Liqiang WU, Yiliang HAN, Xiaoyuan YANG, Minqing ZHANG, 2022. Identity-based threshold proxy re-encryption scheme from lattices and its applications. *Frontiers of Information Technology & Electronic Engineering*, 23(2):258-277. https://doi.org/10.1631/FITEE.2000366

# Identity-based threshold proxy re-encryption scheme from lattices and its applications



**Key words:** Post-quantum cryptography; Threshold proxy re-encryption; Lattices; Robustness; Decentralization

Corresponding author: Yiliang HAN
E-mail: hanyil@163.com
ORCID: https://orcid.org/0000-0002-2116-5408

# Motivation

- From the perspective of practice, centralized proxy re-encryption (PRE) is unsatisfactory. First, the centralized proxy server may not be constantly online; Second, a single proxy node in complete control of the whole re-encryption key may develop into an attractive attack target; Third, the single proxy server causes significant bottlenecks in efficiency.
- From a theoretical perspective, the development of quantum computers has posed serious threats to the security of public-key cryptography. So, PREs against quantum attack are urgently needed.



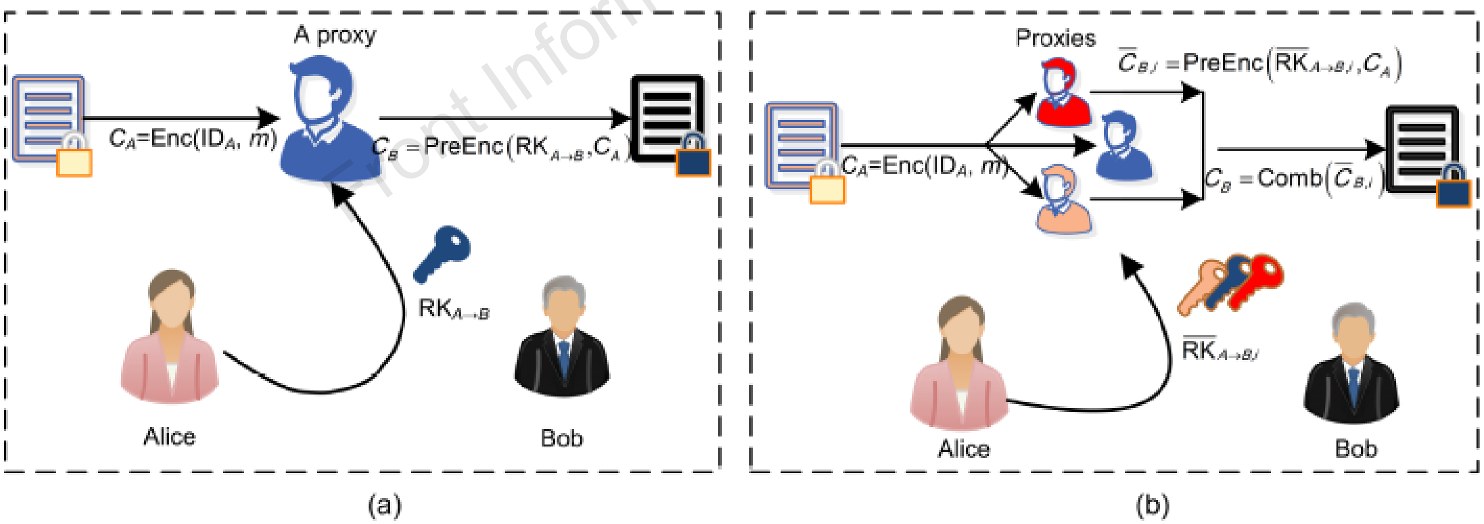


**Fig. 1 Ciphertext conversion in IB-PRE (a) and IB-TPRE (b)**

# Contributions

- We employ Shamir's secret sharing twice. First, it splits a public vector into multiple feature vectors attached to each proxy server. Second, it splits the re-encryption key. The homomorphism between the re-encryption key shares and ciphertext shares makes the overall ciphertext conversion hold, which effectively decentralizes the proxy power.
- A homomorphic signature from lattices is applied to make our scheme robust. Robustness means that a forged or wrong ciphertext share can be detected immediately. Our scheme realizes robustness using the unforgeability of homomorphic signatures.
- Our scheme has many practical applications because of its high availability, low trust, and strong security. Our scheme can be applied in access control in a decentralized environment, such as a file-sharing system based on a blockchain network and a robust key escrow system with threshold cryptography.

