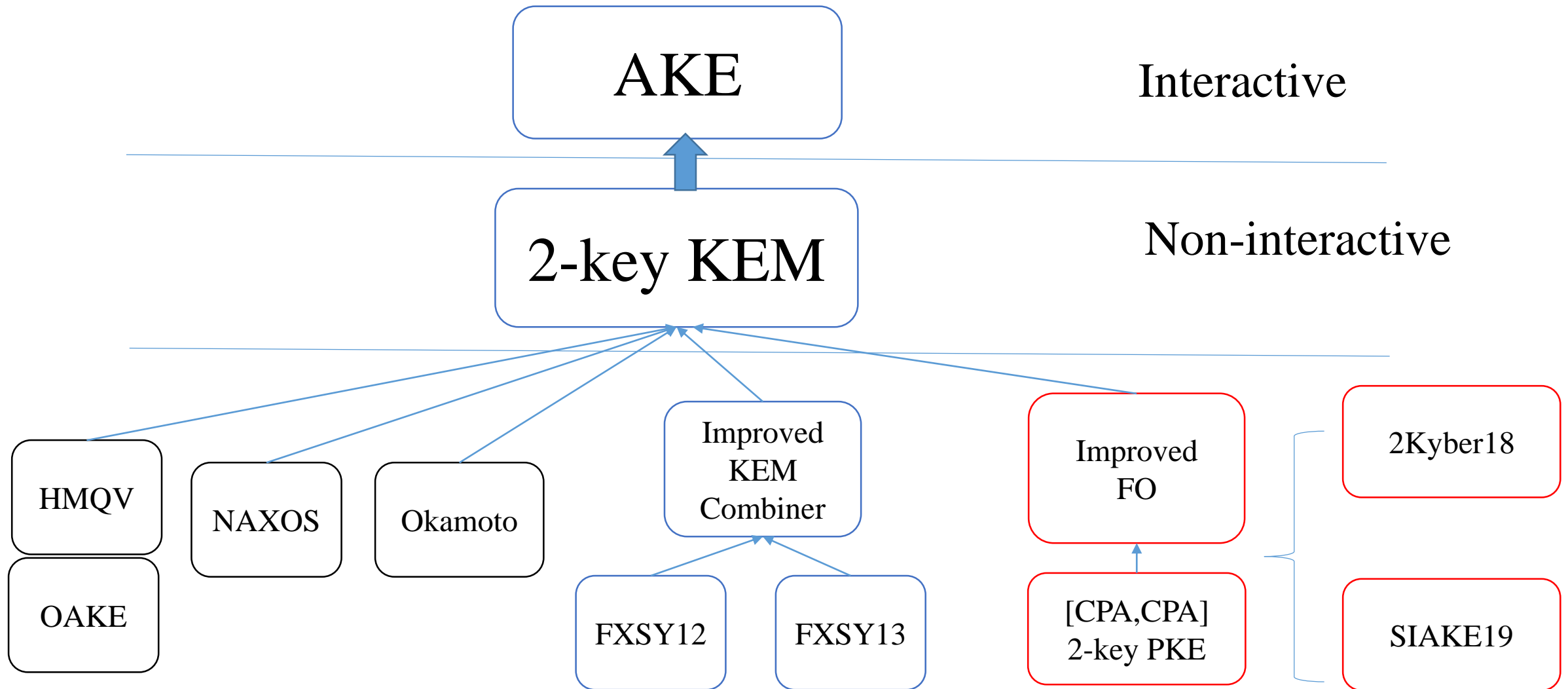


On the Constructions of Implicitly Authenticated Key Exchange

Haiyang Xue

2019.11.20

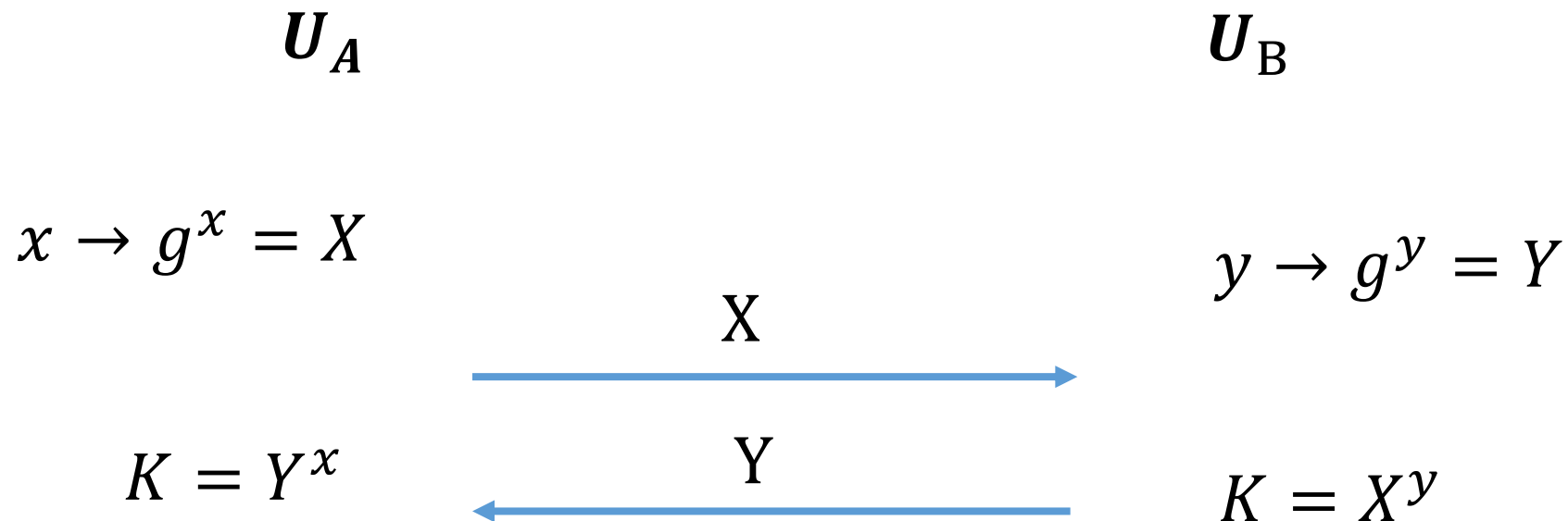
Roadmap



Outline

- Authenticated key exchange
- Motivations & our contributions
- $\text{AKE} \leftarrow \text{2-key KEM} \leftarrow$
- Post-quantum AKE

Diffie-Hellman Key Exchange [DH76]



- Passive secure under DDH assumption
- Adaptive attacks: Man-in-the-middle attack etc.
- Basic and general idea: Authenticated Key Exchange (AKE)

Authenticated Key Exchange

- **Authenticated** Key Exchange (AKE). Binding id with static public key using PKI etc.

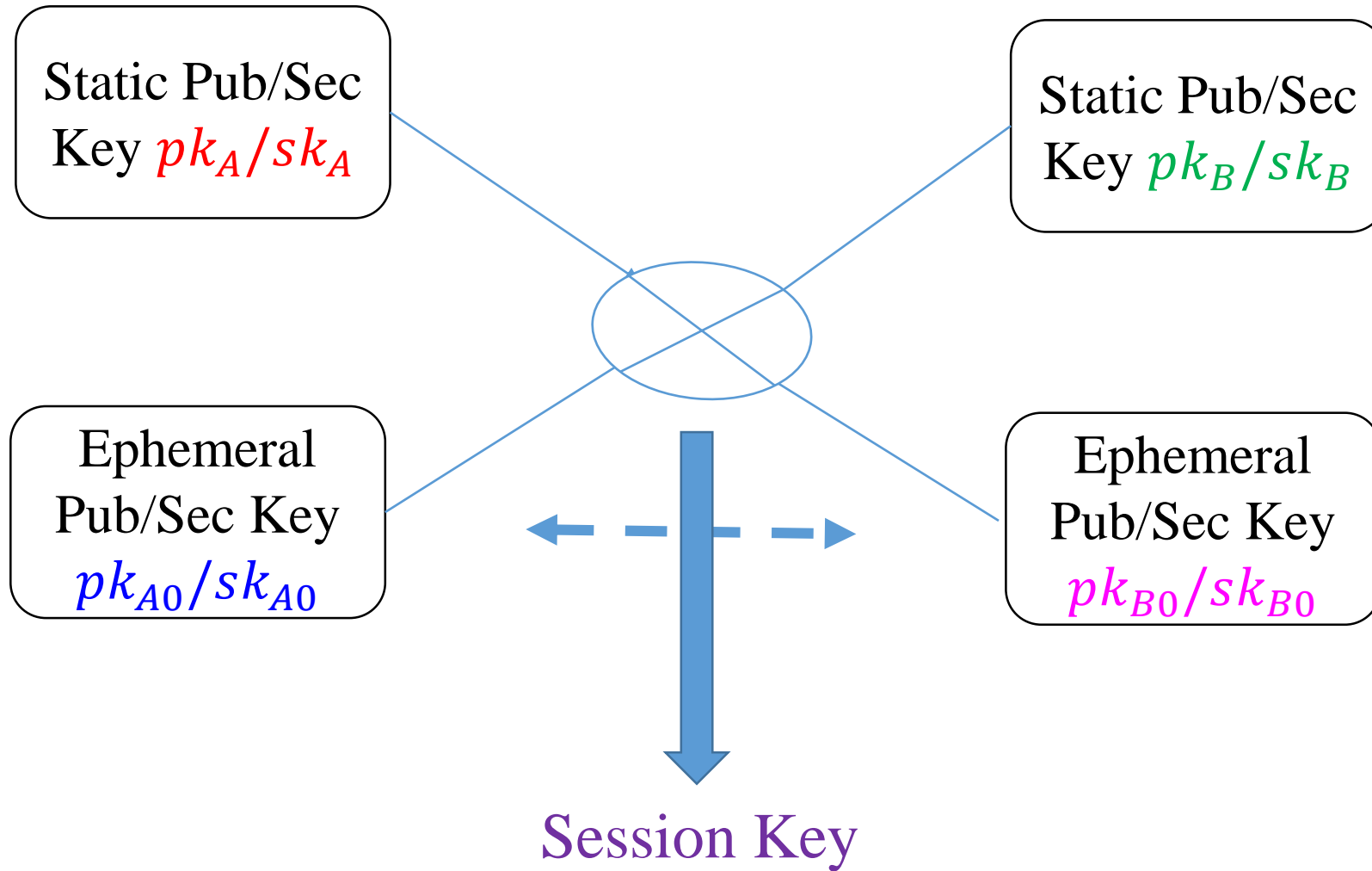
1. Security models

BR model, CK model, HMQV-CK, eCK model, CK+ model

2. Constructions

- **Explicit**: BR, CK01, IKE, Krawczyk03(SIGMA), ..., Peikert14 etc.
- **Implicit**: MTI, MQV, HMQV, OAKE, Okamoto07, NAXOS, BCNP+09, FSXY12-13 etc

