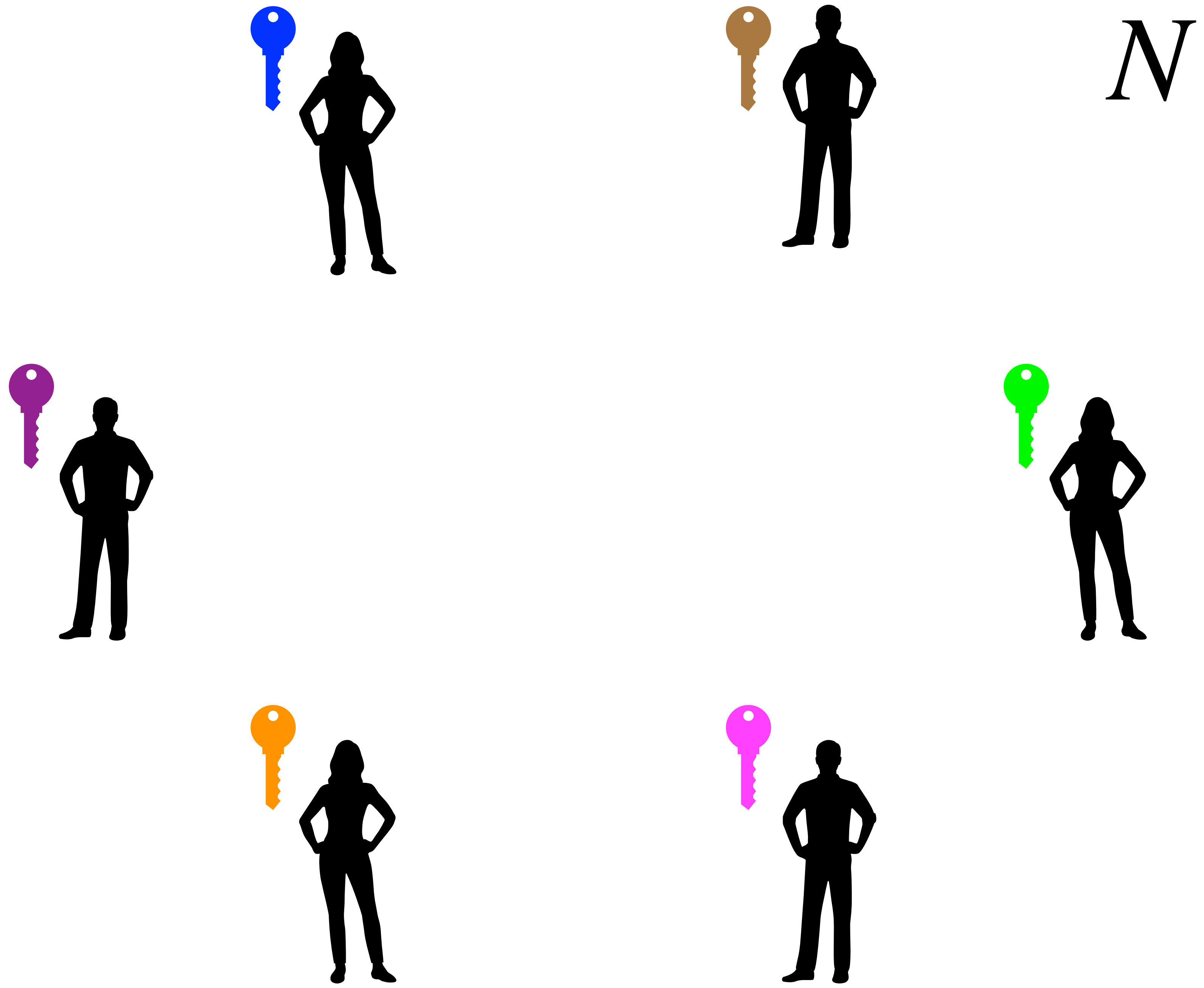
Amber: Lattice-Based Threshold KEM from the BCHK+ Transform

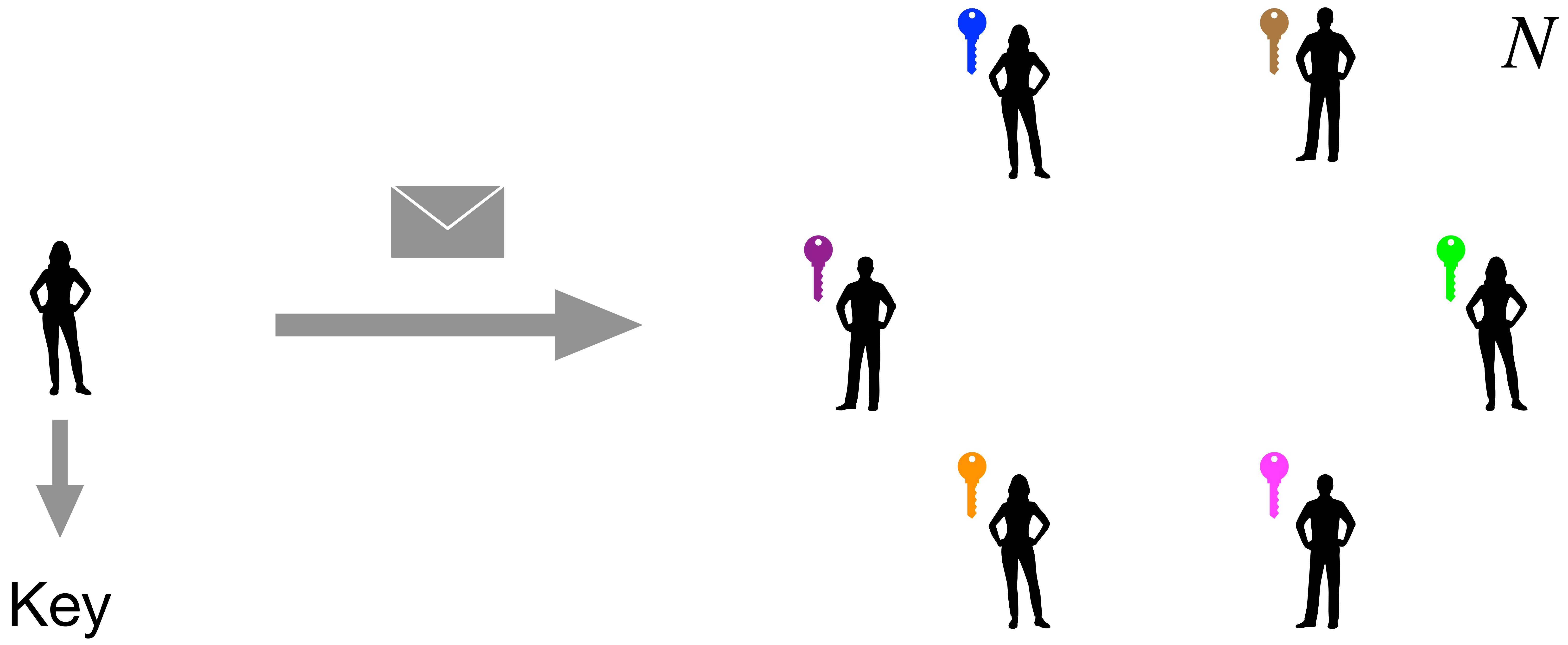
joint work with Katharina Boudgoust, Rafaël del Pino and Thomas Prest

