Keyword Search meets Membership Testing

Sanjit Chatterjee and Sayantan Mukherjee

Indian Institute of Science, Bangalore

10th Dec, 2018, Indocrypt, IHC, New Delhi

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:

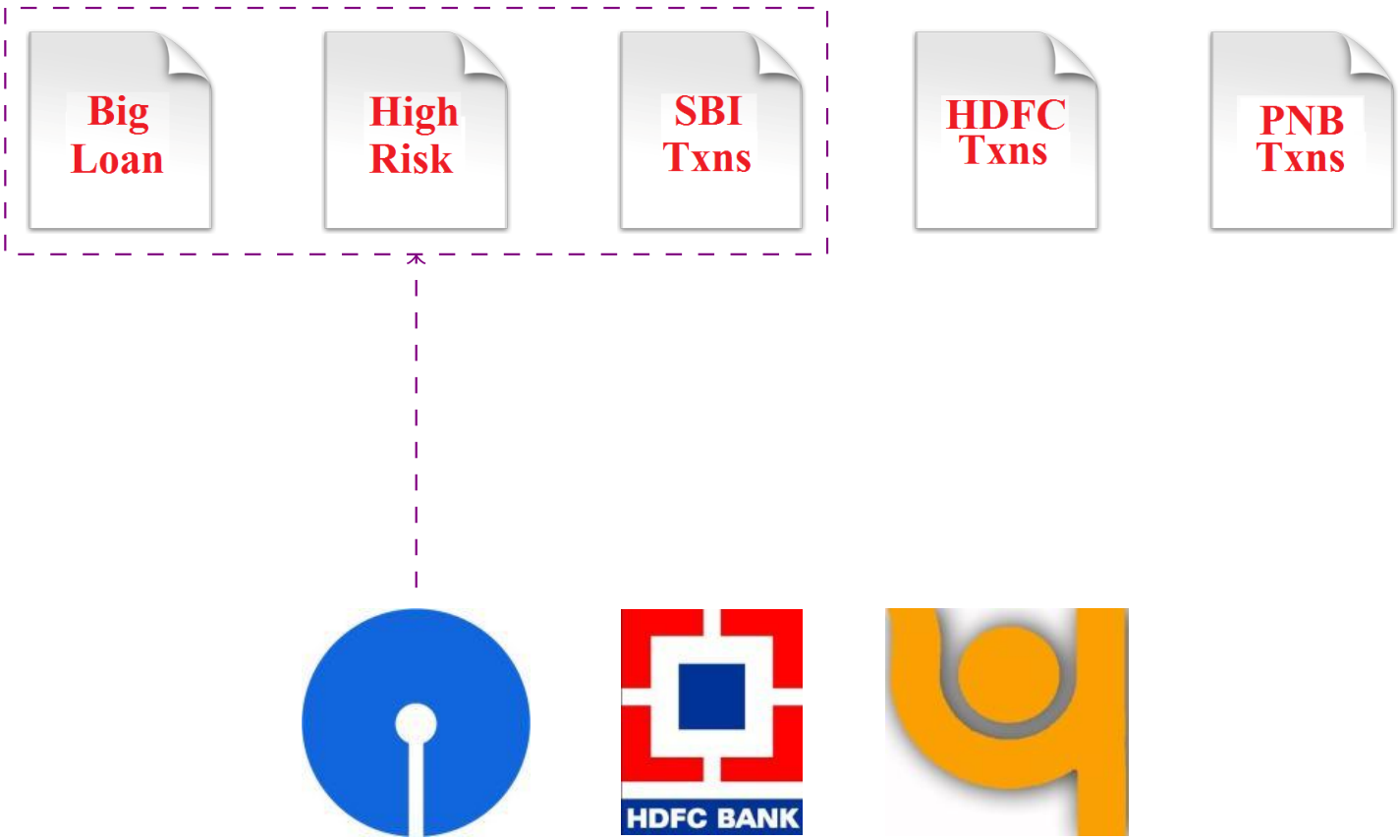
1. Key-Aggregate Searchable Encryption (KASE).
   - [CLW16,PM17] insecure.
2. Broadcast Encryption with Keyword Search (BEKS).
   - [KOR16] only selective security.

Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.

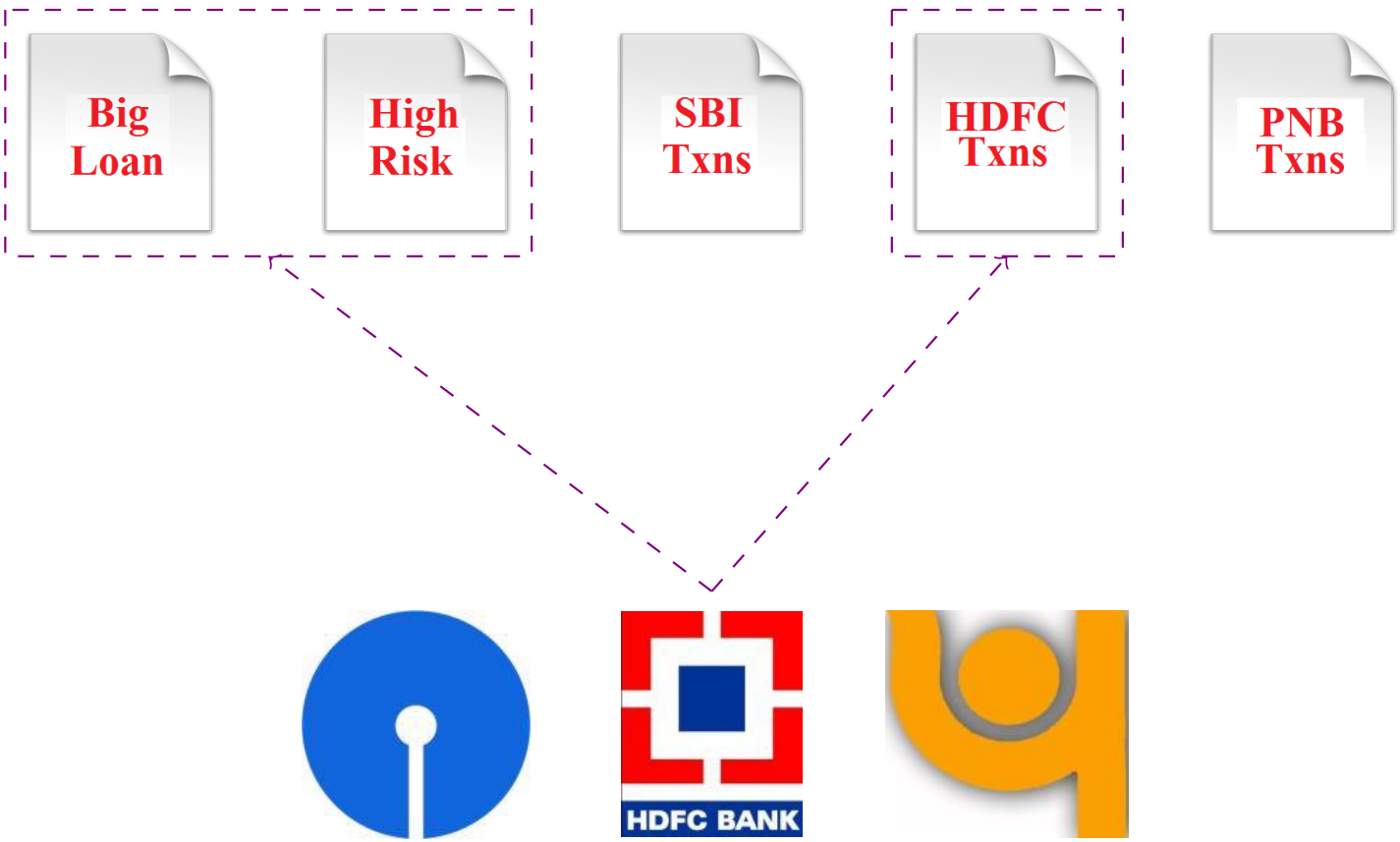
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.



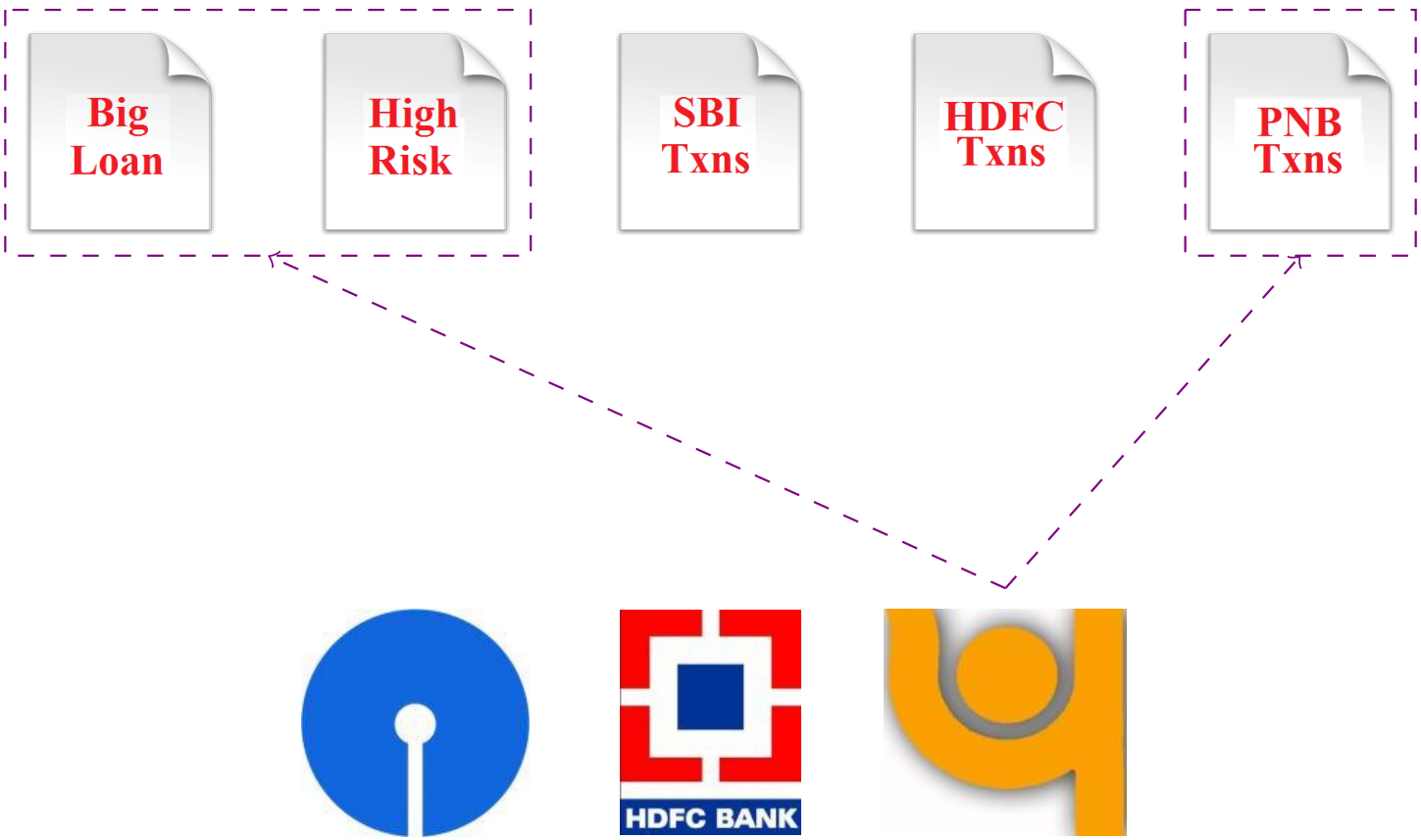
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.