General Structure of AKE

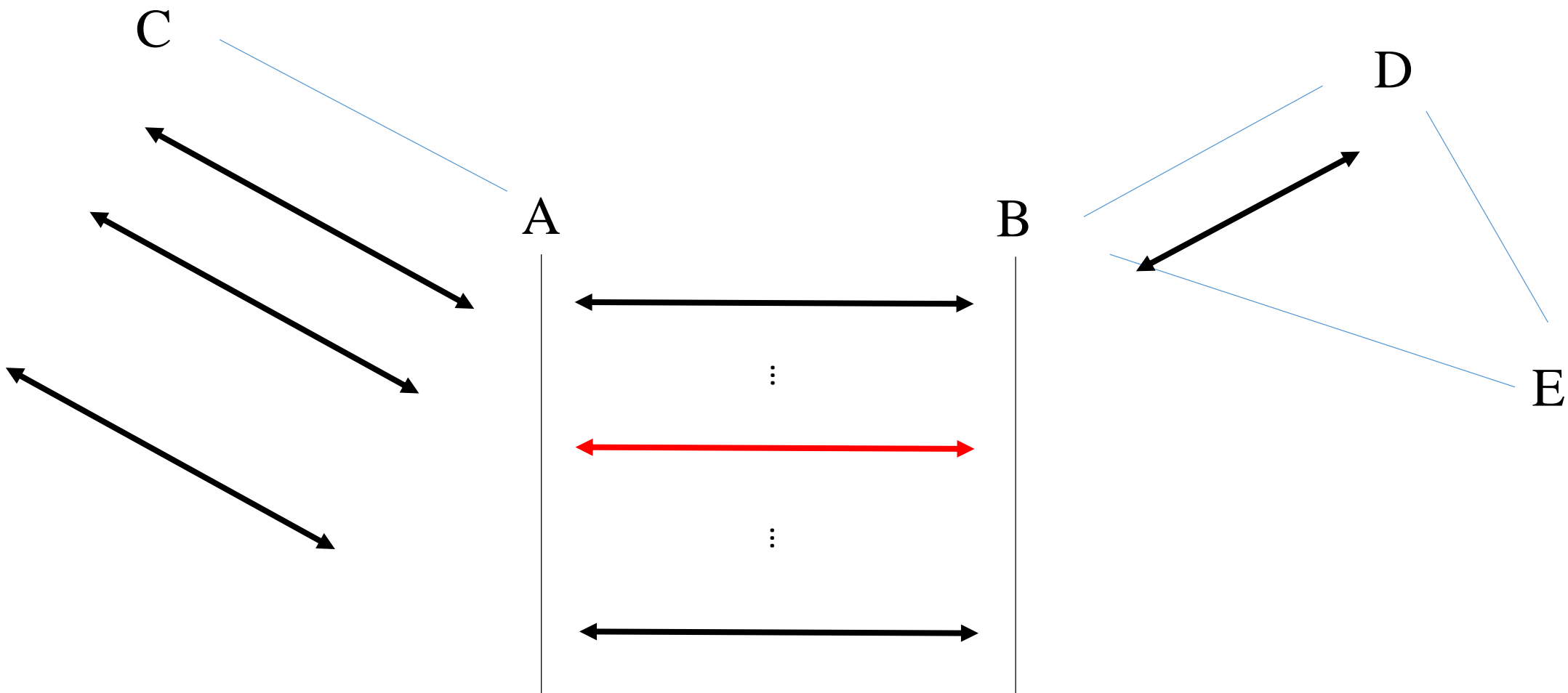


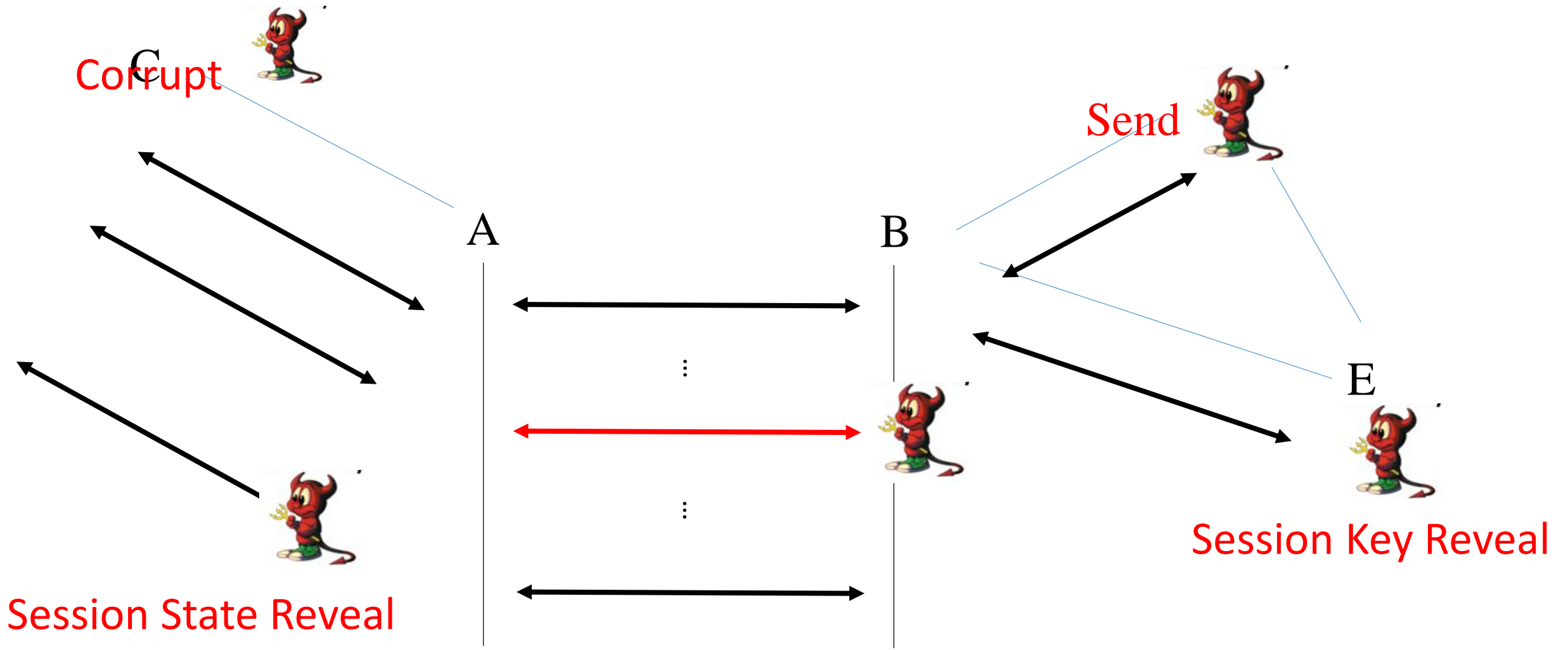
Challenges of AKE

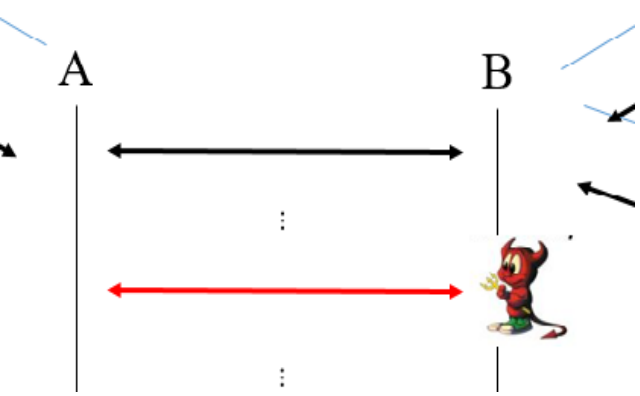
- The models are tedious to describe and difficult to get right;
- just describing a concrete protocol itself can be hard enough;
- the security proofs and checking even more so.

Security of AKE

- Bellare-Rogaway 93 (**BR93**)
indistinguishable type definition
- Canetti-Krawczyk 01(**CK01**)
stronger security (session key, **session state**)
- LaMacchia-Lauter-Mityagin 07 (**eCK**)
stronger (session key, **ephemeral randomness**, wPFS+KCI+MEX)
- Fujioka-Suzuki-Xagawa-Yoneyama 12 (**CK+**)
reform the security of HMQV: CK01+wPFS+KCI+MEX







Event	Case	sid^*	$\overline{sid^*}$	ssk_A	esk_A	esk_B	ssk_B	Security
E_1	1	A	No	\checkmark	\times	-	\times	KCI
E_2	2	A	No	\times	\checkmark	-	\times	MEX
E_3	2	B	No	\times	-	\checkmark	\times	MEX
E_4	1	B	No	\times	-	\times	\checkmark	KCI
E_5	5	A or B	Yes	\checkmark	\times	\times	\checkmark	wPFS
E_6	4	A or B	Yes	\times	\checkmark	\checkmark	\times	MEX
E_{7-1}	3	A	Yes	\checkmark	\times	\checkmark	\times	KCI
E_{7-2}	3	B	Yes	\times	\checkmark	\times	\checkmark	KCI
E_{8-1}	6	A	Yes	\times	\checkmark	\times	\checkmark	KCI
E_{8-2}	6	B	Yes	\checkmark	\times	\checkmark	\times	KCI

\checkmark it may be leaked to adversary; \times it is secure; - means it does not exists

Outline

- Authenticated key exchange
- Motivations & our contributions
- $\text{AKE} \leftarrow \text{2-key KEM} \leftarrow$
- Post-quantum AKE

Constructions of AKE

- Explicit AKE: using additional primitives i.e., **signature** or **MAC**

1. IKE, Canetti-Krawczyk 02

2. SIGMA, Krawczyk 03, **Peikert 14**

3. TLS, Krawczyk 02

Constructions of AKE

- Implicit AKE: **unique** ability so as to compute the resulted session key
 1. **MTI 86**: the first one
 2. **MQV 95**: various attacks
 3. **HMQR 05**: the first provable secure implicit-AKE via gap-DH and KEA
 4. **YZ13**: OAKE
 5. **Okamoto 07**: in standard model from DDH (Hashing Proof Sys.)
 6. **LLM 07**: NAXOS scheme from gap-DBDH
 7. **Boyd et al. 08**: Diffie-Hellman+KEM
 8. **FSXY 12** (2CCA+CPA-KEM, std.), **FSXY 13** (2CCAKEM,RO)
 9. **ZZD+15** HMQR-type based on RLWE with weaker aim

Motivation

- Explicit AKE



SIGMA

Krawczyk 03

- Implicit AKE



???