# Description of the IB-TPRE scheme (1)

**3.1 Key generation algorithms**

1. Setup($1^\lambda$)

Generate a uniformly distributed matrix $A_0 \in \mathbb{Z}_q^{n\times m}$ with its trapdoor $T_{A_0} \in \mathbb{Z}_q^{m\times m}$ by algorithm TrapGen. Choose two uniformly distributed matrices $A_1 \in \mathbb{Z}_q^{n\times m}$ and $B \in \mathbb{Z}_q^{n\times m}$ and a vector $u \in \mathbb{Z}_q^n$. The user dictionary UD and delegation dictionary DD are initialized by empty sets. Publish the public key PP={$A_0$, $A_1$, $B$, $u$} and keep the master private key $\mathrm{MSK} = \{T_{A_0}\} \in \mathbb{Z}_q^{m\times m}$.

2. KeyGen(MSK, id)

When a user id submits a request to PKG for a private key, check the user directory UD first. If there is already a private key record for the user id, retrieve and return it to the user. Otherwise, we perform the following:

(1) Convert the identity id into $H(\mathrm{id}) \in \mathbb{Z}_q^{n\times n}$, obtain the corresponding public key $F_{\mathrm{id}}=(A_0|A_1+H(\mathrm{id})B)$, and calculate the private key $e_{\mathrm{id}} \in \psi_\gamma^{2m}$ by SampleLeft($A_0$, $A_1+H(\mathrm{id})B$, $T_{A_0}$, $u$, $\psi_\gamma$) satisfying $F_{\mathrm{id}}e_{\mathrm{id}}=u$.

(2) Obtain $T_{\mathrm{id}}$=SampleBasisLeft($A_0$, $A_1+H(\mathrm{id})B$, $T_{A_0}$, $s_t$) satisfying $F_{\mathrm{id}}T_{\mathrm{id}}=0$, where $s_t$ is the Gaussian parameter used to generate the user trapdoor.

Return private key $\mathrm{SK_{id}}=(e_{\mathrm{id}}, T_{\mathrm{id}})$ to id and add it to UD.

3. ReKeyGen ($\mathrm{id}_1$, $\mathrm{id}_2$, $\mathrm{SK}_{\mathrm{id}_1}$, $N$, $k$)

On generating the re-encryption between $\mathrm{id}_1$ and $\mathrm{id}_2$, check the delegation directory DD first. If re-encryption key shares are already contained in the directory, distribute these shares to the corresponding proxy server and then exit. Otherwise, compute all the re-encryption key shares {$\mathrm{kFrag}_i$} ($1\le i\le N$) as follows:

(1) Run Shamir's secret sharing to split $u \in \mathbb{Z}_q^n$ to $\overline{u}_i \in \mathbb{Z}_q^n$ ($1\le i\le N$) as feature vectors for each proxy server indexed by $i$, and store them locally (this process needs to be performed only once). For $i\in\{1, 2, \ldots, N\}$, let the feature vector of the $i^{\text{th}}$ proxy server be $\overline{u}_i \in \mathbb{Z}_q^n$, and $\overline{e}_{i,\mathrm{id}_1} \leftarrow \mathrm{SamplePre}\left(F_{\mathrm{id}_1}, T_{\mathrm{id}_1}, \overline{u}_i, \psi_\gamma\right)$ satisfy $F_{\mathrm{id}_1}\overline{e}_{i,\mathrm{id}_1} = \overline{u}_i$.

(2) Choose a matrix $R \in \psi_e^{(2ml)\times n}$, a vector $e \in \psi_e^{2ml}$, and a matrix $E \in \psi_e^{(2ml)\times(2m)}$. Then compute $P = RF_{\mathrm{id}_2} + E \in \mathbb{Z}_q^{(2ml)\times(2m)}$ and $Q = Ru + e \in \mathbb{Z}_q^{2ml}$, where $F_{\mathrm{id}_2} = \left(A_0 \mid A_1 + H(\mathrm{id}_2)B\right)$.

