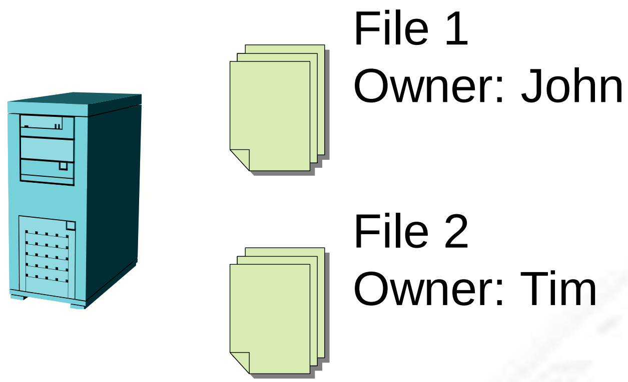
Directly Revocable Key-Policy Attribute-Based Encryption with Verifiable Ciphertext Delegation

Yanfeng Shi, Qingji Zheng, Jiqiang Liu, Zhen Han
Beijing Jiaotong University

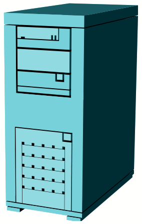
Traditional Encrypted Filesystem

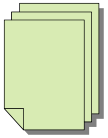


- Encrypted Files stored on Untrusted Server
- Every user can decrypt its own files

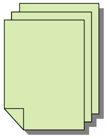
- Files to be shared across different users? Credentials?

Key-Policy Attribute-Based Encryption





File 1
- "Creator: John"
- "Computer Science"
- "Admissions"
- "Date: 04-11-06"



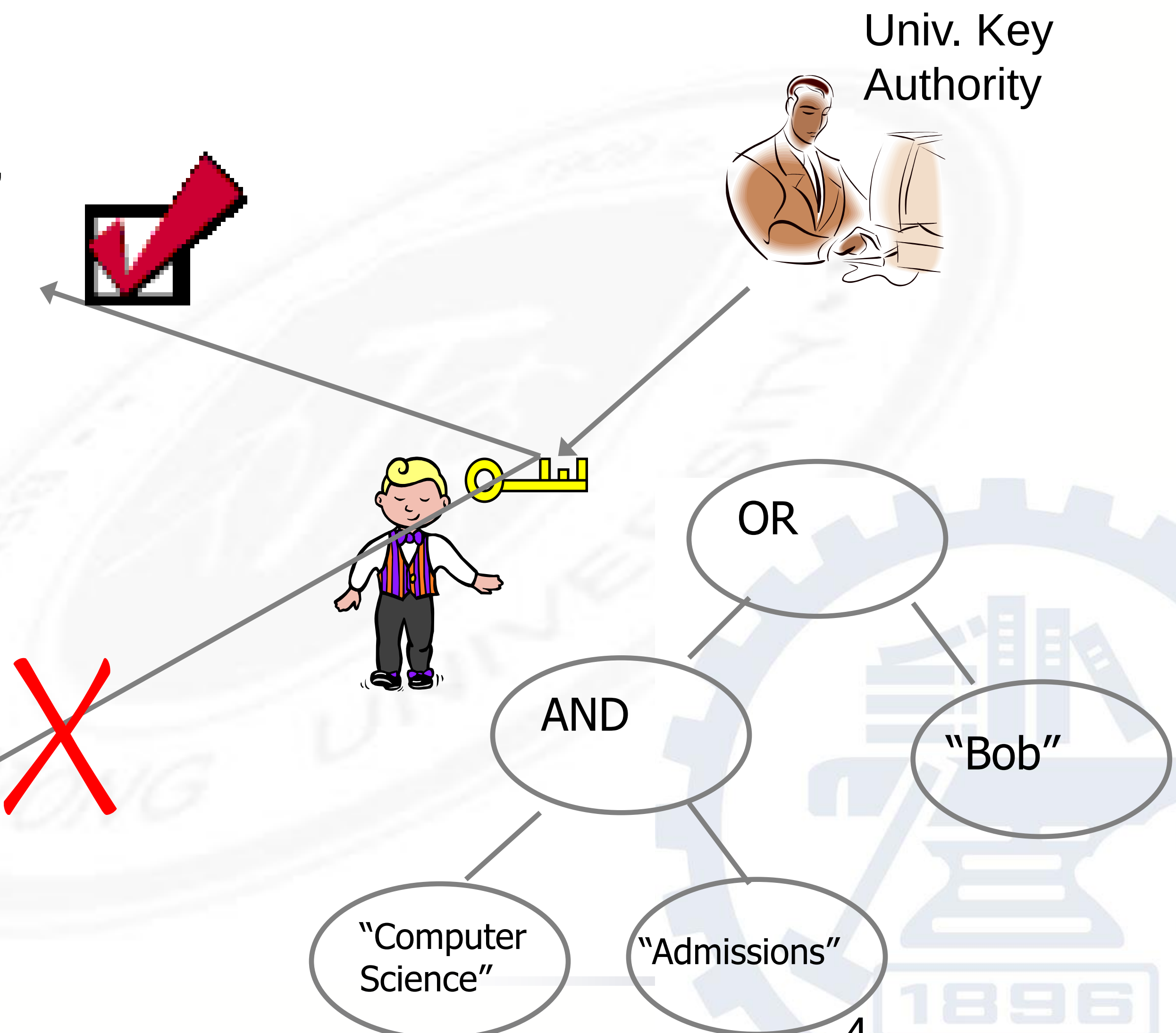
File 2
- "Creator: Tim"
- "History"
- "Admissions"
- "Date: 03-20-05"

