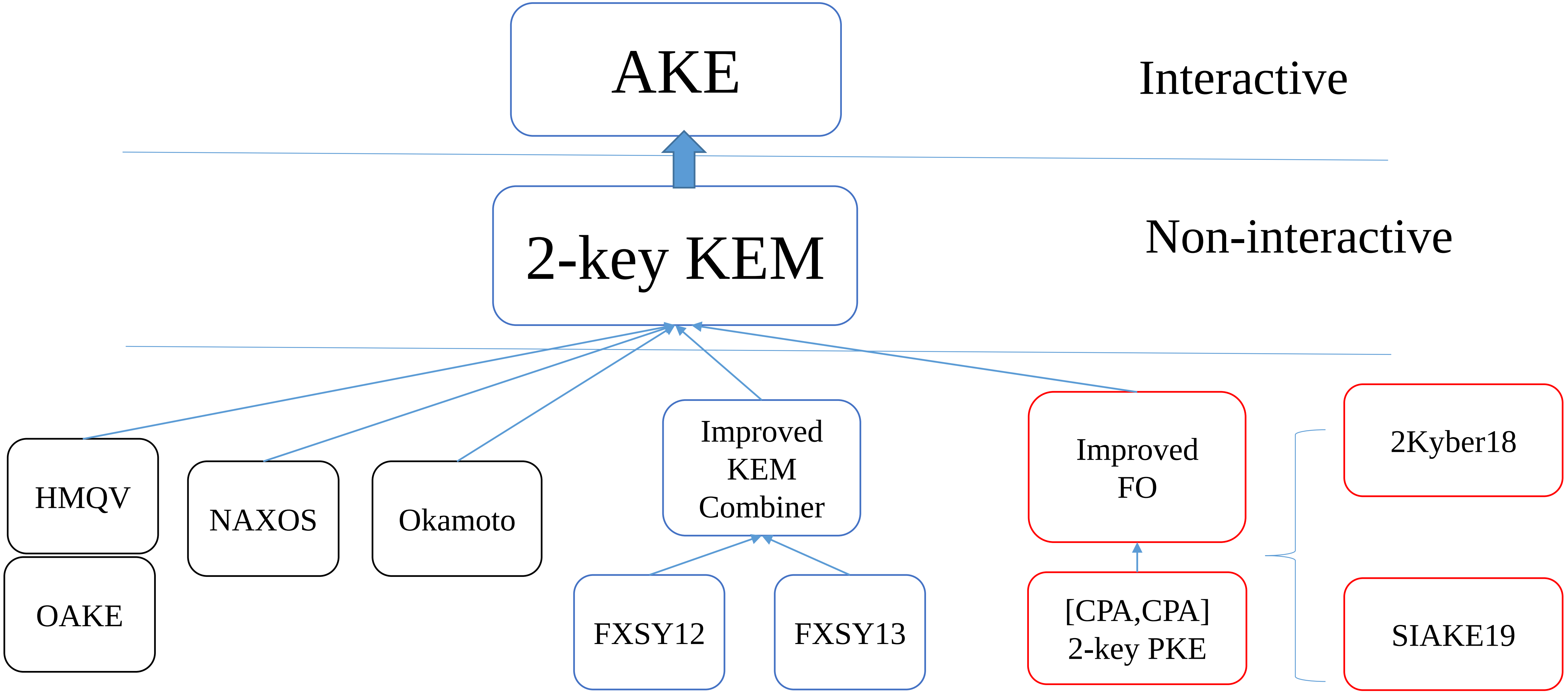
On the Constructions of Implicitly Authenticated Key Exchange

Haiyang Xue

2019.10.12

Roadmap


AKE
Interactive
2-key KEM
Non-interactive
HMQV
OAKE
NAXOS
Okamoto
Improved KEM Combiner
FXSY12
FXSY13
Improved FO
[CPA,CPA] 2-key PKE
2Kyber18
SIAKE19

Outline

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

Diffie-Hellman Key Exchange [DH76]

$\boldsymbol{U_A}$ $\qquad\qquad\qquad\qquad$ $\boldsymbol{U}_{\mathrm{B}}$

$x \rightarrow g^x = X$ $\qquad\qquad\qquad\qquad$ $y \rightarrow g^y = Y$

$$\xrightarrow{\mathrm{X}}$$

$$\xleftarrow{\mathrm{Y}}$$

$K = Y^x$ $\qquad\qquad\qquad\qquad$ $K = X^y$

- 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

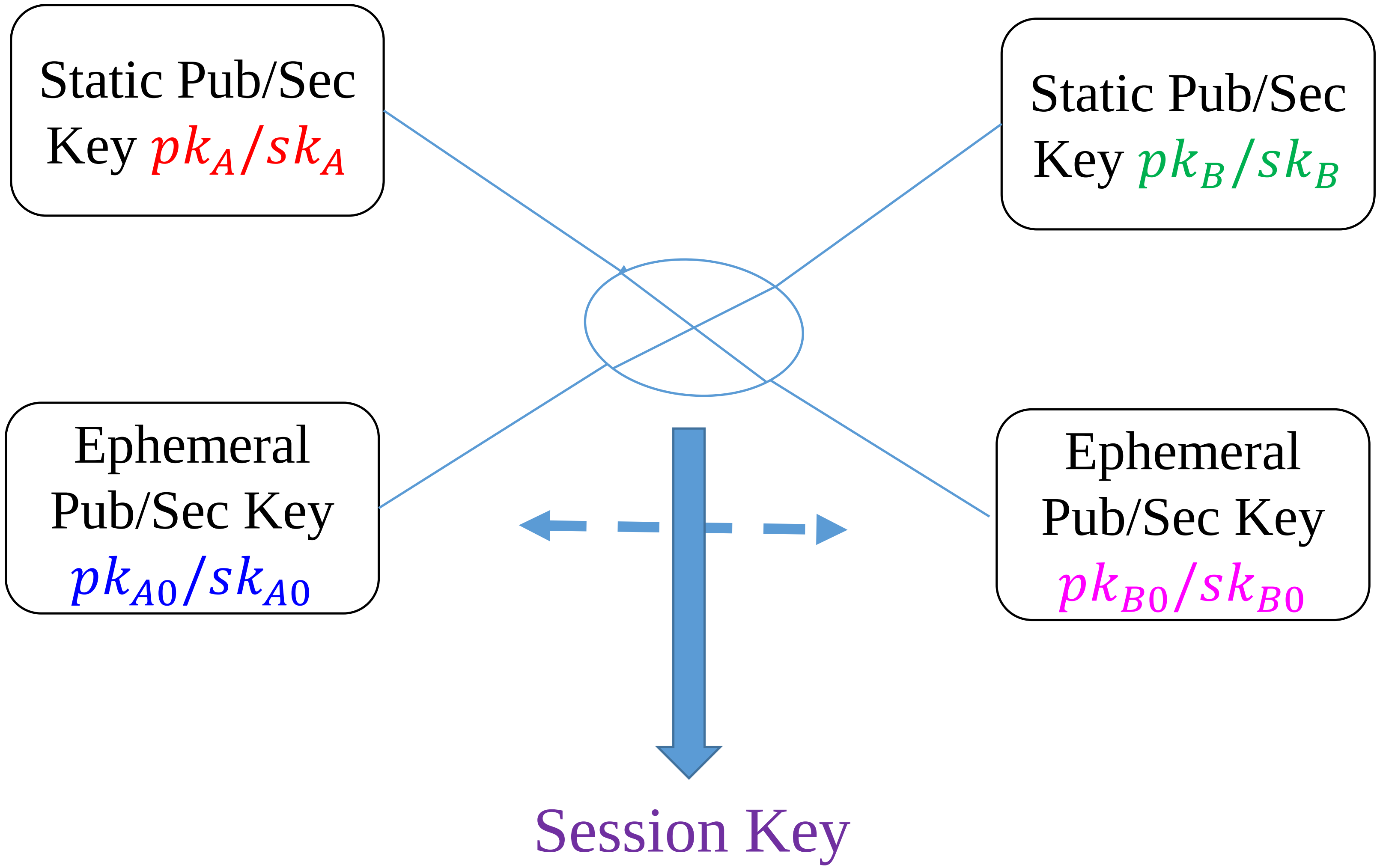
   BR model, CK model, HMQV-CK, eCK model, CK+ model