Presented by: Sasha Lapiha, Royal Holloway @ NIST Workshop on Multi-Party Threshold Schemes. January 28, 2026.

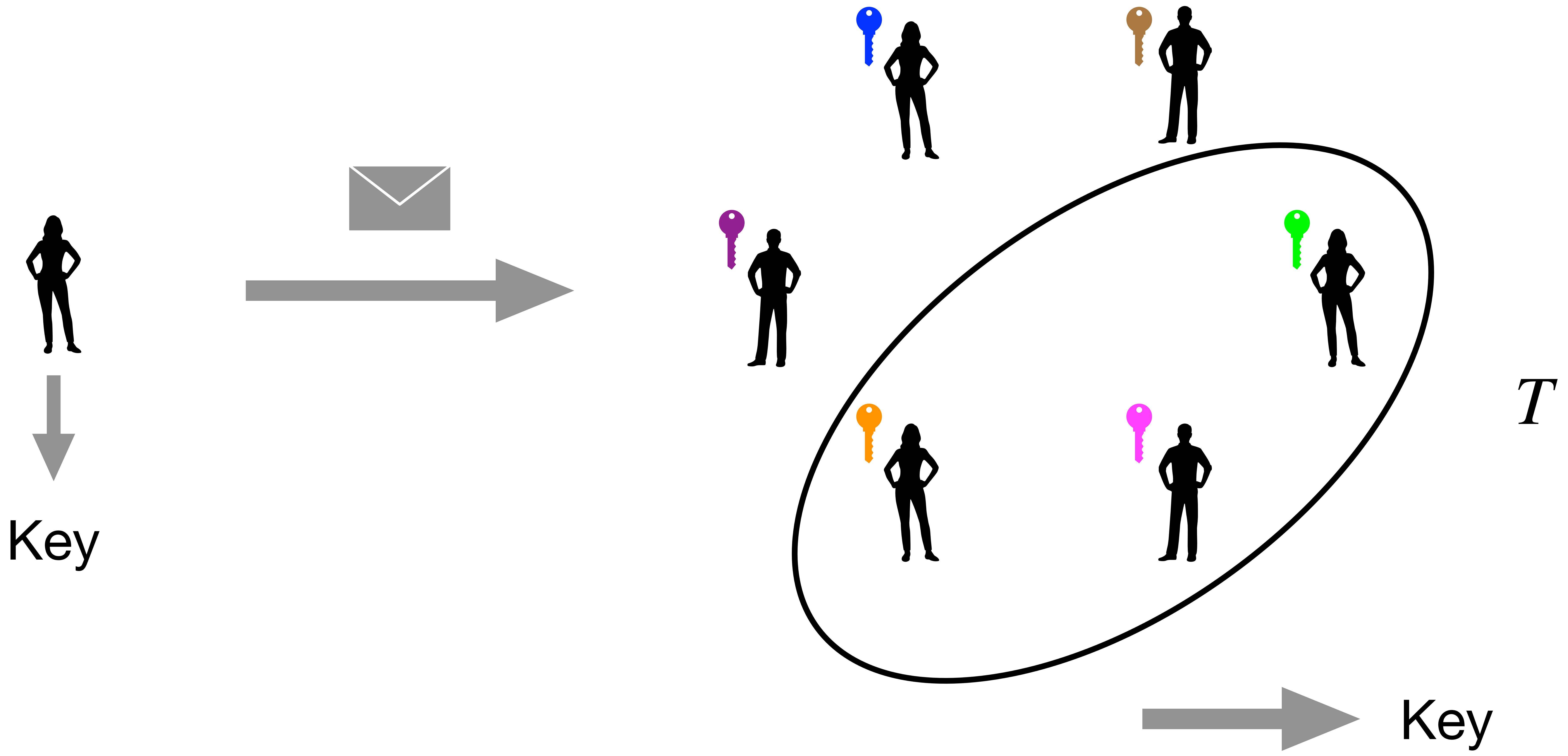
Threshold Decapsulation



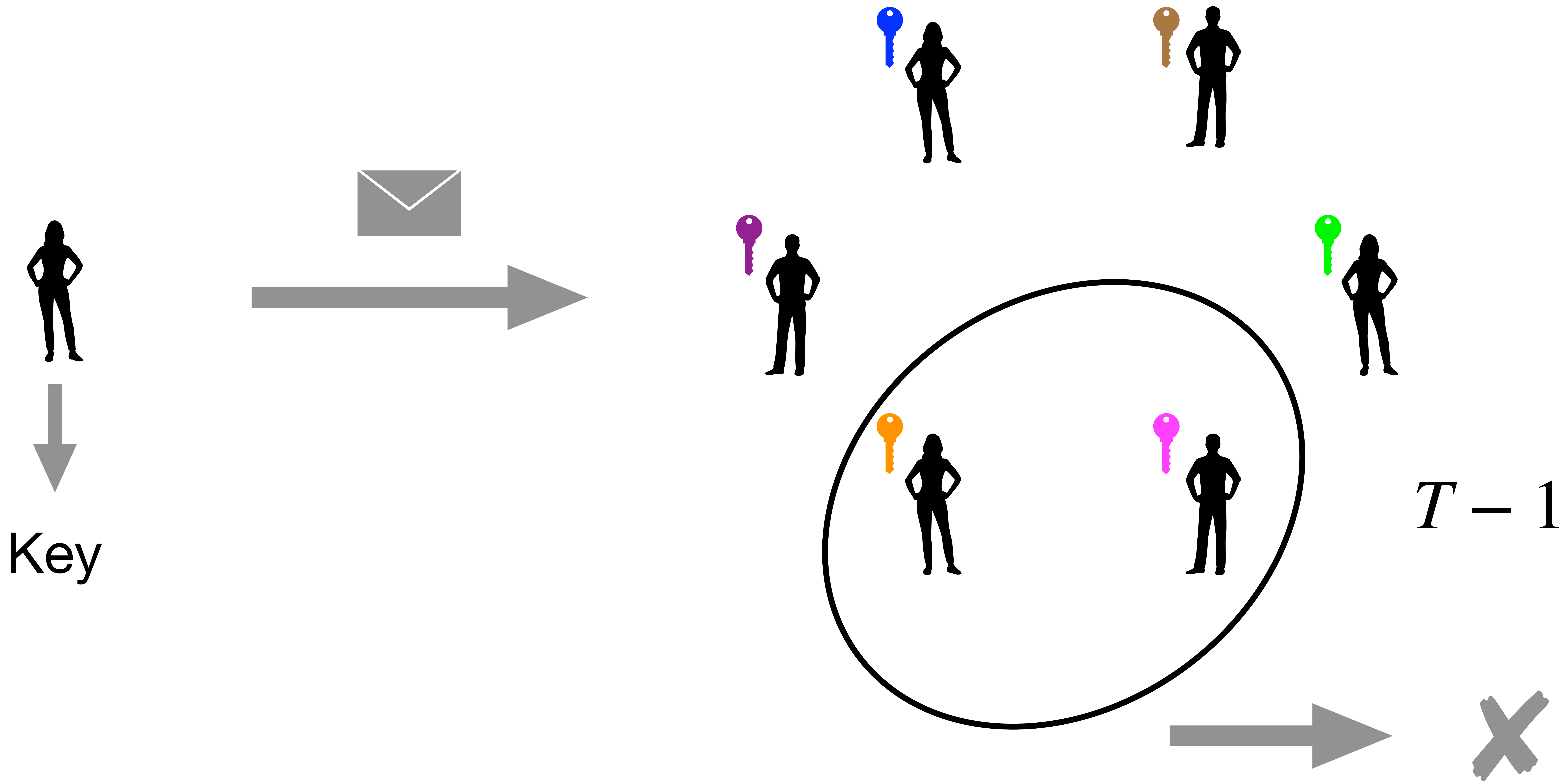
Threshold Decapsulation



Threshold Decapsulation

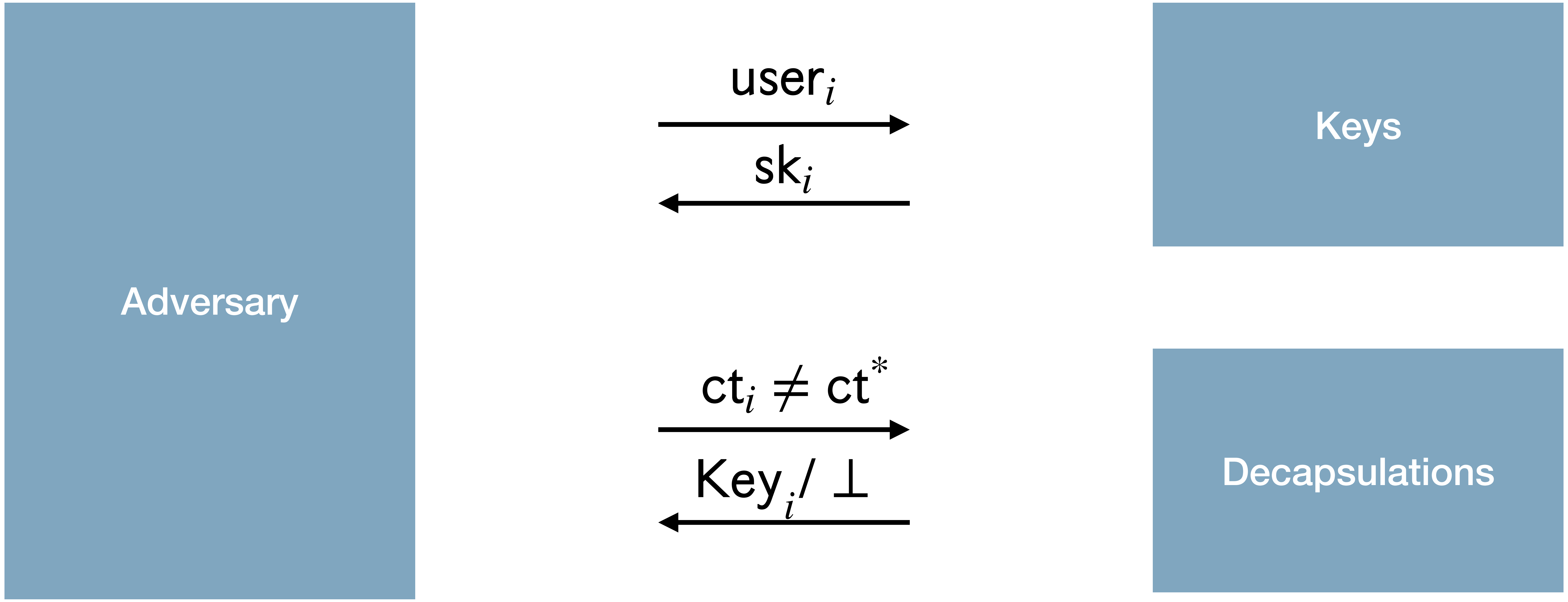


Threshold Decapsulation

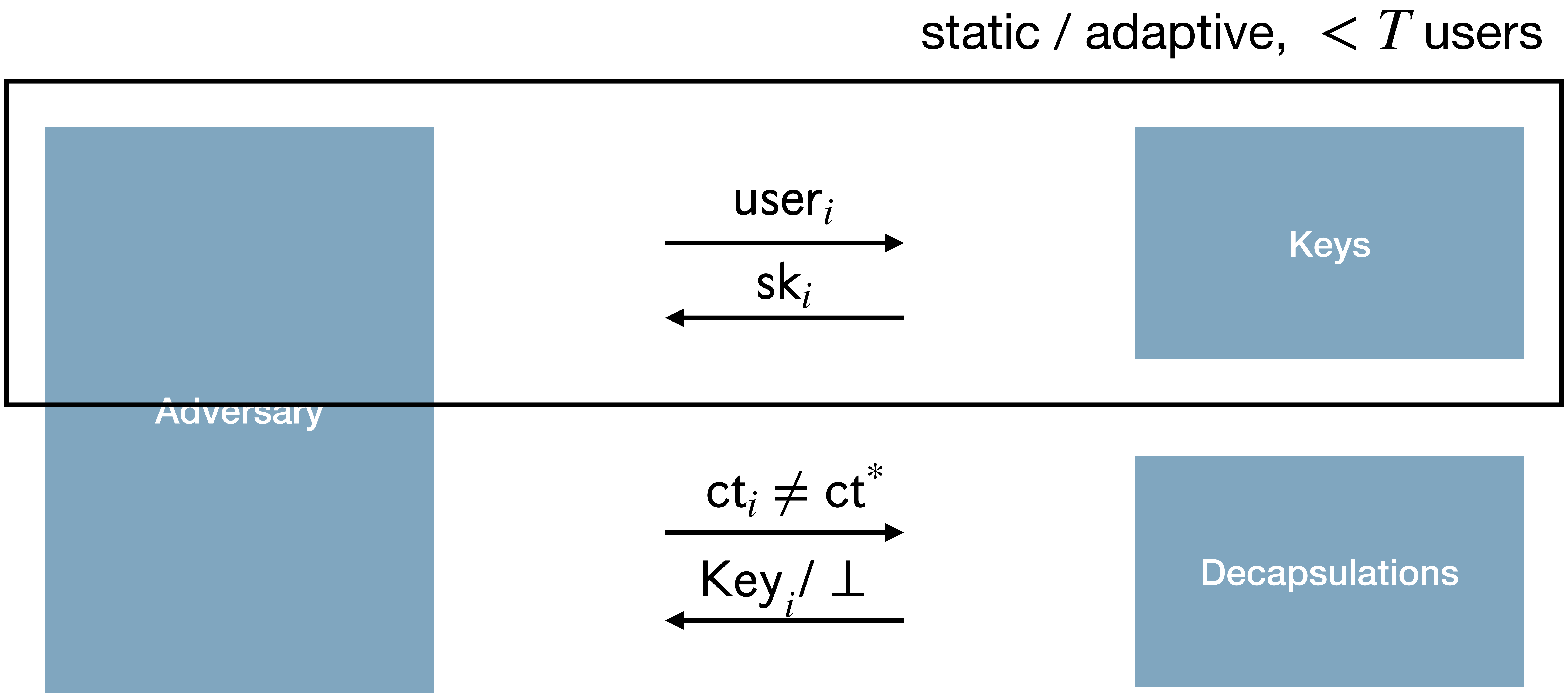


Active Security of Threshold Decapsulation

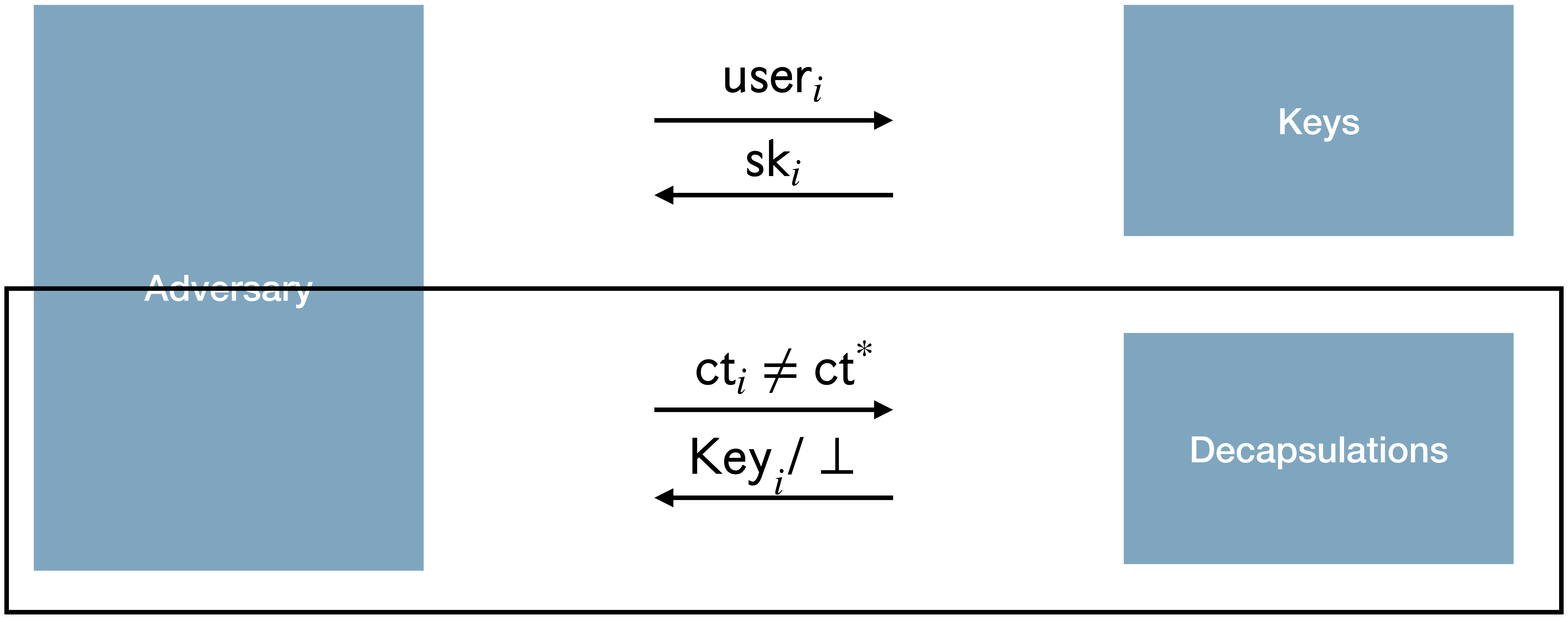
Security Model



Security Model



Security Model



passive / active (CCA2)

FO transform (non-threshold case)

$$\mathsf{msg} \xleftarrow{\$} \{0,1\}^{128}$$
$$\mathsf{ct} = \mathsf{Enc}(\mathsf{msg}, \mathsf{pk})$$

$$\mathsf{msg} = \mathsf{Dec}(\mathsf{ct}, \mathsf{sk})$$

FO transform (non-threshold case)