➢ Label files with attributes

Key-Policy Attribute-Based Encryption

File 1
- “Creator: John”
- “Computer Science”
- “Admissions”
- “Date: 04-11-06”

File 2
- “Creator: Tim”
- “History”
- “Admissions”
- “Date: 03-20-05”



Our Work (1/4): Revocation

- Guarantees
- ○ Non-revoked users can decrypt data.
- ○ Revoked users can't decrypt data added in the future.
- ○ Revoked users can't decrypt data in the past. [22] (After termination, employee shouldn't be able to access anything he doesn't already have)

[22] A. Sahai, H. Seyalioglu, B. Waters, Dynamic credentials and ciphertext delegation for attribute-based encryption, in: CRYPTO, 2012, pp. 199-217.

Our Work (2/4): Revocation

- Non-revoked users can decrypt data.
- Revoked users can’t decrypt data added in the future.
  - ✓ Direct mode: no need to update non-revoked users’ decryption keys.
  - ✓ Indirect mode: need to update all the non-revoked users’ decryption keys.

Our Work (3/4): Revocation

- Revoked users can't decrypt data in the past.
  - ✓ Update the past encrypted data
    - Traditional way: The data owner must download, decrypt, re-encrypt and upload the data stored in the cloud.
    - Outsourcing to cloud: The cloud update the encrypted data- "ciphertext delegation".
      - Unverifiable: the process can't be accountable.
      - Verifiable: the process can be accountable.

Our Work (4/4): Revocation

Scheme	Direct revocation	Ciphertext delegation	Update verifiability	Security assumption
Scheme 1 [2]	√	×	×	n-BDHE
Scheme 2 [2]	√	×	×	r-MEBDH
[1]	√	×	×	DBDH
[5]	×	×	×	DBDH
[22]	×	√	×	three static assumptions
Our solution	√	√	√	$(d+3)$-MDDH

Table 1: Property summary for revocable KPABE schemes in the literature and the solution in this paper. Direct revocation means that the trusted authority can solely update revocation list and there is no need to update non-revoked users' decryption key. Ciphertext delegation means that ciphertexts can be updated by the third party correspondingly when the revocation list is updated. Update verifiability means that the process of the third party updating ciphertexts can be accountable.

