Multi-Key Homomorphic Encryption with Threshold Re-Encryption

Akira Nakashima[1], Yukimasa Sugizaki[1], Hikaru Tsuchida[2], Takuya Hayashi[1], Koji Nuida[3], Toshiyuki Isshiki[1], and Kengo Mori[1]

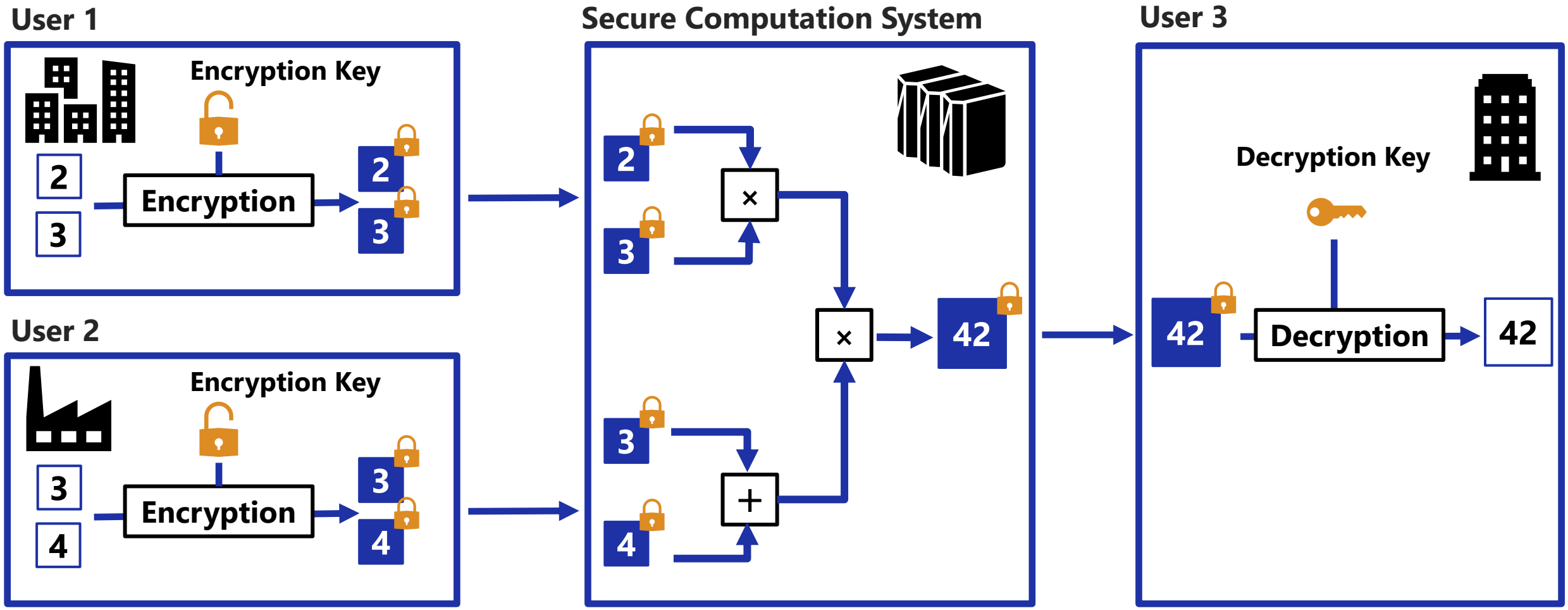
[1] : NEC Corporation [2] : Saitama Institute of Technology [3] : Kyushu University

Table of Contents

Introduction

Fully Homomorphic Encryption (FHE)