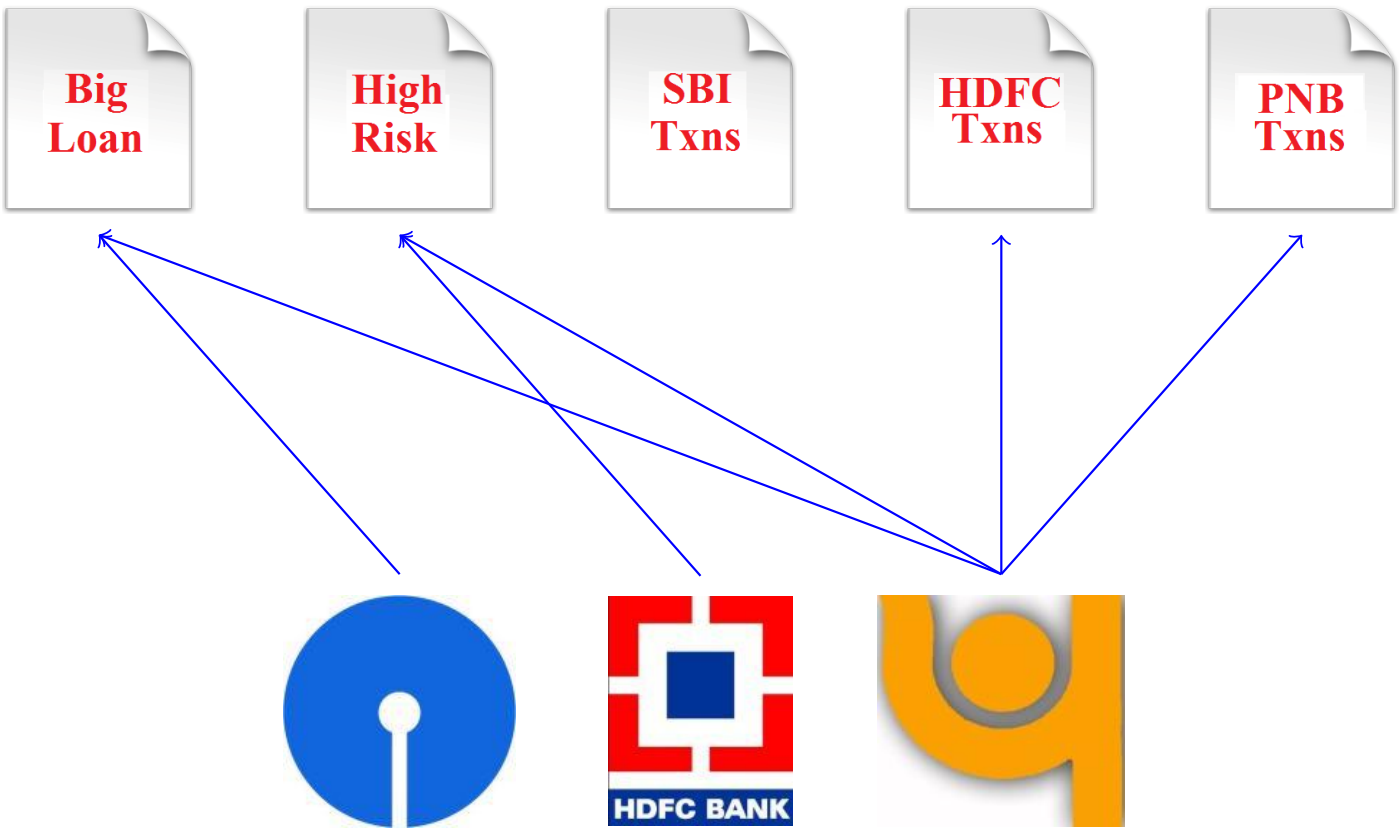
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.



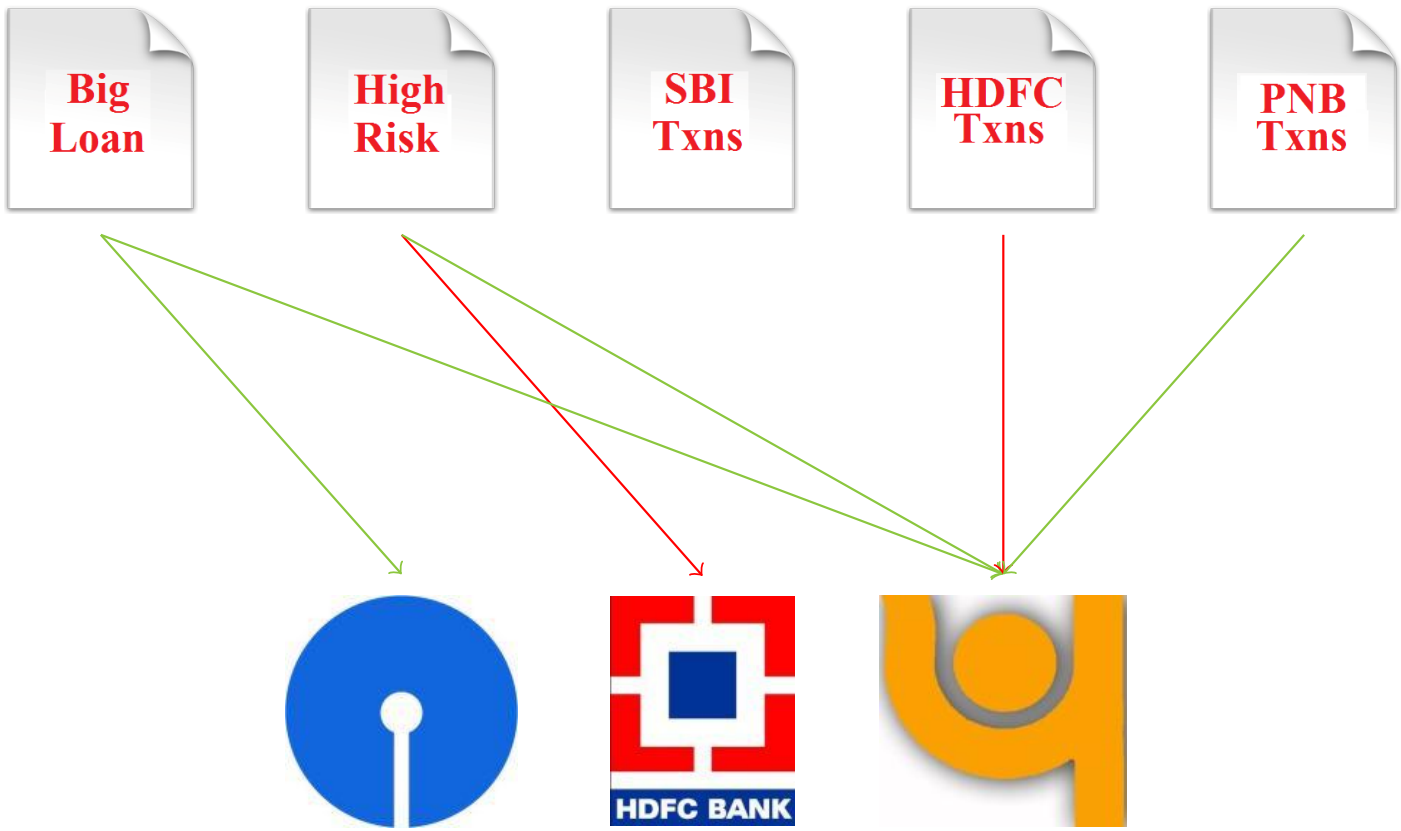
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.
- Users make search request.



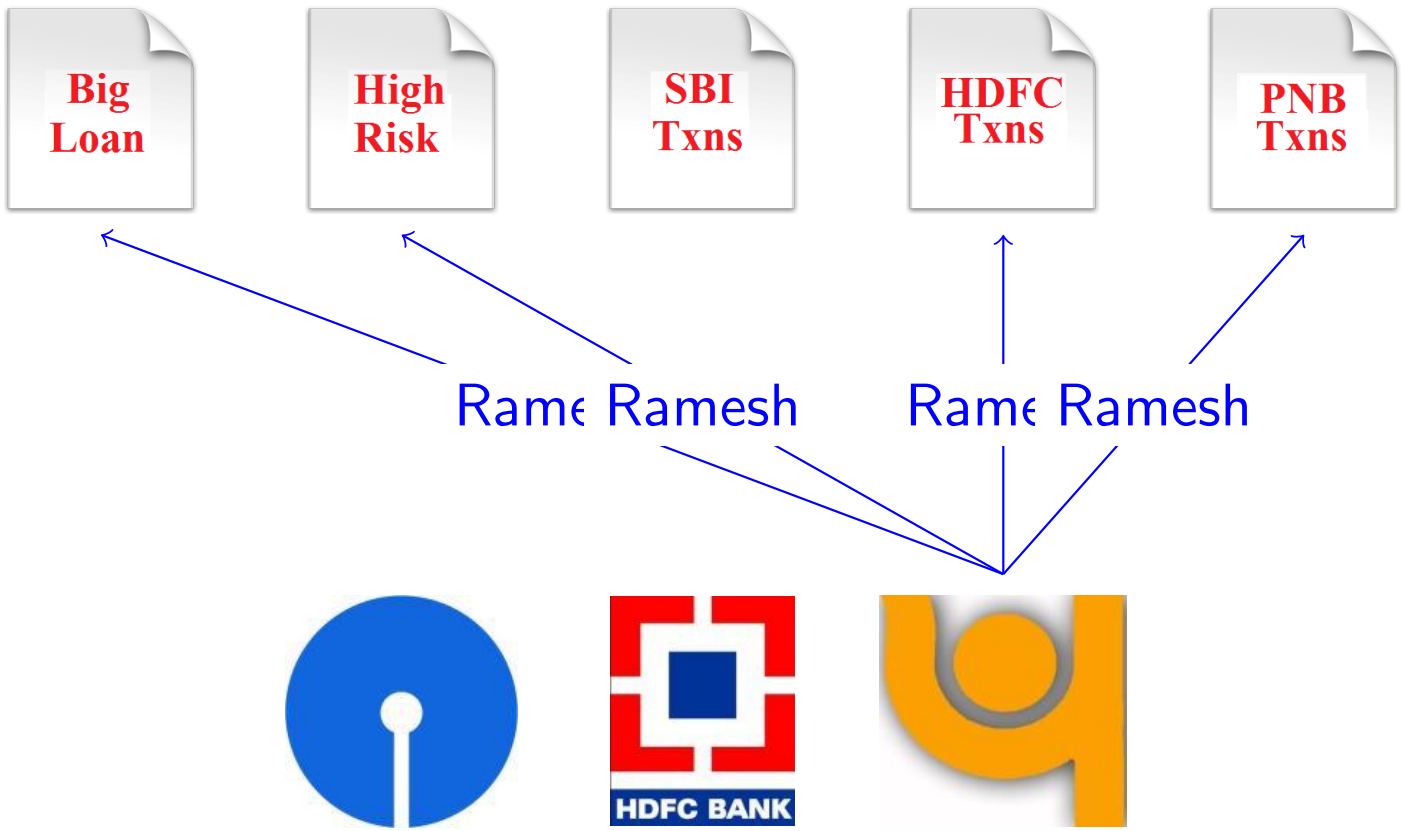
Problem: Key-Aggregate Searchable Encryption (KASE)

- 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 $\perp$.