Motivations

- What is the (non-interactive) core building block of implicit AKE?
- How to grasp and simplify the construction and analysis of implicit AKE?

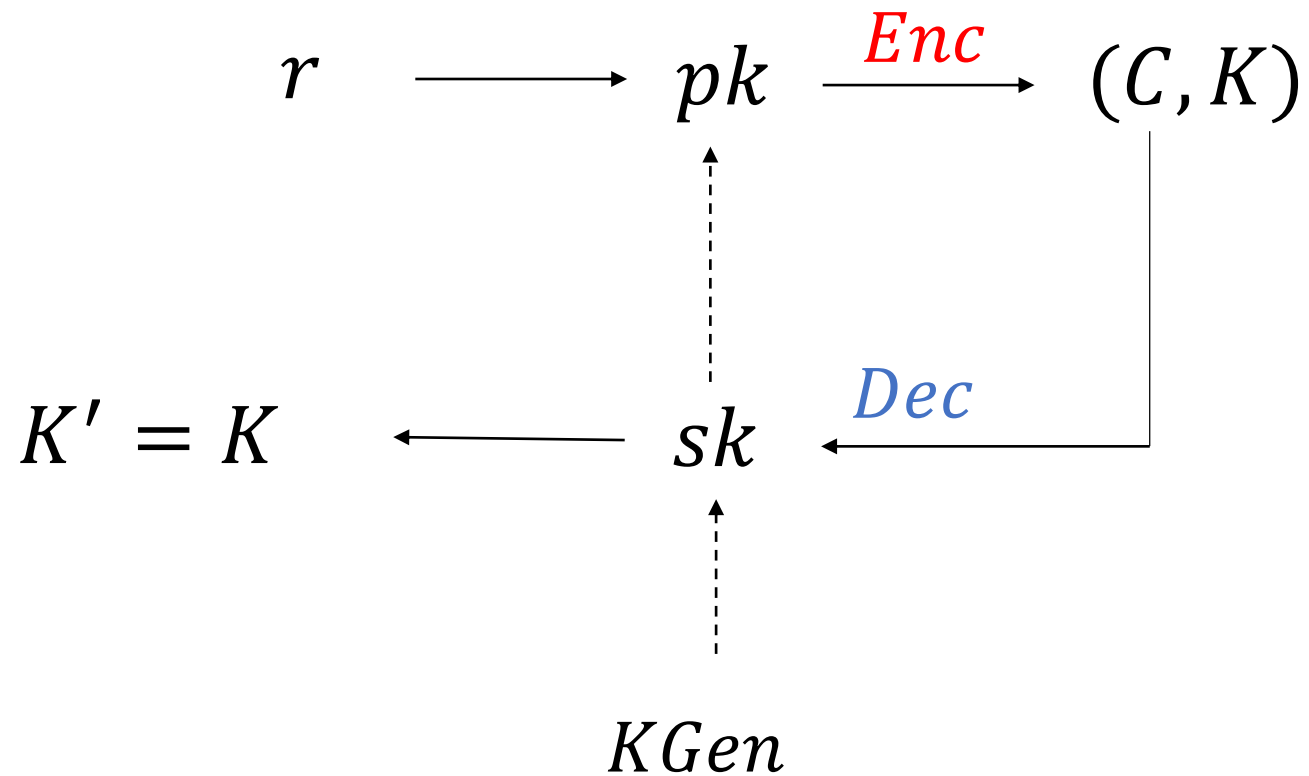
Our Works

- What is the (non-interactive) core building block of implicit AKE?
- propose a new primitive 2-key PKE/KEM
- How to grasp and simplify the construction and analysis of AKE?
- give frames of AKE to understand several well-know AKEs
- construct new AKEs from 2-key PKE/KEM

Outline

- Authenticated key exchange
- Motivations & our contributions
- $\text{AKE} \leftarrow \text{2-key KEM} \leftarrow$
- Post-quantum AKE

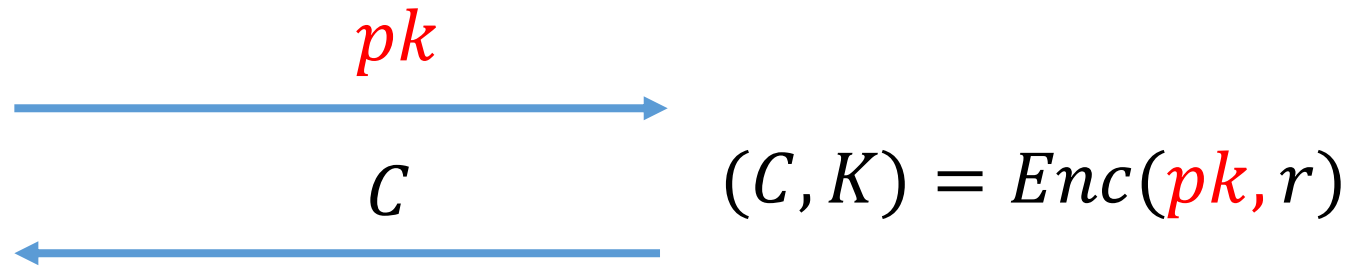
Key Encapsulation Mechanism(KEM)



Key Exchange (transport) and KEM

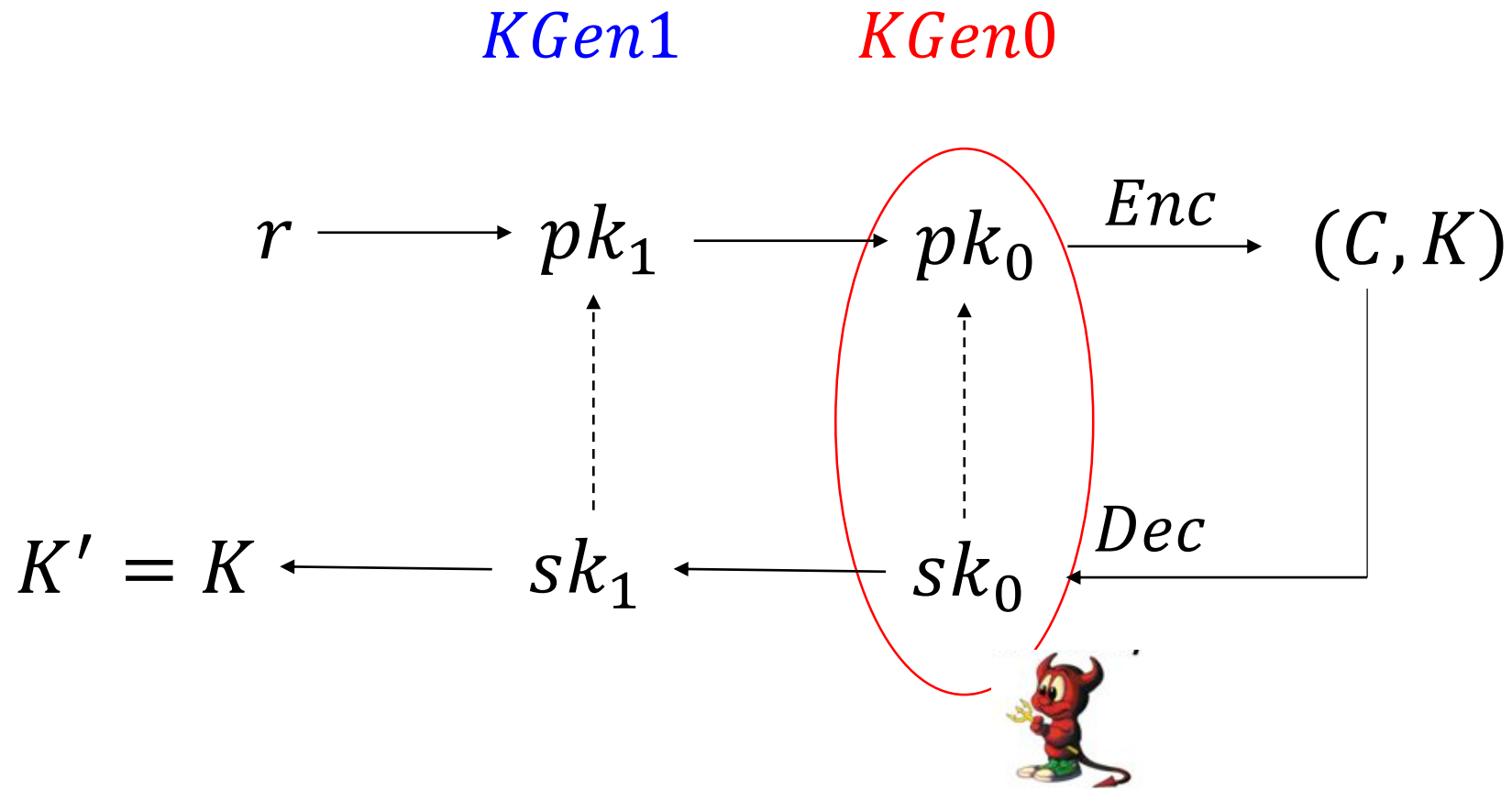
U_A

U_B



$$Dec(sk, C) = K = Enc(pk, r)$$

Our 2-key KEM



It is simple, not a big deal

One-side AKE from 2-key KEM?