(3) Run Shamir's secret sharing on each coefficient from ($P$, $Q$) by the following:

For $i\in\{1, 2, \ldots, 2ml\}$ and $j\in\{1, 2, \ldots, 2m\}$, select a random polynomial $y_{i,j}(x) \in \mathbb{Z}_q[x]$ satisfying $y_{i,j}(0)=P_{i,j}$ with its degree equal to $k-1$. For $i\in\{1, 2, \ldots, 2ml\}$, choose a polynomial $w_i(x) \in \mathbb{Z}_q[x]$ satisfying $w_i(0)=Q_i$ with its degree equal to $k-1$.

# Description of the IB-TPRE scheme (2)

For $1 \le j \le N$, the partial decryption share for the $j^{\text{th}}$ proxy server is

$$(\bar{P}_j, \bar{Q}_j) = \begin{bmatrix} y_{1,1}(j) & \cdots & y_{1,2m}(j) & w_1(j) \\ y_{2,1}(j) & \cdots & y_{2,2m}(j) & w_2(j) \\ \vdots & & \vdots & \vdots \\ y_{2ml,1}(j) & \cdots & y_{2ml,2m}(j) & w_{2ml}(j) \end{bmatrix} \in \mathbb{Z}_q^{(2ml)\times(2m+1)}.$$

(4) The re-encryption key share for the $j^{\text{th}}$ proxy server is $\overline{\mathbf{RK}}_j = \left(\bar{P}_j, \bar{Q}_j - \text{Power2}\left(\bar{e}_{j,\text{id}_1}\right)\right)$ $(1 \le j \le N)$.

(5) Generate verification and signing keys (hsvk, hssk) by HS.KeyGen. Select $N$ keys $\text{prfk}_1, \text{prfk}_2, \ldots, \text{prfk}_N$ independently by the pseudo-random function $F_{\text{prfk}}$. For $i \in \{1, 2, \ldots, N\}$, set $X_i = (\overline{\mathbf{RK}}_i, \text{prfk}_i)$ and compute $\bar{\sigma}_i = \text{HS.Sign}(\text{hssk}, X_i)$.

Publish hsvk to verify the signature in the subsequent process. Send shares $\{\mathbf{kFrag}_i\} = \left\{\overline{\mathbf{RK}}_i, \text{prfk}_i, \bar{\sigma}_i\right\}$ $(1 \le i \le N)$ to the proxy server with index $i$ over a secure channel, and add them to DD.

**3.2 Encryption and decryption algorithms**

1. Encrypt(id, $M$)

To encrypt a bit $M \in \{0, 1\}$, it works as follows:

(1) On input of the user $\text{id} \in \mathbb{Z}_q^n$, encode it as $F_{\text{id}} = (A_0 \mid A_1 + H(\text{id})B) \in \mathbb{Z}_q^{n\times(2m)}$.

(2) Choose a uniform vector $s \in \mathbb{Z}_q^{n\times 1}$ randomly.

(3) Choose a uniform integer matrix $R_{\text{id}} \in \{-1, 1\}^{m\times m}$ randomly.

(4) Choose $x \in \psi_e$ and $y \in \psi_e^{1\times m}$, and then calculate $z = yR_{\text{id}} \in \psi_e^{1\times m}$.

(5) Calculate $c_1 = s^{\mathrm{T}} F_{\text{id}} + \eta(y \mid z) \in \mathbb{Z}_q^{1\times(2m)}$ and $c_2 = s^{\mathrm{T}} u + \eta x + M\lfloor q/2 \rfloor \in \mathbb{Z}_q$. Output ciphertext $C = (c_1, c_2) \in \mathbb{Z}_q^{2m+1}$.

2. Dec($C_{\text{id}}$, $e_{\text{id}}$)

On input of a ciphertext $C_{\text{id}} = (c_1, c_2)$ and the corresponding private key $e_{\text{id}}$, compute $M' = c_2 - c_1 e_{\text{id}}$. If $M' \in (\lfloor q/4 \rfloor, \lfloor 3q/4 \rfloor)$, set $M' = 1$; otherwise, $M' = 0$.

**3.3 Ciphertext share processing algorithms**

1. $\text{PreEnc}\left(C_{\text{id}_1}, \{\text{kFrag}_i\}\right)$

The input contains a user $\text{id}_1$'s ciphertext $C_{\text{id}_1} = (c_{1,\text{id}_1}, c_{2,\text{id}_1})$ and a PRE key share $\{\mathbf{kFrag}_i\} = \left\{\overline{RK}_i = \left(\bar{P}_i, \bar{Q}_i - \text{Power2}(\bar{e}_{i,\text{id}_1})\right), \text{prfk}_i, \bar{\sigma}_i\right\}$.