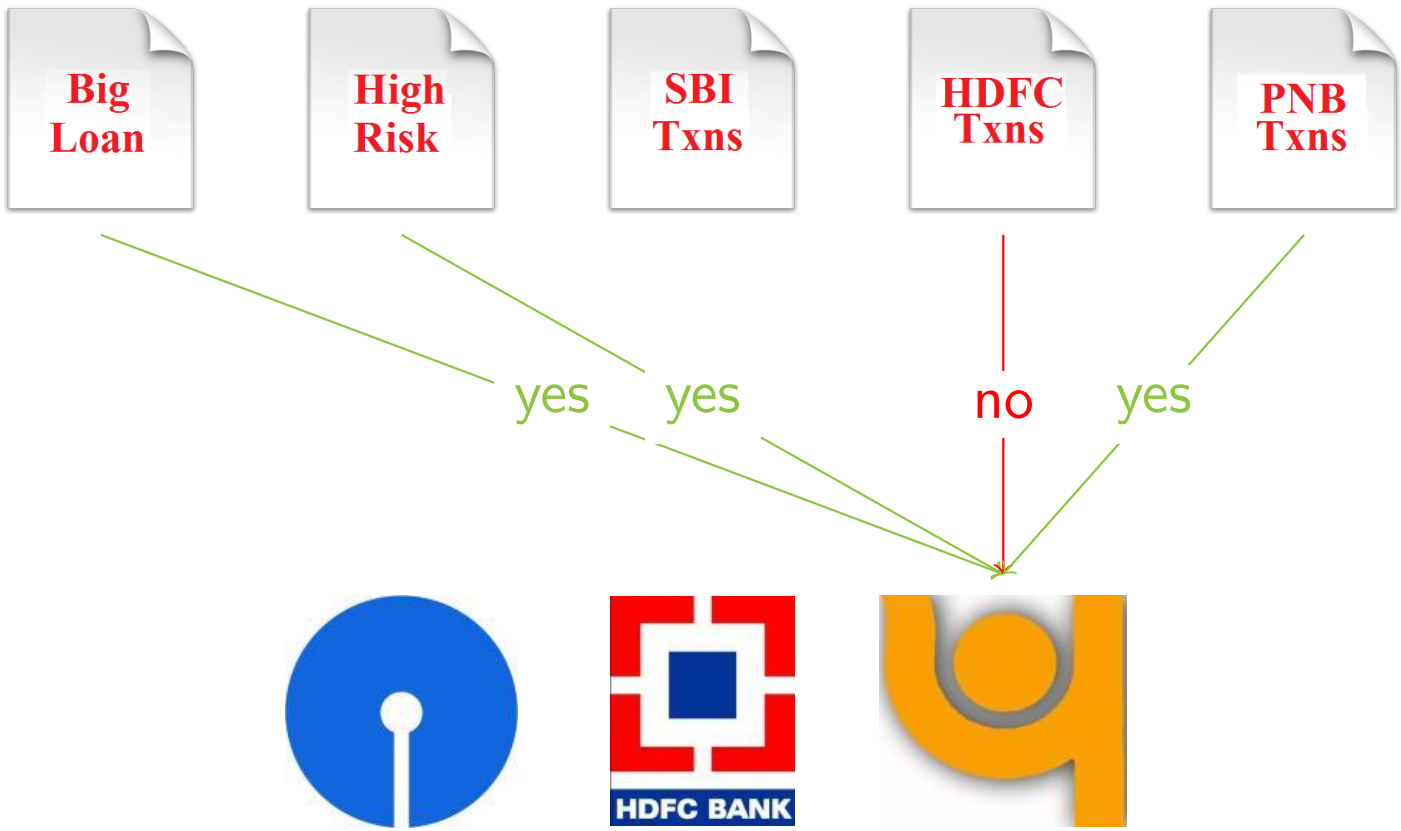
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.
- Users make search request.



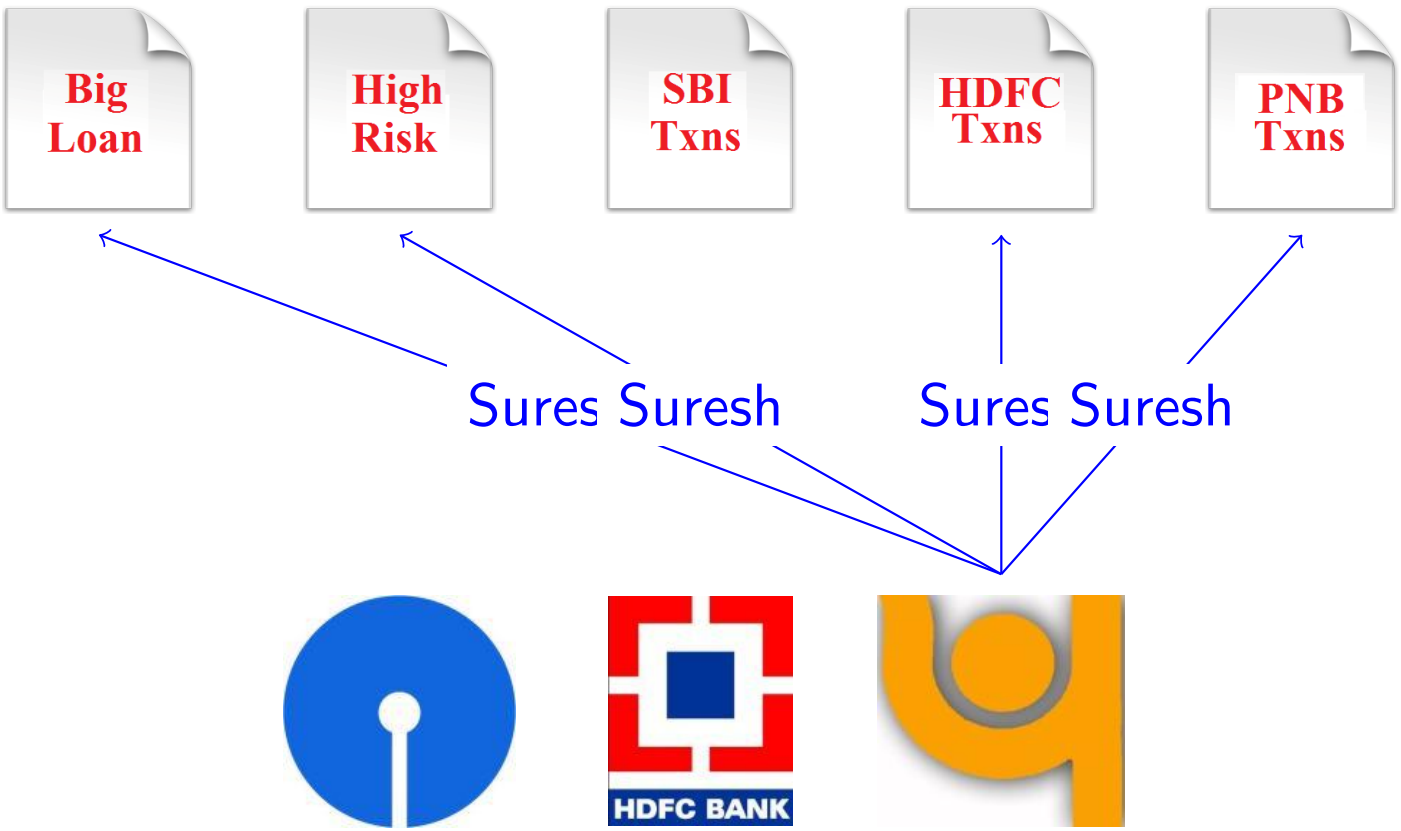
Problem: Key-Aggregate Searchable Encryption (KASE)

- 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 $\perp$.



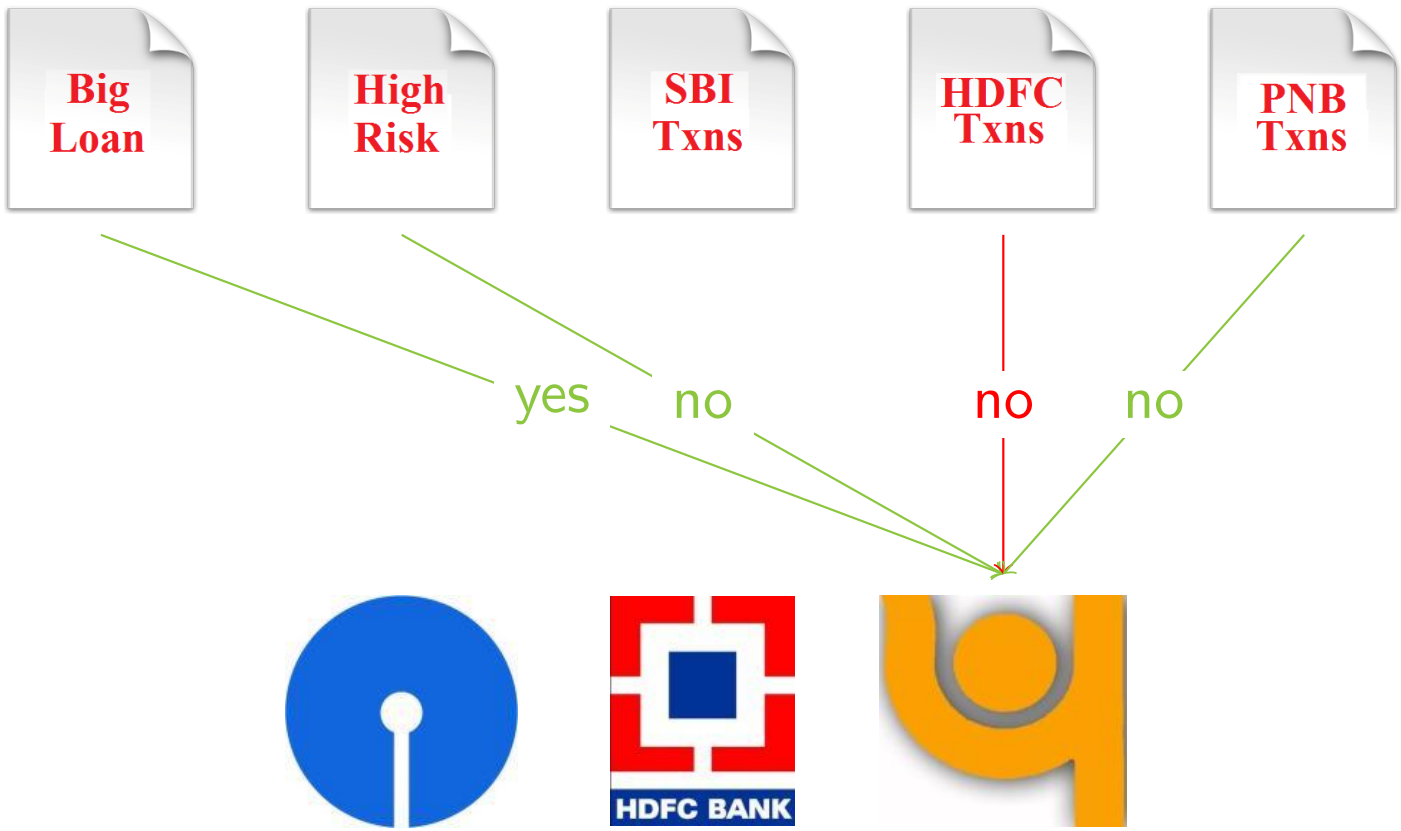
Problem: Key-Aggregate Searchable Encryption (KASE)

- Let five files and three users.
- Each user can search a permitted set of files only.
- Users make search request.