$\mathsf{msg} \xleftarrow{\$} \{0,1\}^{128}$
$\mathsf{ct} = \mathsf{Enc}(\mathsf{msg}, \mathsf{pk})$

$\mathsf{msg} = \mathsf{Dec}(\mathsf{ct}, \mathsf{sk})$

$\mathsf{msg} \xleftarrow{\$} \{0,1\}^{128}$
$\mathsf{rand} = \mathsf{PRG}(\mathsf{msg})$
$\mathsf{ct} = \mathsf{Enc}(\mathsf{msg}, \mathsf{pk}; \mathsf{rand})$

FO transform (non-threshold case)

$\mathsf{msg} \xleftarrow{\$} \{0,1\}^{128}$
$\mathsf{ct} = \mathsf{Enc}(\mathsf{msg}, \mathsf{pk})$

$\mathsf{msg} = \mathsf{Dec}(\mathsf{ct}, \mathsf{sk})$

$\longrightarrow$

$\mathsf{msg} \xleftarrow{\$} \{0,1\}^{128}$
$\mathsf{rand} = \mathsf{PRG}(\mathsf{msg})$
$\mathsf{ct} = \mathsf{Enc}(\mathsf{msg}, \mathsf{pk}; \mathsf{rand})$

$\mathsf{msg} = \mathsf{Dec}(\mathsf{ct}, \mathsf{sk})$
$\mathsf{rand} = \mathsf{PRG}(\mathsf{msg})$