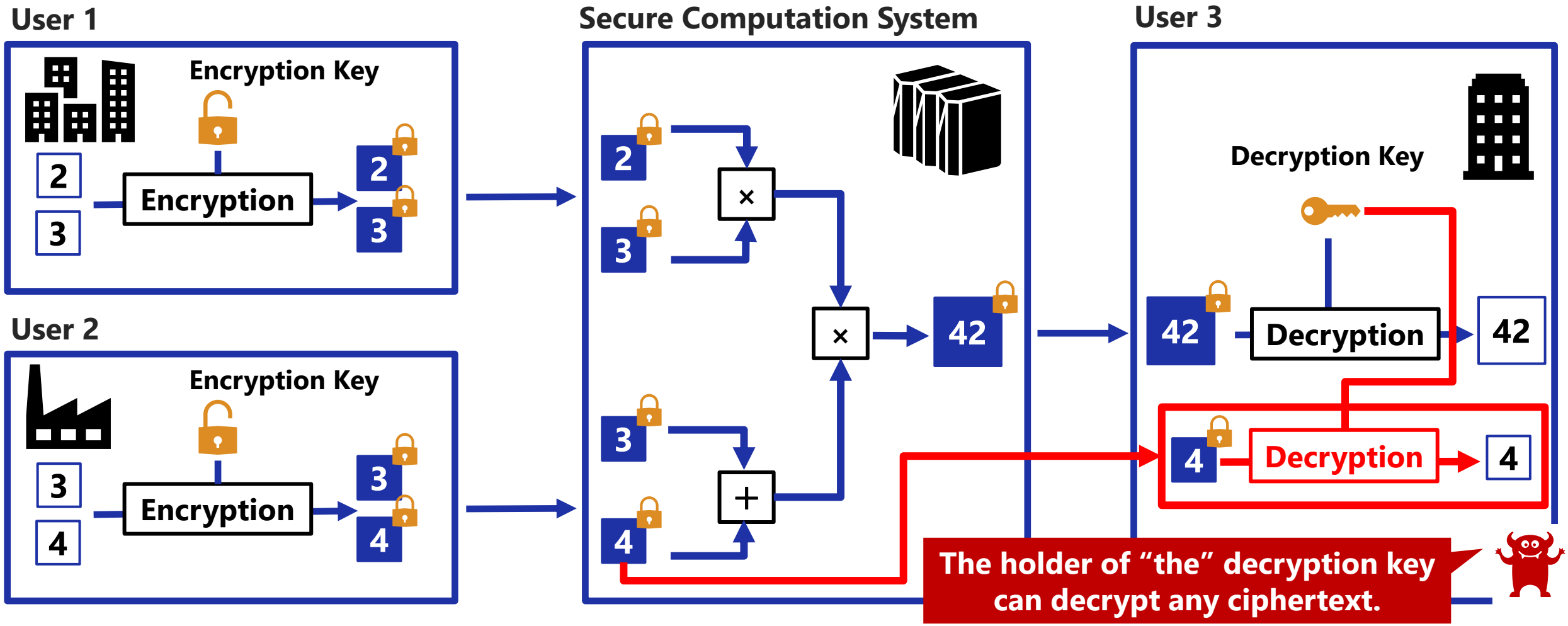
- FHE enables computation on data that are being encrypted.



A. Nakashima et al., Multi-Key Homomorphic Encryption with TRE, SAC 2024

Issue of FHE

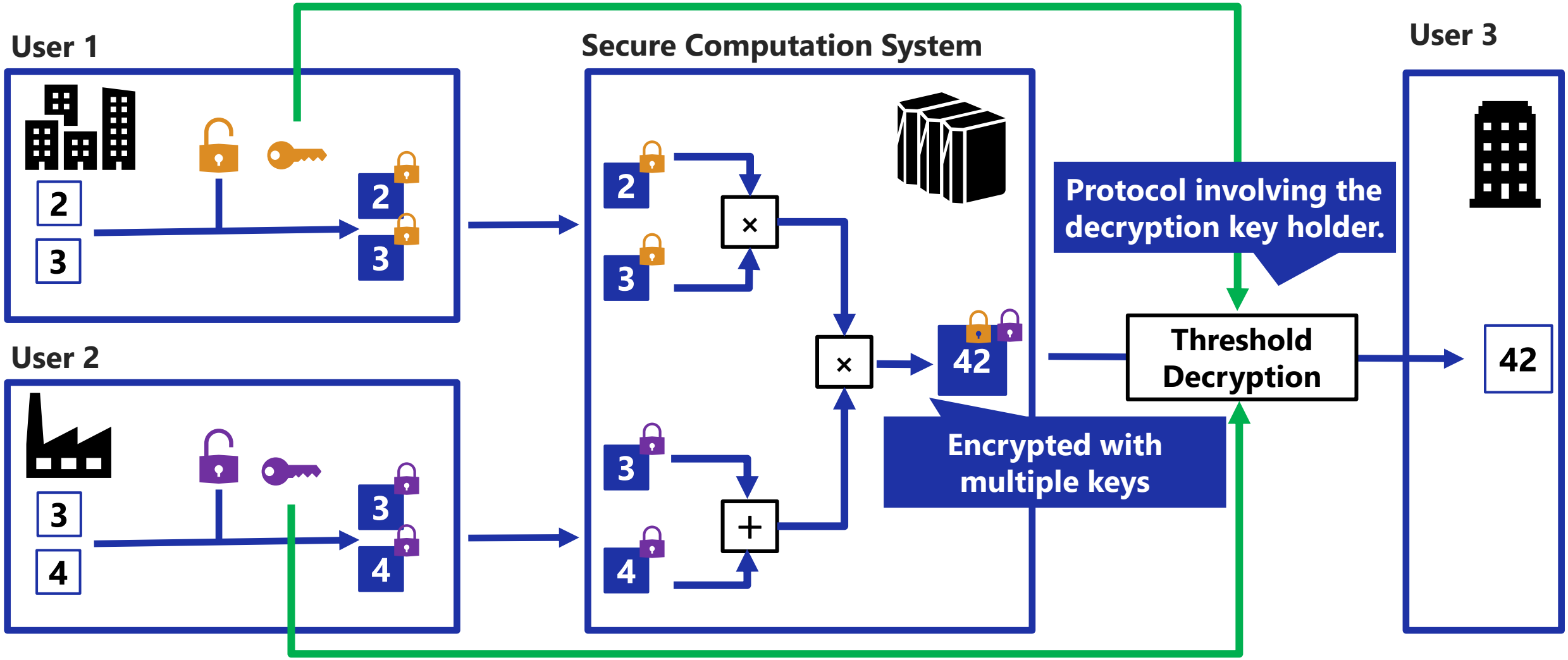
- Ciphertexts in the system must correspond to a single decryption key.