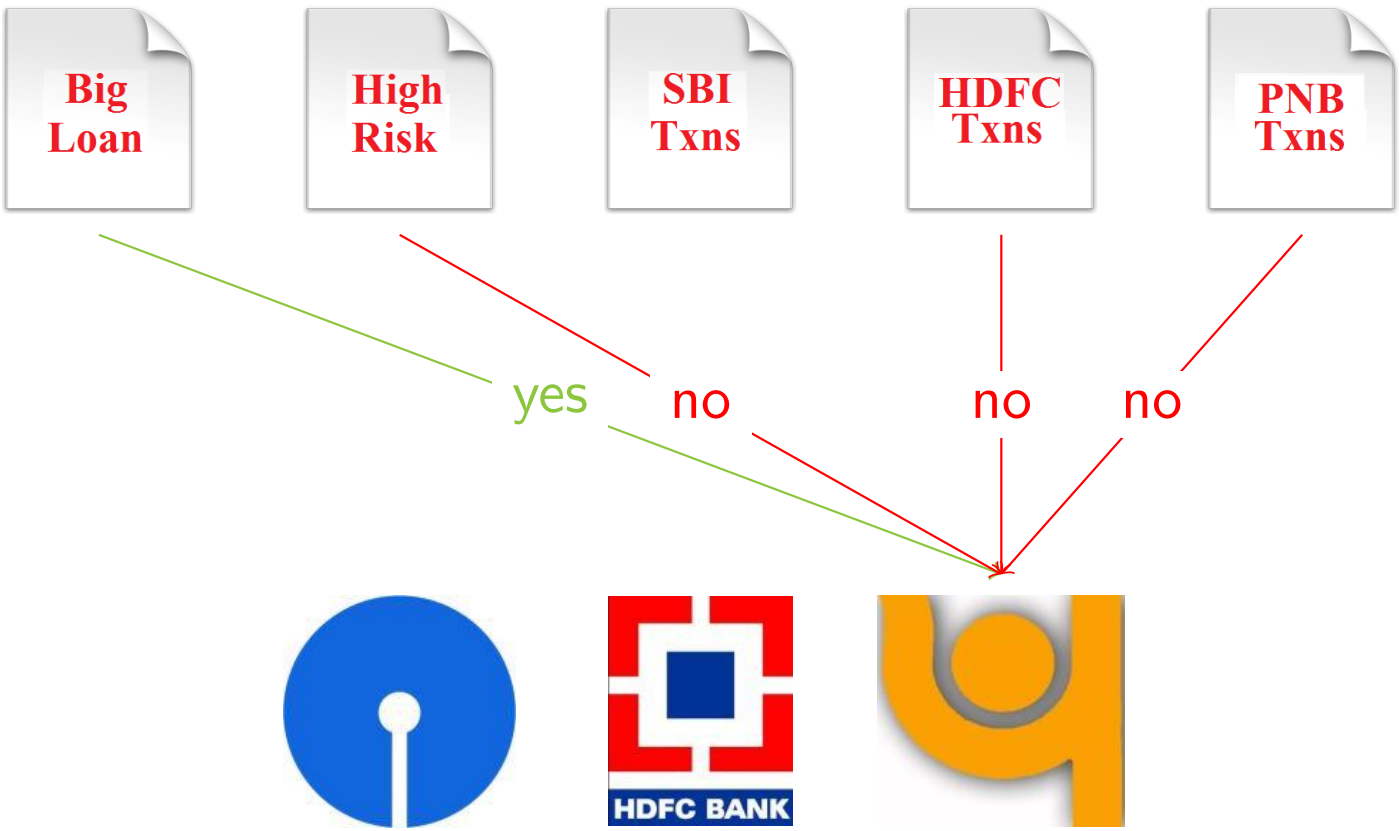
Problem: Key-Aggregate Searchable Encryption (KASE)

- 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 $\perp$.



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.



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 ($S \subset \mathcal{F}$).
   - If file z $\in S$, the user can search if file z contains keyword ω.

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 ($S \subset \mathcal{F}$).
   - If file $z \in S$, the user can search if file z contains keyword ω.
   - 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 ($S \subset \mathcal{F}$).
   - If file $z \in S$, the user can search if file z contains keyword ω.
   - Key-Policy
     - SrchEnc takes $y = (z, \omega')$ outputs ciphertext CT_y.
     - 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 ω.
   - Ciphertext-Policy
     - SrchEnc takes $y = (S, \omega')$ outputs ciphertext CT_y.
     - Trapdoor takes $x = (z, \omega)$ 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 ($S \subset \mathcal{F}$).
   - If file $z \in S$, the user can search if file z contains keyword ω.
   - Key-Policy
     - SrchEnc takes $y = (z, \omega')$ outputs ciphertext CT_y.
     - Trapdoor takes $x = (S, \omega)$ outputs trapdoor/key SK_x.
     - $|SK_x|$ should be constant.
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 ω.
   - Ciphertext-Policy
     - SrchEnc takes $y = (S, \omega')$ outputs ciphertext CT_y.
     - Trapdoor takes $x = (z, \omega)$ outputs trapdoor/key SK_x.
     - $|CT_y|$ should be constant.
3. Output 1 if $(z \in S)$ and $(\omega = \omega')$.

Searchable Encryption (SE) [BGOP04]

For a predicate $\mathsf{R} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, SE is collection of four PPT.

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

Security: Given 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, ω) is encoded to $(w_0 \cdot \omega + \sum_{i \in S} w_i)$.
  - (z, ω') is encoded to $(w_0 \cdot \omega' + w_{\mathsf{z}})$.
- Encoding-based intuition:
  - Correctness: if $(\mathsf{z} \in S)$ and $(\omega = \omega')$:
    $(w_0 \cdot \omega + \sum_{i \in S} w_i) \in LS(w_0 \cdot \omega' + w_{\mathsf{z}}, (w_i)_{i \in S \setminus (\mathsf{z})})$.

Intuition

- Compact merge of BB-IBE and GW-Hash.
  - (S, ω) is encoded to $(w_0 \cdot \omega + \sum_{i \in S} w_i)$.
  - (z, ω') is encoded to $(w_0 \cdot \omega' + w_{\mathsf{z}})$.
- Encoding-based intuition:
  - Correctness: if $(\mathsf{z} \in S)$ and $(\omega = \omega')$:
    $(w_0 \cdot \omega + \sum_{i \in S} w_i) \in LS(w_0 \cdot \omega' + w_{\mathsf{z}}, (w_i)_{i \in S \setminus (\mathsf{z})})$.
  - Security: if $(\mathsf{z} \notin S)$:
    $(w_0 \cdot \omega + \sum_{i \in S} w_i) \notin LS(w_0 \cdot \omega' + w_{\mathsf{z}}, (w_i)_{i \in S \setminus (\mathsf{z})})$.
  - Security: if $(\mathsf{z} \in S)$ and $(\omega \neq \omega')$:
    $(w_0 \cdot \omega + \sum_{i \in S} w_i) \notin LS(w_0 \cdot \omega' + w_{\mathsf{z}}, (w_i)_{i \in S \setminus (\mathsf{z})})$.
- Extend Tag-based IBE construction technique [RS14].

Algorithm 1 Construction

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

1: $(p, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_\mathsf{T}, e) \leftarrow \mathcal{G}_{\mathsf{abg}}(1^\lambda)$
2: $(g_1, g_2) \hookleftarrow \mathsf{G}_1 \times \mathsf{G}_2$, $g_\mathsf{T} \leftarrow \mathsf{G}_\mathsf{T}$
3: $\alpha_1, \alpha_2, c, d, (u_i, v_i)_{i\in[n+1]} \hookleftarrow \mathbb{Z}_p$
4: $b \hookleftarrow \mathbb{Z}_p^\times$, $g_\mathsf{T}^\alpha = e(g_1, g_2)^{(\alpha_1 + b\alpha_2)}$
5: $\left(g_1^{w_i} = g_1^{u_i + bv_i}\right)_{i\in[n+1]}$, $g_1^w = g_1^{c+bd}$
6: $\mathsf{msk} = (g_2, g_2^c, \alpha_1, \alpha_2, d, (u_i, v_i)_{i\in[n+1]})$
7: $\mathsf{mpk} = \left(g_1, g_1^b, \left(g_1^{w_i}\right)_{i\in[n+1]}, g_1^w, g_\mathsf{T}^\alpha\right)$

$\underline{\mathsf{SrchEnc}(\mathsf{mpk}, \mathsf{y} = (\mathsf{z}, \omega'))}$

1: $s, (t_i)_{i\in[n]} \hookleftarrow \mathbb{Z}_p$
2: $\mathfrak{K} = e(g_1, g_2)^{\alpha s}$, $\mathsf{C}_0 = g_1^s$, $\mathsf{C}_1 = g_1^{bs}$
$\mathsf{C}_{2,i} = \begin{cases} g_1^{s(w_{n+1}\omega' + w_\mathsf{z} + wt_\mathsf{z})} & \text{if } i = \mathsf{z} \\ g_1^{s(w_i + wt_i)} & \text{otherwise} \end{cases}$
3: $\mathsf{CT}_\mathsf{y} = (\mathsf{C}_0, \mathsf{C}_1, \left(\mathsf{C}_{2,i}, t_i\right)_{i\in[n]})$

$\underline{\mathsf{Trapdoor}(\mathsf{msk}, \mathsf{x} = (S, \omega))}$

1: $r \hookleftarrow \mathbb{Z}_p$
2: $\mathsf{K}_1 = g_2^r, \mathsf{K}_2 = g_2^{cr}, \mathsf{K}_4 = g_2^{dr}$
$\mathsf{K}_3 = g_2^{\alpha_1 + r(u_{n+1}\omega + \sum_{i\in S} u_i)}$
$\mathsf{K}_5 = g_2^{\alpha_2 + r(v_{n+1}\omega + \sum_{i\in S} v_i)}$
3: $\mathsf{SK}_\mathsf{x} = (\mathsf{K}_1, \mathsf{K}_2, \mathsf{K}_3, \mathsf{K}_4, \mathsf{K}_5)$

$\underline{\mathsf{Test}((\mathsf{SK}_\mathsf{x}, S), (\mathsf{CT}_\mathsf{y}, \mathfrak{K}))}$

1: $A = e\left(\prod_{i\in S} \mathsf{C}_{2,i}, \mathsf{K}_1\right)$
2: $B = e\left(\mathsf{C}_0, \mathsf{K}_3 \prod_{i\in S} \mathsf{K}_2^{t_i}\right) e\left(\mathsf{C}_1, \mathsf{K}_5 \prod_{i\in S} \mathsf{K}_4^{t_i}\right)$
3: Output 1 iff $\mathfrak{K} = B/A$

Algorithm 2 Construction

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

1: $(p, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_\mathsf{T}, e) \leftarrow \mathcal{G}_{\mathsf{abg}}(1^\lambda)$
2: $(g_1, g_2) \hookleftarrow \mathsf{G}_1 \times \mathsf{G}_2$, $g_\mathsf{T} \leftarrow \mathsf{G}_\mathsf{T}$
3: $\alpha_1, \alpha_2, c, d, (u_i, v_i)_{i\in[n+1]} \hookleftarrow \mathbb{Z}_p$
4: $b \hookleftarrow \mathbb{Z}_p^\times$, $g_\mathsf{T}^\alpha = e(g_1, g_2)^{(\alpha_1 + b\alpha_2)}$
5: $\left(g_1^{w_i} = g_1^{u_i + bv_i}\right)_{i\in[n+1]}$, $g_1^w = g_1^{c+bd}$
6: $\mathsf{msk} = (g_2, g_2^c, \alpha_1, \alpha_2, d, (u_i, v_i)_{i\in[n+1]})$
7: $\mathsf{mpk} = \left(g_1, g_1^b, \left(g_1^{w_i}\right)_{i\in[n+1]}, g_1^w, g_\mathsf{T}^\alpha\right)$

