecy but not enough
  - Statistical attacks on access patterns reveal 80% of search queries to an encrypted email repository [IKK12]
Oblivious Random Access Machine (ORAM)

- Originally proposed to protect computer software from reverse engineering attacks [Gol87]
- Preserves data secrecy and access pattern obliviousness
  - One logical access $\rightarrow$ many real accesses
  - Requires periodic shuffling
- Mostly optimized for single-client setting
Sharing Oblivious Storage

- Four security requirements
  1. Server-side data secrecy (SDS)
  2. Server-side access pattern obliviousness (SAPO)
  3. Client-side data secrecy (CDS)
  4. Client-side access pattern obliviousness (CAPO)
Sharing Oblivious Storage

- Four security requirements
  1. Server-side data secrecy (SDS)
  2. Server-side access pattern obliviousness (SAPO)
  3. Client-side data secrecy (CDS)
  4. Client-side access pattern obliviousness (CAPO)

- **Stateful vs. Stateless** ORAM
  - Stateful ORAM → not suitable for multi-user setting
  - Stateless ORAM → no client-side security guarantees
Sharing Oblivious Storage

- Four security requirements
  1. Server-side data secrecy (SDS)
  2. Server-side access pattern obliviousness (SAPO)
  3. Client-side data secrecy (CDS)
  4. Client-side access pattern obliviousness (CAPO)

- Multi-User ORAM [JWQ14]
  - Provides CAPO but not CDS
  - Utilizes a sequence of proxies between clients and server
Sharing Oblivious Storage

- Four security requirements
  1. Server-side data secrecy (SDS)
  2. Server-side access pattern obliviousness (SAPO)
  3. Client-side data secrecy (CDS)
  4. Client-side access pattern obliviousness (CAPO)

- Our result additionally satisfies CDS
- Clients can now impose fine-grained access control over data

- We use oABE as an underlying building block
Alice is a Doctor

Third-party storage
Alice is a Doctor – with Stateless ORAM
Alice is a Doctor – with Stateless ORAM

Third-party storage

CDS: Client-side Data Secrecy
CAPO: Client-side Access Pattern Obliviousness
Alice is a Doctor – with Multi-User ORAM
Alice is a Doctor – with Multi-User ORAM

Hospital servers

Third-party storage
Alice is a Doctor – with Multi-User ORAM

Third-party storage

Hospital servers

No CDS

CDS: Client-side Data Secrecy
Alice is a Doctor – with Oblivious Group Storage [FNP15]
Alice is a Doctor – with Oblivious Group Storage [FNP15]
Alice is a Doctor – with Oblivious Group Storage [FNP15]
$1^\lambda, M, N \xrightarrow{} \text{Setup} \xrightarrow{} \text{Non-interactive}

\text{KeyGen} \xrightarrow{} \text{Non-interactive}

\text{MSK} \quad \text{MPK} \quad \text{AK}_1 \quad \ldots \quad \text{AK}_M

\text{ST}_0

\text{CK}
OGS – Setting

Server

Anonymizers

Client

Interactive

Write

Server

Anonymizers

MPK

Message Position CK
OGS – Setting

Server

ST<sub>t</sub>

Anonymizers

AK<sub>1,t</sub>

AK<sub>M,t</sub>

Client

Position

CK

Interactive

MPK

Server

ST<sub>t+1</sub>

Anonymizers

AK<sub>1,t+1</sub>

AK<sub>M,t+1</sub>

Client

Message

or X
OGS – Security Model

- Honest-but-curious (OBC) model

- Adversary is allowed to corrupt
  - Server
  - Clients
  - Anonymizers
    - At least one anonymizer should remain honest

- A game between a PPT adversary and a challenger
  - Adversary tries to distinguish between two read queries by two honest clients
  - Adversary wins the game if he can guess better than tossing a coin
OGS – Construction (intuition)

Security Guarantees:    CDS   CAPO   SDS   SAPO

Clients

Anonymizers

Server
OGS – Construction (intuition)