U_A
 pk_1

U_B

pk_0

C

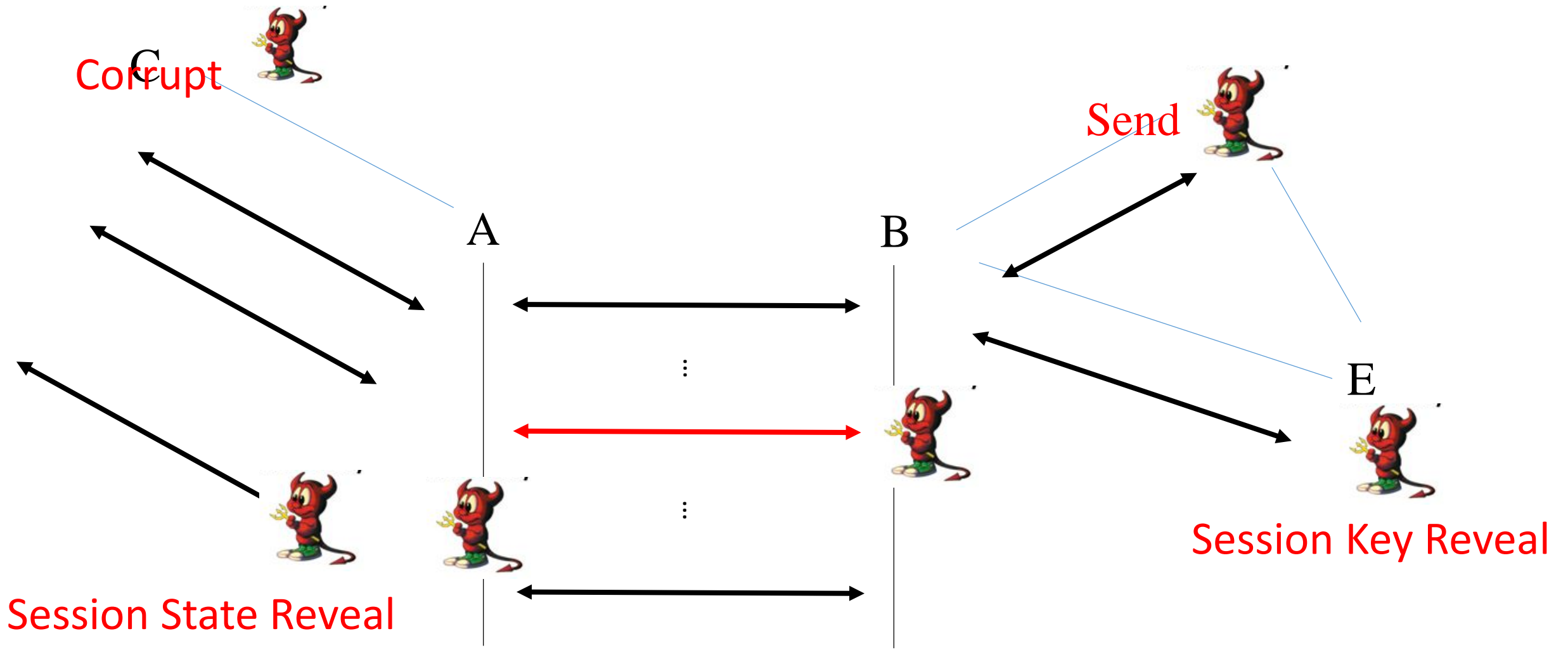
$(C, K) = Enc(pk_1, pk_0, R_B)$

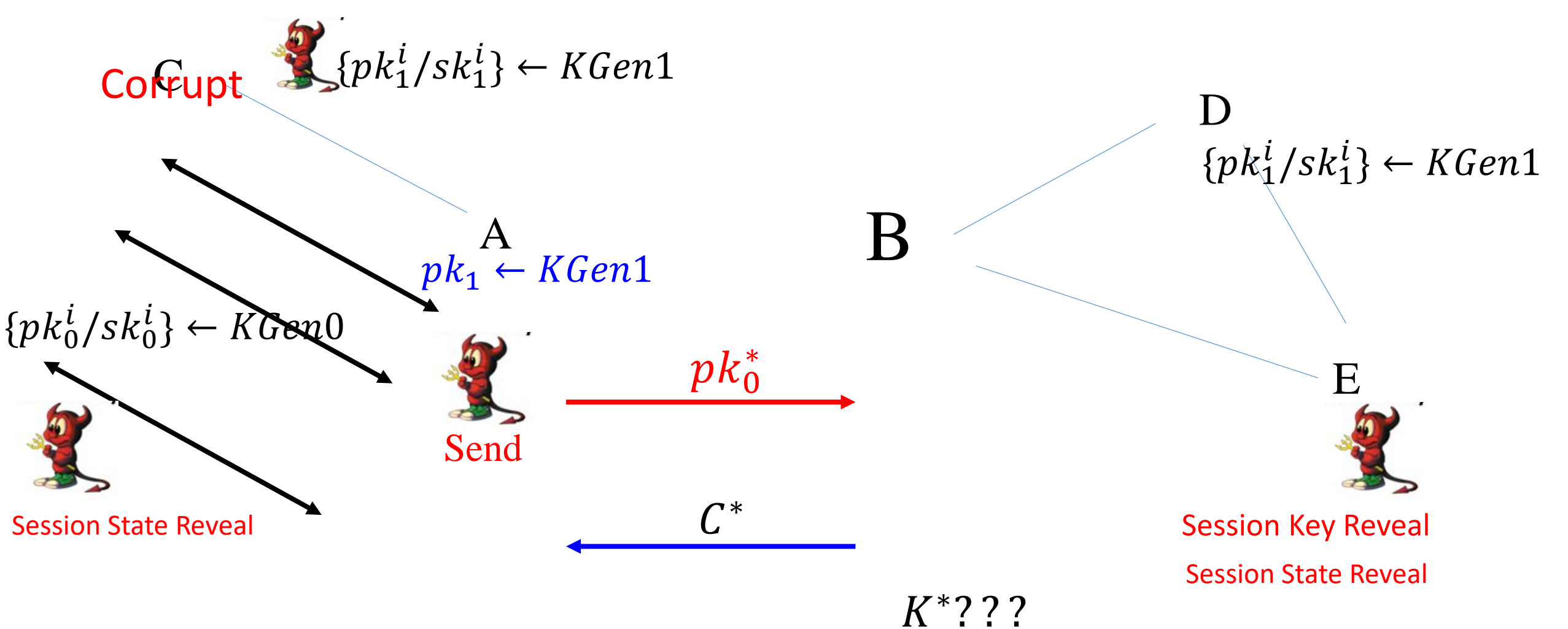
$Dec(sk_1, sk_0, C) = K$

The key point is how to define its security to fit the requirement of AKE

CK+ security

sid*	$\overline{\text{sid}^*}$	ssk_A	esk_A	esk_B	ssk_B	Bounds
A	No	\checkmark	\times	-	\times	$\text{Adv}_{2\text{KEM}}^{[\text{OW-CCA}, \cdot]}, pk_1 = pk_B, pk_0^* = cpk_0$
A	No	\times	\checkmark	-	\times	$\text{Adv}_{2\text{KEM}}^{[\text{OW-CCA}, \cdot]}, pk_1 = pk_B, pk_0^* = cpk_0$
B	No	\times	-	\checkmark	\times	$\text{Adv}_{2\text{KEM}}^{[\text{OW-CCA}, \cdot]}, pk_1 = pk_A, pk_0^* \leftarrow \mathcal{A}$
B	No	\times	-	\times	\checkmark	$\text{Adv}_{2\text{KEM}}^{[\text{OW-CCA}, \cdot]}, pk_1 = pk_A, pk_0^* \leftarrow \mathcal{A}$
A/B	Yes	\checkmark	\times	\times	\checkmark	$\text{Adv}_{2\text{KEM}}^{[\cdot, \text{OW-CPA}]}, pk_0 = pk_0(\text{sid}^*) pk_1^* \in [L_1]_1$
A/B	Yes	\times	\checkmark	\checkmark	\times	$\text{Adv}_{2\text{KEM}}^{[\text{OW-CCA}, \cdot]}, pk_1 = pk_A, pk_0^* \in [L_0]_1$
A	Yes	\checkmark	\times	\checkmark	\times	$\text{Adv}_{2\text{KEM}}^{[\text{OW-CCA}, \cdot]}, pk_1 = pk_B, pk_0^* = cpk_0$
B	Yes	\times	\checkmark	\times	\checkmark	$\text{Adv}_{2\text{KEM}}^{[\text{OW-CCA}, \cdot]}, pk_1 = pk_A, pk_0^* \in [L_0]_1$
A	Yes	\times	\checkmark	\times	\checkmark	$\text{Adv}_{2\text{KEM}}^{[\text{OW-CCA}, \cdot]}, pk_1 = pk_A, pk_0^* \in [L_0]_1$
B	Yes	\checkmark	\times	\checkmark	\times	$\text{Adv}_{2\text{KEM}}^{[\text{OW-CCA}, \cdot]}, pk_1 = pk_B, pk_0^* = cpk_0$





$$SK = H(sid, K_A, K^*)$$

DecO

pk'_0, C'

If $pk'_0 \in L$
 $K' = Dec(sk_1, sk'_0, C')$

Send Adv

$[CPA, \cdot]$

C

A

\mathcal{B}

$pk_1 \leftarrow KGen1$

Send



pk_0^*



C^*



$K^* ???$

$SK = H(sid, K_A, K^*)$

pk_0^i



pk_0^i, sk_0^i



C'