A. Nakashima et al., Multi-Key Homomorphic Encryption with TRE, SAC 2024

Multi-Key FHE (MK-FHE)

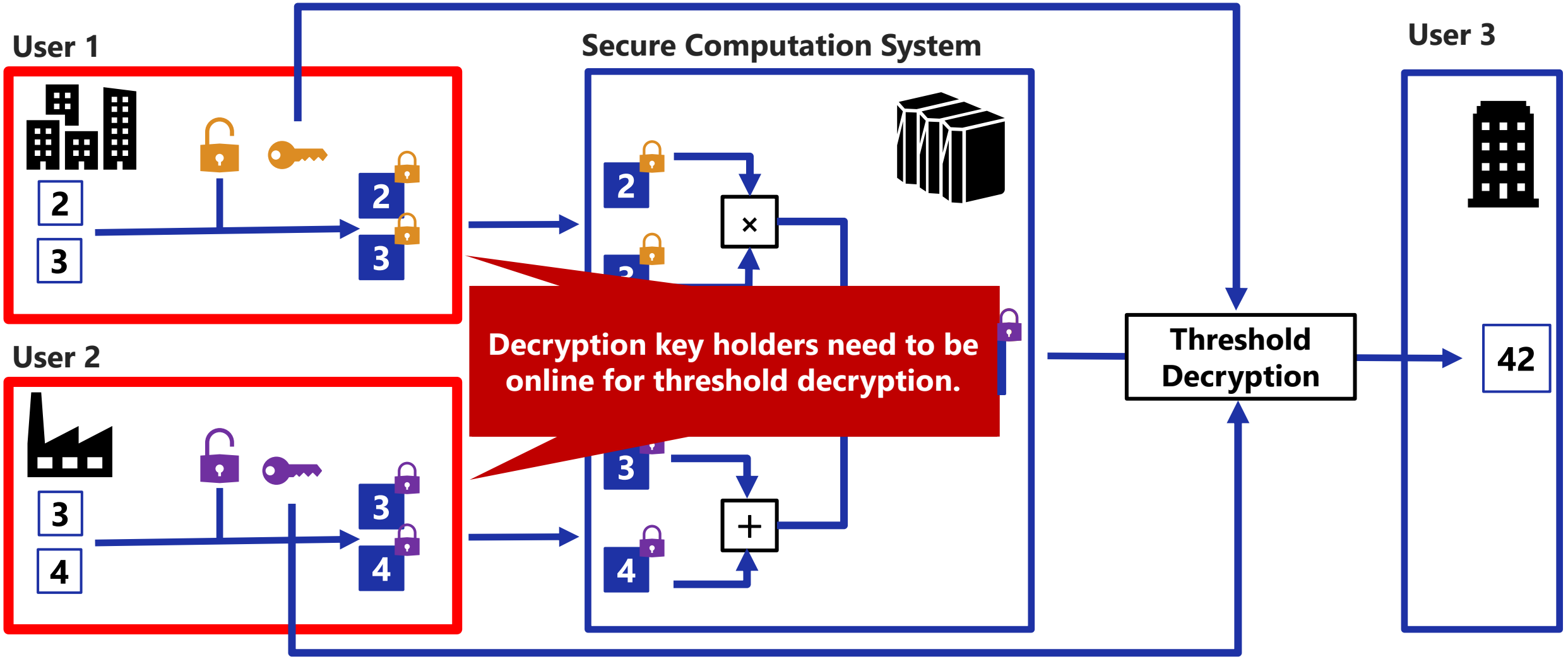
- MK-FHE allows computation on ciphertexts under different keys.