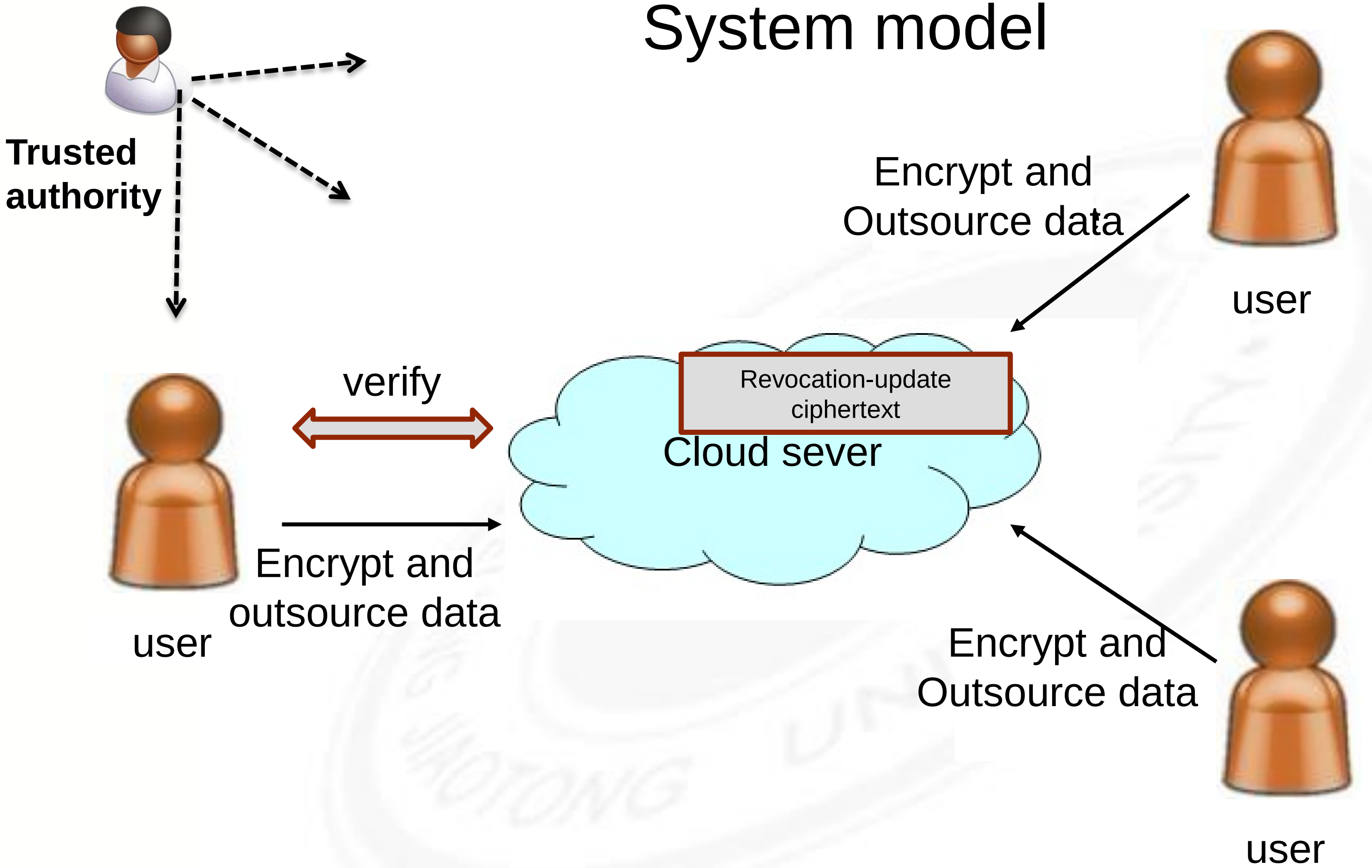
[1] N. Attrapadung, H. Imai, Attribute-based encryption supporting direct/indirect revocation modes, in: IMA Int. Conf., 2009, pp. 278-300.

[2] N. Attrapadung, H. Imai, Conjunctive broadcast and attribute-based encryption, in: Pairing-Based Cryptography–Pairing 2009, Springer, 2009, pp. 248–265.

[5] A. Boldyreva, V. Goyal, V. Kumar, Identity-based encryption with efficient revocation, in: ACM Conference on Computer and Communications Security, 2008, pp. 417-426.

[22] A. Sahai, H. Seyalioglu, B. Waters, Dynamic credentials and ciphertext delegation for attribute-based encryption, in: CRYPTO, 2012, pp. 199-217.

System model



Techniques

- Techniques
  - Multilinear Maps
  - Subset cover

Multilinear Maps

$d+3$: $(G_0, G_1, \dots, G_{d+2})$ order p

$d+2$ mappings $e_i: G_0 \times G_i \rightarrow G_{i+1}, i = 0, \dots, d+1$

Properties:

- ✓ Given generator $g_0 \in G_0$, then $g_{i+1} = e_i(g_0, g_i)$ is the generator of G_{i+1}
- ✓ $e_i({g_0}^{\alpha}, {g_i}^{\beta}) = e_i(g_0, g_i)^{\alpha\beta}$
- ✓ e_i can be efficiently computed

