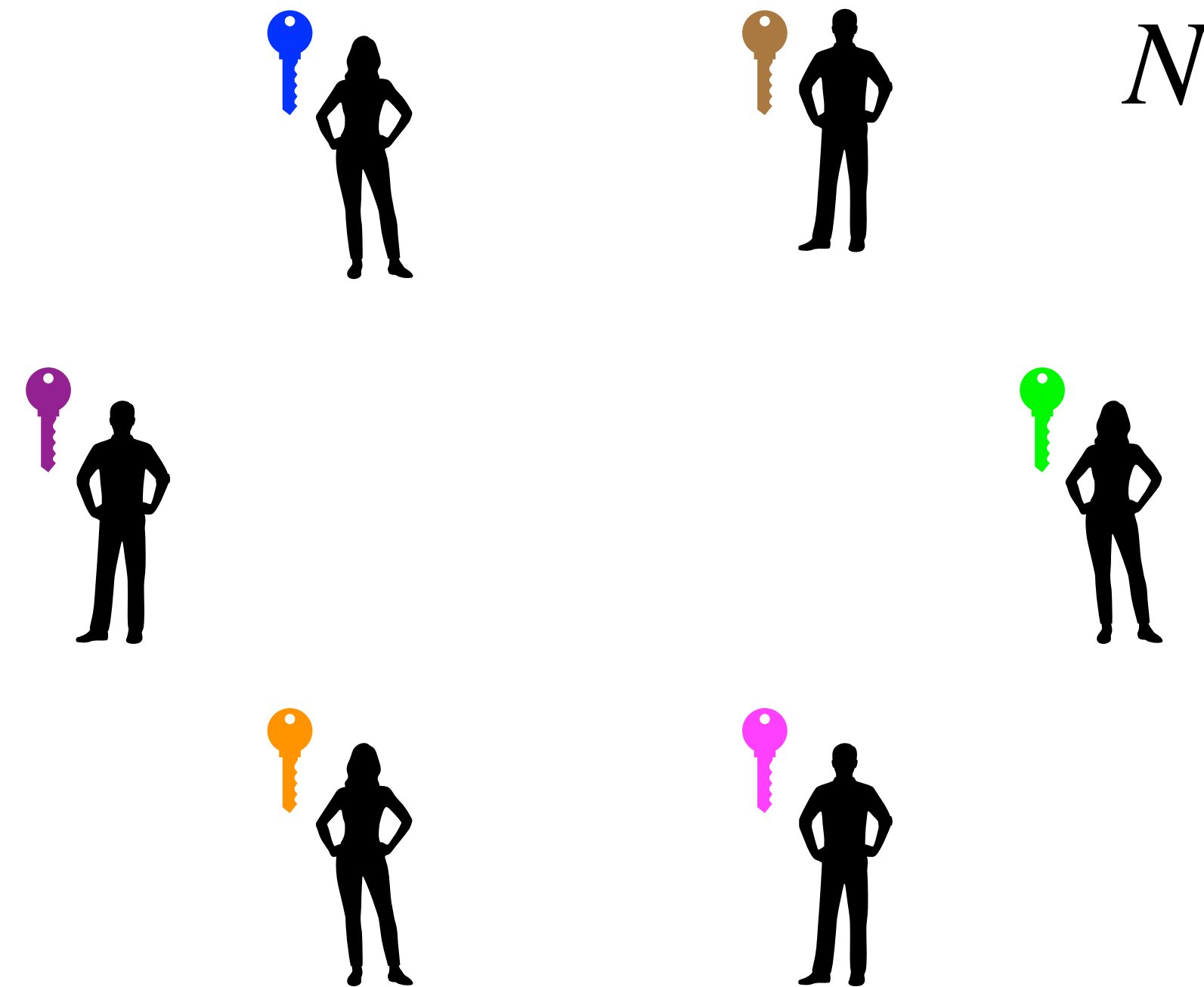


Amber: Lattice-Based Threshold KEM from the BCHK+ Transform

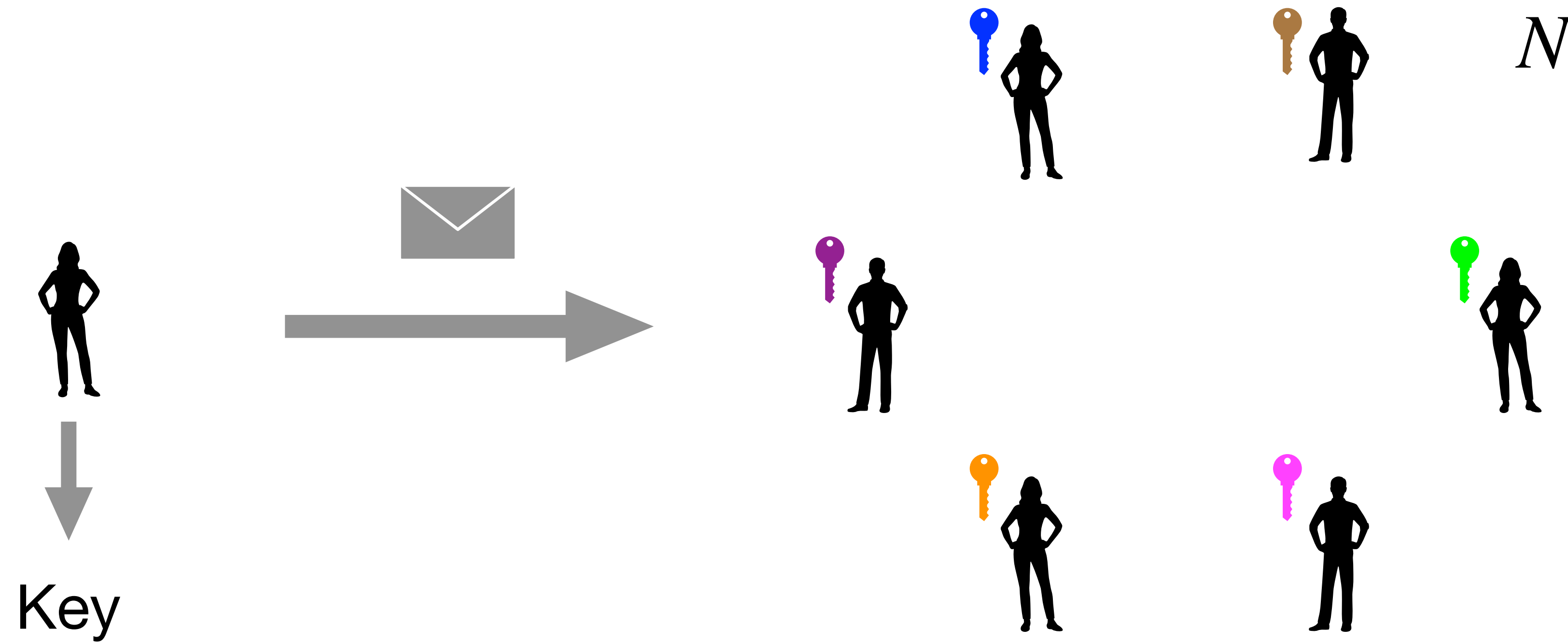
joint work with Katharina Boudgoust, Rafaël del Pino and Thomas Prest