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

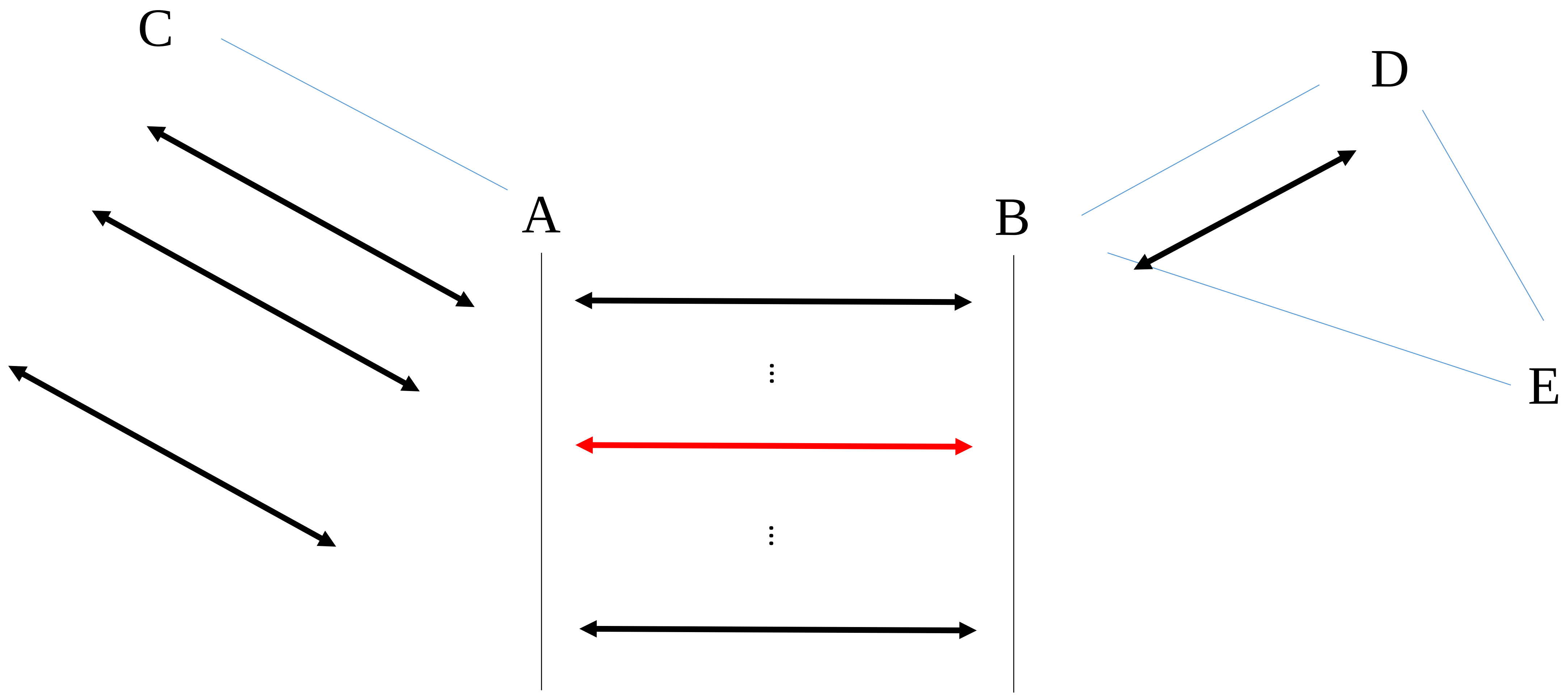
- Bellare-Rogaway 93 (BR93)

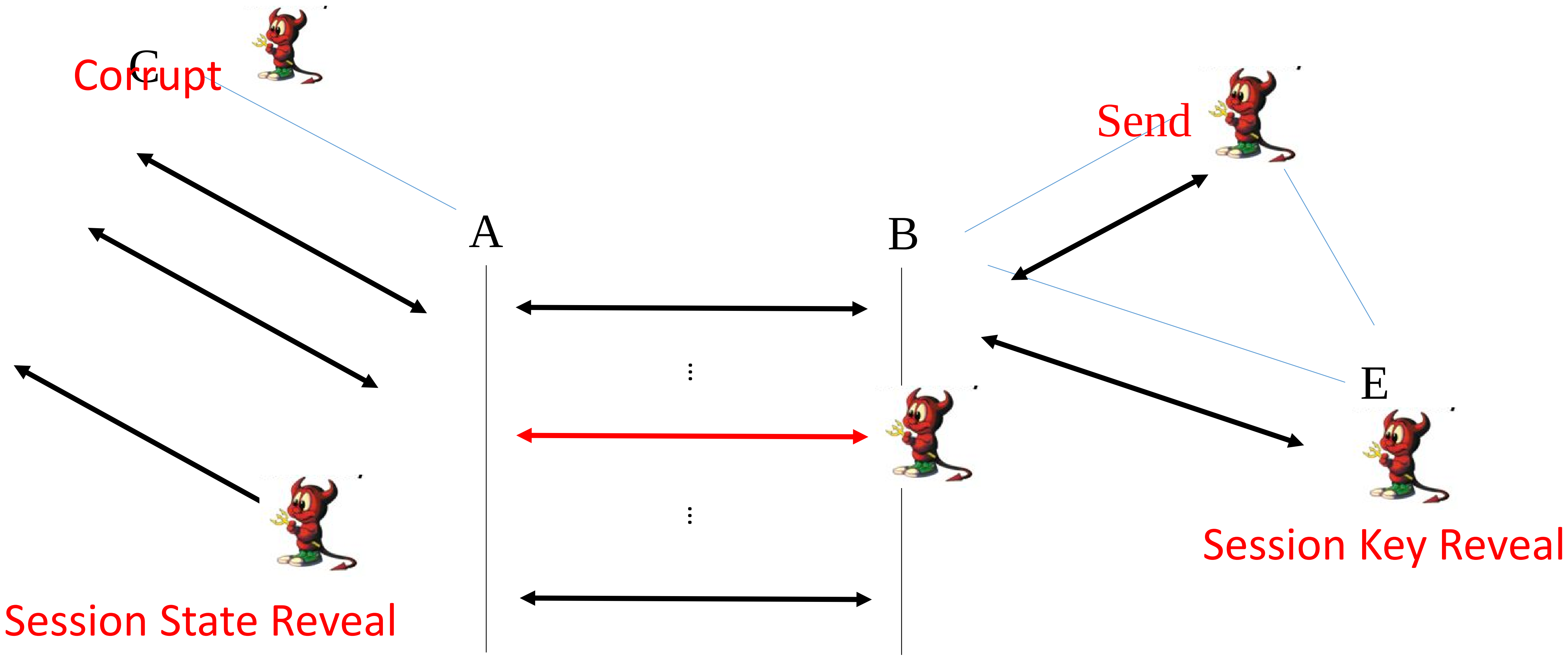
  indistinguishable type definition
- Canetti-Krawczyk 01(CK01)

  stronger security (session key, session state)
- LaMacchia-Lauter-Mityagin 07 (eCK)