Subset Cover

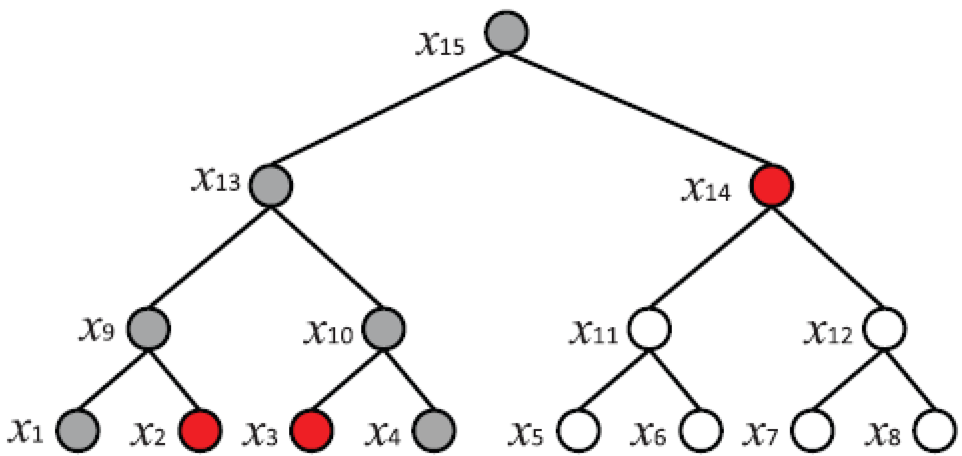


Figure 1: Subset cover technique to encode the revocation list. Given the revocation list $\mathsf{R} = \{x_1, x_4\}$, the nodes of $\mathsf{path}(x_1)$ and $\mathsf{path}(x_4)$ are marked (in gray color), and then $\mathsf{cover}(\mathsf{R}) = \{x_2, x_3, x_{14}\}$ (in red color).

System Setup

$\alpha, \beta \xleftarrow{R} Z_p, H_1: \{0,1\}^* \rightarrow Z_p, H_2: \{0,1\}^* \rightarrow G_0,$

$h_j \xleftarrow{R} G_{d+1}, 0 \leq j \leq max, max$ is the maximum number of attributes. Define $Q(y) = \prod_{j=0}^{max} \left(h_j^{y^j}\right), y \in Z_p$

pm=$(e_0, \dots, e_{d+1}, G_0, \dots, G_{d+2}, g_0, \dots, g_{d+2}, e_{d+1}(g_0, g_{d+1})^{\alpha}, g_0^{\beta}, H_1, H_2, h_0, \dots, h_{max})$

mk=(α, β)

北京交通大學
BEIJING JIAOTONG UNIVERSITY

Key Generation

mk=(α,β)

(1) $\alpha_1, v_2, v_3, \ldots, v_k \xleftarrow{R} Z_p$, set α_2, $s.t. \alpha = \alpha_1 + \alpha_2 \bmod p$.

(2) $\mathbf{v} = (\alpha_1, v_2, v_3, \ldots, v_k)$,

for $i = 1, \ldots, l$, compute $\lambda_{\pi(i)} = \mathrm{M}_i \cdot \mathbf{v}$, M is an $l \times k$ matrix;

$$D_i^{(1)} = g_{d+1}^{\lambda_{\pi(i)}} Q(H_1(\pi(i)))^{r_i}, D_i^{(2)} = g_0^{r_i}, \; r_i \xleftarrow{R} Z_p;$$

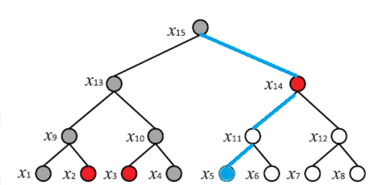
Access control policy

(3) Let $P_{x_{i_0}} = e_0(H_2(x_{i_0}), g_0^{\beta})$, compute $P_{x_{i_j}} = e_j(H_2(x_{i_j}), P_{x_{i_{j-1}}})$,

where $j = 1, \ldots, d$. (path(uid)=$\{x_{i_0}, \ldots x_{i_d}\}$)

Path(uid) is recorded using mullinear maps

$$D^{(3)} = g_{d+1}^{\alpha_2} P_{\mathrm{uid}}^{t}, D^{(4)} = g_0^{t}, \; t \xleftarrow{R} Z_p;$$



sk$=$(uid, (M,π), $(D_i^{(1)}, D_i^{(2)})_{i\in[1,l]}, D^{(3)}, D^{(4)}$)