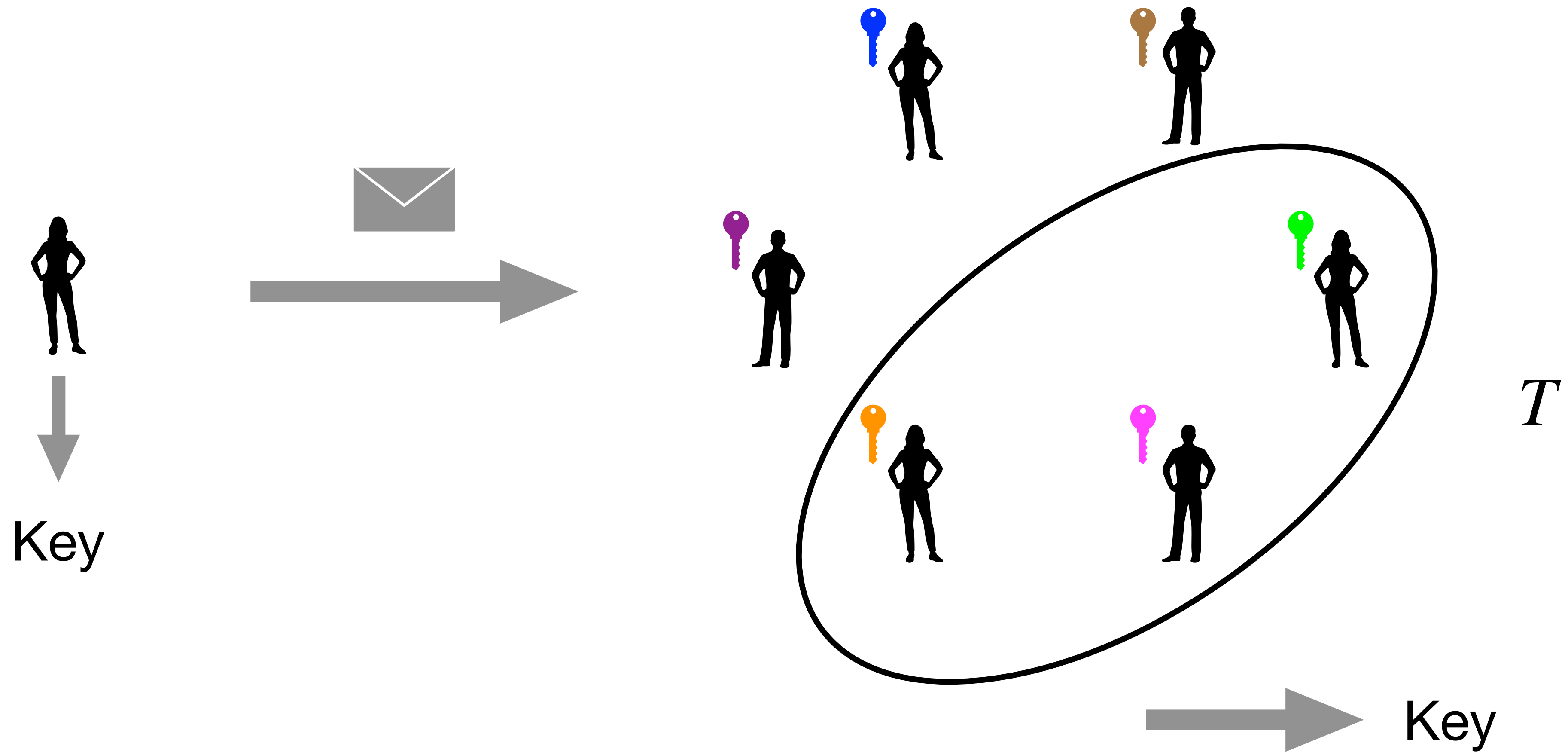
Threshold Decapsulation



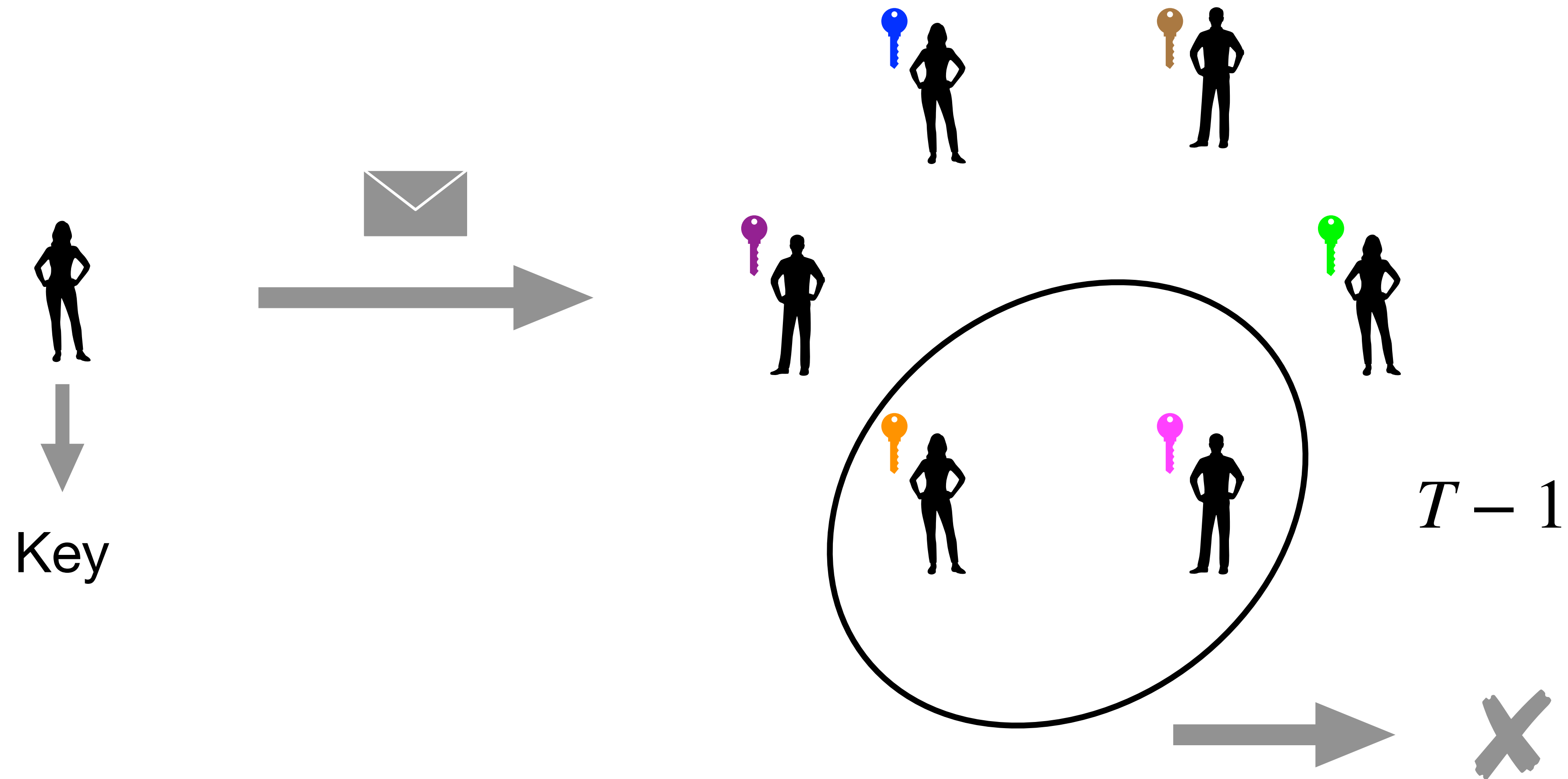
Threshold Decapsulation



Threshold Decapsulation

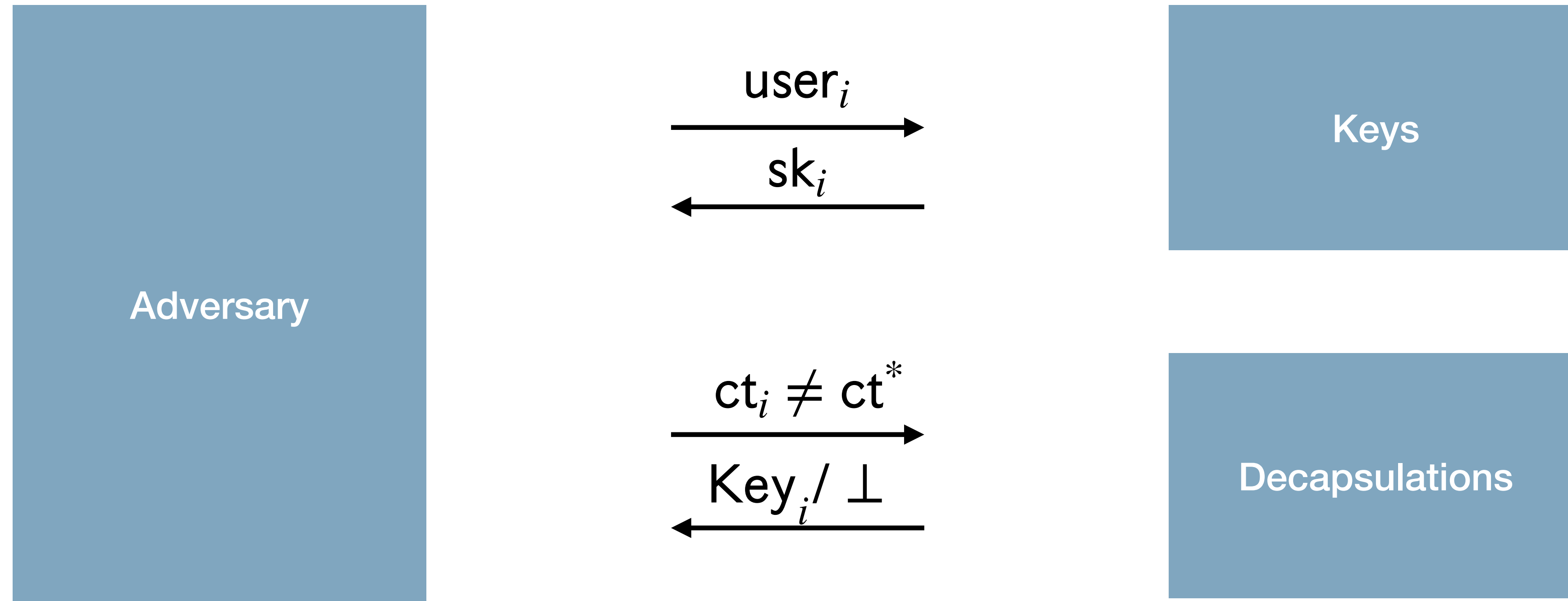


Threshold Decapsulation



Active Security of Threshold Decapsulation

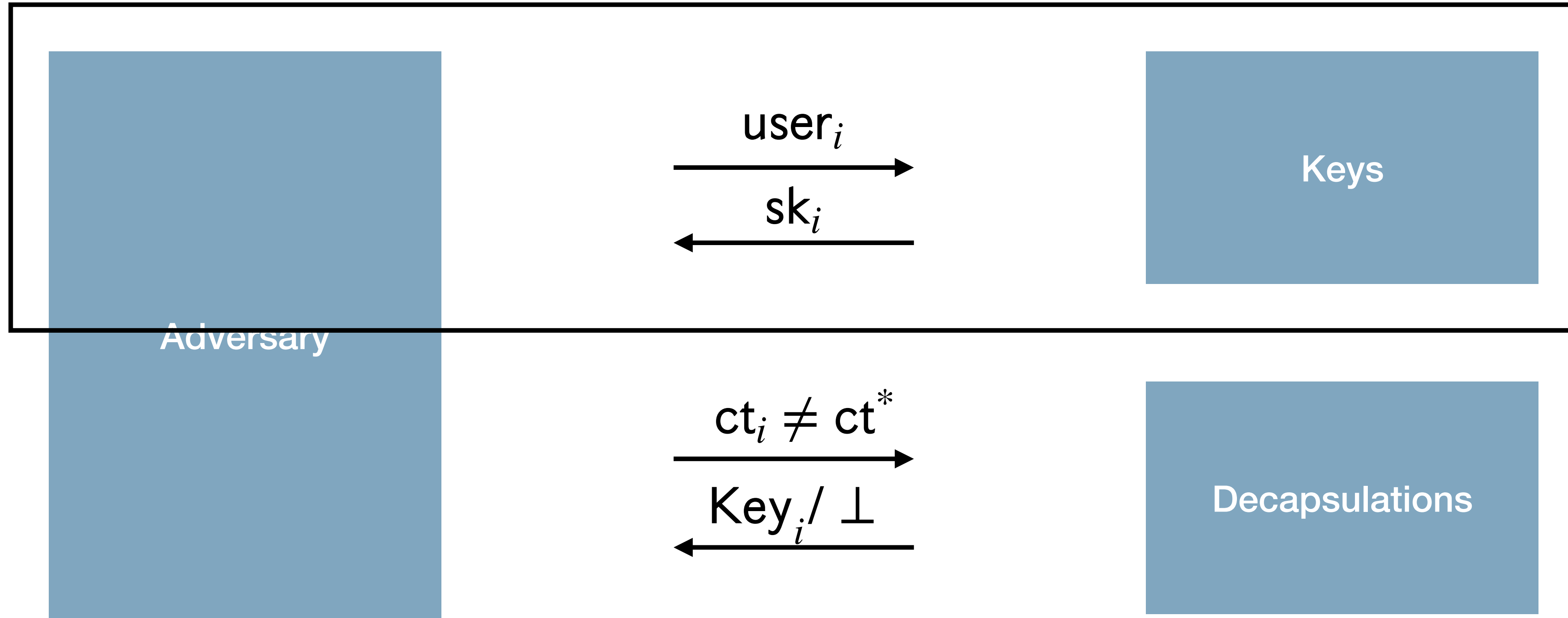
Security Model



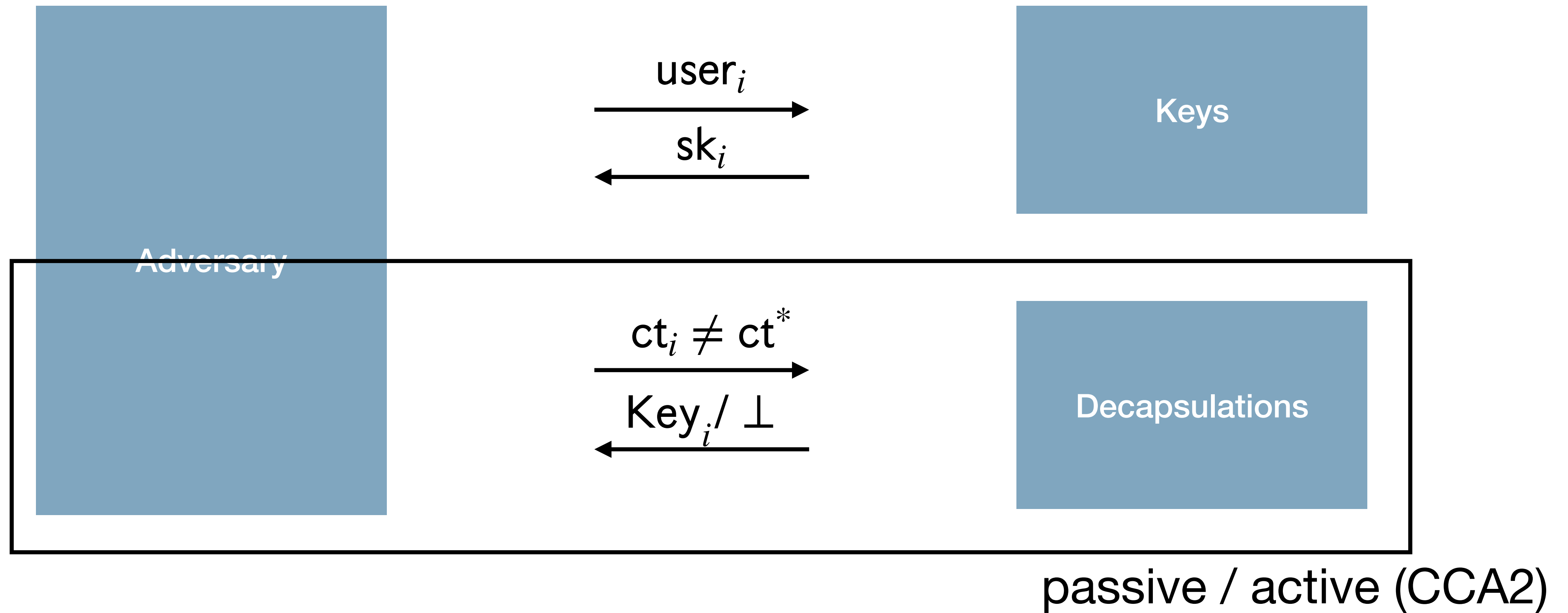
Security Model



static / adaptive, $< T$ users



Security Model



FO transform (non-threshold case)