$\mathsf{Enc}(\mathsf{msg}, \mathsf{pk}; \mathsf{rand}) == \mathsf{ct}?$

Previous work

MPC
[CCMS21], [KLO+19]

FHE
[BGG+18]

ZK Proofs
[DLN+21]

Generic ?

BCHK Transform

BCHK Transform

[CHK04],[BCHK07],[BBH07]

Identity-based Encryption (IBE) + One-time Signature (OTS) = CCA2 PKE

Identity-based Encryption

$$(\mathsf{sk}, \mathsf{pp}) \leftarrow \text{ibe-KeyGen}()$$
$$\mathsf{ct} = \text{ibe-Enc}(\mathsf{msg}, \mathsf{pp}, \mathsf{id})$$

then

$$\mathsf{sk}_{\mathsf{id}} = \text{ibe-Extract}(\mathsf{sk}, \mathsf{id})$$
$$\mathsf{msg} = \text{ibe-Dec}(\mathsf{ct}, \mathsf{sk}_{\mathsf{id}})$$

BCHK Transform

$\mathsf{Encrypt}(\mathsf{msg}, \mathsf{pp}) :$

$(\mathsf{sig\text{-}sk}, \mathsf{sig\text{-}vk}) \leftarrow \mathsf{sig\text{-}KeyGen}()$