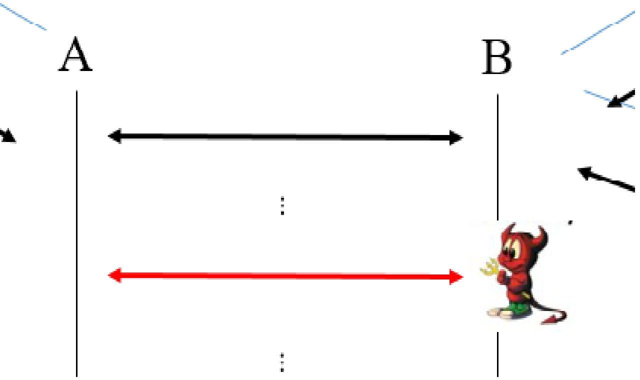
  stronger (session key, ephemeral randomness,wPFS+KCI+MEX)
- Fujioka-Suzuki-Xagawa-Yoneyama 12 (CK+)

  reform the security of HMQV: CK01+wPFS+KCI+MEX


C
D
A
B
E



C
Corrupt
Send
A
B
E
Session Key Reveal
Session State Reveal



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

√ it may be leaked to adversary; × it is secure; - means it does not exists

Outline

- Authenticated key exchange
- Motivations & our contributions
- AKE ← 2-key KEM ←
- 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. **HMQV 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** HMQV-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
- AKE ← 2-key KEM ←
- Post-quantum AKE

Key Encapsulation Mechanism(KEM)

$$
\begin{tikzcd}
r \arrow[r] & pk \arrow[r, "\textcolor{red}{Enc}"] & (C,K) \arrow[dl, "\textcolor{blue}{Dec}"] \\
K' = K & sk \arrow[l] \arrow[u, dashed] & \\
& KGen \arrow[u, dashed] &
\end{tikzcd}
$$

Key Exchange (transport) and KEM

$\boldsymbol{U}_{\mathrm{A}}$ $\qquad\qquad\qquad\qquad$ $\boldsymbol{U}_{\mathrm{B}}$

$$\xrightarrow{\quad pk \quad}$$

$$\xleftarrow{\quad C \quad} \qquad (C, K) = Enc(pk, r)$$

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

Our 2-key KEM

$$
\begin{tikzcd}
 & \color{blue}{KGen1} & \color{red}{KGen0} & \\
r \arrow[r] & pk_1 \arrow[r] & pk_0 \arrow[r, "Enc"] & (C,K) \arrow[dl, "Dec"] \\
K' = K & sk_1 \arrow[l] \arrow[u, dashed] & sk_0 \arrow[l] \arrow[u, dashed] &
\end{tikzcd}
$$

It is simple, not a big deal

One-side AKE from 2-key KEM?