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

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

FO transform (non-threshold case)



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

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



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

FO transform (non-threshold case)



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

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

$msg = Dec(ct, sk)$

$msg = Dec(ct, sk)$
 $rand = PRG(msg)$
 $Enc(msg, pk; rand) = = ct?$



Previous work



MPC
[CCMS21], [KLO+19]

FHE
[BGG+18]

ZK Proofs
[DLN+21]

Generic ?

BCHK Transform

BCHK Transform

[CHK04],[BCHK07],[BBH07]



Identity-based Encryption



```
(sk, pp) ← ibe-KeyGen()  
ct = ibe-Enc(msg, pp, id)
```

then

```
skid = ibe-Extract(sk, id)  
msg = ibe-Dec(ct, skid)
```

BCHK Transform



Encrypt(msg, pp) :

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

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

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

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

BCHK Transform



$ct_0 = \text{ibe-Enc}(\text{msg}, \text{id} = \text{sig-vk})$

$(\text{sig-sk}, \text{sig-vk}) \leftarrow \text{sig-KeyGen}()$
 $\text{sig} = \text{Sign}(ct_0, \text{sig-sk})$

BCHK Transform



Encrypt(msg, pp) :

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

Decrypt(ct, sk) :

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

BCHK Transform



Encrypt(msg, pp) :