A. Nakashima et al., Multi-Key Homomorphic Encryption with TRE, SAC 2024

Issue of MK-FHE

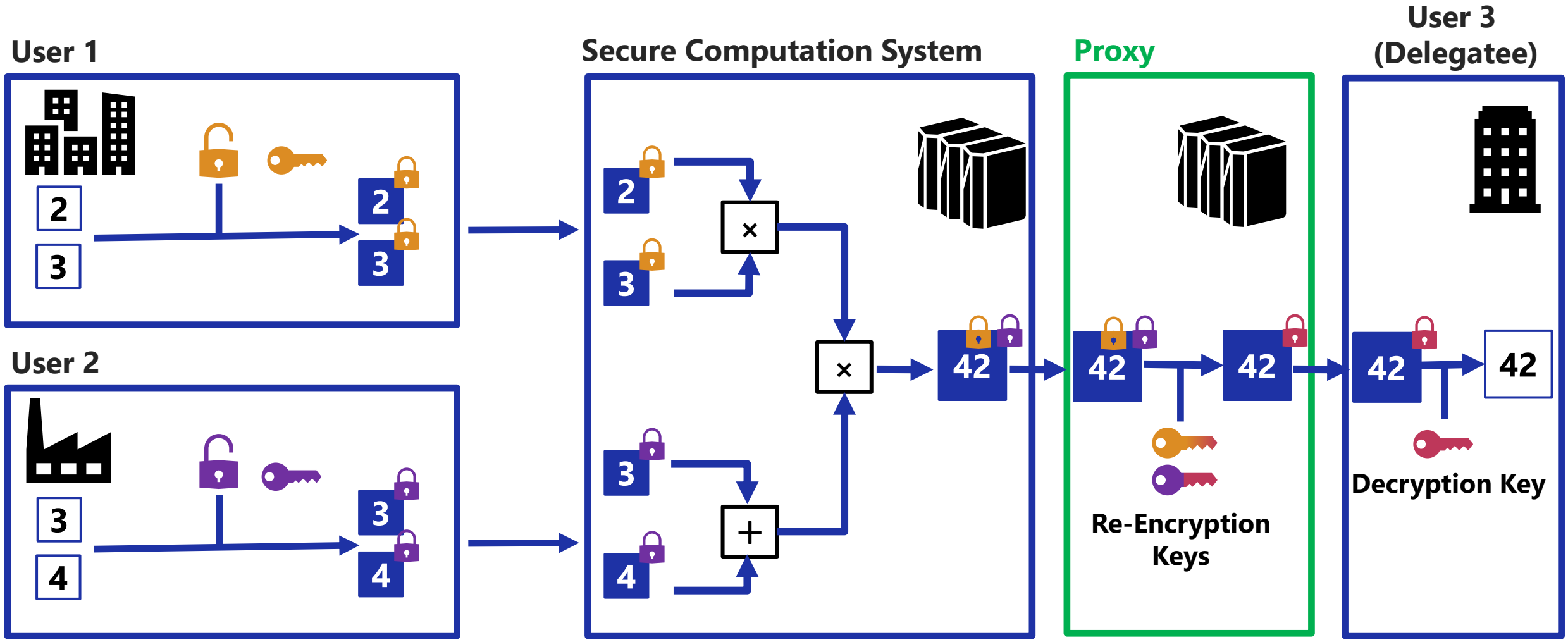
- Threshold decryption requires some work by decryption key holders.



A. Nakashima et al., Multi-Key Homomorphic Encryption with TRE, SAC 2024

MK-FHE with Proxy Re-Encryption (PRE)[Y+18]