Security Guarantees: CDS CAPO SDS SAPO

Clients

Anonymizers

Server
OGS – Construction (intuition)

Security Guarantees: CDS CAPO SDS SAPO

ORAM Protocol

- \( x_l, y_l \in \mathbb{F}_q \setminus \{0\} \)
- \( p, m \in \mathbb{G}_q \)
- \( E((x_l, y_l), (p, m)) = (p^{x_l}, m^{y_l}) \)
- \((x_l, y_l)\) is used only once

Anonymizers

Server

Clients

\[ T = \log D \]
OGS – Construction (intuition)

Security Guarantees: CDS **CAPO** SDS SAPO

**M-ORAM [JWQ14] Protocol**

- $x_l^k, y_l^k, s_k \in F_q \setminus \{0\}$
- $x_l = \prod_{k=1}^{M} x_l^k, \ y_l = \prod_{k=1}^{M} y_l^k, \ w_i = \prod_{k=1}^{M} h(s_k, i)$

Clients

Anonymizers

Server
OGS – Construction (intuition)

Security Guarantees: CDS [CAPO] SDS SAPO

M-ORAM [JWQ14] Protocol
Write Algorithm (simplified)

\( p, m \)

\( w_1 \)

\( w_2 \)

\( \vdots \)

\( w_N \)

Clients

Anonymizers

Server
OGS – Construction (intuition)

Security Guarantees:

- CDS
- CAPO
- SDS
- SAPO

M-ORAM [JWQ14] Protocol

Write Algorithm (simplified)

Clients

\[ p, m \]
\[ r', r'' \leftarrow F_q \setminus \{0\} \]

Anonymizers

\[
\begin{align*}
&w_1 \\
&w_2 \\
&\vdots \\
&w_N
\end{align*}
\]

\[
\begin{align*}
&x_1^1, y_1^1 \\
&x_2^1, y_2^1 \\
&\vdots \\
&x_L^1, y_L^1 \\
&s_1
\end{align*}
\]

\[
\begin{align*}
&x_1^2, y_1^2 \\
&x_2^2, y_2^2 \\
&\vdots \\
&x_L^2, y_L^2 \\
&s_2
\end{align*}
\]

\[
\begin{align*}
&x_1^M, y_1^M \\
&x_2^M, y_2^M \\
&\vdots \\
&x_L^M, y_L^M \\
&s_M
\end{align*}
\]

Server
OGS – Construction (intuition)

Security Guarantees:  CDS  CAPO  SDS  SAPO

M-ORAM [JWQ14] Protocol
Write Algorithm (simplified)

\[ w_1, w_2, \ldots, w_N \]

Anonymizers

Server
OGS – Construction (intuition)

Security Guarantees: CDS CAPO SDS SAPO

M-ORAM [JWQ14] Protocol
Write Algorithm (simplified)

\[
(p^{r'w_1x_1^1/h(s_1, 1)}, m^{r''w_1y_1^1/h(s_1, 1)})
\]

\[
\begin{align*}
&x_1^1, y_1^1 \\
&x_2^1, y_2^1 \\
&\vdots \\
&s_1 \\
&x_L^1, y_L^1
\end{align*}
\]

\[
\begin{align*}
&x_1^2, y_1^2 \\
&x_2^2, y_2^2 \\
&\vdots \\
&s_2 \\
&x_L^2, y_L^2
\end{align*}
\]

\[
\begin{align*}
&x_1^M, y_1^M \\
&x_2^M, y_2^M \\
&\vdots \\
&s_M \\
&x_L^M, y_L^M
\end{align*}
\]

Clients

Anonymizers

Server
OGS – Construction (intuition)

Security Guarantees: CDS **CAPO** SDS SAPO

**M-ORAM** [JWQ14] Protocol

Write Algorithm (simplified)

\[(p^{r'w_1x_1^1x_1^2/(h(s_1,1)h(s_2,1))}, m^{r''w_1y_1^1y_1^2/(h(s_1,1)h(s_2,1))})\]