pk_1, L



$pk_1 \leftarrow KGen1,$

$L = \{pk_0^i, sk_0^i, r_0^i\} \leftarrow KGen0$

pk_0^*



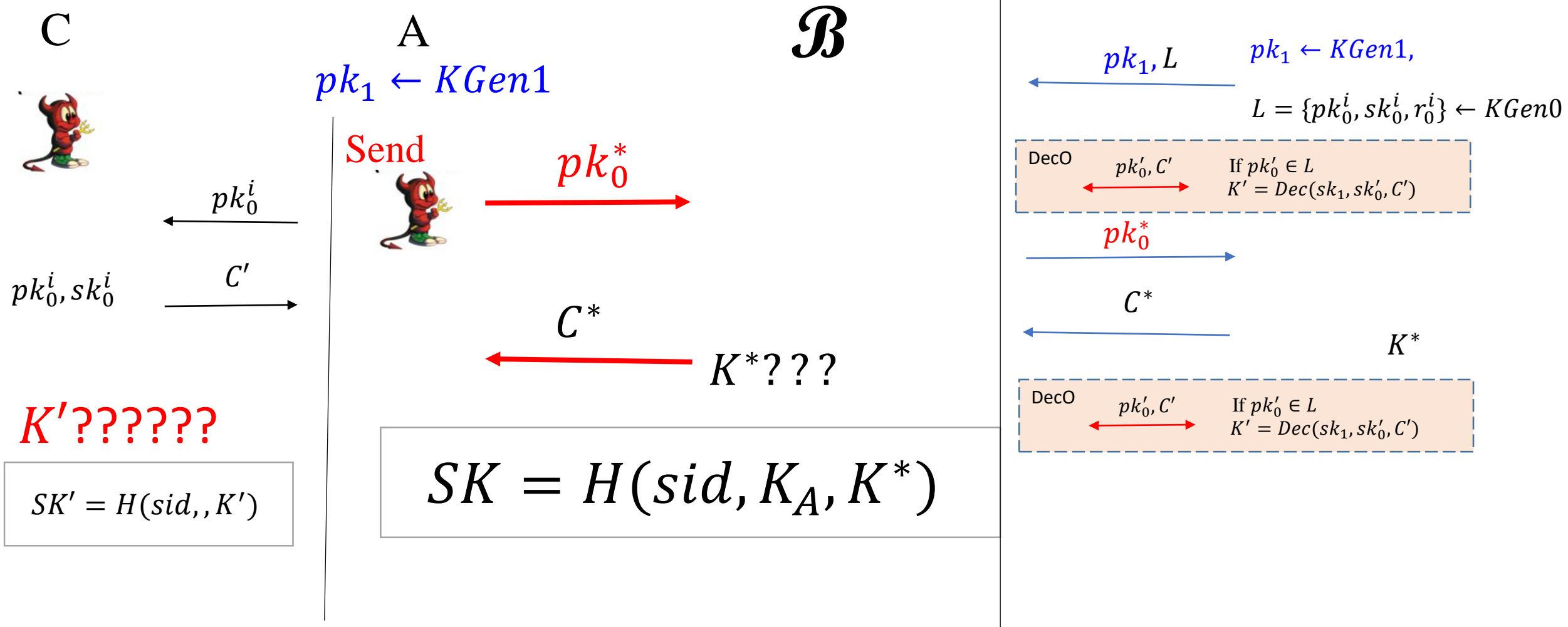
C^*



K^*

Send Adv + Session Key/State Reveal

[CCA, ·]



Case 6-10

[CCA, ·]

C

A

B

$pk_1 \leftarrow KGen1$



pk_0^i

pk_0^i, sk_0^i

C'

$K'??????$

$SK' = H(sid, K')$

pk_0^*

C^*

$K^*????$

$SK = H(sid, K_A, K^*)$

$pk_0^* \quad pk_1, L \quad pk_1 \leftarrow KGen1,$
 $L = \{pk_0^i, sk_0^i, r_0^i\} \leftarrow KGen0$

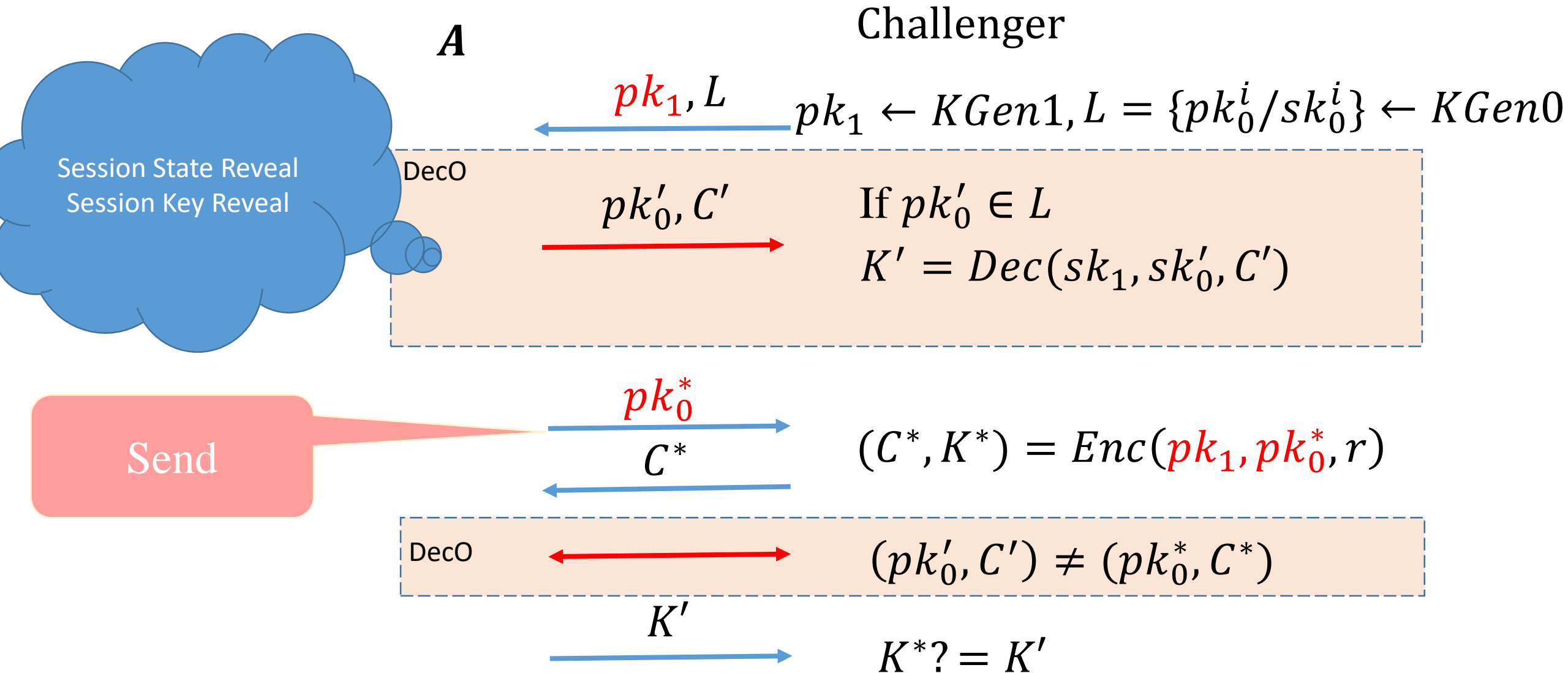
DecO pk_0', C' If $pk_0' \in L$
 $K' = Dec(sk_1, sk_0', C')$