$\mathsf{ct}_0 = \mathsf{ibe\text{-}Enc}(\mathsf{msg}, \mathsf{id} = \mathsf{sig\text{-}vk})$
$\mathsf{sig} = \mathsf{Sign}(\mathsf{ct}_0, \mathsf{sig\text{-}sk})$
$\mathsf{ct} = (\mathsf{ct}_0, \mathsf{sig}, \mathsf{sig\text{-}vk})$

BCHK Transform

$\mathsf{ct}_0 = \text{ibe-Enc}(\mathsf{msg}, \mathsf{id} = \text{sig-vk})$

$(\text{sig-sk}, \text{sig-vk}) \leftarrow \text{sig-KeyGen}()$

$\mathsf{sig} = \mathsf{Sign}(\mathsf{ct}_0, \text{sig-sk})$

BCHK Transform

$\mathsf{Encrypt}(\mathsf{msg}, \mathsf{pp}):$

$(\text{sig-sk}, \text{sig-vk}) \leftarrow \text{sig-KeyGen}()$
$\mathsf{ct}_0 = \text{ibe-Enc}(\mathsf{msg}, \mathsf{id} = \text{sig-vk})$
$\mathsf{sig} = \mathsf{Sign}(\mathsf{ct}_0, \text{sig-sk})$
$\mathsf{ct} = (\mathsf{ct}_0, \mathsf{sig}, \text{sig-vk})$

