Keyword Search meets Membership Testing

Sanjit Chatterjee and Sayantan Mukherjee

Indian Institute of Science, Bangalore
Keyword Search meets Membership Testing

Scenario:
- Files (and keywords) are stored in the cloud as ciphertext.
- To search files by keywords, user gives trapdoor/key.

We deal with two variants:
1. Key-Aggregate Searchable Encryption (KASE).
   - [CLW16,PM17] insecure.
2. Broadcast Encryption with Keyword Search (BEKS).
   - [KOR16] only selective security.
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.

- Big Loan
- High Risk
- SBI Txns
- HDFC Txns
- PNB Txns
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a **permitted set** of files only.
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.
- Users make search request.
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.
- Users make search request.
- If permitted, gets back search result. Else gets ⊥.
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.
- Users make search request.
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.
- Users make search request.
- If permitted, gets back search result. Else gets $\bot$.

<table>
<thead>
<tr>
<th>User</th>
<th>Big Loan</th>
<th>High Risk</th>
<th>SBI Txns</th>
<th>HDFC Txns</th>
<th>PNB Txns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ramesh</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Suresh</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a **permitted set** of files only.
- Users make **search** request.
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.
- Users make search request.
- If permitted, gets back search result. Else gets ⊥.
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a **permitted set** of files only.
- Users make **search** request.
- If permitted and keyword in file, gets back **yes**. Else **no**.
1. Key-Aggregate Searchable Encryption (KASE):
   - Files are stored encrypted in the cloud.
   - Each user can search only a subset of files \( S \subset \mathcal{F} \).
   - If file \( z \in S \), the user can search if file \( z \) contains keyword \( \omega \).
Keyword Search Meets Membership Testing

1. Key-Aggregate Searchable Encryption (KASE):
   - Files are stored encrypted in the cloud.
   - Each user can search only a subset of files ($S \subseteq \mathcal{F}$).
   - If file $z \in S$, the user can search if file $z$ contains keyword $\omega$.
   - **Key-Policy**
     - SrchEnc takes $y = (z, \omega')$ outputs ciphertext $CT_y$.
     - Trapdoor takes $x = (S, \omega)$ outputs trapdoor/key $SK_x$.

3. Output 1 if ($z \in S$) and ($\omega = \omega'$).
Keyword Search Meets Membership Testing

1. Key-Aggregate Searchable Encryption (KASE):
   - Files are stored encrypted in the cloud.
   - Each user can search only a subset of files ($S \subseteq F$).
   - If file $z \in S$, the user can search if file $z$ contains keyword $\omega$.
   - **Key-Policy**
     - $\text{SrchEnc}$ takes $y = (z, \omega')$ outputs ciphertext $CT_y$.
     - $\text{Trapdoor}$ takes $x = (S, \omega)$ outputs trapdoor/key $SK_x$.

2. Broadcast Encryption with Keyword Search (BEKS):
   - Files are stored encrypted for few privileged users ($S$).
   - If user $z \in S$, (s)he can search if file $f$ contains keyword $\omega$.
   - **Ciphertext-Policy**
     - $\text{SrchEnc}$ takes $y = (S, \omega')$ outputs ciphertext $CT_y$.
     - $\text{Trapdoor}$ takes $x = (z, \omega)$ outputs trapdoor/key $SK_x$.

3. Output 1 if ($z \in S$) and ($\omega = \omega'$).
1. Key-Aggregate Searchable Encryption (KASE):
   - Files are stored encrypted in the cloud.
   - Each user can search only a subset of files ($S \subset F$).
   - If file $z \in S$, the user can search if file $z$ contains keyword $\omega$.

2. Broadcast Encryption with Keyword Search (BEKS):
   - Files are stored encrypted for few privileged users ($S$).
   - If user $z \in S$, (s)he can search if file $f$ contains keyword $\omega$.

3. Output 1 if ($z \in S$) and ($\omega = \omega'$).
For a predicate $R : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$, SE is collection of four PPT.

- **KeyGen**: Outputs public parameter $\text{mpk}$ and master secret key $\text{msk}$.
- **Trapdoor**: It takes $\text{mpk}$, $\text{msk}$ and a trapdoor-index $x \in \mathcal{X}$ and outputs a trapdoor $\text{SK} \in \mathcal{SK}$ corresponding to $x$.
- **SrchEnc**: It takes $\text{mpk}$ and a data-index $y \in \mathcal{Y}$ and outputs ciphertext $\text{CT} \in \mathcal{C}$ corresponding to $y$ and encapsulation key $\mathcal{K} \in \mathcal{K}$.
- **Test**: It takes $\text{mpk}$, $\text{SK}$ and $(\text{CT}, \mathcal{K})$ as input. Outputs $b \in \{0, 1\}$.