C^*

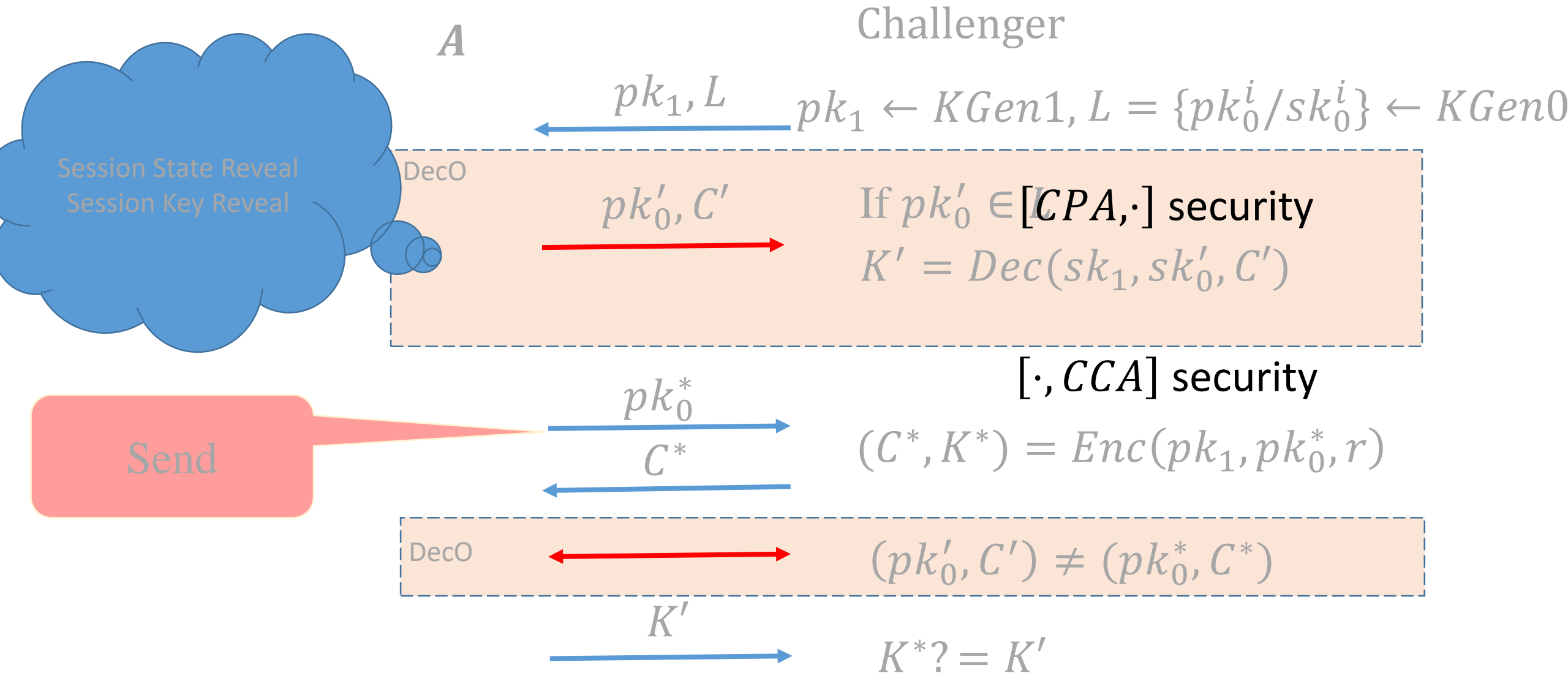
K^*

DecO pk_0', C' If $pk_0' \in L$
 $K' = Dec(sk_1, sk_0', C')$

$[CCA, \cdot]$ Security of 2-key KEM



$[CCA, \cdot]$ Security of 2-key KEM



One-side AKE from [CCA, CPA] 2-key KEM

U_A pk_{A1}

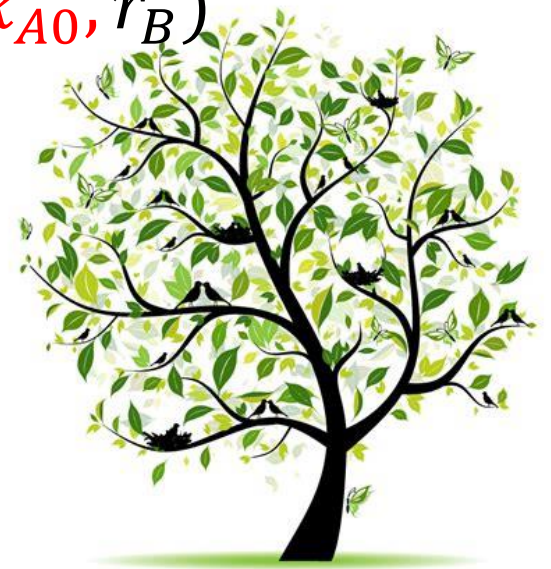
U_B

pk_{A0}

C

$(C, K) = Enc(pk_{A1}, pk_{A0}, r_B)$

$K = Dec(sk_{A1}, sk_{A0}, C)$



The other side AKE from $[CCA, CPA]$ 2-key KEM

U_A

U_B pk_{B1}

pk_{B0}

$(C_B, K_B) = Enc(pk_{B1}, pk_{B0})$

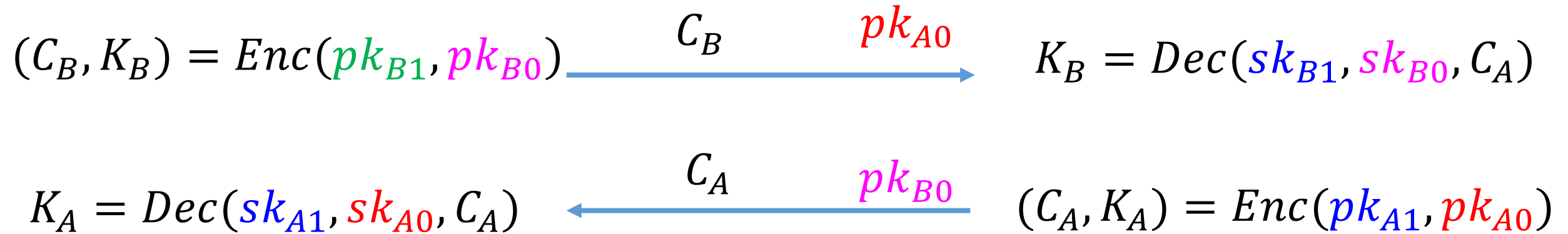
C_B

$K_B = Dec(sk_{B1}, sk_{B0}, C_A)$

Main AKE frame? $\leftarrow [CCA, CPA]$ 2-key KEM

U_A pk_{A1}

U_B pk_{B1}



$$K = \text{Hash}(\text{sid}, K_A, K_B) \text{ or } \text{PRF}(K_B) \oplus \text{PRF}(K_A)$$

Several AKE frames with Tricks

$U_A \quad pk_{A1}$

$U_B \quad pk_{B1}$

$$(C_B, K_B) = Enc(pk_{B1}, pk_{B0}) \xrightarrow{pk_{A0} \quad C_B} K_B = Dec(sk_{B1}, sk_{B0}, C_A)$$

$$K_A = Dec(sk_{A1}, sk_{A0})$$

Trick 1

All the randomness for *Enc* and *KGen0*
is generated from both *ephemeral secret* r_{A0}
and *static secret key* sk_A

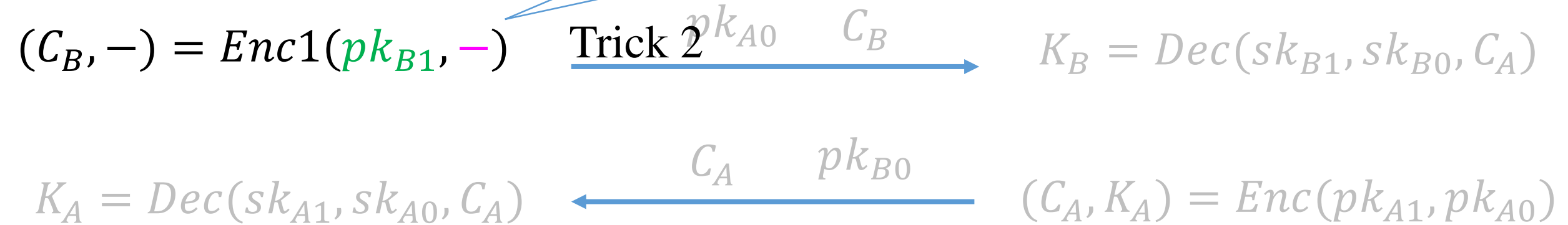
$$) = Enc(pk_{A1}, pk_{A0})$$

$$K = Hash(sid, K_A, K_B) \text{ or } PRF(K_B) \oplus PRF(K_A)$$

Several AKE frames with Tricks

U_A pk_{A1}

2-key KEM is public key pk_{B0} independent



$$K = \text{Hash}(\text{sid}, K_A, K_B) \text{ or } \text{PRF}(K_B) \oplus \text{PRF}(K_A)$$

Several AKE frames with Tricks

$U_A \quad pk_{A1}$

$U_B \quad pk_{B1}$

$$(C_B, K_B) = Enc(pk_{B1}, pk_{B0}) \xrightarrow[\epsilon_B]{pk_{A0}} K_B = Dec(sk_{B1}, sk_{B0}, C_A)$$

$$K_A = Dec(sk_{A1}, sk_{A0}, C_A) \xleftarrow[\epsilon_A]{pk_{B0}} (C_A, K_A) = Enc(pk_{A1}, pk_{A0})$$

Trick 3

C_B can be publicly computed from pk_{A0}
 C_A can be publicly computed from pk_{B0}

$$K = Hash(sk_{A0} \parallel A \parallel B \parallel sk_{B0} \parallel B \parallel A \parallel sk_{A1} \parallel A)$$

Understanding HMQV-**A** based on 2-key KEM

$$U_A \quad A = g^a$$

$$U_B$$

$$X = g^x$$

$$d = h(X, B)$$

$$K_A = (YB^e)^{x+ad}$$

X



YB^e



$$Y = g^y, C_A = YB^e$$

$$e = h(Y, A)$$

$$K_B = (XA^d)^{y+be}$$

Understanding HMQV-**B** based on 2-key KEM

U_A

$U_B \quad B = g^b$

$$X = g^x, C_B = XA^d$$

$$d = h(X, B)$$

$$K_A = (YB^e)^{x+ad}$$

XA^d

$$Y = g^y$$

$$e = h(Y, A)$$

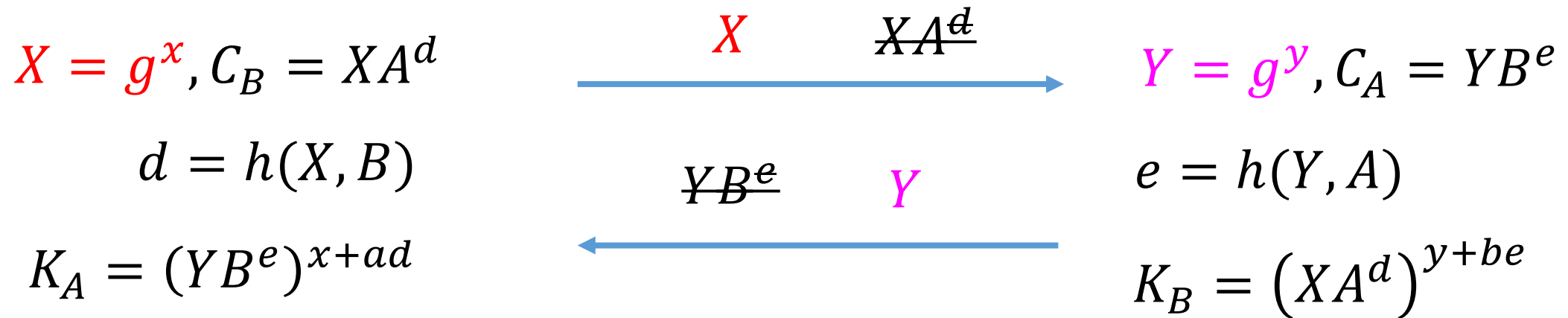
$$K_B = (XA^d)^{y+be}$$

Y

Understanding HMQV based on 2-key KEM

$$U_A \quad A = g^a$$

$$U_B \quad B = g^b$$



$$K = \text{Hash}(A, B, X, Y, K_A, K_B)$$

HMQRV-2KEM

- $(a, A) \leftarrow KGen1$
- $(x, X) \leftarrow KGen0$
- $\left(K = (XA^d)^{(y+be)}, C = YB^e \right) \leftarrow Enc(A, X; y, b; B)$
where $e = h(X, B), d = h(Y, A)$

Understanding AKE

- Every well-known implicit AKE implies a 2-key KEM
 - **HMQRV(&OAKE)**: 2-key KEM from gap-DH and KEA
 - **LLM07**: (aka. NAXOS) 2-key KEM from gap-DH
 - **Okamoto 07**: 2-key KEM from DDH (modified Cramer-Shoup)
 - **FSXY12**, improved KEM combiner in std. model
 - **FSXY13**, improved KEM combiner in RO model



Generic constructions of 2-key KEM

- CCA secure $(C_1, K_1) = \text{Enc}(\textcolor{red}{pk}_1)$, and $(C_0, K_0) = \text{Enc}(\textcolor{red}{pk}_0)$

$$C = C_1 | C_0, K = f(K_1, K_0, C)$$

- GHP18, CCA secure when f is a hash (in RO) or PRF function (in std.).