$\mathsf{Decrypt}(\mathsf{ct}, \mathsf{sk}):$

$\mathsf{Verify}(\mathsf{ct}_0, \mathsf{sig}, \text{sig-vk}) == 1\,?$
$\mathsf{sk}_{\mathsf{id}} = \text{ibe-Extract}(\mathsf{sk}, \mathsf{id} = \text{sig-vk})$
$\mathsf{msg} = \text{ibe-Dec}(\mathsf{ct}_0, \mathsf{sk}_{\mathsf{id}})$

BCHK Transform

$\mathsf{Encrypt}(\mathsf{msg}, \mathsf{pp}):$

$(\mathsf{sig\text{-}sk}, \mathsf{sig\text{-}vk}) \leftarrow \mathsf{sig\text{-}KeyGen}()$
$\mathsf{ct}_0 = \mathsf{ibe\text{-}Enc}(\mathsf{msg}, \mathsf{id} = \mathsf{sig\text{-}vk})$
$\mathsf{sig} = \mathsf{Sign}(\mathsf{ct}_0, \mathsf{sig\text{-}sk})$
$\mathsf{ct} = (\mathsf{ct}_0, \mathsf{sig}, \mathsf{sig\text{-}vk})$

$\mathsf{Decrypt}(\mathsf{ct}, \mathsf{sk}):$

$\mathsf{Verify}(\mathsf{ct}_0, \mathsf{sig}, \mathsf{sig\text{-}vk}) == 1\,?$
$\mathsf{sk}_{\mathsf{id}} = \mathsf{ibe\text{-}Extract}(\mathsf{sk}, \mathsf{id} = \mathsf{sig\text{-}vk})$
$\mathsf{msg} = \mathsf{ibe\text{-}Dec}(\mathsf{ct}_0, \mathsf{sk}_{\mathsf{id}})$

Threshold BCHK

[BBH07]

Threshold IBE + OTS = Threshold PKE

Consistency / Robustness

$$\begin{array}{ccc} & \mathsf{ct} & \\ \swarrow & & \searrow \\ \{\mathsf{dec}_i\}_{i \in \mathsf{user-set}_0} & & \{\mathsf{dec}_i\}_{i \in \mathsf{user-set}_1} \\ \downarrow & & \downarrow \\ \mathsf{Key}_0 & & \mathsf{Key}_1 / \perp \end{array}$$

BCHK+ = BCHK + FO

Threshold IBE + OTS = Threshold KEM

\+ FO

BCHK+ Transform

Encrypt(msg, pp) :

msg $\leftarrow \{0,1\}^{128}$
(sig-sk, sig-vk) $\leftarrow$ sig-KeyGen()
rand = PRG(msg)
ct_0 = ibe-Enc(msg, sig-vk; rand)
sig = Sign(ct_0, sig-sk)

ct = (ct_0, sig, sig-vk)

PartDecrypt(ct, sk_i) :

Verify(ct_0, sig, sig-vk) == 1?
dec_i = ibe-Extract(sig-vk, sk_i)

Combine($\{\text{dec}_i\}$) :

sk_{id} = Combine($\{\text{dec}_i\}$)
msg = ibe-Decrypt(ct, sk_{id})
rand = PRG(msg)
Enc(msg, pk; rand) == ct?

BCHK+ Transform

Encrypt(msg, pp) :

$\text{msg} \leftarrow \{0,1\}^{128}$
$(\text{sig-sk}, \text{sig-vk}) \leftarrow \text{sig-KeyGen}()$
$\text{rand} = \text{PRG}(\text{msg})$
$\text{ct}_0 = \text{ibe-Enc}(\text{msg}, \text{sig-vk}; \text{rand})$
$\text{sig} = \text{Sign}(\text{ct}_0, \text{sig-sk})$

$\text{ct} = (\text{ct}_0, \text{sig}, \text{sig-vk})$

PartDecrypt(ct, sk_i) :

$\text{Verify}(\text{ct}_0, \text{sig}, \text{sig-vk}) == 1\,?$
$\text{dec}_i = \text{ibe-Extract}(\text{sig-vk}, \text{sk}_i)$

secret sensitive

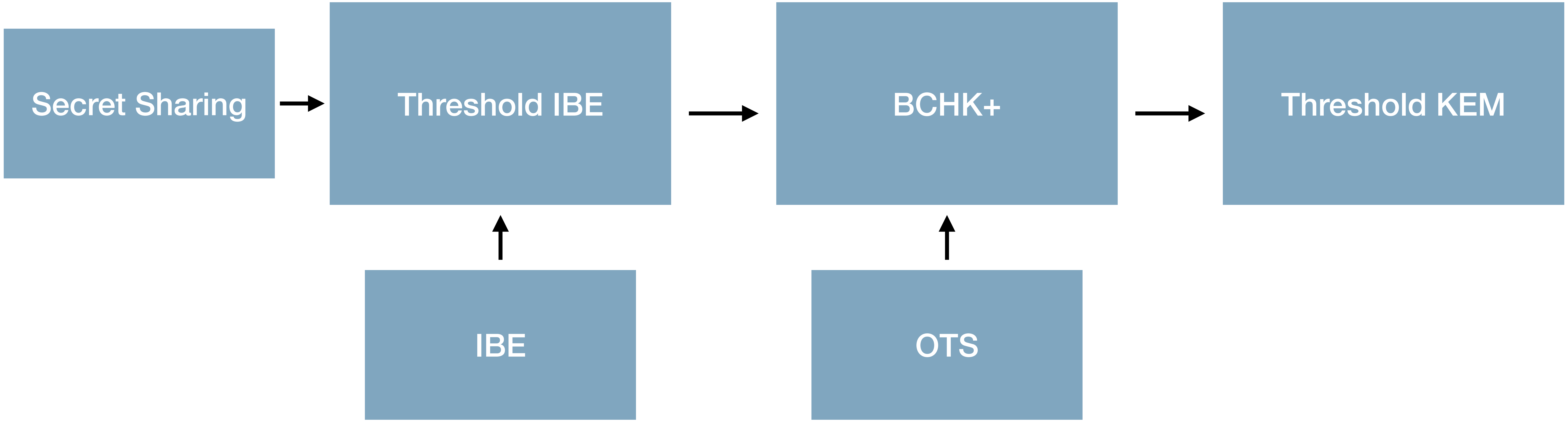
Combine($\{\text{dec}_i\}$) :