$\boldsymbol{U}_{\mathrm{A}}$

pk_1

$\boldsymbol{U}_{\mathrm{B}}$

$$pk_0 \longrightarrow$$

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

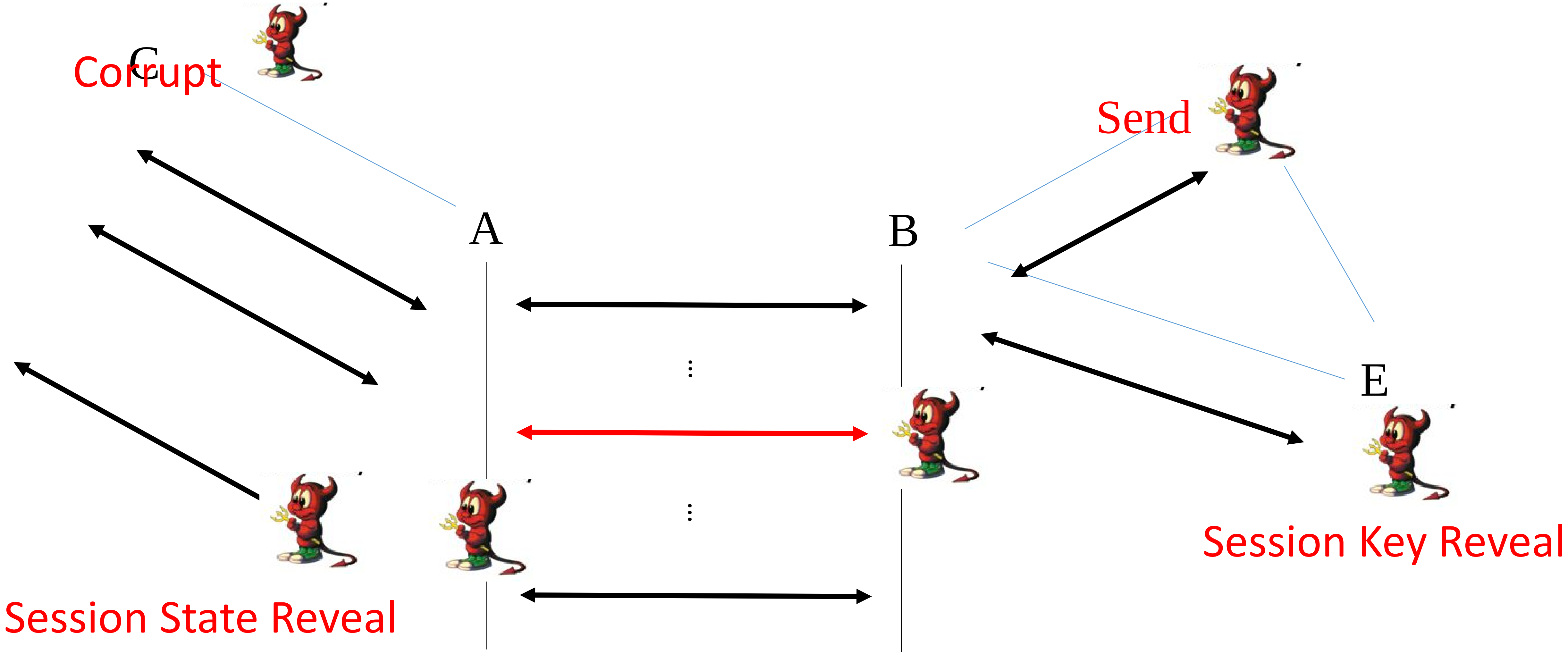
$$C \longleftarrow$$

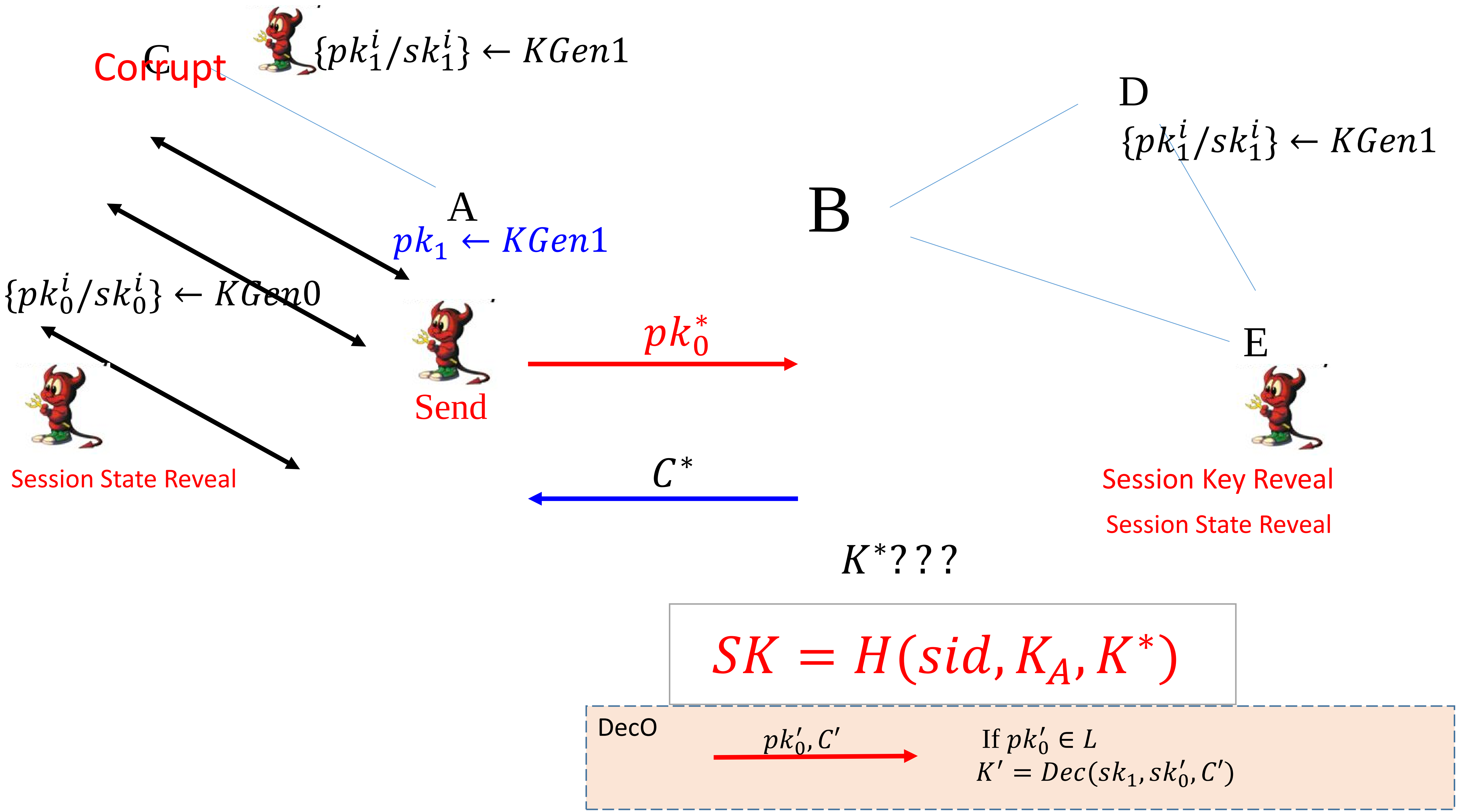
$$Dec(sk_1, sk_0, C) = K$$

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

CK+ security

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


C
Corrupt
Send
A
B
E
⋮
⋮
Session Key Reveal
Session State Reveal



C
Corrupt
$\{pk_1^i/sk_1^i\} \leftarrow KGen1$
D
$\{pk_1^i/sk_1^i\} \leftarrow KGen1$
A
$pk_1 \leftarrow KGen1$
B
$\{pk_0^i/sk_0^i\} \leftarrow KGen0$
pk_0^*
Send
E
C^*
Session State Reveal
Session Key Reveal
Session State Reveal
$K^*???$
$SK = H(sid, K_A, K^*)$
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^*

pk_0^i

C'

pk_0^i, sk_0^i

C^*

$K^*???$

$$\boxed{SK = H(sid, K_A, K^*)}$$

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

$[CCA,\cdot]$

C

A

$pk_1 \leftarrow KGen1$

Send

$\mathcal{B}$

pk_0^*

pk_0^i

pk_0^i, sk_0^i

C'