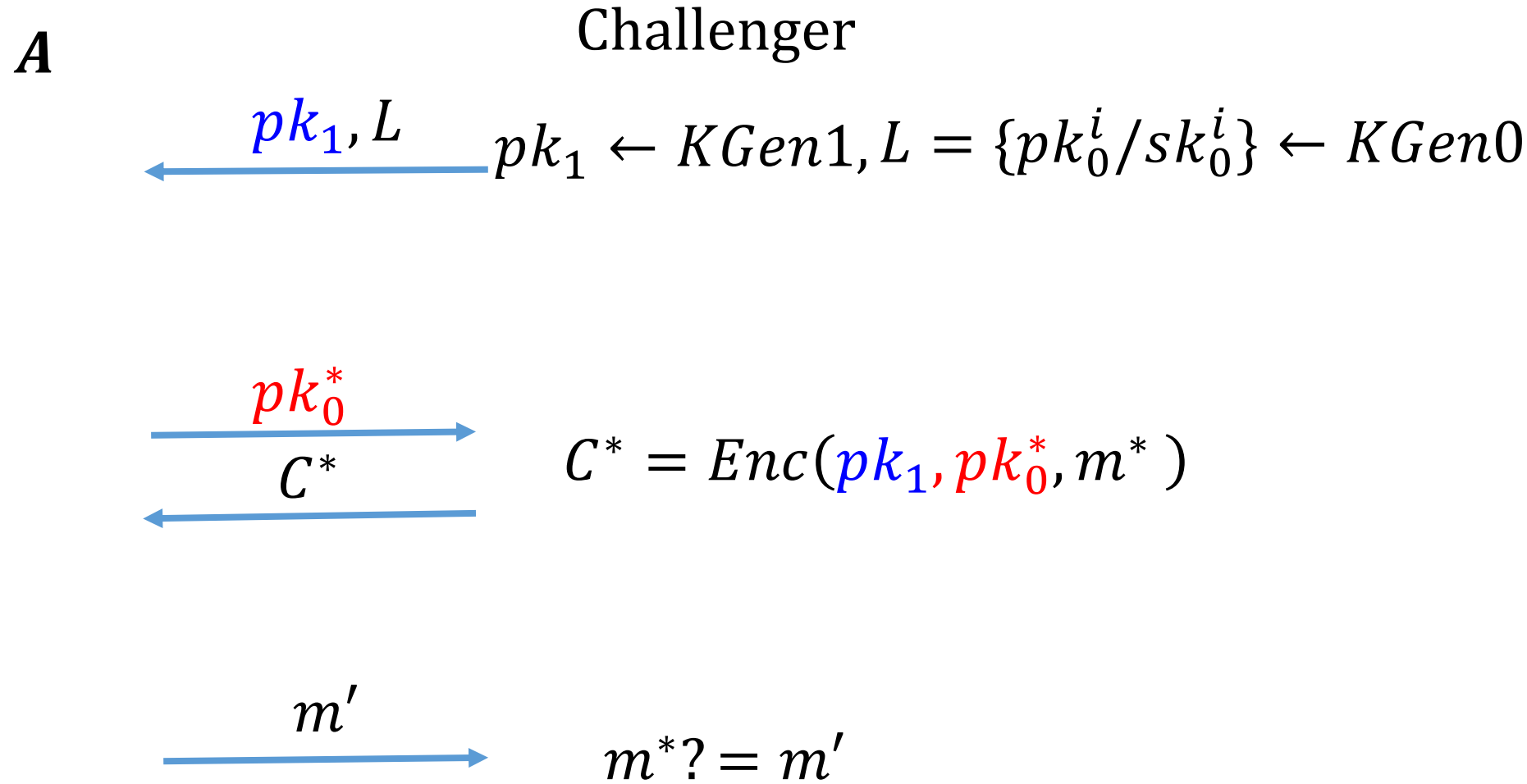
- $\textcolor{red}{[CCA, .]secure}$

- $C = C_1 | C_0, K = f(\textcolor{red}{pk}_0, K_1, K_0, C)$

More Generic Constructions of 2-key KEM

- Fujioka-Okamoto ?

$[CPA, \cdot]$ Security of 2-key PKE



More Generic Constructions of 2-key KEM

- Fujioka-Okamoto ?

- For CPA secure 2PKE
- $(pk_1, sk_1) \leftarrow Gen1,$
- $C = Encrypt(pk_1, m, r);$
- $m = Decrypt(sk_1, C);$

- Is FO secure?
- $(pk_1, sk_1) \leftarrow Gen1,$
- $C = Encrypt(pk_1, m, r);$
 $r = G(m)$
- $m = Decrypt(sk_1, C)$
If $C = Encrypt(pk_0, m, G(m))$
 $K = H(m)$ else \perp

More Generic Constructions of 2-key KEM

- Classical Fujioaka-Okamoto transformation does not work for $[CCA, \cdot]$ security
- Improved FO transformation by putting public key in hashing step to generate K

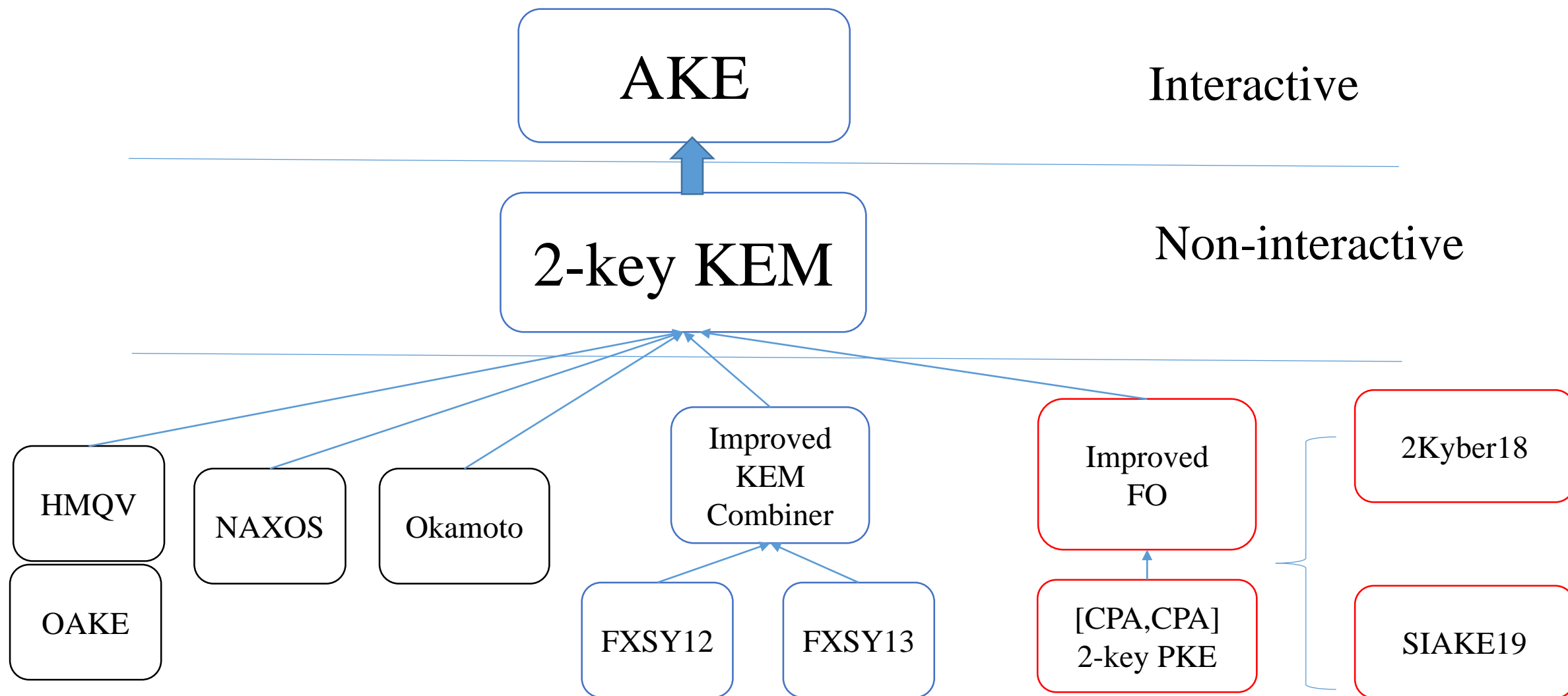
More Generic Constructions of 2-key KEM

- Improved FO transformation by putting public key in hashing step to generate K

- For $[CPA, CPA]$ secure 2PKE
- $(pk_1, sk_1) \leftarrow Gen1,$
- $(pk_0, sk_0) \leftarrow Gen0;$
- $C = \text{Encrypt}(pk_1, pk_0, m, r);$
- $m = \text{Decrypt}(sk_1, sk_0, C);$

- Is FO $[CCA, CCA]$ secure?
- $(pk_1, sk_1) \leftarrow Gen1,$
- $(pk_0, sk_0) \leftarrow Gen0;$
- $C = \text{Encrypt}(pk_1, pk_0, m, r);$
 $r = G(m)$
- $m = \text{Decrypt}(sk_1, sk_0, C)$
If $C = \text{Encrypt}(pk_1, pk_0, m, G(m))$
 $K = H(pk_1, pk_0, m, C)$