$\text{sk}_{\text{id}} = \text{Combine}(\{\text{dec}_i\})$
$\text{msg} = \text{ibe-Decrypt}(\text{ct}, \text{sk}_{\text{id}})$
$\text{rand} = \text{PRG}(\text{msg})$
$\text{Enc}(\text{msg}, \text{pk}; \text{rand}) == \text{ct}\,?$

public

Amber: The Lattice Construction

The Blueprint



The Ciphertext: Dual Regev Encryption

$\mathsf{ct} = (\mathbf{u}, v)$, where $\mathbf{u} = \mathbf{F}_{\mathsf{id}} \cdot s + \mathbf{e}$ and $v = r \cdot s + e' + \mathsf{Encode}(\mathsf{msg})$

The Ciphertext

$\mathsf{ct} = (\mathbf{u}, v)$, where $\mathbf{u} = \mathbf{F}_{\mathsf{id}} \cdot s + \mathbf{e}$ and $v = r \cdot s + e' + \mathsf{Encode}(\mathsf{msg})$

the key: $\mathbf{x}$ short s.t. $\mathbf{x}^t \cdot \mathbf{F}_{\mathsf{id}} = r \bmod q$

The Ciphertext

$\mathrm{ct} = (\mathbf{u}, v)$, where $\mathbf{u} = \mathbf{F}_{\mathrm{id}} \cdot s + \mathbf{e}$ and $v = r \cdot s + e' + \mathrm{Encode}(\mathrm{msg})$

the key: $\mathbf{x}$ short s.t. $\mathbf{x}^t \cdot \mathbf{F}_{\mathrm{id}} = r \bmod q$

$\mathrm{msg} = \mathrm{Round}(v - \mathbf{x}^t \cdot \mathbf{u})$

Starting IBE

ROHIBE [CHKP10]

- Dual Regev with $\mathbf{F}_{\mathsf{id}} = [\mathbf{A} \mid H(\mathsf{id})]$
- $\mathsf{sk}_{\mathsf{id}}$ is short $\mathbf{x}$ s.t. $\mathbf{x}^t \cdot \mathbf{F}_{\mathsf{id}} = r \bmod q$
- Master sk: short trapdoor for $\mathbf{A}$

Starting IBE

ROHIBE [CHKP10]

- Dual Regev with $\mathbf{F}_{\mathsf{id}} = [\mathbf{A} \mid H(\mathsf{id})]$
- $\mathsf{sk}_{\mathsf{id}}$ is short $\mathbf{x}$ s.t. $\mathbf{x}^t \cdot \mathbf{F}_{\mathsf{id}} = r \bmod q$
- Master sk: short trapdoor for $\mathbf{A}$
  - Key extraction - trapdoor sampling - non-linear

Threshold IBE

ROHIBE

Plover [EEN+24] + Key Secret Sharing [BCP+25],[PKM+24]

- $\mathbf{x} = c \cdot \mathbf{s} + \mathbf{p}$
- Schnorr-type protocol to compute $c \cdot \mathbf{s} + \mathbf{p}$

Secret Sharing

Secret key: short vector **s**

Secret Sharing

Shamir Secret Sharing + masks or Vandermonde Secret Sharing

Secret Sharing

Secret key: short vector **s**

Secret Sharing

Shamir Secret Sharing + masks　　or　　Vandermonde Secret Sharing

$$\mathbf{s} = \sum \lambda_i \mathbf{s}_i + \mathbf{m}_i$$

$$\sum \mathbf{m}_i = 0$$

Secret Sharing

Secret key: short vector **s**

Secret Sharing

Shamir Secret Sharing + masks   or   Vandermonde Secret Sharing

$$\mathbf{s} = \sum \lambda_i \mathbf{s}_i + \mathbf{m}_i$$

$$\sum \mathbf{m}_i = 0$$

$$\mathbf{s} = \sum \mathbf{s}_i$$

Security Implications

Variant	Reconstruction	Corruptions	Robustness	Group Size
Shamir SS	$\mathbf{s} = \sum \lambda_i \mathbf{s}_i + \mathbf{m}_i$	Adaptive	No	Large (< 1024)
Vandermonde SS	$\mathbf{s} = \sum \mathbf{s}_i$	Static	Yes	Medium (< 64)

The parameters

Variant	Rounds	Decryption Queries	Pk size (bytes)	Ct size (bytes)
Shamir SS	3	2^46	6 688	28 544
Vandermonde SS	3	2^25	7 456	29 056