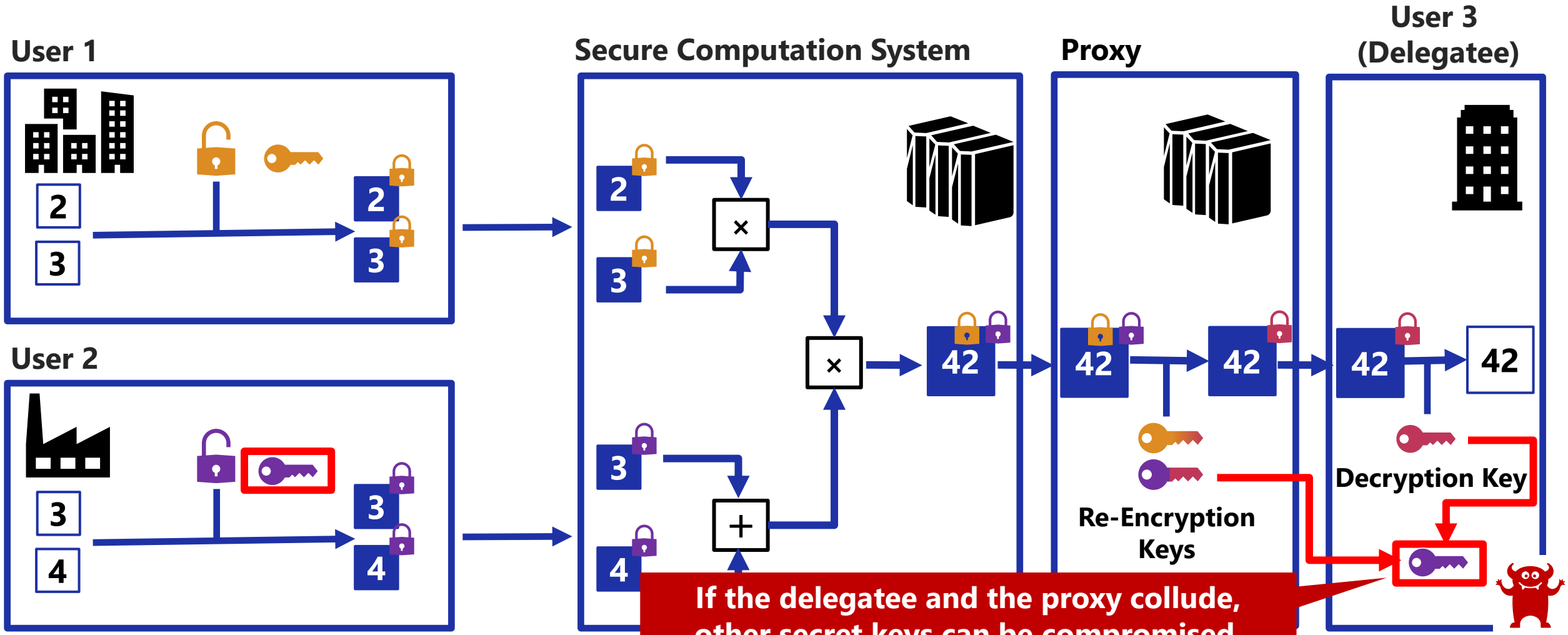
- Proxy converts a multi-key ciphertext into a single-key ciphertext.



[Y+18] S. Yasuda et al., Multi-key homomorphic proxy re-encryption, ISC 2018, pp.328-346.

Issue of MK-FHE with PRE [Y+18]

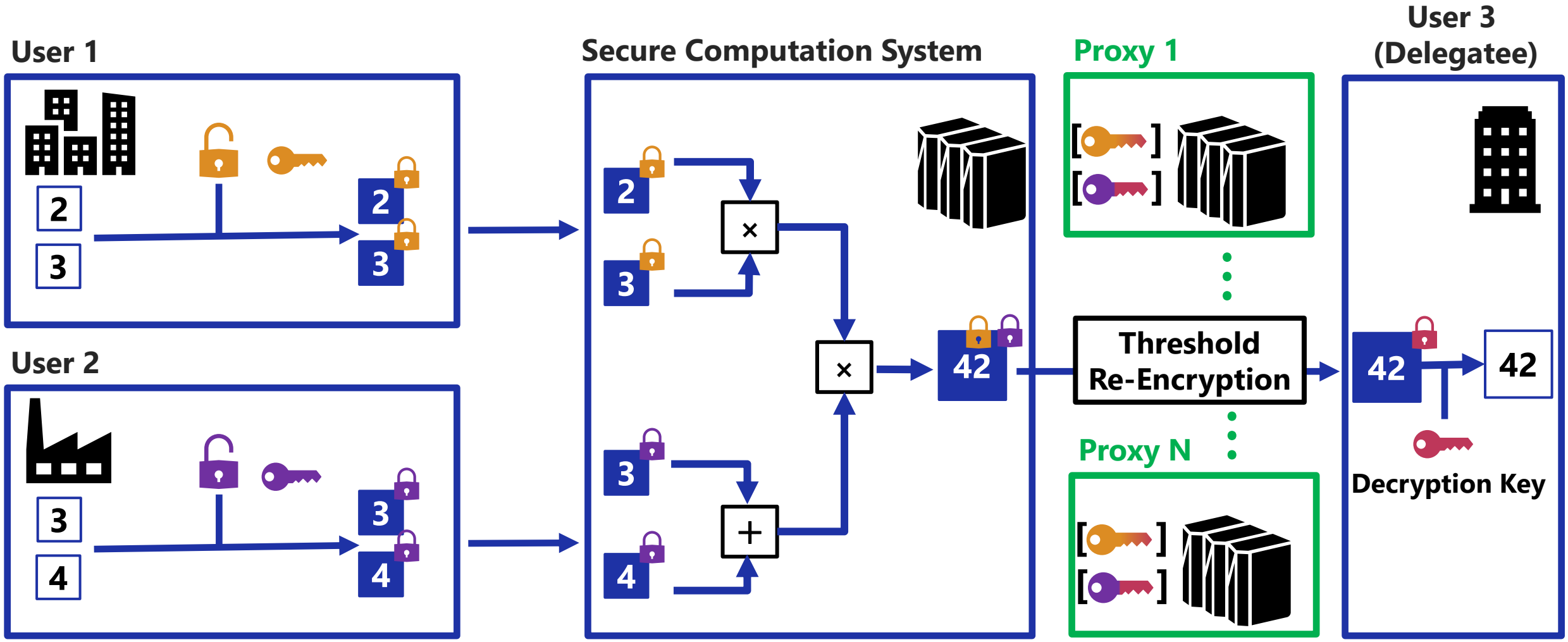
- Re-encryption keys can be a cause of leakage of secret keys.