(1) Compute $c'_{1,\text{id}_1} = \text{Bits}\left(c_{1,\text{id}_1}\right) \cdot \bar{P}_i \in \mathbb{Z}_q^{2m}$ and $c'_{2,\text{id}_2} = c_{2,\text{id}_2} + \text{Bits}\left(c_{1,\text{id}_1}\right) \cdot \left(\bar{Q}_i - \text{Power2}(\bar{e}_{i,\text{id}_1})\right) \in \mathbb{Z}_q$.

(2) Compute $(e'_1, e'_2) = F_{\text{prfk}_i}(c_{1,\text{id}_1}, c_{2,\text{id}_1}) \in \mathbb{Z}_r^{2m+1}$.

# Description of the IB-TPRE scheme (3)

(3) Compute $\bar{c}_{1,\mathrm{id}_2} = c'_{1,\mathrm{id}_2} + \eta e'_1$ and $\bar{c}_{2,\mathrm{id}_2} = c'_{2,\mathrm{id}_2} + \eta e'_2$. Set $\bar{C}_{i,\mathrm{id}_2} = \left(\bar{c}_{1,\mathrm{id}_2}, \bar{c}_{2,\mathrm{id}_2}\right)$.

(4) Homomorphically evaluate the signature $\bar{\sigma}_{i,\mathrm{id}_2} = \mathrm{HS.SignEval}\left(g_{C_{\mathrm{id}_1}}, \bar{\sigma}_i\right)$, where circuit $g_{C_{\mathrm{id}_1}}$ is defined as follows:

$$g_{C_{\mathrm{id}_1}}\left(\overline{\mathrm{RK}}_i = \left(\bar{P}_i, \bar{Q}_i - \mathrm{Power2}(\bar{e}_{i,\mathrm{id}_1})\right), \mathrm{prfk}_i\right)$$
$$= \left(\mathrm{Bits}\left(c_{1,\mathrm{id}_1}\right)\cdot \bar{P}_i, c_{2,\mathrm{id}_1} + \mathrm{Bits}\left(c_{1,\mathrm{id}_1}\right)\cdot\left(\bar{Q}_i\right.\right.$$
$$\left.\left.-\mathrm{Power2}(\bar{e}_{i,\mathrm{id}_1})\right)\right) + \eta\cdot F_{\mathrm{prfk}_i}(c_{1,\mathrm{id}_1}, c_{2,\mathrm{id}_1}) \in \mathbb{Z}_q^{2m+1}.$$

Output the ciphertext share $\left\{C_{\mathrm{cFrag},i}\right\} = \left\{C_{i,\mathrm{id}_2}, \bar{\sigma}_{i,\mathrm{id}_2}\right\}$.

2. $\mathrm{Verify}\left(\mathrm{hsvk}, C_{\mathrm{cFrag},i}\right)$

On input of the verification key hsvk and a ciphertext share $\left\{C_{\mathrm{cFrag},i}\right\} = \left\{\bar{C}_{i,\mathrm{id}_2}, \bar{\sigma}_{i,\mathrm{id}_2}\right\}$, the verification algorithm outputs $\mathrm{HS.Verify}\left(\mathrm{hsvk}, \bar{C}_{i,\mathrm{id}_2}, g_{C_{\mathrm{id}_1}}, \bar{\sigma}_{i,\mathrm{id}_2}\right)$.

3. $\mathrm{Comb}\left(C_{\mathrm{id}_1}, \left\{C_{\mathrm{cFrag},i}\right\}_{i\in S}\right)$

Assume that the participant set is $S$, and $k'=|S|$ represents the valid share size. If $k'<k$, output $\perp$; otherwise, calculate a whole ciphertext.

(1) For each ciphertext share $\{C_{\mathrm{cFrag},i}\}$ $(i\in S)$, check Verify(hsvk, $\{C_{\mathrm{cFrag},i}\}$). If the verification fails, output $\perp$ and exit.