北京交通大學
BEIJING JIAOTONG UNIVERSITY

Encryption

pm=$(e_0, \dots, e_{d+1}, G_0, \dots, G_{d+2}, g_0, \dots, g_{d+2}, e_{d+1}(g_0, g_{d+1})^{\alpha}, g_0{}^{\beta}, H_1, H_2, h_0, \dots, h_{max})$

Encrypt the massage m $\in G_{d+2}$ under attribute set S

(1) $C^{(1)} = \mathrm{m}e_{d+1}(g_0, g_{d+1})^{\alpha s}, C^{(2)} = g_0^s, s \overset{R}{\leftarrow} Z_p;$

The Attribute set

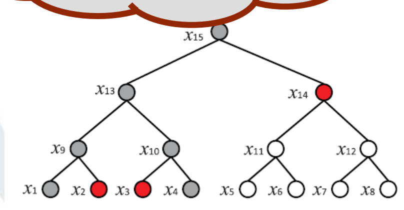
(2) $\boxed{C_{\mathrm{at}}^{(3)} = Q(H_1(\mathrm{at}))^{s}, \mathrm{at} \in \mathrm{S};}$

(3) path$(x) = \{x_{i_0}, \dots, x_{i_{depth(x)}}\}, x_{i_0} = root \wedge x_{i_{depth(x)}} = x, x \in$ $cover(\mathrm{R}), cover(\mathrm{R})$ is the cover set of revocation list R.

$P_{x_{i_0}} = e_0(H_2(x_{i_0}), g_0^{\beta})$, compute $P_{x_{i_j}} = e_j(H_2(x_{i_j}), P_{x_{i_{j-1}}})$,

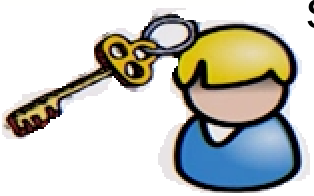
where $j = 1, \dots, dept(x)$, set $\boxed{C_x^{(4)} = P_x^s.}$

The set cover nodes



cph=(S,R,$C^{(1)}, C^{(2)}, \{C_{\mathrm{at}}^{(3)}\}_{\mathrm{at}\in\mathrm{S}}, \{C_x^{(4)}\}_{x\in cover(\mathrm{R})}$)

Decryption Part I



sk $= (\mathrm{uid}, (\mathrm{M},\pi), (D_i^{(1)}, D_i^{(2)})_{i\in[1,l]}, D^{(3)}, D^{(4)})$

For each “satisfied” node (uid $\notin$ R $\wedge$ S satisfies (M,π)) perform a computation

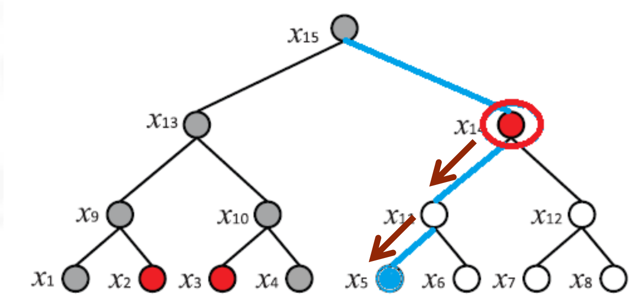
(1) With uid $\notin$ R, exist a node x $s.t.$ $x \in (\mathrm{path(uid)} \cap \mathrm{cover(R)})$, *suppose*

$$\mathrm{path(uid)} = \left\{x_{i_0}, \dots, x_{i_{depth(x)}}, \dots, x_{i_d}\right\}, x_{i_d} = \mathrm{uid} \wedge x_{i_{depth(x)}} = x;$$

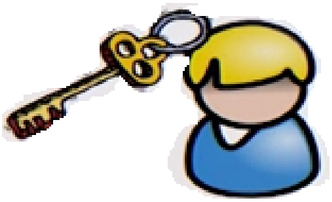
(2) Let $P'_{x_{i_{depth(x)}}} = C_x^{(4)}$, compute $P'_{x_{i_{j+1}}} = e_{j+1}(H_2\left(x_{i_{j+1}}\right), P'_{x_{i_j}})$ for $j = depth(x), \dots, d-1$;

(3) $P'_{uid} = P'_{x_{i_d}}$

Extend ciphertext of x to uid (using multilinear maps)