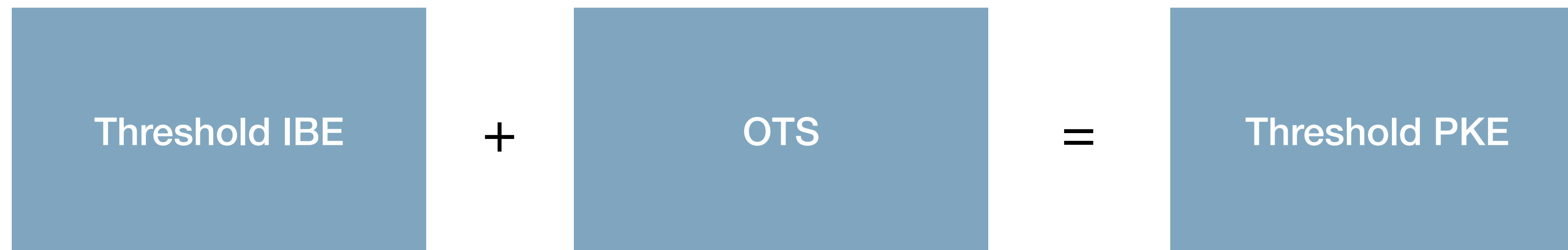
$(\text{sig-sk}, \text{sig-vk}) \leftarrow \text{sig-KeyGen}()$
 $\text{ct}_0 = \text{ibe-Enc}(\text{msg}, \text{id} = \text{sig-vk})$
 $\text{sig} = \text{Sign}(\text{ct}_0, \text{sig-sk})$
 $\text{ct} = (\text{ct}_0, \text{sig}, \text{sig-vk})$

Decrypt(ct, sk) :

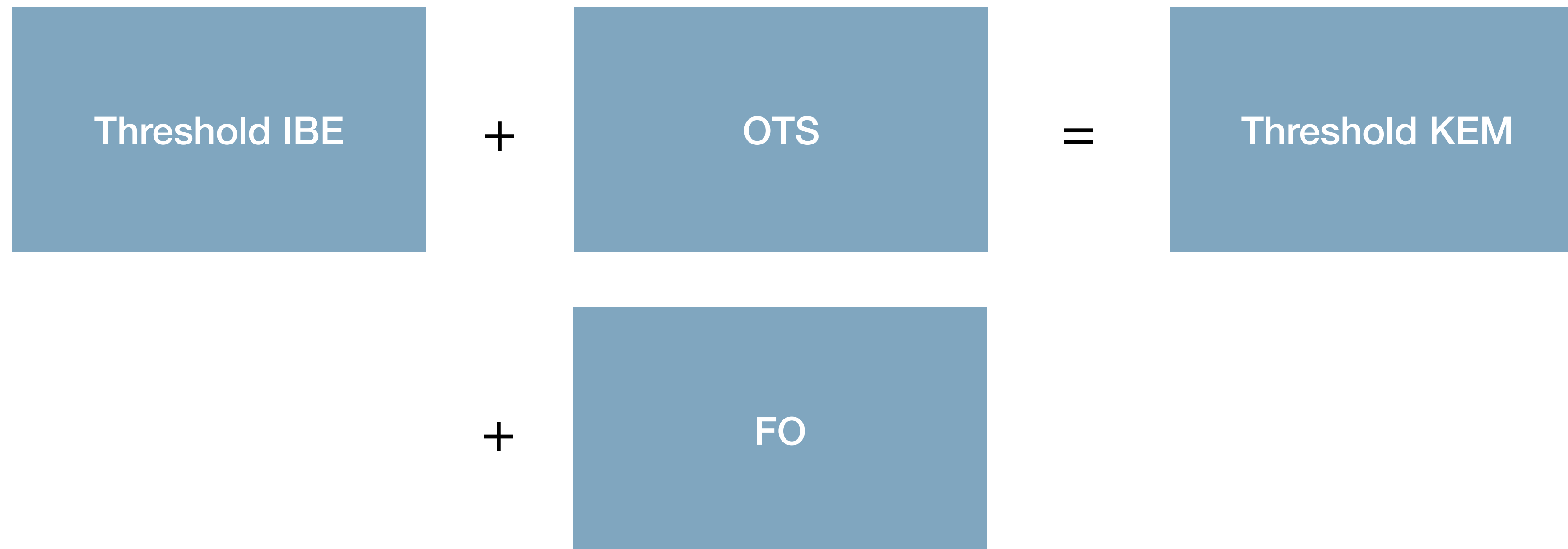
$\text{Verify}(\text{ct}_0, \text{sig}, \text{sig-vk}) = = 1?$
 $\text{sk}_{\text{id}} = \text{ibe-Extract}(\text{sk}, \text{id} = \text{sig-vk})$
 $\text{msg} = \text{ibe-Dec}(\text{ct}_0, \text{sk}_{\text{id}})$

Threshold BCHK

[BBH07]



BCHK₊ = BCHK + 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($\{dec_i\}$) :