(2) Parse $C_{\mathrm{cFrag},i} = \left\{\bar{C}_{i,\mathrm{id}_2}, \bar{\sigma}_{i,\mathrm{id}_2}\right\}$, and then take the proxy server's index $i$ and $\bar{C}_{i,\mathrm{id}_2} = \left(\bar{c}_{1,\mathrm{id}_2}, \bar{c}_{2,\mathrm{id}_2}\right)$. Calculate each Lagrange coefficient as

$$\lambda_i = \prod_{j\in S, i\neq j} \frac{-j}{i-j}.$$

At last, recover a whole ciphertext as

$$C_{\mathrm{id}_2} = \left(c_{1,\mathrm{id}_2}, c_{2,\mathrm{id}_2}\right)$$
$$= \sum_{i\in S}\lambda_i\left(\bar{c}_{1,\mathrm{id}_2}, \bar{c}_{2,\mathrm{id}_2}\right)_i + \left(0, \left(1-\sum_{i\in S}\lambda_i\right)c_{2,\mathrm{id}_1}\right).$$

# Description of the IB-TPRE scheme (4)

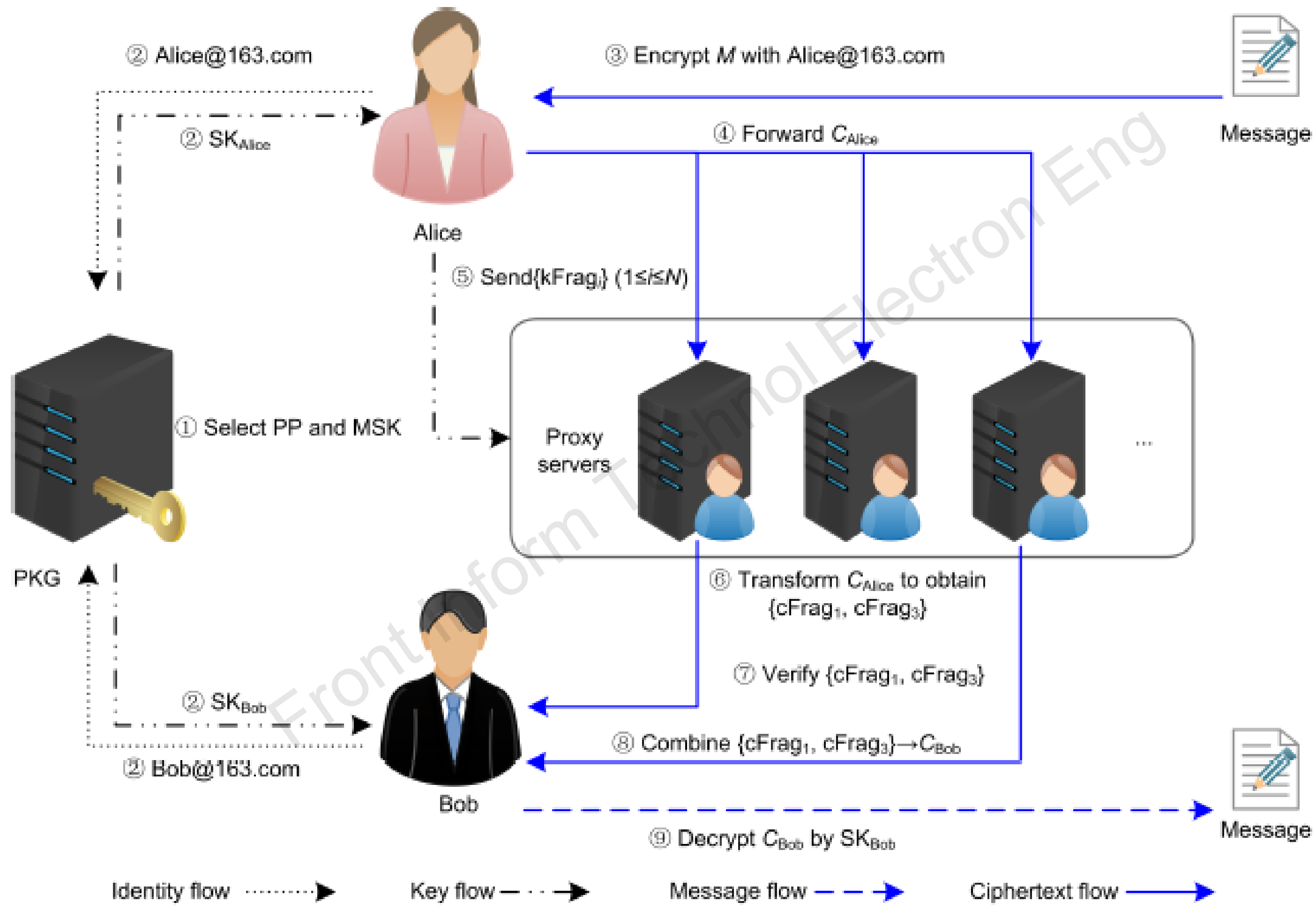


Fig. 2 Workflow of our proposed IB-TPRE scheme

# Analysis (1)

**Theorem 2** (Correctness) Assume $m=\lceil 6n\log_2 q \rceil$, $\sigma \geq \|\tilde{T}_{A_0}\|\omega\left(\sqrt{\log_2 n}\right)$, and $B_\gamma$ ($B_e$) the upper bound of the discrete Gaussian distribution $\psi_\gamma$ ($\psi_e$). The output of the pseudo-random function $F_{\mathrm{prfk}}$ is a uniform distribution with its upper bound $B_r$, and set $q \geq 8\max\left\{4ml, \sqrt{2m}k(N!)^3\right\}B_e B_\gamma + 8\sqrt{2m}k(N!)^3 B_r B_\gamma$.

The probability of successful decryption of our new IB-TPRE is close to 1 in a single hop.

**Theorem 3** (Security) The IB-TPRE system is IND-sID-CPA secure if the LWE assumption holds.

**Theorem 4** (Robustness) If a homomorphic signature scheme $\Pi_{\mathrm{HS}}$=(HS.KeyGen, HS.Sign, HS.SignEval, HS.Verify) satisfies unforgeability, the proposed IB-TPRE scheme is robust.

# Analysis (2)

**Table 1 Strategies to answer ReKeyGen queries**

| Delegator | Delegatee | User processing | Returned content |
|---|---|---|---|