Anonymizers

Server

Clients
OGS – Construction (intuition)

Security Guarantees:  CDS  CAPO  SDS  SAPO

M-ORAM [JWQ14] Protocol
Write Algorithm (simplified)

\[
(p, m, r', r'') \quad (p'w_1^{x_1^1x_2^1...x_M^M/(h(s_1, 1)h(s_2, 1)...h(s_M, 1)), m, r''w_1^{y_1^1y_2^1...y_M^M/(h(s_1, 1)h(s_2, 1)...h(s_M, 1))})
\]
OGS – Construction (intuition)

Security Guarantees: CDS [CAPO] SDS SAPO

M-ORAM [JWQ14] Protocol
Write Algorithm (simplified)

Anonymizers

Server

Clients

\[
\begin{align*}
&x_1^1, y_1^1, x_2^1, y_2^1, \ldots, x_L^1, y_L^1, s_1 \\
&x_1^2, y_1^2, x_2^2, y_2^2, \ldots, x_L^2, y_L^2, s_2 \\
&\vdots \\
&x_1^M, y_1^M, x_2^M, y_2^M, \ldots, x_L^M, y_L^M, s_M
\end{align*}
\]
OGS – Construction (intuition)

Security Guarantees: CDS CAPO SDS SAPO

M-ORAM [JWQ14] Protocol
Write Algorithm (simplified)

\[
P, m, r', r'' \\
p'^x, m'^y \\
\]

Anonymizers

\[
x_1^1, y_1^1 \\
x_2^1, y_2^1 \\
\vdots \\
x_L^1, y_L^1 \\
S_1 \\
\]
\[
x_1^2, y_1^2 \\
x_2^2, y_2^2 \\
\vdots \\
x_L^2, y_L^2 \\
S_2 \\
\]
\[
x_1^M, y_1^M \\
x_2^M, y_2^M \\
\vdots \\
x_L^M, y_L^M \\
S_M \\
\]

Server

Clients
OGS – Construction (intuition)

Security Guarantees: CDS  CAPO  SDS  SAPO

M-ORAM [JWQ14] Protocol
Write Algorithm (simplified)
OGS – Construction (intuition)

Security Guarantees:  

- CDS
- **CAPO**
- SDS
- SAPO

**M-ORAM** [JWQ14] Protocol

Write Algorithm (simplified)

- Point is, **anonymizers can be distributed**
- Actual Write algorithm is more complicated
- Intuition behind Read algorithm is similar

Clients

\[
\begin{align*}
&\text{w}_1: (x^1_1, y^1_1) \quad (x^1_2, y^1_2) \quad \ldots \quad (x^1_M, y^1_M) \\
&\text{w}_2: (x^2_1, y^2_1) \quad (x^2_2, y^2_2) \quad \ldots \quad (x^2_M, y^2_M) \\
&\vdots \\
&\text{w}_N: (x^N_1, y^N_1) \quad (x^N_2, y^N_2) \quad \ldots \quad (x^N_M, y^N_M)
\end{align*}
\]

Anonymizers

Server
OGS – Construction (intuition)

Security Guarantees:  

- CDS
- CAPO
- SDS
- SAPO

Clients
Anonymizers
Server
OGS – Construction (intuition)

Security Guarantees:  

- CDS  
- CAPO  
- SDS  
- SAPO

**oABE [FaPe12] Scheme**

- Before Write algorithm, set $\bar{m} \leftarrow \text{oABE.Encrypt}(S, m)$
- Store $\bar{m}$ at server
- After Read algorithm, set $m \leftarrow \text{oABE.Decrypt}(SK_i, \bar{m})$
# OGS vs. the State of the Art

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Stateful ORAM</th>
<th>Stateless ORAM</th>
<th>M-ORAM</th>
<th>OGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allows multiple clients to share a cloud storage</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Allows clients to hide their data access patterns with the cloud storage from other clients</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Resists collusion of revoked clients with malicious cloud storage provider</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Allows clients to hide their data stored at a cloud storage from revoked clients</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>
Thank You!