[Y+18] S. Yasuda et al., Multi-key homomorphic proxy re-encryption, ISC 2018, pp.328-346.

Proposal : MK-FHE with Threshold Re-Encryption (TRE)

- MK-FHE-TRE allows re-encryption while distributing re-encryption keys to N proxies.



Our Contribution

- We propose **MK-FHE with threshold re-encryption** (MK-FHE-TRE).
  - Decryption keys of the delegators are not compromised unless the delegatee colludes with **all proxies**.
  - We instantiate MK-FHE-TRE based on the multi-key variant of BFV by Chen et al. [C+19].

[C+19] H. Chen et al., Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference, ACM CCS 2019, pp.395-412.

Our Scheme

Notations

- The set of users : $\mathcal{U} = \{U_1, \dots, U_k\}$.
- The set of proxies : $\mathcal{P} = \{P_1, \dots, P_N\}$.
- Quotient ring : $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ for an integer q.
  - Plaintext space : $R_t = R/(t \cdot R)$ for $t \ll q$.
- Shares of N-out-of-N additive secret sharing : $[x] = ([x]_1, \dots, [x]_N)$.
  - $x \equiv x_1 + \cdots + x_N \mod q \ \ (x_i \in R_q)$.
  - $[x]_i = x_i$.

MK-BFV Scheme [C+19]

- **DecKey :** A small polynomial $s \in R_q$.
- **EncKey** : For a decryption key s, the encryption key is $(p_0, p_1) \in R_q^2$ such that

$$p_0 + p_1 \cdot s \approx 0 \pmod{q}.$$

- **Enc :** For a plaintext $m \in R_t$, the ciphertext is $(c_0, c_1) \in R_q^2$ such that

$$c_0 + c_1 \cdot s \approx \lfloor q/t \rceil \cdot m \pmod{q}.$$

- **Eval :** For $({c_0}^{(1)}, {c_1}^{(1)})$ under s_1 and $({c_0}^{(2)}, {c_1}^{(2)})$ under s_2, homomorphic operations are performed to extended ciphertexts $\left({c_0}^{(1)}, {c_1}^{(1)}, 0\right)$ and $\left({c_0}^{(2)}, 0, {c_1}^{(2)}\right)$.

Extend in the same way for a larger number of related keys.

- **Dec :** For an extended ciphertext $(c_0, c_1, \ldots, c_k)$, compute $\left\lfloor \frac{t}{q} \cdot \left(c_0 + \sum_{i=1}^{k} c_i \cdot s_i\right) \right\rceil \bmod t.$

[C+19] H. Chen et al., Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference, ACM CCS 2019, pp.395-412.

MK-BFV Scheme [C+19]

- **DecKey :** A small polynomial $s \in R_q$.
- **EncKey** : For a decryption key s, the encryption key is $(p_0, p_1) \in R_q^2$ such that

$$p_0 + p_1 \cdot s \approx 0 \pmod{q}.$$

- **Enc :** For a plaintext $m \in R_t$, the ciphertext is $(c_0, c_1) \in R_q^2$ such that

$$c_0 + c_1 \cdot s \approx \lfloor q/t \rfloor \cdot m \pmod{q}.$$

Conputed by threshold decryption;

- **PartDec:** User U_i computes $\mu_i \leftarrow c_i \cdot s_i + e_i$ $(i = 1, \dots k)$ and sends to a user U_D.
- **Merge:** U_D computes $\left\lfloor \frac{t}{q} \cdot \left(c_0 + \sum_{i=1}^{k} \mu_i\right) \right\rceil$ and obtains the plaintext.

- **Dec :** For an extended ciphertext $(c_0, c_1, \dots, c_k)$, compute $\left\lfloor \frac{t}{q} \cdot \left(c_0 + \sum_{i=1}^{k} c_i \cdot s_i\right) \right\rceil \bmod t$.

[C+19] H. Chen et al., Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference, ACM CCS 2019, pp.395-412.