| $\mathrm{id}_1 \notin (\Gamma_\mathrm{H} \cup \Gamma_\mathrm{C})$ | $\mathrm{id}_2 \in \Gamma_\mathrm{H}$<br>$\mathrm{id}_2 \notin (\Gamma_\mathrm{H} \cup \Gamma_\mathrm{C})$ | No action<br>Add $\mathrm{id}_2$ to $\Gamma_\mathrm{H}$ | Obtain $\mathbf{RK}_{\mathrm{id}_1} \leftarrow \mathrm{KeyGen}$ with $\boldsymbol{T_B}$, compute re-encryption key shares $\{\mathrm{kFrag}_i\}$ $(1 \le i \le N)$ by ReKeyGen, and return all the shares |
| | $\mathrm{id}_2 \in \Gamma_\mathrm{C}$ | Add $\mathrm{id}_1$ to $\Gamma_\mathrm{H}$ | Obtain $\mathbf{RK}_{\mathrm{id}_1} \leftarrow \mathrm{KeyGen}$ with $\boldsymbol{T_B}$, compute re-encryption key shares $\{\mathrm{kFrag}_i\}$ $(1 \le i \le N)$ by ReKeyGen, and return only the first $k-1$ shares |
| $\mathrm{id}_1 \in \Gamma_\mathrm{H}$ | $\mathrm{id}_2 \in \Gamma_\mathrm{H}$ | No action | If $\mathrm{id}_1 \ne \mathrm{id}^*$, obtain $\mathbf{RK}_{\mathrm{id}_1} \leftarrow \mathrm{KeyGen}$ with $\boldsymbol{T_B}$ and compute re-encryption key shares $\{\mathrm{kFrag}_i\}$ $(1 \le i \le N)$ by ReKeyGen. Otherwise, randomly generate a temporary $\mathbf{RK}_{\mathrm{id}_1 \to \mathrm{id}_2}$ and divide it into $N$ shares. At last, return all of them to $N$ proxy servers |
| | $\mathrm{id}_2 \in \Gamma_\mathrm{C}$<br>$\mathrm{id}_2 \notin (\Gamma_\mathrm{H} \cup \Gamma_\mathrm{C})$ | No action<br>No action | If $\mathrm{id}_1 \ne \mathrm{id}^*$, obtain $\mathbf{RK}_{\mathrm{id}_1} \leftarrow \mathrm{KeyGen}$ with $\boldsymbol{T_B}$ and compute re-encryption key shares $\{\mathrm{kFrag}_i\}$ $(1 \le i \le N)$ by ReKeyGen. Otherwise, randomly generate a temporary $\mathbf{RK}_{\mathrm{id}_1 \to \mathrm{id}_2}$ and divide it into $N$ shares. At last, return only the first $k-1$ shares |
| $\mathrm{id}_1 \in \Gamma_\mathrm{C}$ | | No action | $\perp$ |