C^*

$K^*???$

$K'??????$

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

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

pk_1, L

$pk_1 \leftarrow KGen1,$

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

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

pk_0^*

C^*

K^*

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

Case 6-10

$[CCA,\cdot]$

C

A

$\mathcal{B}$

$pk_1 \leftarrow KGen1$

pk_0^i

pk_0^i, sk_0^i $\quad C'$

pk_0^*

C^*

$K^*???$

$K'??????$

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

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

$pk_0^* \quad pk_1, L$

$pk_1 \leftarrow KGen1,$

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

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

C^*

K^*

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

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

A — Challenger

Session State Reveal
Session Key Reveal

pk_1, L ←

$pk_1 \leftarrow KGen1, L = \{pk_0^i / sk_0^i\} \leftarrow KGen0$

DecO

pk_0', C' →

If $pk_0' \in L$

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

Send

pk_0^* →

$(C^*, K^*) = Enc(pk_1, pk_0^*, r)$

C^* ←

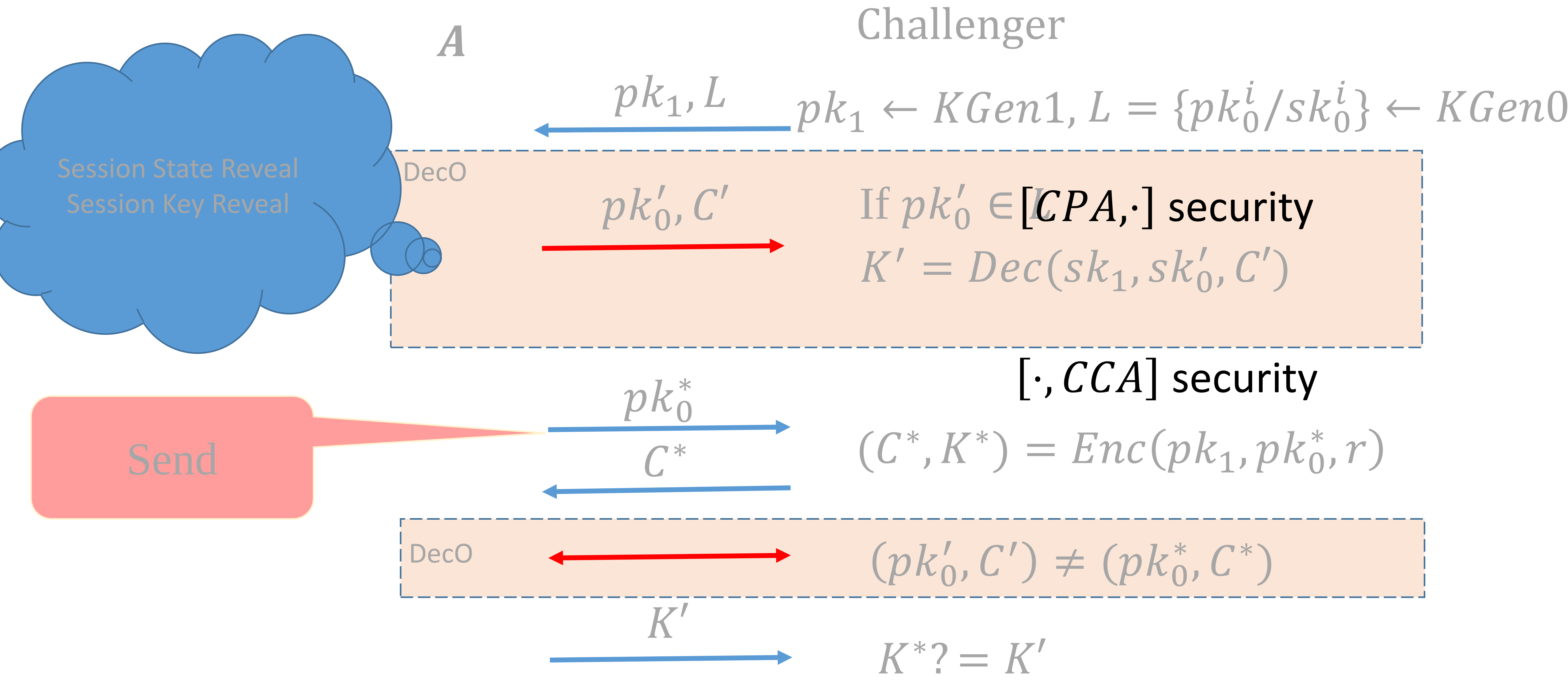
DecO ↔

$(pk_0', C') \neq (pk_0^*, C^*)$

K' →

$K^* ?= K'$

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



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

$\boldsymbol{U}_{\mathrm{A}}$ pk_{A1}

$\boldsymbol{U}_{\mathrm{B}}$

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

$\boldsymbol{U}_{\mathrm{A}}$ $\boldsymbol{U}_{\mathrm{B}}$ pk_{B1}

pk_{B0}

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

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

$\boldsymbol{U}_{\mathrm{A}}$ pk_{A1} $\qquad\qquad$ $\boldsymbol{U}_{\mathrm{B}}$ pk_{B1}

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

$$K_A = Dec(sk_{A1}, sk_{A0}, C_A) \xleftarrow{\quad C_A \qquad pk_{B0} \quad} (C_A, K_A) = Enc(pk_{A1}, pk_{A0})$$

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

Several AKE frames with Tricks

$\boldsymbol{U}_{\mathrm{A}}$ pk_{A1}

$\boldsymbol{U}_{\mathrm{B}}$ pk_{B1}

pk_{A0} C_B

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

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

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

Trick 1

$K_A = Dec(sk_{A1}, sk$ … $) = Enc(pk_{A1}, pk_{A0})$

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

Several AKE frames with Tricks

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

U_A pk_{A1}

$(C_B, -) = Enc1(pk_{B1}, -)$ Trick 2 $\xrightarrow{pk_{A0} \quad C_B}$ $K_B = Dec(sk_{B1}, sk_{B0}, C_A)$

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

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

Several AKE frames with Tricks

$\boldsymbol{U}_{\mathrm{A}} \quad pk_{A1}$

$\boldsymbol{U}_{\mathrm{B}} \quad pk_{B1}$

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

$K_A = Dec(sk_{A1}, sk_{A0}, C_A) \xleftarrow{\cancel{C_A} \quad pk_{B0}} (C_A, K_A) = Enc(pk_{A1}, pk_{A0})$

Trick 3

C_B can be publicly computed from pk_{A0}

C_A can be publicly computed from pk_{B0}

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

Understanding HMQV-A based on 2-key KEM

$\boldsymbol{U}_{\mathrm{A}}$ $A = g^a$

$\boldsymbol{U}_{\mathrm{B}}$

$X = g^x$

$\xrightarrow{X}$

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

$d = h(X, B)$

$e = h(Y, A)$

$\xleftarrow{YB^e}$

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

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

Understanding HMQV-B based on 2-key KEM

$\boldsymbol{U}_{\mathrm{A}}$

$\boldsymbol{U}_{\mathrm{B}}$ $B = g^b$

$X = g^x, C_B = XA^d$ $\xrightarrow{XA^d}$ $Y = g^y$

$d = h(X, B)$ $e = h(Y, A)$

$\xleftarrow{Y}$

$K_A = (YB^e)^{x+ad}$ $K_B = \left(XA^d\right)^{y+be}$

Understanding HMQV based on 2-key KEM

$\boldsymbol{U}_{\mathrm{A}}$ $A = g^a$

$\boldsymbol{U}_{\mathrm{B}}$ $B = g^b$

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

$d = h(X, B)$

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

X $\quad \cancel{XA^d}$ $\longrightarrow$

$\longleftarrow$ $\cancel{YB^e}$ $\quad Y$

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

$e = h(Y, A)$

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

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

HMQV-2KEM

- $(a, A) \leftarrow KGen1$
- $(x, X) \leftarrow KGen0$

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

Understanding AKE

- Every well-known implicit AKE implies a 2-key KEM
  - HMQV(&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 $(C_1, K_1) = Enc(pk_1), and\ (C_0, K_0) = Enc(pk_0)$

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

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

- [CCA, .]secure
  - $C = C_1 | C_0, K = f(pk_0, K_1, K_0, C)$

More Generic Constructions of 2-key KEM

- Fujioka-Okamoto ?

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

$\boldsymbol{A}$ Challenger

$\xleftarrow{pk_1, L}$ $pk_1 \leftarrow KGen1, L = \{pk_0^i / sk_0^i\} \leftarrow KGen0$