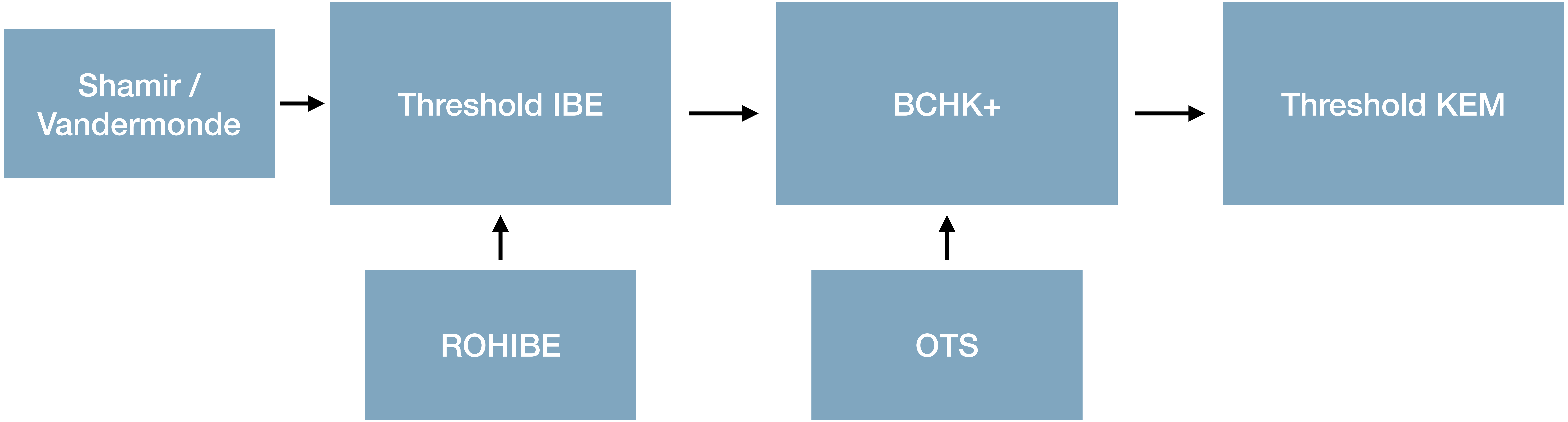
The number are stated for T = 32, N = 64 and security level of 128 bits.

Future work / Your feedback

- Optimised implementation
  - Reach out if you are interested!
- Community feedback
  - Robust/med group/low queries VS adaptive/large group
  - Use cases
- Further reducing the sizes

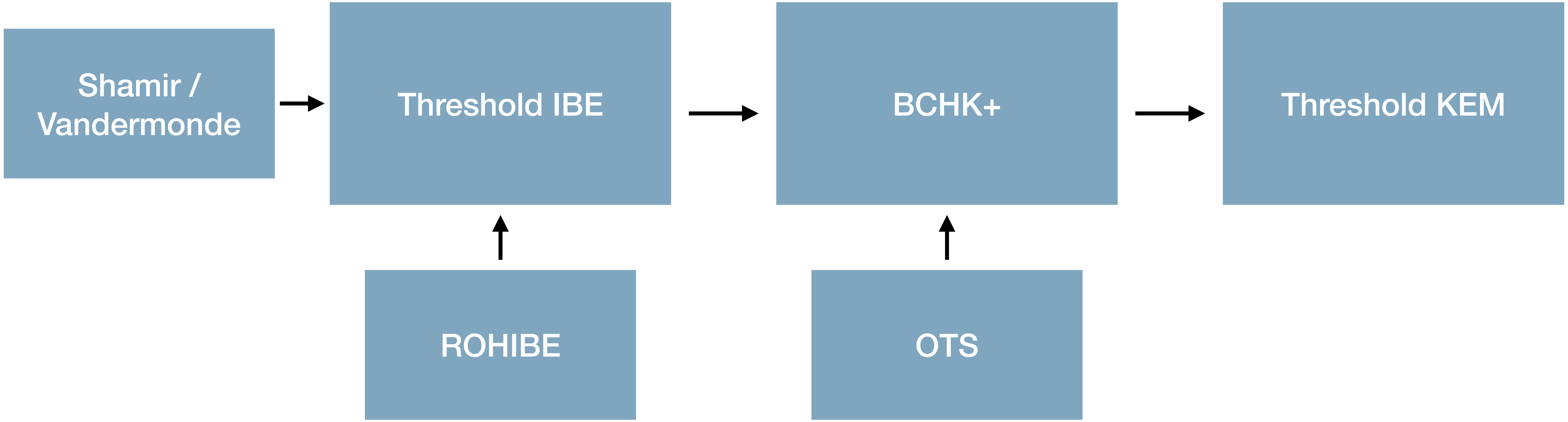
Summary

"Amber: Lattice-Based Threshold KEM with Active Security" | ia.cr/2025/1958, ia.cr/2026/021



Summary

“Amber: Lattice-Based Threshold KEM with Active Security” | ia.cr/2025/1958, ia.cr/2026/021



Thank you!

References

[BGG+18] — Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. CRYPTO 2018

[KLO+19] — Michael Kraitsberg, Yehuda Lindell, Valery Osheter, Nigel P. Smart, and Younes Talibi Alaoui. Adding distributed decryption and key generation to a ring-LWE based CCA encryption scheme. ACISP 2019

[DLN+21] — Julien Devevey, Benoit Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. Non-interactive CCA2-secure threshold cryptosystems: Achieving adaptive security in the standard model without pairings. PKC 2021

[CCMS21] — Kelong Cong, Daniele Cozzo, Varun Maram, and Nigel P. Smart. Gladius: LWR based efficient hybrid public key encryption with distributed decryption. ASIACRYPT 2021

[BBH07] — Dan Boneh, Xavier Boyen, and Shai Halevi. Chosen ciphertext secure public key threshold encryption without random oracles. SIAM J. Comput. 2007

[CHK04] — Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. EUROCRYPT 2004

[CHKP10] — David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. EUROCRYPT 2010

[EEN+24] — Muhammed F. Esgin, Thomas Espitau, Guilhem Niot, Thomas Prest, Amin Sakzad, and Ron Steinfeld. Plover: Masking-friendly hash-and-sign lattice signatures. EUROCRYPT 2024

[PKM+24] — Rafaël Del Pino, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, and Markku-Juhani O. Saarinen. Threshold raccoon: Practical threshold signatures from standard lattice assumptions. EUROCRYPT 2024

[BCP+25] — Giacomo Borin, Sofía Celi, Rafaël del Pino, Thomas Espitau, Guilhem Niot, and Thomas Prest. Threshold signatures reloaded: ML-DSA and enhanced raccoon with identifiable aborts. Paper 2025/1166, 2025.