Decryption Part II



(4) S satisfies (M,π), exist c_i, $s.t.$ $\sum_{\pi(i)\in S} c_i \mathrm{M}_i$=(1,0,…,0), then

$$K = \prod_{\pi(i)\in S} \underbrace{\boxed{\left(\frac{e_{d+1}(C^{(2)}, D_i^{(1)})}{e_{d+1}(D_i^{(2)}, C_{\pi(i)}^{(3)})}\right)^{c_i}}}_{\text{S satisfies (M, } \pi\text{)}} \cdot \underbrace{\boxed{\frac{e_{d+1}(C^{(2)}, D^{(3)})}{e_{d+1}(D^{(4)}, P'_{uid})}}}_{\text{uid} \notin \text{R}}$$

(5) $m = C^{(1)} / K.$

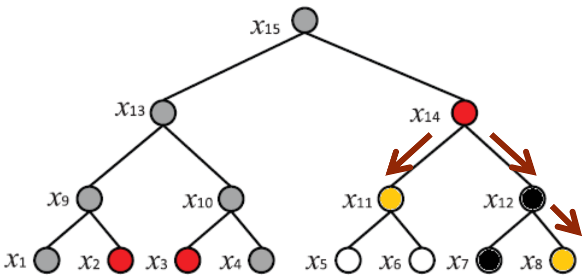
Update

Given a new revocation list R’, update as follows:

- If exists $x \in \mathrm{cover}(\mathrm{R})\ s.t.\ x = x'$, set $\tilde{C}^{(4)}_{x'} = C^{(4)}_{x}$;
- Otherwise exists $x \in \mathrm{cover}(\mathrm{R})\ s.t.\ x$ is an ancestor of x',$\mathrm{path}(x') = \mathrm{path}(x) \cup \{x_{i_{depth(x)}}, \ldots, x_{i_{depth(x')}}\}$, where $x_{i_{depth(x)}} = x, x_{i_{depth(x')}} = x'$, set $P'_{x_{i_{j+1}}} = e_{j+1}\left(H_2\left(x_{i_{j+1}}\right), P'_{x_{i_j}}\right)$ and $\tilde{C}^{(4)}_{x'} = P'_{x'}$;
- Let $\tilde{C}^{(1)} = C^{(1)}, \tilde{C}^{(2)} = C^{(2)}, \tilde{C}^{(3)}_{\mathrm{at}} = C^{(3)}_{\mathrm{at}}$.

Updated ciphtext:

$$\text{cph'} = (S, R', \tilde{C}^{(1)}, \tilde{C}^{(2)}, \{\tilde{C}^{(3)}_{\mathrm{at}}\}_{\mathrm{at} \in S}, \{\tilde{C}^{(4)}_{x'}\}_{x' \in \mathrm{cover}(R')})$$



X_7 is the new revoked user, the ciphertext part for x_{14} needs to update to x_{11} and x_8

北京交通大学
BEIJING JIAOTONG UNIVERSITY

Verification

Verify the following equations:

$$C^{(1)} = \tilde{C}^{(1)}, C^{(2)} = \tilde{C}^{(2)}$$

$$\forall \mathrm{at} \in S, \tilde{C}^{(3)}_{\mathrm{at}} = C^{(3)}_{\mathrm{at}},$$

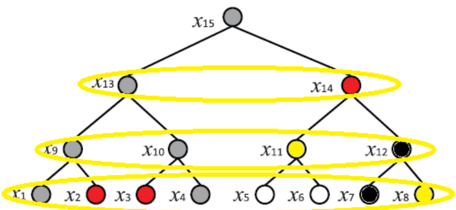
$$\forall \mathrm{x} \in \mathrm{cover(R)} \cap \mathrm{cover(R')}, C^{(4)}_{x} = \tilde{C}^{(4)}_{x}$$

Check each level

If hold, proceed to verify whether $\exists i,\ s.t.$

$$\boxed{e_{depth(x')+1}\left(C^{(2)}, \prod_{i=1}^{\eta} P_{x'_i}^{c_i}\right) = e_{depth(x')+1}\left(g_0, \prod_{i=0}^{\eta} (\tilde{C}^{(4)}_{x'})^{c_i}\right),}$$

where $c_1, \ldots, c_\eta \xleftarrow{R} Z_p, x'_j \in \mathrm{cover(R')} - \mathrm{cover(R)}$,and $depth(x'_j) = i, i = 1, \ldots, d.$



Thanks!