$\xrightarrow{pk_0^*}$

$\xleftarrow{C^*}$ $C^* = Enc(pk_1, pk_0^*, m^*)$

$\xrightarrow{m'}$ $m^* ?= m'$

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 $\perp$

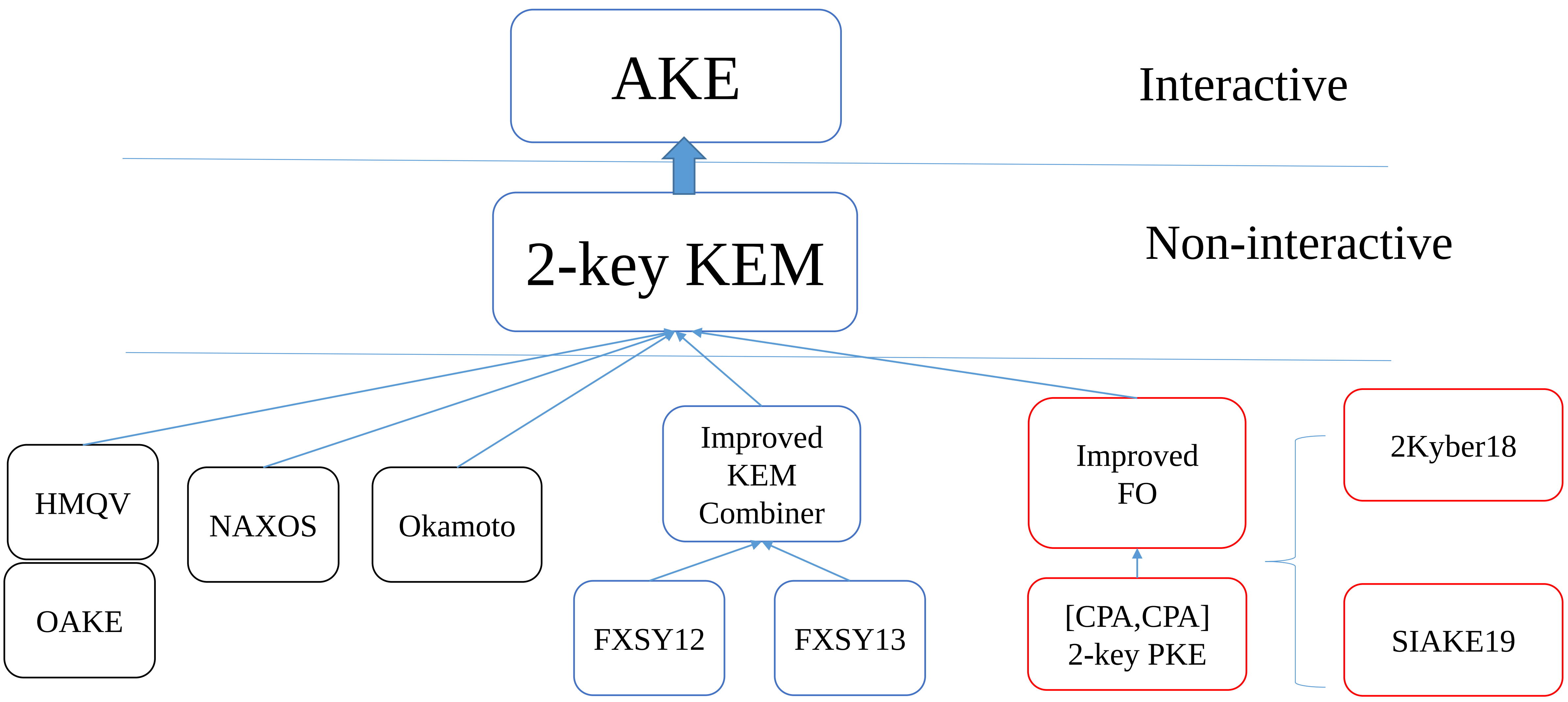
More Generic Constructions of 2-key KEM

- Classical Fujioka-Okamoto transformation does not work for $[CCA,\cdot]$ seurity
- 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

Roadmap


AKE
Interactive
2-key KEM
Non-interactive
HMQV
OAKE
NAXOS
Okamoto
Improved KEM Combiner
FXSY12
FXSY13
Improved FO
[CPA,CPA] 2-key PKE
2Kyber18
SIAKE19

2-key PKE

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

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

2Kyber18

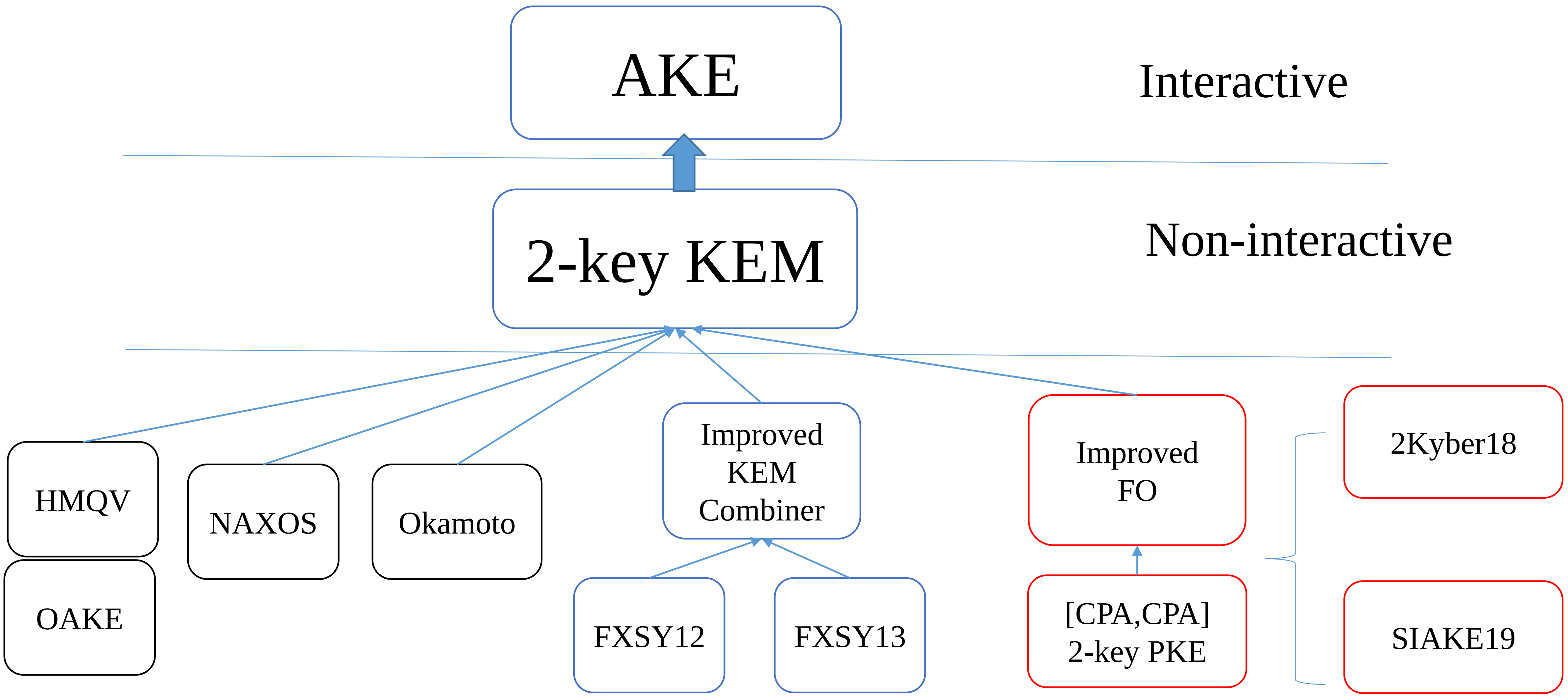
$$g^r, H(h_1^r) \oplus m_1, H(h_1^r) \oplus m_1$$