Our Scheme : MK-BFV-TRE

- In our proposed scheme, **DecKey, EncKey, Enc, Eval** are identical to the MK-BFV.

- We add the following algorithms for threshold re-encryption;
  - **ReKeyGen :** Each user U_i $(i = 1, \dots, k)$ generates a re-encryption key $rk_{i \to D}$ for a delegatee U_D, and distributes the share of $rk_{i \to D}$ to the proxies $\mathcal{P} = (P_1, \dots, P_N)$.
  - **ReEnc :** By using the share $[rk_{i \to D}]$, the proxies $\mathcal{P}$ convert a multi-key ciphertext into a single-key ciphertext decryptable by U_D.
  - **Dec :** The delegatee U_D decrypts the ciphertext without threshold decryption.

Proposal : ReKeyGen

1. The delegatee U_D generates a masking key $r_{i\to D} \in R_q$ uniformly, and sends it to users U_i $(i = 1, \ldots, k)$ via secure channel.

2. Each user U_i $(i = 1, \ldots, k)$ computes a re-encryption key $rk_{i\to D} \coloneqq s_i - r_{i\to D} \pmod q$, and distributes the share $[rk_{i\to D}]$ to the proxies $\mathcal{P}$.

Even if all proxies are corrupted, s_i is not compromised, since masking key $r_{i\to D}$ conceals U_i's decryption key s_i.

Proposal : ReEnc

- Consider to re-encrypt an extend ciphertext $ct = (c_0, c_1, \dots, c_k) \in R_q^{k+1}$.
  - $c_0 + c_1 \cdot s_1 + \cdots + c_k \cdot s_k \equiv \lfloor q/t \rfloor \cdot m + e \pmod q$.

$rk_{i\to D} \equiv s_i - r_{i\to D} \pmod q$

1. Proxies $\mathcal{P}$ compute
   $[c_0'] \leftarrow c_0 + \sum_{i=1}^{k} c_i \cdot [rk_{i\to D}] = \left[\lfloor q/t \rfloor \cdot m + e - \sum_{i=1}^{k} c_i \cdot r_{i\to D}\right]$
   and add smudging noise same as threshold decryption.

2. Proxies $\mathcal{P}$ reconstruct c_0' and send the new ciphertext $(c_0', c_1, \dots, c_k)$ to the delegatee U_D.

Proposal : Dec (after Re-Encryption)

1. By using the new ciphertext $(c_0', c_1, \ldots, c_k)$, and masking keys $r_{1\to D}, \ldots, r_{k\to D}$, the deligatee U_D computes

$$\mu \leftarrow c_0' + \sum_{i=1}^{k} c_i \cdot r_{i\to D}\,.$$

$\mu \approx \lfloor q/t \rceil \cdot m \pmod q$ since $c_0' \approx \lfloor q/t \rceil \cdot m - \sum_{i=1}^{k} c_i \cdot r_{i\to D} \pmod q$.

2. By computing $m \leftarrow \left\lfloor \frac{t}{q} \cdot \mu \right\rceil \bmod t$, U_D obtains the plaintext without threshold decryption.

Implementation and Experimental Setting

- We implemented our scheme over our BFV library.
- We compare the execution time of our scheme and the multi-key BFV [C+19] on Intel Xeon Silver 4114 CPU.
- We also estimated communication time with these settings:
  - LAN: 10 Gbps throughput, 0.5 ms latency.
  - WAN: 72 Mbps throughput, 72 ms latency (based on measurements between AWS US East and West).

Experiment

- Threshold re-encryption is relatively slower than threshold decryption, however decryption itself is considerably faster.

Threshold Decryption [C+19]

#Users ($= k$)		2	4	8
Exec. [ms]		26.9	27.6	28.9
Comm. [ms]	LAN	2.04	4.75	10.2
	WAN	290	677	1451

Re-Encryption (Ours)

#Users ($= k$)		2				4				8			
#Proxies ($= N$)		1	2	4	8	1	2	4	8	1	2	4	8
Exec. [ms]		250				298				393			
Comm. [ms]	LAN	2.03	2.71	4.07	6.78	3.39	4.07	5.42	8.14	6.10	6.78	8.14	10.8
	WAN	290	387	580	967	484	580	774	1169	871	967	1161	1548

Decryption (Ours)

#Users ($= k$)	2	4	8
Exec. [ms]	0.05	0.95	1.85

[C+19] H. Chen et al., Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference, ACM CCS 2019, pp.395-412.

Conclusion

- We proposed **MK-FHE with threshold re-encryption.**
  - Decryption keys of the delegators are not compromised unless the delegatee colludes with **all proxies.**
  - We instantiated MK-FHE-TRE by extending the multi-key BFV scheme by Chen et al. [C+19].
  - We implemented our scheme and confirmed that the decryption process runs in shorter time compared to threshold decryption proposed in [C+19]

[C+19] H. Chen et al., Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference, ACM CCS 2019, pp.395-412.