$sk_{id} = Combine(\{dec_i\})$

$msg = ibe-Decrypt(ct, sk_{id})$

$rand = PRG(msg)$

$Enc(msg, pk; rand) == ct?$

BCHK+ Transform



```
Encrypt(msg, pp) :  
msg ← {0,1}128  
(sig-sk, sig-vk) ← sig-KeyGen()  
rand = PRG(msg)  
ct0 = ibe-Enc(msg, sig-vk; rand)  
sig = Sign(ct0, sig-sk)  
  
ct = (ct0, sig, sig-vk)
```

```
PartDecrypt(ct, ski) :
```

```
Verify(ct0, sig, sig-vk) == 1?  
deci = ibe-Extract(sig-vk, ski)
```

secret
sensitive

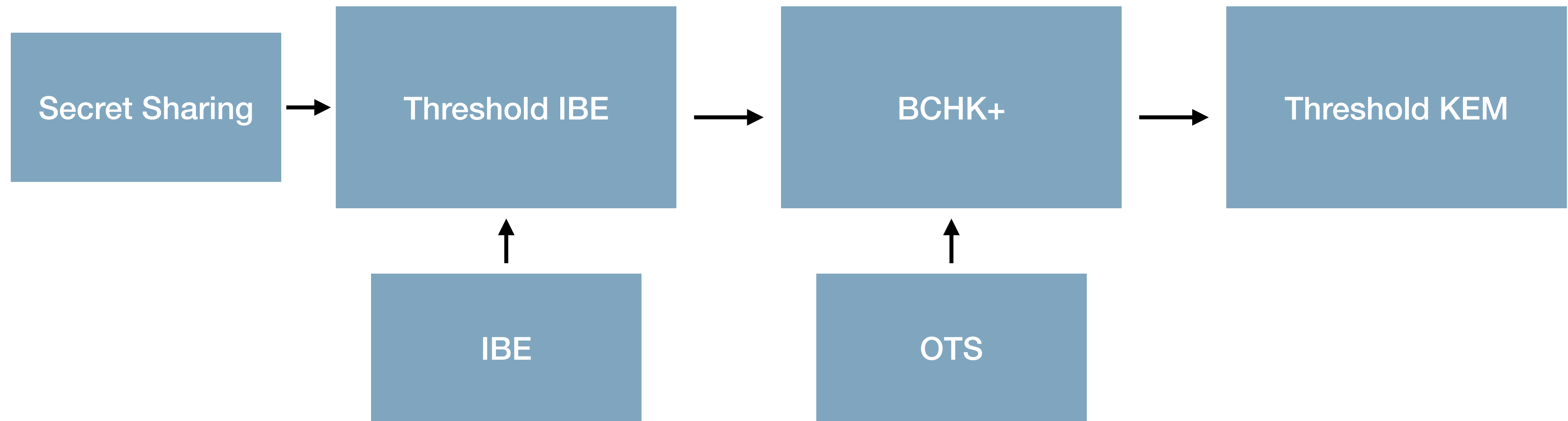
```
Combine({deci}) :
```

```
skid = Combine({deci})  
msg = ibe-Decrypt(ct, skid)  
rand = PRG(msg)  
Enc(msg, pk; rand) == ct?
```

public

Amber: The Lattice Construction

The Blueprint



The Ciphertext: Dual Regev Encryption



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

The Ciphertext



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

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

The Ciphertext



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

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

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

Starting IBE

ROHIBE [CHKP10]



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

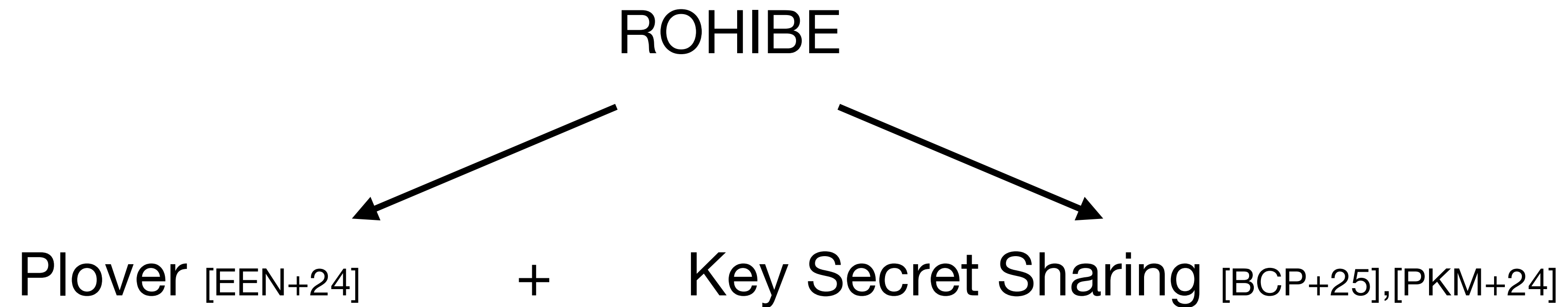
Starting IBE

ROHIBE [CHKP10]



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

Threshold IBE

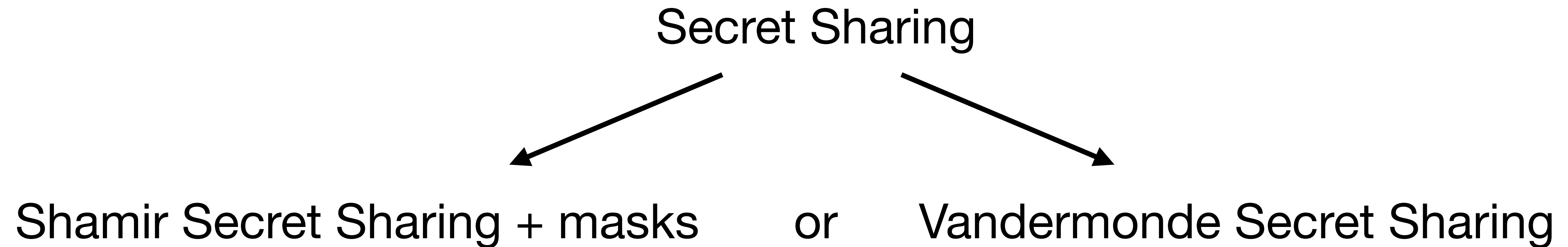


- $\mathbf{x} = c \cdot \mathbf{s} + \mathbf{p}$
- Schnorr-type protocol to compute $c \cdot \mathbf{s} + \mathbf{p}$