SIAKE19

Outline

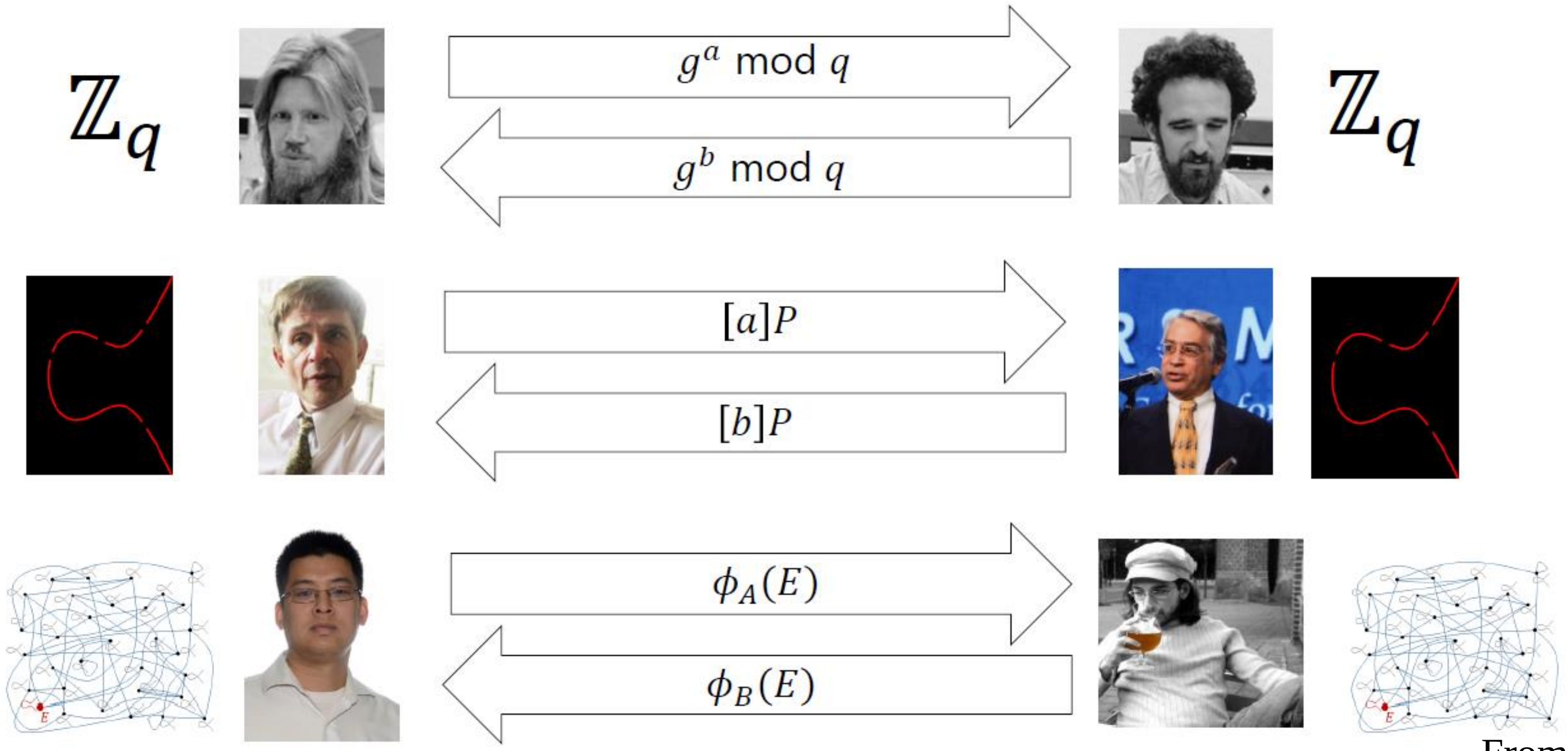
- Authenticated key exchange
- Motivations & our contributions
- AKE $\leftarrow$ 2-key KEM $\leftarrow$
- Post-quantum AKE

Roadmap



AKE
Interactive
2-key KEM
Non-interactive
HMQV
OAKE
NAXOS
Okamoto
Improved KEM Combiner
FXSY12
FXSY13
Improved FO
[CPA,CPA] 2-key PKE
2Kyber18
SIAKE19

SIDH Key Exchange



From Croatia’s slides

[JAC+18] Jao, D., Azarderakhsh, R., Campagna, M., et al: Supersingular Isogeny Key Encapsulation. NIST Round 2.

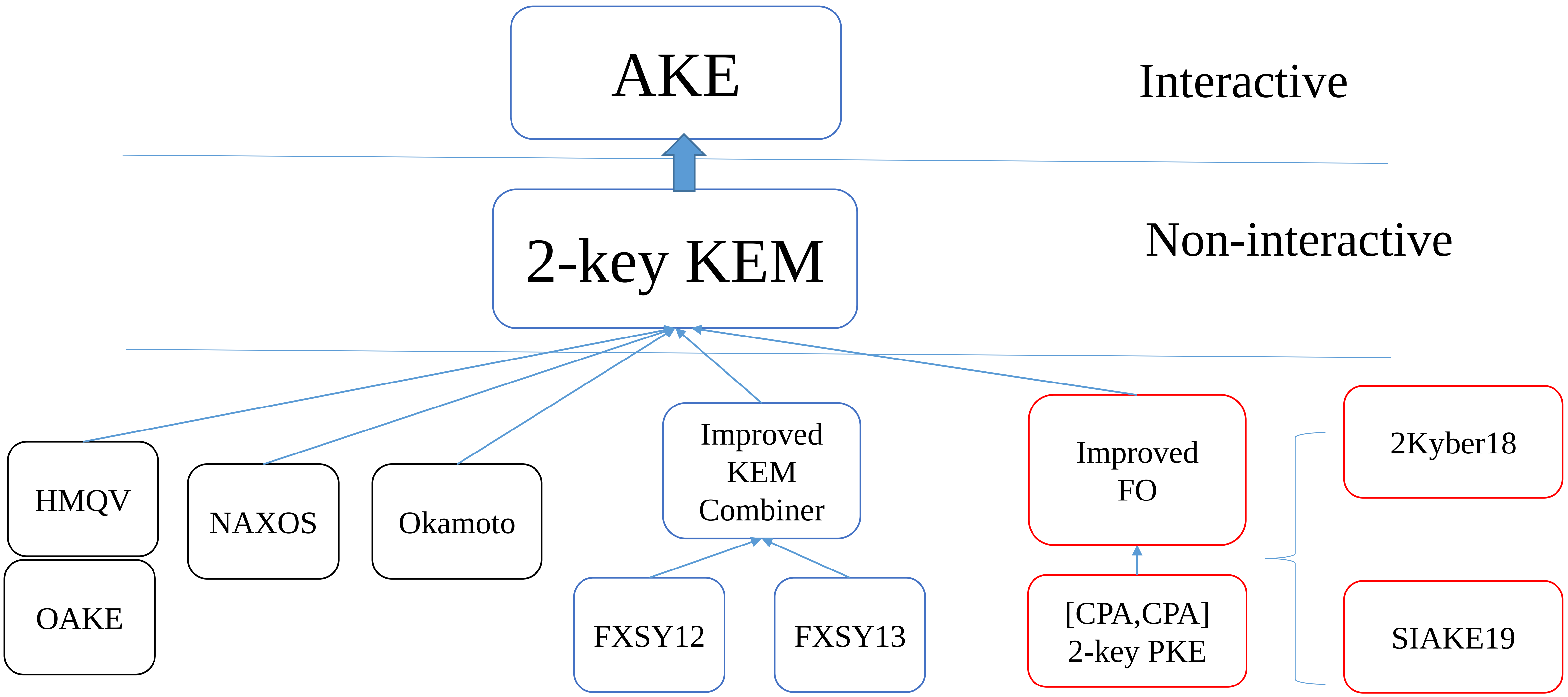
2-key PKE from SIDH

$$g^r, H \circ \mathrm{j}(h_1^r) \oplus m_1, H \circ \mathrm{j}(h_0^r) \oplus m_0$$

18	1	1

国内算法竞赛，SIAKE

Roadmap



AKE
Interactive
2-key KEM
Non-interactive
HMQV
OAKE
NAXOS
Okamoto
Improved KEM Combiner
FXSY12
FXSY13
Improved FO
[CPA,CPA] 2-key PKE
2Kyber18
SIAKE19

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???