Security: Given $\text{CT}$, no PPT can get any new info about $y$. 
Tools

- Identity-Based Encryption [BB04].
- Broadcast Encryption [GW08].
- Dual System Encryption [Wat09].
- Prime-order Predicate Encryption [CGW15,AT16].
- Tag-based IBE from QA-NIZK [JR13].
Intuition

- Compact merge of BB-IBE and GW-Hash.
  - $(S, \omega)$ is encoded to $(w_0 \cdot \omega + \sum_{i \in S} w_i)$.
  - $(z, \omega')$ is encoded to $(w_0 \cdot \omega' + w_z)$.
- Encoding-based intuition:
  - Correctness: if $(z \in S)$ and $(\omega = \omega')$:
    $$(w_0 \cdot \omega + \sum_{i \in S} w_i) \in LS(w_0 \cdot \omega' + w_z, (w_i)_{i \in S \setminus \{z\}}).$$
Intuition

- Compact merge of BB-IBE and GW-Hash.
  - \((S, \omega)\) is encoded to \((w_0 \cdot \omega + \sum_{i \in S} w_i)\).
  - \((z, \omega')\) is encoded to \((w_0 \cdot \omega' + w_z)\).

- Encoding-based intuition:
  - Correctness: if \((z \in S)\) and \((\omega = \omega')\):
    \((w_0 \cdot \omega + \sum_{i \in S} w_i) \in LS(w_0 \cdot \omega' + w_z, (w_i)_{i \in S \setminus (z)})\).
  - Security: if \((z \notin S)\):
    \((w_0 \cdot \omega + \sum_{i \in S} w_i) \notin LS(w_0 \cdot \omega' + w_z, (w_i)_{i \in S \setminus (z)})\).
  - Security: if \((z \in S)\) and \((\omega \neq \omega')\):
    \((w_0 \cdot \omega + \sum_{i \in S} w_i) \notin LS(w_0 \cdot \omega' + w_z, (w_i)_{i \in S \setminus (z)})\).

- Extend Tag-based IBE construction technique [RS14].
Key-Aggregate Searchable Encryption (KASE)

**Algorithm 1 Construction**

**KeyGen**(1\(^\lambda\), \(\mathcal{U}, \mathcal{W}\))

1: \((p, G_1, G_2, G_T, e) \leftarrow G_{abg}(1\(^\lambda\))

2: \((g_1, g_2) \leftarrow G_1 \times G_2, g_T \leftarrow G_T

3: \alpha_1, \alpha_2, c, d, (u_i, v_i)_{i \in [n+1]} \leftarrow \mathbb{Z}_p

4: \quad b \leftarrow \mathbb{Z}_p^\times, g_T^\alpha = e(g_1, g_2)^{\alpha_1 + b\alpha_2}

5: \quad (g_1^{w_i} = g_1^{u_i + bv_i})_{i \in [n+1]}, g_1^w = g_1^{c + bd}

6: \quad msk = (g_2, g_2^c, \alpha_1, \alpha_2, d, (u_i, v_i)_{i \in [n+1]})

7: \quad mpk = (g_1, g_1^b, (g_1^{w_i})_{i \in [n+1]}, g_1^w, g_T^\alpha)

**SrchEnc**(mpk, \(y = (z, \omega')\))

1: \(s, (t_i)_{i \in [n]} \leftarrow \mathbb{Z}_p

2: \(K = e(g_1, g_2)^{\alpha s}, C_0 = g_1^s, C_1 = g_1^{bs}

3: \quad C_{2,i} = \begin{cases} g_1^{s(w_{i+1}\omega' + w_z + wt_z)} & \text{if } i = z \\ g_1^{s(w_i + wt_i)} & \text{otherwise} \end{cases}

4: \quad CT_y = (C_0, C_1, (C_{2,i}, t_i)_{i \in [n]})

**Trapdoor**(msk, \(x = (S, \omega)\))

1: \(r \leftarrow \mathbb{Z}_p

2: \quad K_1 = g_2^r, K_2 = g_2^{cr}, K_4 = g_2^{dr}

3: \quad K_2 = g_2^{\alpha_2 + r(v_{n+1}\omega + \sum_{i \in S} v_i)}

4: \quad K_5 = g_2

5: \quad SK_x = (K_1, K_2, K_3, K_4, K_5)

**Test**((SK\(_x\), S), (CT\(_y\), \(R\)))

1: \(A = e\left(\prod_{i \in S} C_{2,i}, K_1\right)

2: \quad B = e\left(C_0, K_3 \prod_{i \in S} K_2^{t_i}\right) e\left(C_1, K_5 \prod_{i \in S} K_4^{t_i}\right)