Secret Sharing



Secret key: short vector \mathbf{s}



Secret Sharing



Secret key: short vector \mathbf{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	$s = \sum \lambda_i s_i + \mathbf{m}_i$	Adaptive	No	Large (< 1024)
Vandermonde SS	$s = \sum s_i$	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.

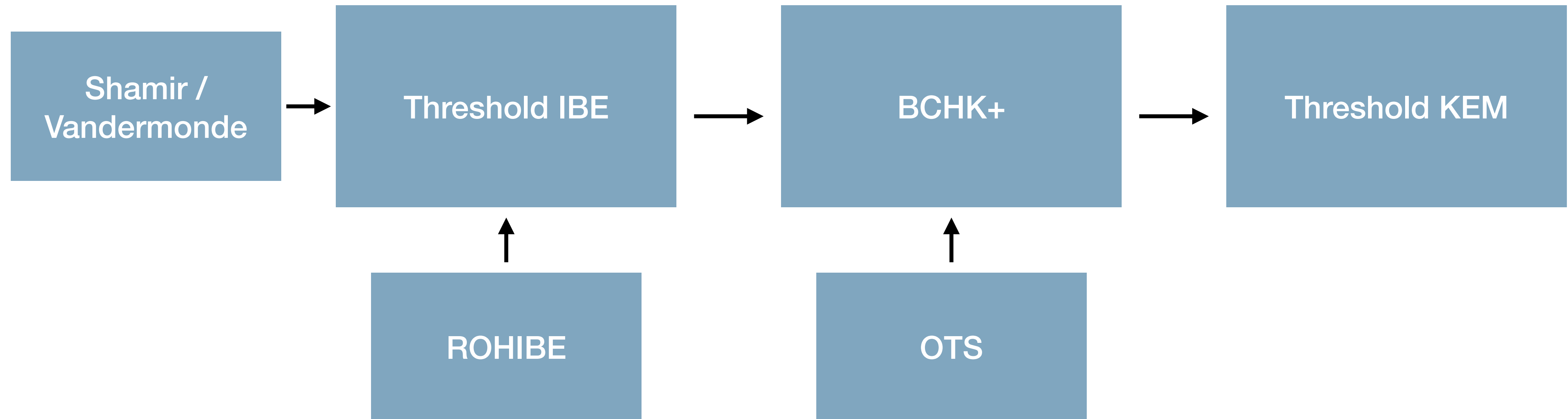
Future work / Your feedback



- ◆ 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

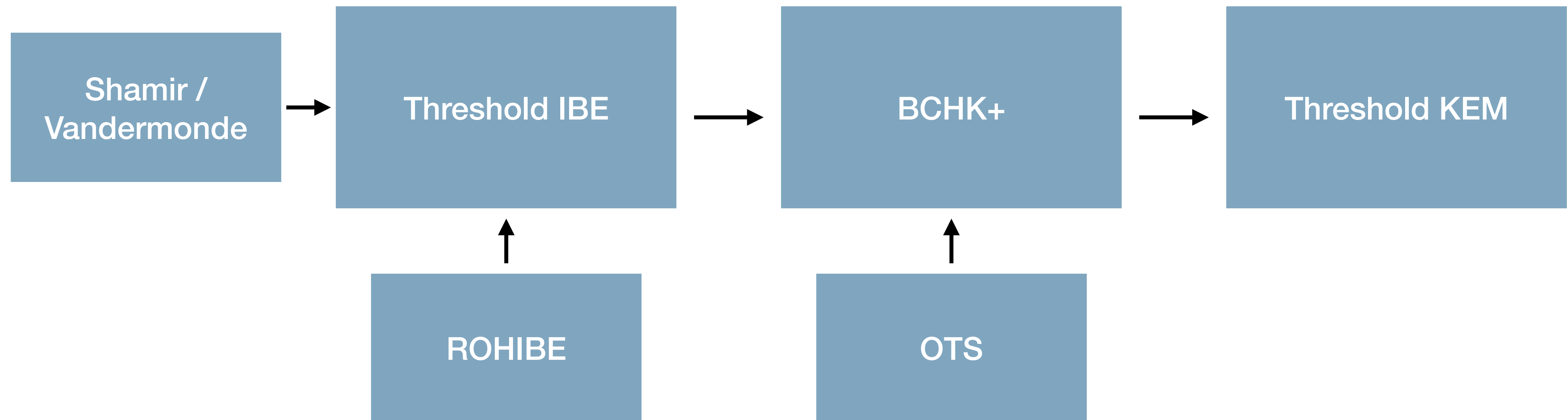
Summary

“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.