# Applications

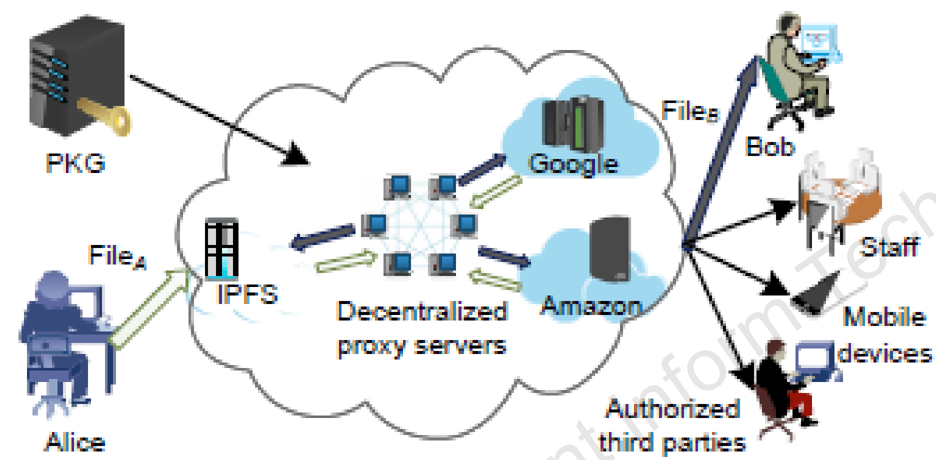


**Fig. 3  A file-sharing system based on a blockchain network**

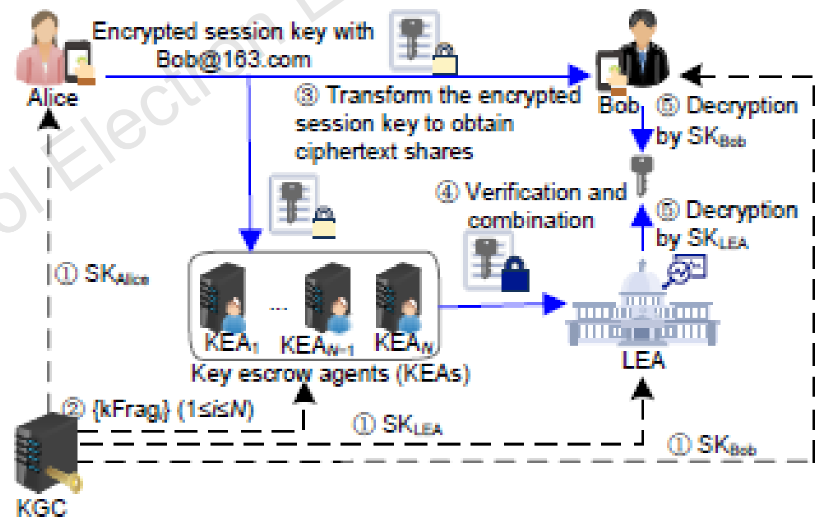


**Fig. 4  A robust key escrow system with threshold cryptography**

# Conclusions

- We propose an IB-TPRE over lattices by combining the threshold proxy re-encryption, identity-based encryption, and homomorphic signature.
- Our IB-TPRE has the advantages of high availability, low trust, and strong security.
- Our IB-TPRE provides encryption and cryptographic access control performed by a decentralized network.

吴立强，硕士，讲师。2012年毕业于武警工程大学，现于国防科技大学攻读博士学位。主要研究方向为基于格的密码学和可证明安全理论。



韩益亮，博士，武警工程大学教授，博士生导师。1999年毕业于武警工程学院，2012年于西安交通大学获博士学位。从事密码与信息安全教学科研工作，主要研究领域包括公钥密码、网络舆情分析等，发明了广义签名技术。主持国家和省级科研课题10余项，发表论文90余篇，获省部级奖励15项。2016年获中国科协“求是”杰出青年实用工程奖。