3: \quad Output 1 iff \(R = B/A\)
Algorithm 2 Construction

**KeyGen**($1^\lambda, U, \mathcal{W}$)

1. $(p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}_{\text{abg}}(1^\lambda)$
2. $(g_1, g_2) \leftarrow G_1 \times G_2, g_T \leftarrow G_T$
3. $\alpha_1, \alpha_2, c, d, (u_i, v_i)_{i \in [n+1]} \leftarrow \mathbb{Z}_p$
4. $b \leftarrow \mathbb{Z}_p^\times, g_T^\alpha = e(g_1, g_2)^{(\alpha_1 + b\alpha_2)}$
5. $(g_{w_i} = g_1^{u_i + bv_i})_{i \in [n+1]}, g_1^{w} = g_1^{c + bd}$
6. $\text{msk} = (g_2, g_2^c, \alpha_1, \alpha_2, d, (u_i, v_i)_{i \in [n+1]})$
7. $\text{mpk} = (g_1, g_1^b, (g_1^{w_i})_{i \in [n+1]}, g_1^w, g_T^\alpha)$

**SrchEnc**($\text{mpk}, y = (S, \omega')$)

1. $s, \bar{t} \leftarrow \mathbb{Z}_p$
2. $\mathcal{R} = e(g_1, g_2)^{\alpha s}, C_0 = g_1^s, C_1 = g_1^{bs}$
3. $C_2 = g_1$
4. $\text{CT}_y = (C_0, C_1, C_2, \bar{t})$

**Trapdoor**($\text{msk}, x = (z, \omega)$)

1. $r \leftarrow \mathbb{Z}_p$
2. $K_1 = g_2^r, K_2 = g_2^{cr}, K_4 = g_2^{dr}$
3. $K_{3,i} = \begin{cases} g_2^\alpha r(u_{n+1} \omega + u_z) & \text{if } i = z \\ g_2^{ru_i} & \text{otherwise} \end{cases}$
4. $K_{5,i} = \begin{cases} g_2^{\alpha_2 r(v_{n+1} \omega + v_z)} & \text{if } i = z \\ g_2^{rv_i} & \text{otherwise} \end{cases}$
5. $\text{SK}_x = (K_1, K_2, (K_{3,i})_{i \in [n]}, K_4, (K_{5,i})_{i \in [n]})$

**Test**($\text{SK}_x, (\text{CT}_y, \mathcal{R}, S)$)

1. $A = e(C_2, K_1)$
2. $B = e\left(C_0, K_2^{\bar{t} \prod_{i \in S} K_{3,i}}\right) e\left(C_1, K_4^{\bar{t} \prod_{i \in S} K_{5,i}}\right)$
3. Output 1 iff $\mathcal{R} = B / A$
Are the Solutions Any Good?

1. KASE:
   ▶ Constant-size trapdoor: $5G_2$.
   ▶ **Fully anonymous secure** under SXDH.
   ▶ Hides both the keyword $\omega'$ and user $z$ in the ciphertext.
   ▶ Search requires only three pairings. Took on average 0.07 sec for $|U| = 10,000$ and $|S| = 5000$.

2. BEKS:
   ▶ Constant-size ciphertext: $3G_1 + G_T + \mathbb{Z}_p$.
   ▶ **Fully anonymous secure** under SXDH.
   ▶ Hides the keyword $\omega'$ in ciphertext.
   ▶ Search requires only three pairings. Took on average 0.4 sec for $|U| = 10,000$ and $|S| = 5000$. 
1. **KASE:**
   - Constant-size trapdoor: $5G_2$.
   - **Fully anonymous secure** under SXDH.
   - Hides both the keyword $\omega'$ and user $z$ in the ciphertext.
   - Search requires only three pairings. Took on average 0.07 sec for $|U| = 10,000$ and $|S| = 5000$.
   - Large ciphertext.

2. **BEKS:**
   - Constant-size ciphertext: $3G_1 + G_T + \mathbb{Z}_p$.
   - **Fully anonymous secure** under SXDH.
   - Hides the keyword $\omega'$ in ciphertext.
   - Search requires only three pairings. Took on average 0.4 sec for $|U| = 10,000$ and $|S| = 5000$.
   - Large trapdoor.
Thank you!
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a **privileged set** of users.
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.
- Users make search request.
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.
- Users make search request.
- If permitted, gets back search result. Else gets ⊥.
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.
- Users make search request.

![Diagram with files and users]
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.
- Users make search request.
- If permitted, gets back search result. Else gets ⊥.
Let five files and three users.

Each file has a **privileged set** of users.

Users make **search** request.
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.
- Users make search request.
- If permitted, gets back search result. Else gets ⊥.

![Diagram of files and permissions]

Big Loan
High Risk
SBI Txns
HDFC Txns
PNB Txns

- yes
- no
- no
- no

HDFC BANK

12 / 12
Let five files and three users.
Each file has a privileged set of users.
Users make search request.
If permitted and keyword in file, gets back yes. Else no.