Roadmap



2-key PKE

$$g^{r_1}, h_1^{r_1} \oplus m_1 \parallel g^{r_2}, h_2^{r_2} \oplus m_2$$

$$g^{r_1}, g^{r_2}, h_1^{r_1} \oplus h_2^{r_2} \oplus m_2$$

2Kyber18

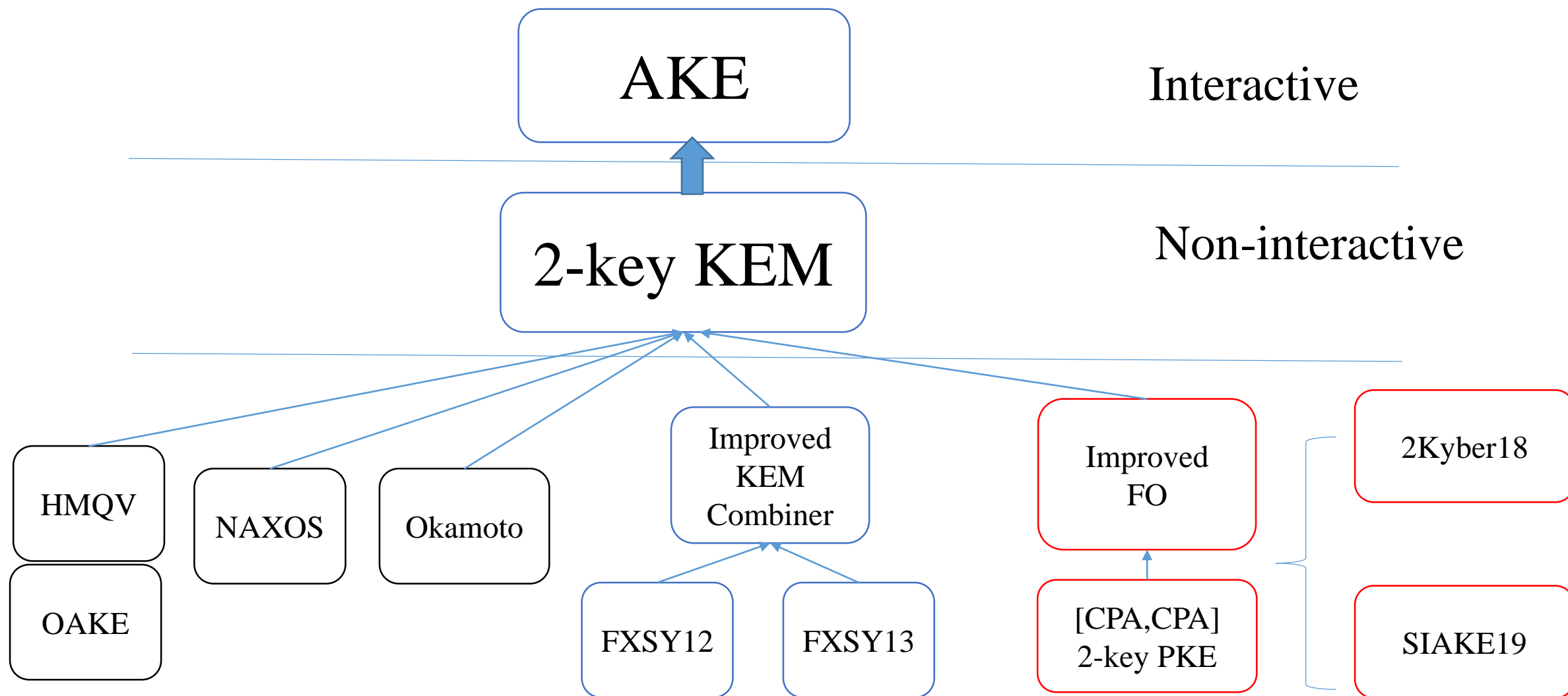
$$g^r, H(h_1^r) \oplus m_1, H(h_1^r) \oplus m_1$$

SIAKE19

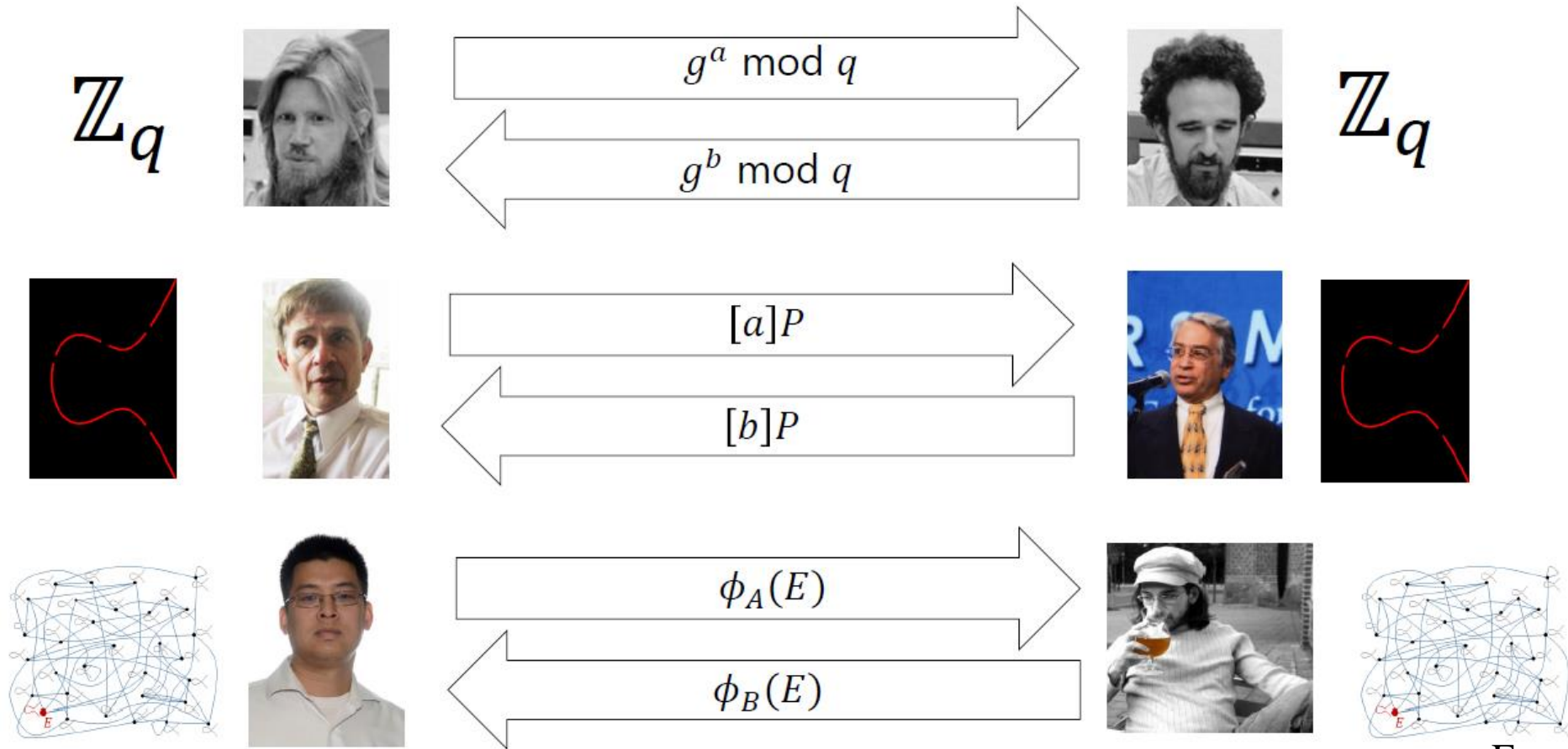
Outline

- Authenticated key exchange
- Motivations & our contributions
- $\text{AKE} \leftarrow \text{2-key KEM} \leftarrow$
- Post-quantum AKE

Roadmap



SIDH Key Exchange



From Croatia's slides

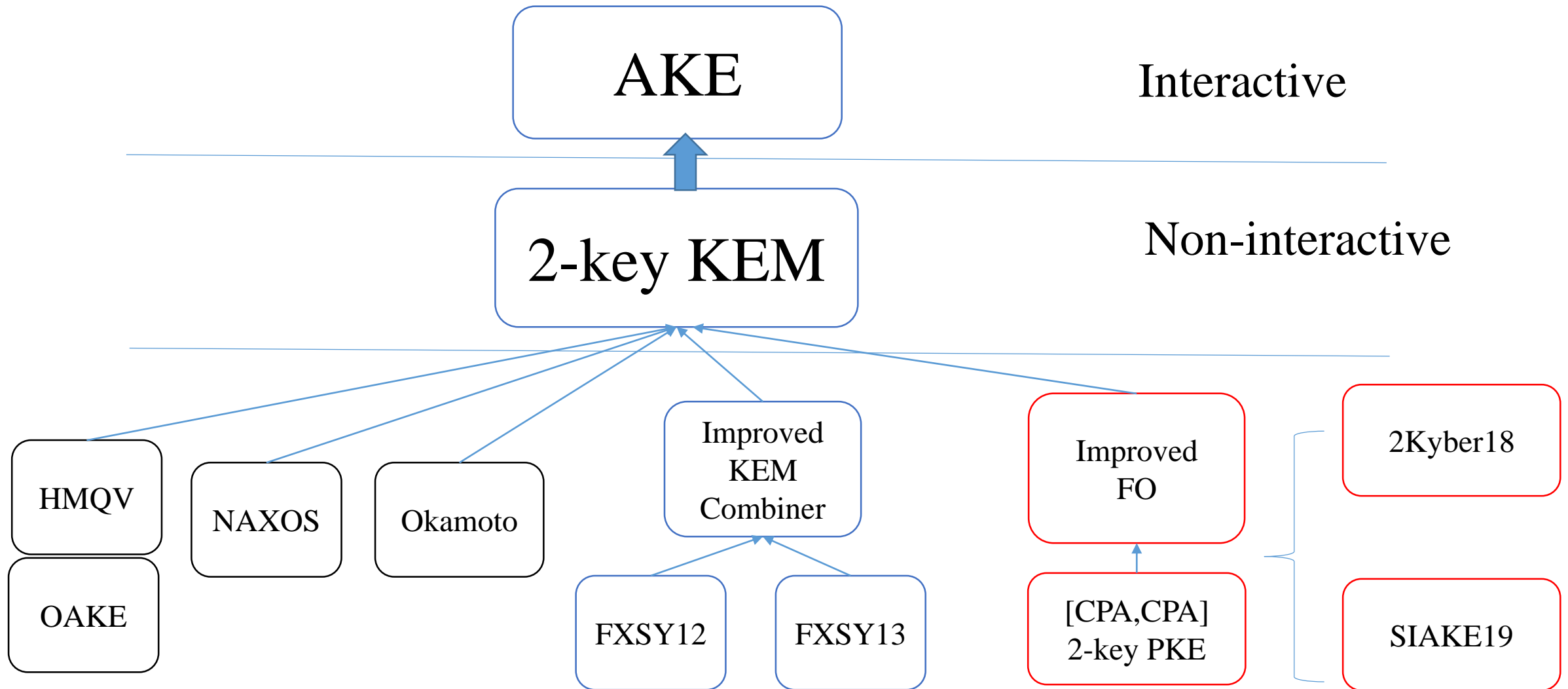
2-key PKE from SIDH

$$g^r, H \circ j(\mathcal{h}_1^r) \oplus m_1, H \circ j(\mathcal{h}_0^r) \oplus m_0$$

18	1	1
----	---	---

国内算法竞赛，SIAKE，第一轮密钥交换类第一

Roadmap



Conclusion

- [CCA, CPA] secure 2-key KEM and its (generic) constructions
- Understand *HMQV*, *NAXOS*, *Okamoto*, *FSXY12-3* etc. via 2-key KEM
- New Constructions based on lattice and SIDH

Haiyang Xue, Xianhui Lu, Bao Li, Bei Liang, Jingnan He, [Understanding and Constructing AKE via Double-key Key Encapsulation Mechanism](#), ASIACRYPT 2018

Xiu Xu, Haiyang Xue, Kunpeng Wang, Man Ho Au, Song Tian, [Strongly Secure Authenticated Key Exchange from Supersingular Isogenies](#), ASIACRYPT 2019

Future Work

- The security in the Quantum Random Oracle Model???