$\underline{\mathsf{Trapdoor}(\mathsf{msk}, \mathsf{x} = (\mathsf{z}, \omega))}$

1: $r \hookleftarrow \mathbb{Z}_p$
2: $\mathsf{K}_1 = g_2^r, \mathsf{K}_2 = g_2^{cr}, \mathsf{K}_4 = g_2^{dr}$

$$\mathsf{K}_{3,i} = \begin{cases} g_2^{\alpha_1 + r(u_{n+1}\omega + u_\mathsf{z})} & \text{if } i = \mathsf{z} \\ g_2^{ru_i} & \text{otherwise} \end{cases}$$

$$\mathsf{K}_{5,i} = \begin{cases} g_2^{\alpha_2 + r(v_{n+1}\omega + v_\mathsf{z})} & \text{if } i = \mathsf{z} \\ g_2^{rv_i} & \text{otherwise} \end{cases}$$

3: $\mathsf{SK}_\mathsf{x} = (\mathsf{K}_1, \mathsf{K}_2, \left(\mathsf{K}_{3,i}\right)_{i\in[n]}, \mathsf{K}_4, \left(\mathsf{K}_{5,i}\right)_{i\in[n]})$

$\underline{\mathsf{SrchEnc}(\mathsf{mpk}, \mathsf{y} = (S, \omega'))}$

1: $s, \ddot{t} \hookleftarrow \mathbb{Z}_p$
2: $\mathfrak{K} = e(g_1, g_2)^{\alpha s}, \mathsf{C}_0 = g_1^s, \mathsf{C}_1 = g_1^{bs}$

$$\mathsf{C}_2 = g_1^{s\left(w_{n+1}\omega' + \sum_{j\in S} w_j + w\ddot{t}\right)}$$

3: $\mathsf{CT}_\mathsf{y} = (\mathsf{C}_0, \mathsf{C}_1, \mathsf{C}_2, \ddot{t})$

$\underline{\mathsf{Test}(\mathsf{SK}_\mathsf{x}, (\mathsf{CT}_\mathsf{y}, \mathfrak{K}, S))}$

1: $A = e\left(\mathsf{C}_2, \mathsf{K}_1\right)$
2: $B = e\left(\mathsf{C}_0, \mathsf{K}_2^{\ddot{t}} \prod_{i\in S} \mathsf{K}_{3,i}\right) e\left(\mathsf{C}_1, \mathsf{K}_4^{\ddot{t}} \prod_{i\in S} \mathsf{K}_{5,i}\right)$
3: Output 1 iff $\mathfrak{K} = B/A$

Are the Solutions Any Good?

1. KASE:
   - Constant-size trapdoor: $5\mathsf{G}_2$.
   - Fully anonymous secure under SXDH.
   - Hides both the keyword ω' and user z in the ciphertext.
   - Search requires only three pairings. Took on average 0.07 sec for $|\mathcal{U}| = 10{,}000$ and $|S| = 5000$.

2. BEKS:
   - Constant-size ciphertext: $3\mathsf{G}_1 + \mathsf{G}_\mathsf{T} + \mathbb{Z}_p$.
   - Fully anonymous secure under SXDH.
   - Hides the keyword ω' in ciphertext.
   - Search requires only three pairings. Took on average 0.4 sec for $|\mathcal{U}| = 10{,}000$ and $|S| = 5000$.

Anything lacking?

1. KASE:
   - Constant-size trapdoor: $5\mathsf{G}_2$.
   - Fully anonymous secure under SXDH.
   - Hides both the keyword ω' and user z in the ciphertext.
   - Search requires only three pairings. Took on average 0.07 sec for $|\mathcal{U}| = 10{,}000$ and $|S| = 5000$.
   - Large ciphertext. ×
2. BEKS:
   - Constant-size ciphertext: $3\mathsf{G}_1 + \mathsf{G}_\mathsf{T} + \mathbb{Z}_p$.
   - Fully anonymous secure under SXDH.
   - Hides the keyword ω' in ciphertext.
   - Search requires only three pairings. Took on average 0.4 sec for $|\mathcal{U}| = 10{,}000$ and $|S| = 5000$.
   - Large trapdoor. ×

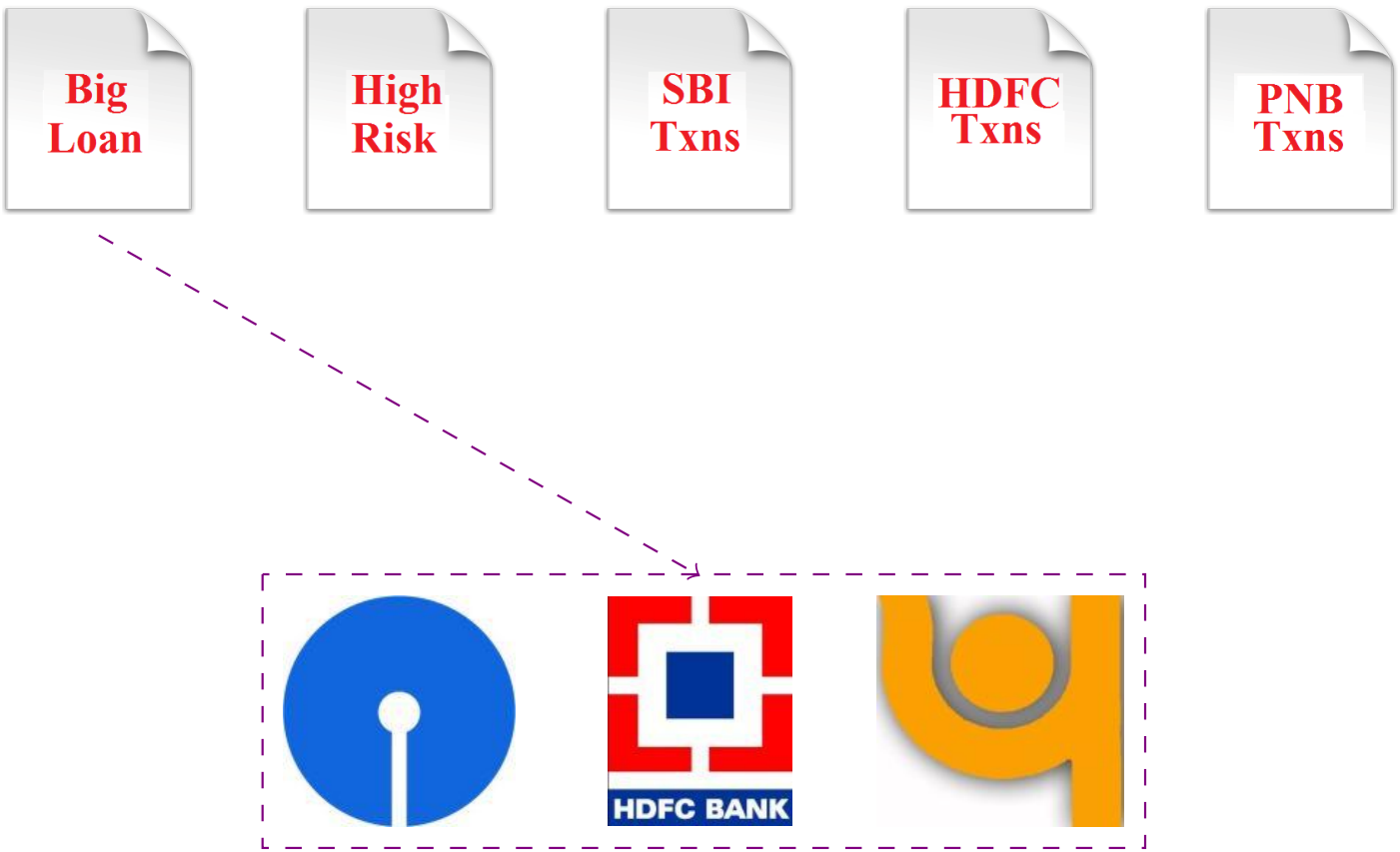
Thank you!

Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.

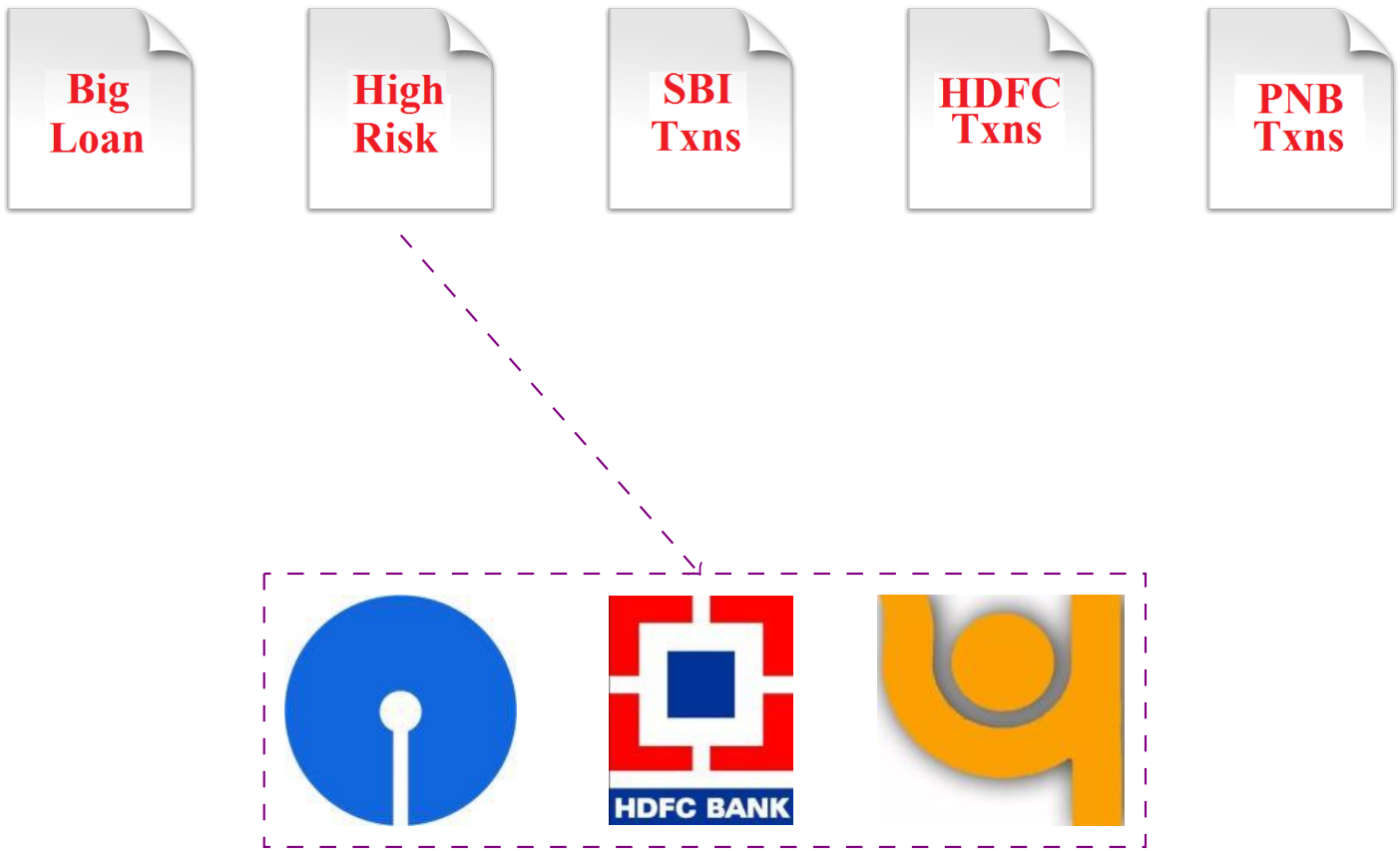

Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.