\Orchestrating a brighter world

NEC creates the social values of safety, security,
fairness and efficiency to promote a more sustainable world
where everyone has the chance to reach their full potential.

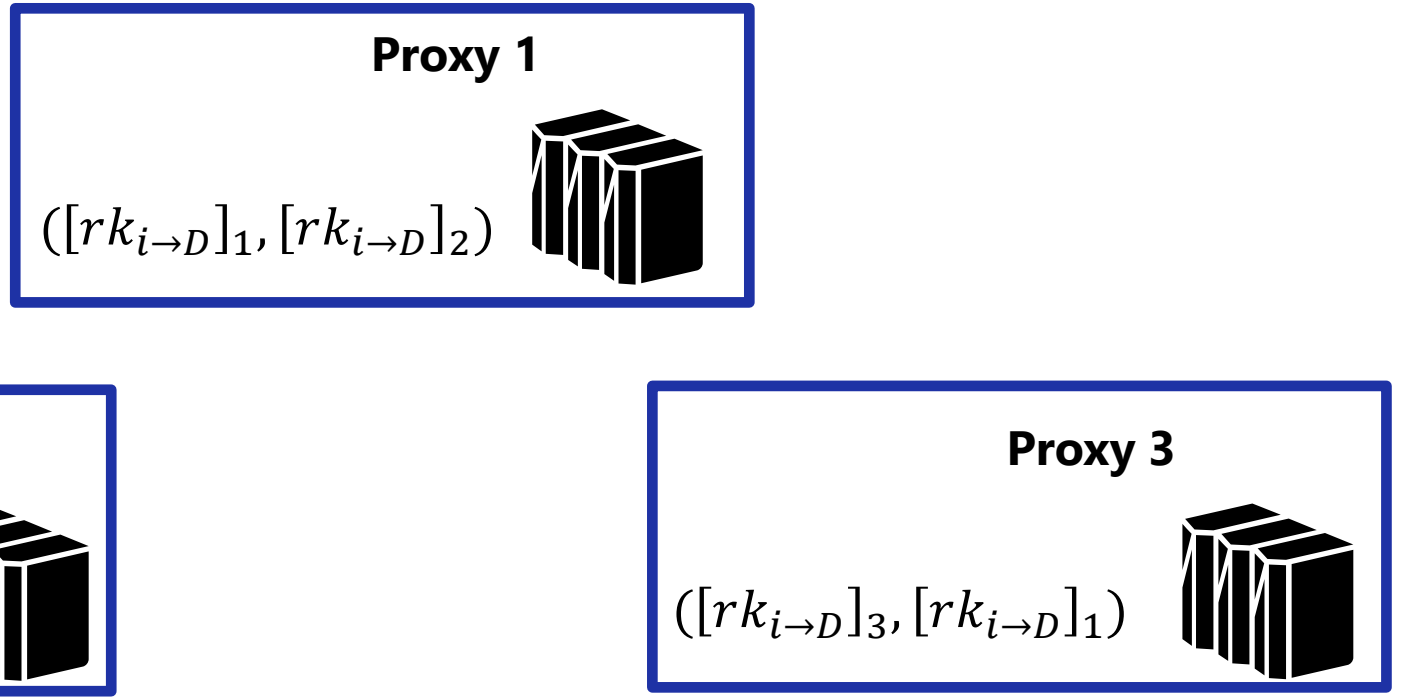

Orchestrating a brighter world
NEC

Extension to t-out-of-N

- We can straightforwardly extend the re-encryption using (N, N)-ASS to t-out-of-N replicated secret sharing.

The example of replicated secret sharing; $(3, 3)$ to $(2, 3)$:

Assume $rk_{i\to D} = [rk_{i\to D}]_1 + [rk_{i\to D}]_2 + [rk_{i\to D}]_3$



Experiment setting in detail

- We used the ciphertext parameters below for the number of users $k = 1, 2, 4, 8$.

Parameter	Value
Ring dimension: n	8192
Length of ciphertext modulus: $\log q$	220
Standard deviation for noise σ_{err}	3.2
Standard deviation for smudging noise σ_{smdg}	2^{20}
Plaintext modulus: t	256
Multiplicative depth: L	5
Security parameter: λ	128

Experiment for ReKeyGen

- The execution time appears linear regarding the number of proxies N.

Re-Encryption Key Generation (Ours)

#Proxies ($= N$)		1	2	4	8
Exec. [ms]		51.2	102	203	409
Comm. [ms]	LAN	1.36	2.03	3.39	6.10
	WAN	193	290	484	871