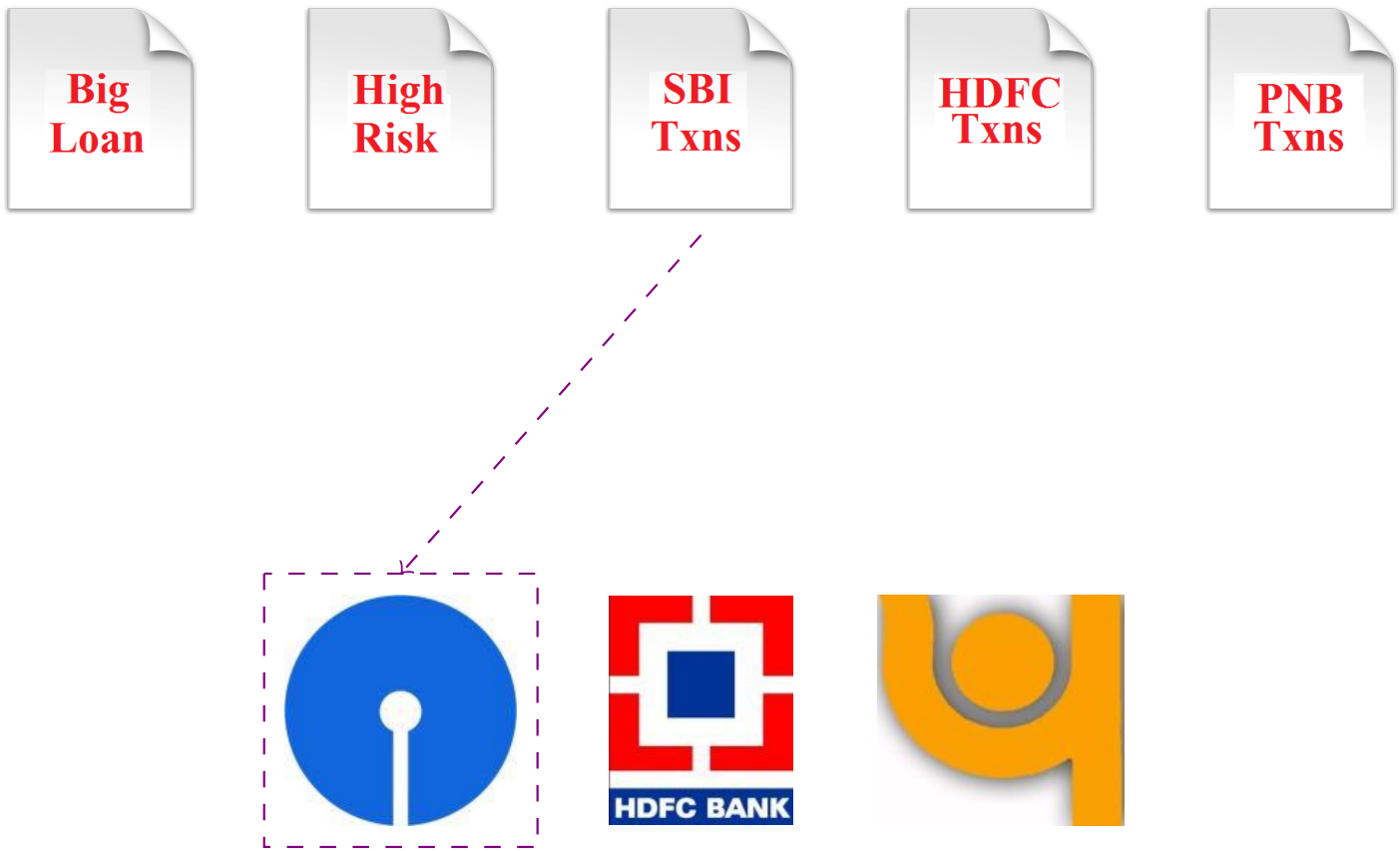
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.



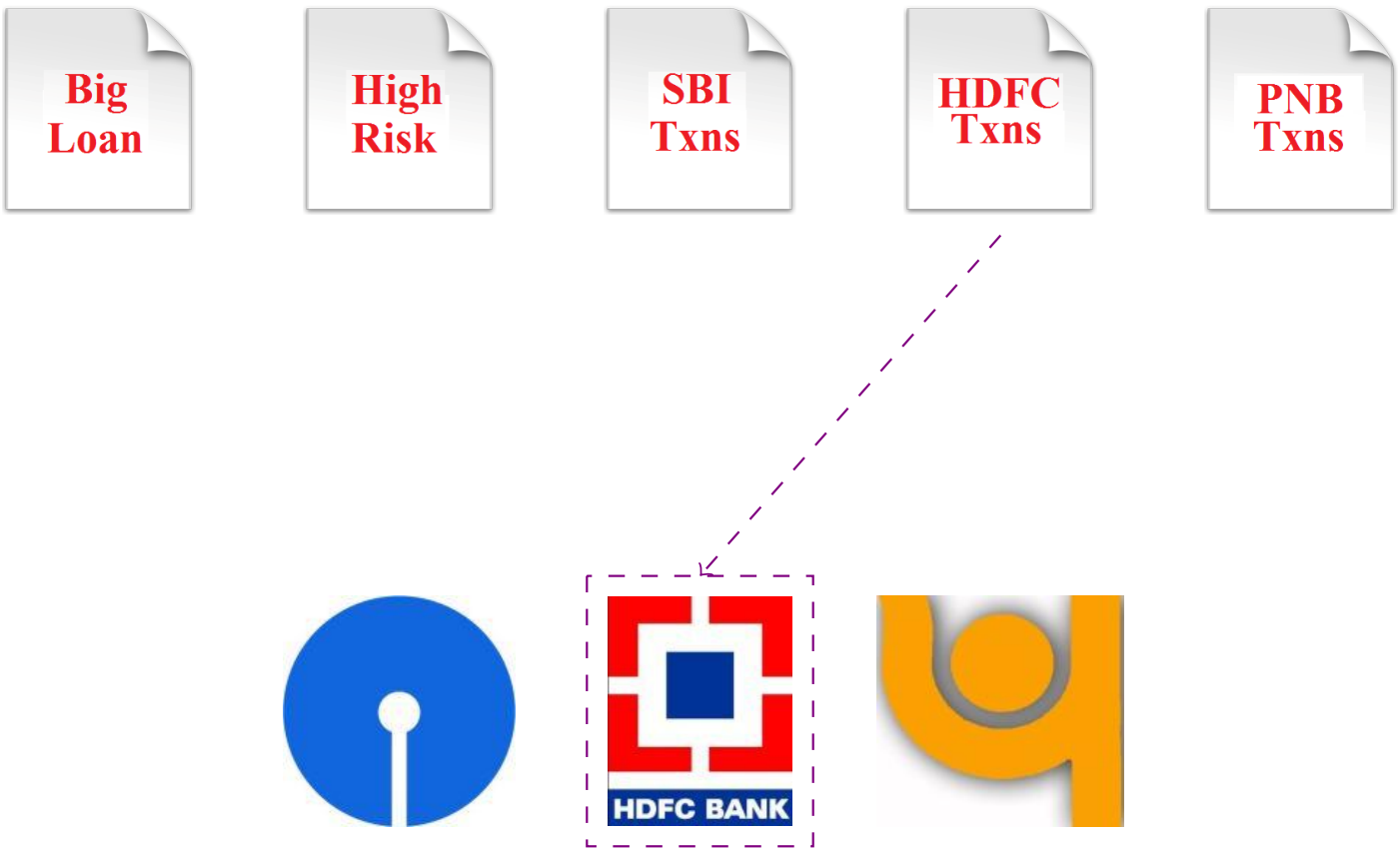
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.



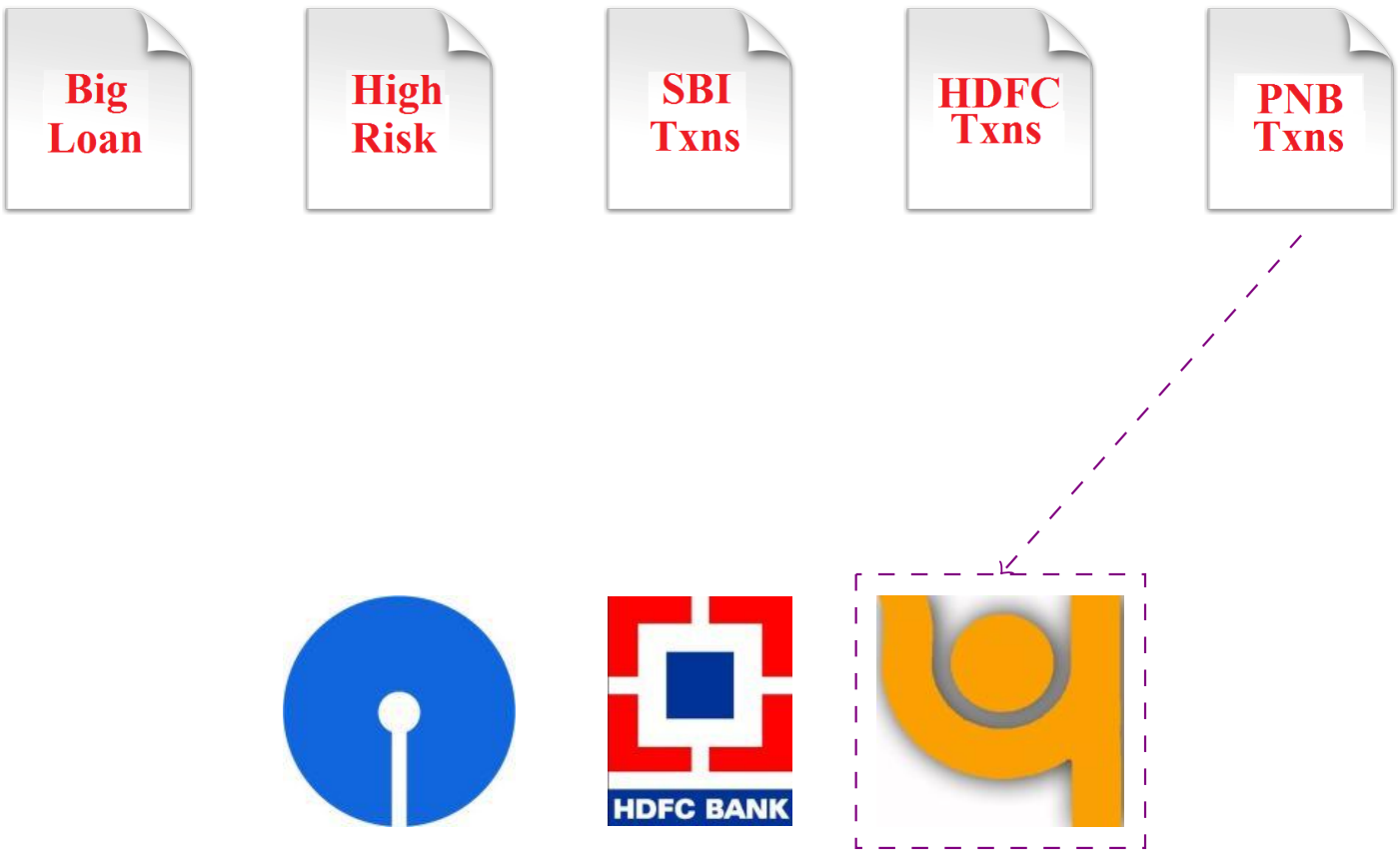
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.



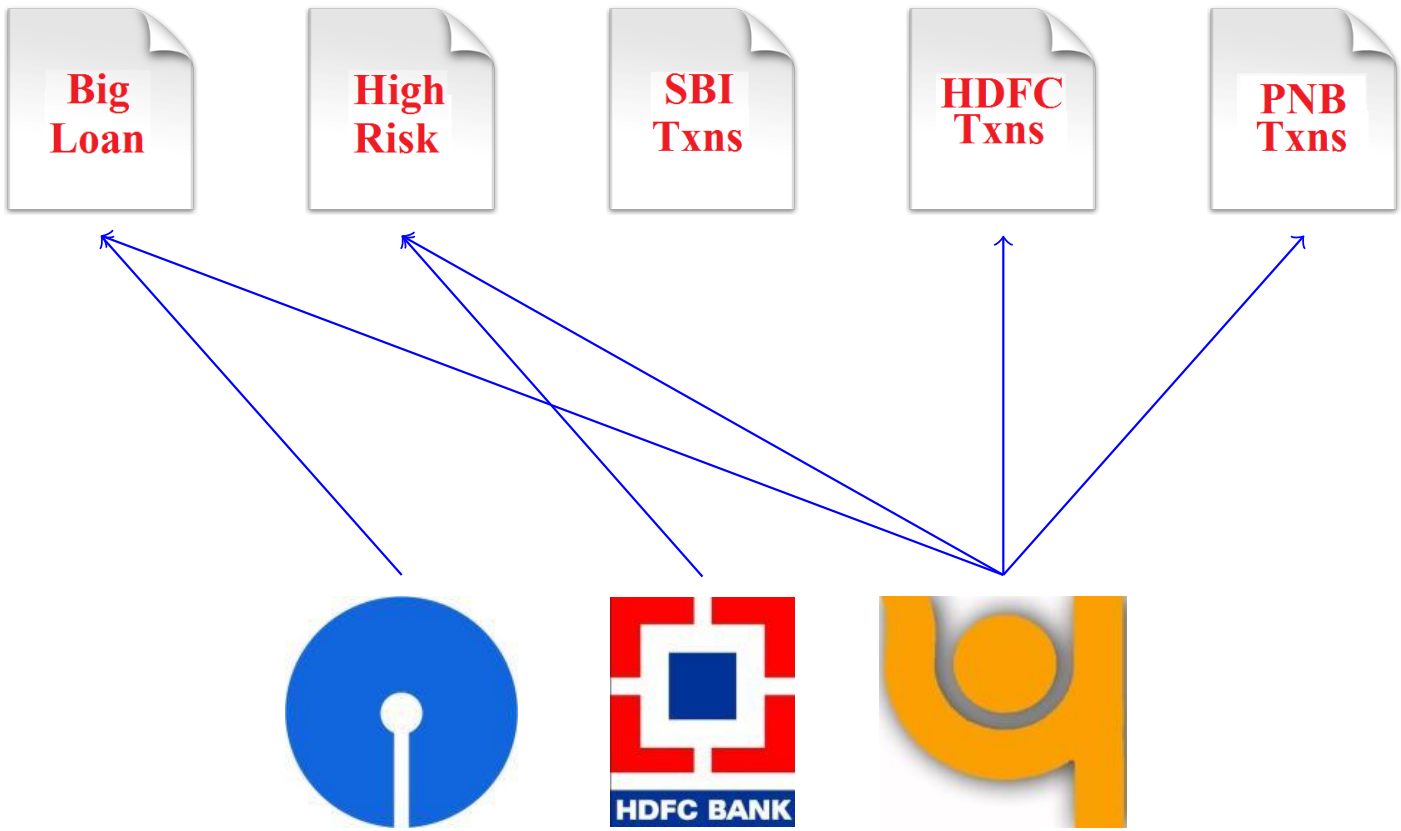
Problem: Broadcast Encryption with Keyword Search (BEKS)

- Let five files and three users.
- Each file has a privileged set of users.



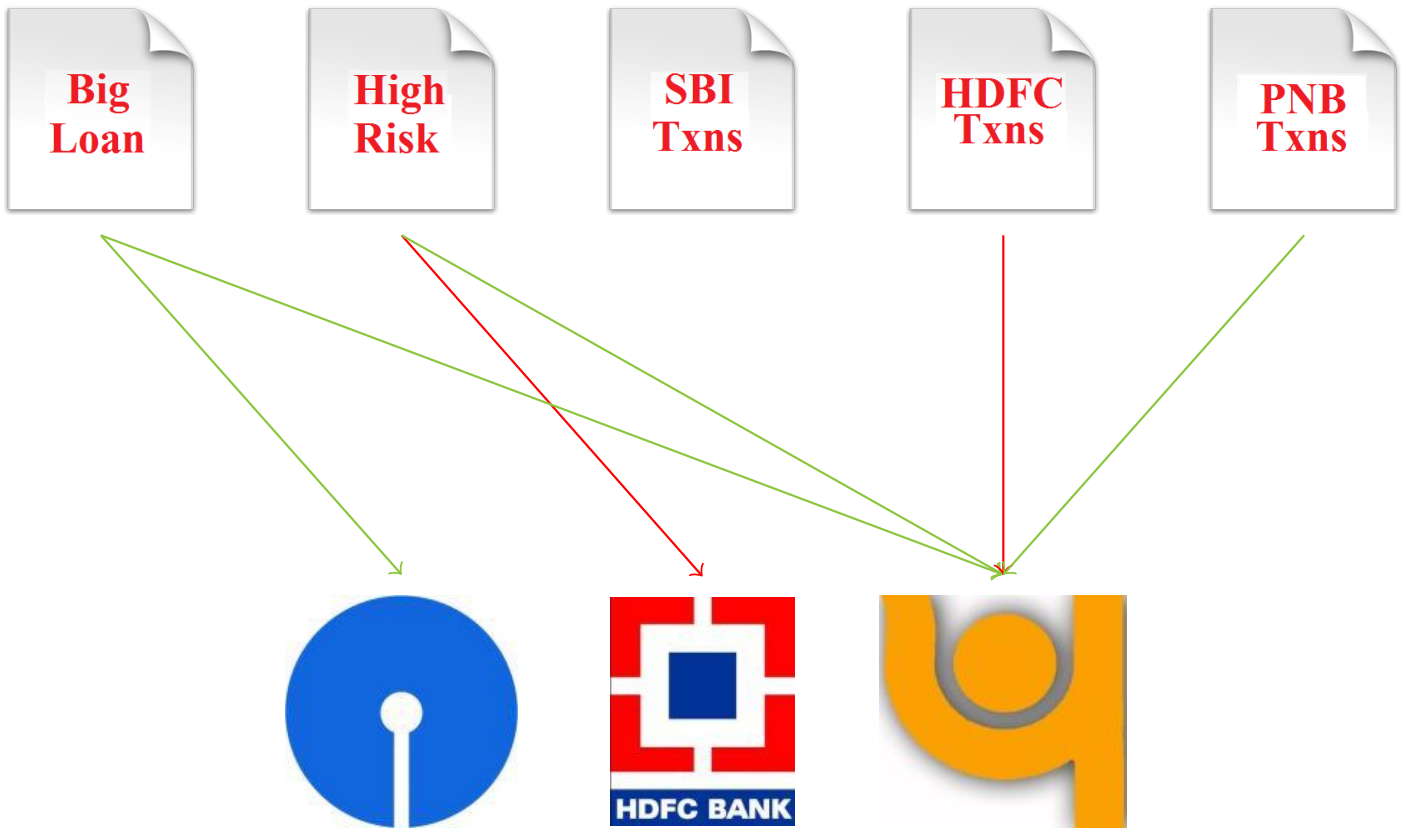
Problem: Broadcast Encryption with Keyword Search (BEKS)

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



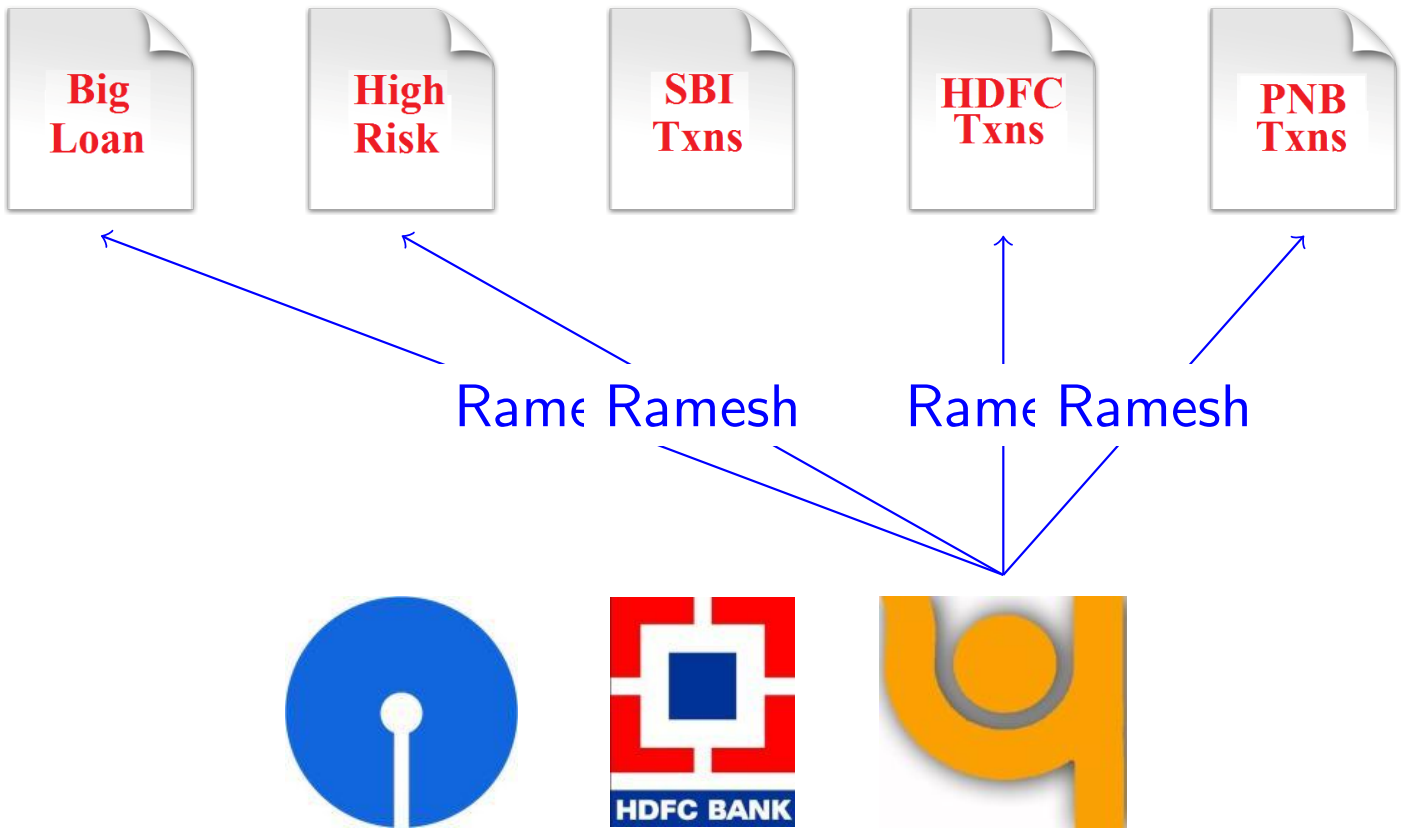
Problem: Broadcast Encryption with Keyword Search (BEKS)

- 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 $\perp$.



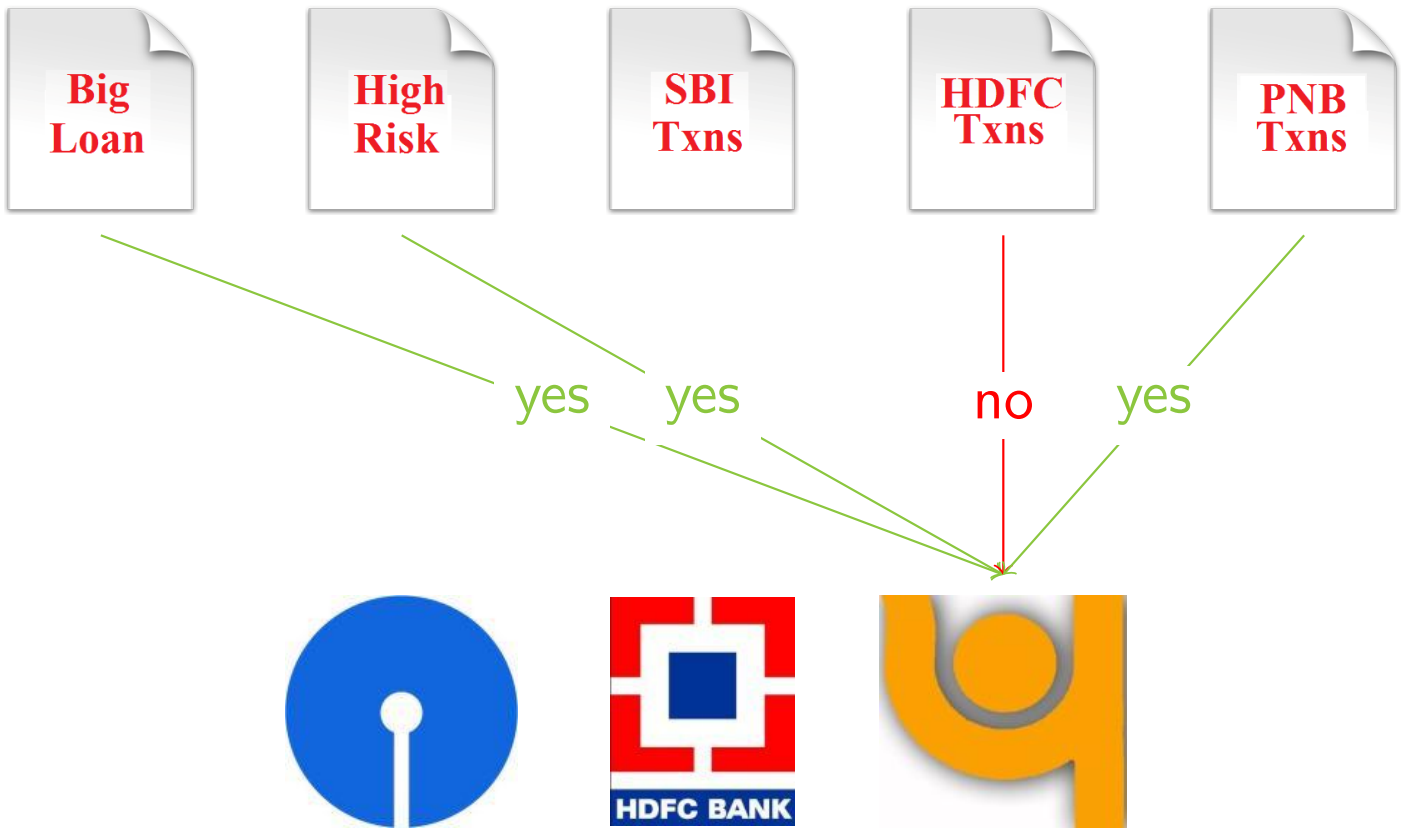
Problem: Broadcast Encryption with Keyword Search (BEKS)

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



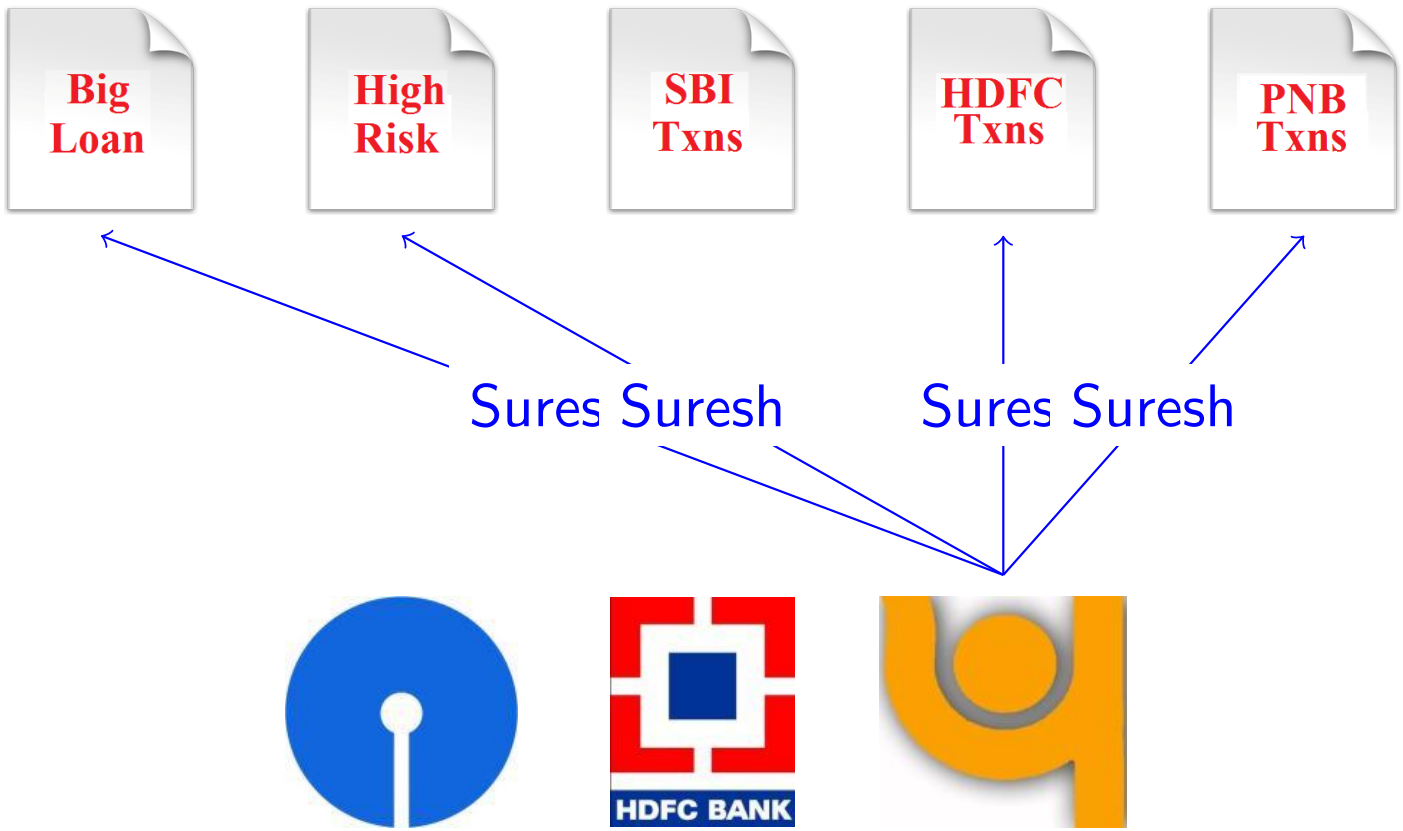
Problem: Broadcast Encryption with Keyword Search (BEKS)

- 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 $\perp$.



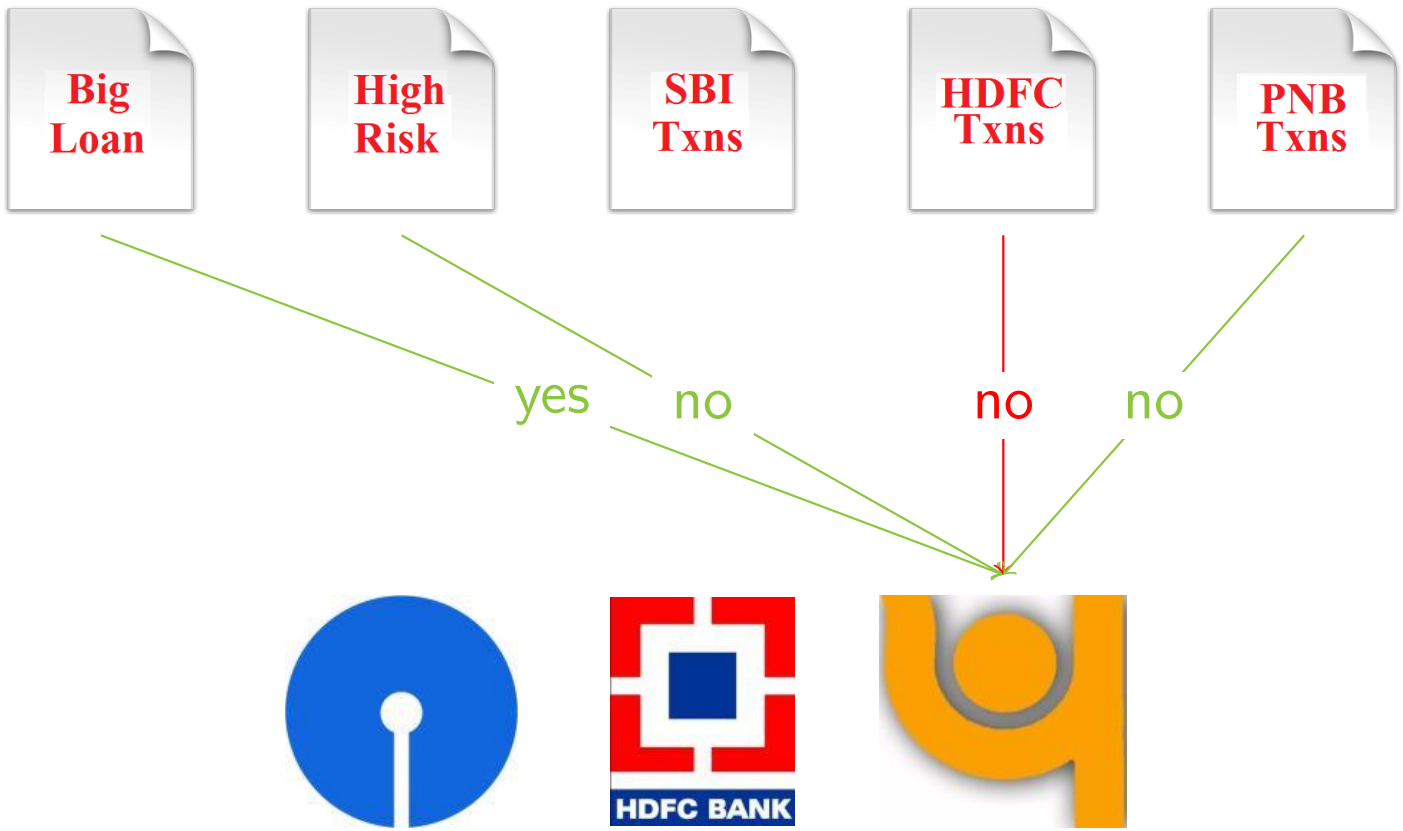
Problem: Broadcast Encryption with Keyword Search (BEKS)

- 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 $\perp$.



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 and keyword in file, gets back yes. Else no.