FO Transformation

- 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 $C = Encrypt(pk_0, m, G(m))$

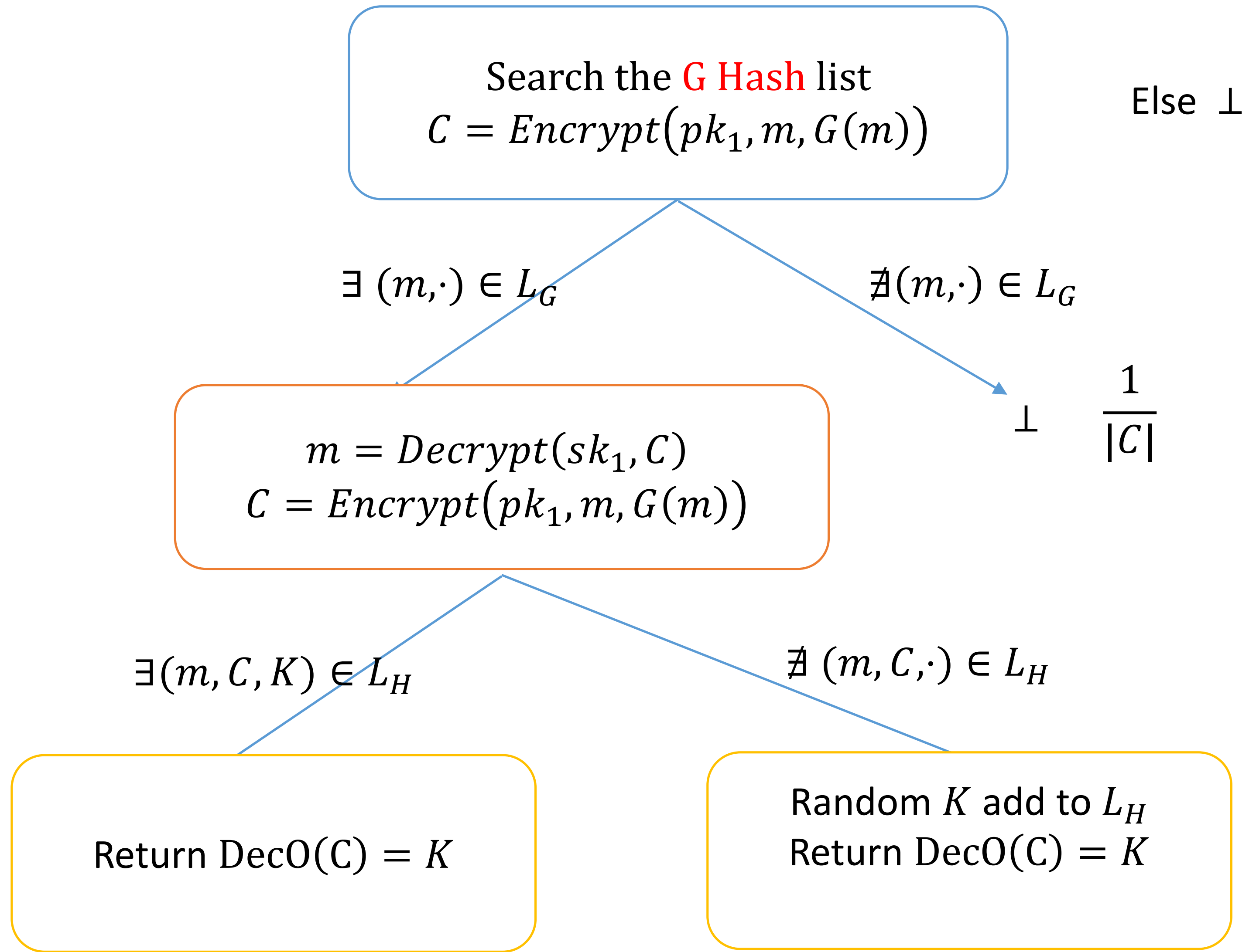
  $K = H(m)$ else $\perp$

Without sk_1 or sk_0, how to answer DecO?

$$DecO(C)$$

1. $m = Decrypt(sk_1, C)$
2. If $C \neq Encrypt(pk_1, m, G(m)), \perp$
3. else $K = H(m)$

$C^*; K^* = \mathrm{H}(\mathrm{m}^*)$



If there is decryption failure?

$C = Encrypt(pk_1, pk_0, m, G(m))$
But $m \neq Decrypt(sk_1, sk_0, C)$

$+q_G\ \epsilon$

Search the G Hash list
$C = Encrypt(pk_1, pk_0, m, G(m))$

Else $\perp$

$\exists\ (m,\cdot) \in L_G$

$\nexists (m,\cdot) \in L_G$

$\perp$ $\quad \frac{1}{|C|}$

$m = Decrypt(sk_1, sk_0, C)$
$C = Encrypt(pk_1, pk_0, m, G(m))$

$\exists (m, C, K) \in L_H$

$\nexists\ (m, C,\cdot) \in L_H$

Return $\mathrm{DecO(C)} = K$

Random K add to L_H
Return $\mathrm{DecO(C)} = K$

$C^*; K^* = \mathrm{H(m^*)}$

Improved FO Transformation

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

- Is FO $[CCA, CCA]$ secure?
- $(pk_1, sk_1) \leftarrow Gen1,$
- $(pk_0, sk_0) \leftarrow Gen0;$
- $C = Encrypt(pk_1, pk_0, m, r);$

  $r=G(m)$
- $m = Decrypt(sk_1, sk_0, C)$

  If $C = Encrypt(pk_1, pk_0, m, G(m))$

  $K = H(pk_1, pk_0, m, C)$

How about $DecO(pk_0', C')$

$DecO(pk_0', C)$ and $DecO(pk_0'', C)$ will get the same answer.

$+q_G\ \epsilon$

Search the Hash list
$C = Encrypt(pk_1, pk_0, m, G(m))$

Else $\perp$

$\exists\ (m,\cdot) \in L_G$

$\nexists (m,\cdot) \in L_G$

$\perp$ $\quad \frac{1}{|C|}$

$m = Decrypt(sk_1, sk_0, C)$
$C = Encrypt(pk_1, pk_0, m, G(m))$

$\exists (m, C, K) \in L_H$

$\nexists\ (m, C, \cdot) \in L_H$

Return $\mathrm{DecO(C)} = K$

Random K add to L_H
Return $\mathrm{DecO(C)} = K$

How about $DecO(pk_0', C')$

Since $K = H(pk_0, m, C)$
If $pk_0' \notin L$ $\perp$ else

Search the Hash list
$C = Encrypt(pk_1, pk_0, m, G(m))$

Else $\perp$

$+q_G\ \epsilon$

$\exists\ (m,\cdot) \in L_G$

$\nexists (m,\cdot) \in L_G$

$\perp$ $\quad \frac{1}{|C|}$

$m = Decrypt(sk_1, sk_0, C)$
$C = Encrypt(pk_1, pk_0, m, G(m))$

$\exists (pk_0', m, C, K) \in L_H$

$\nexists\ (pk_0', m, C, \cdot) \in L_H$

Return DecO(C) $= K$

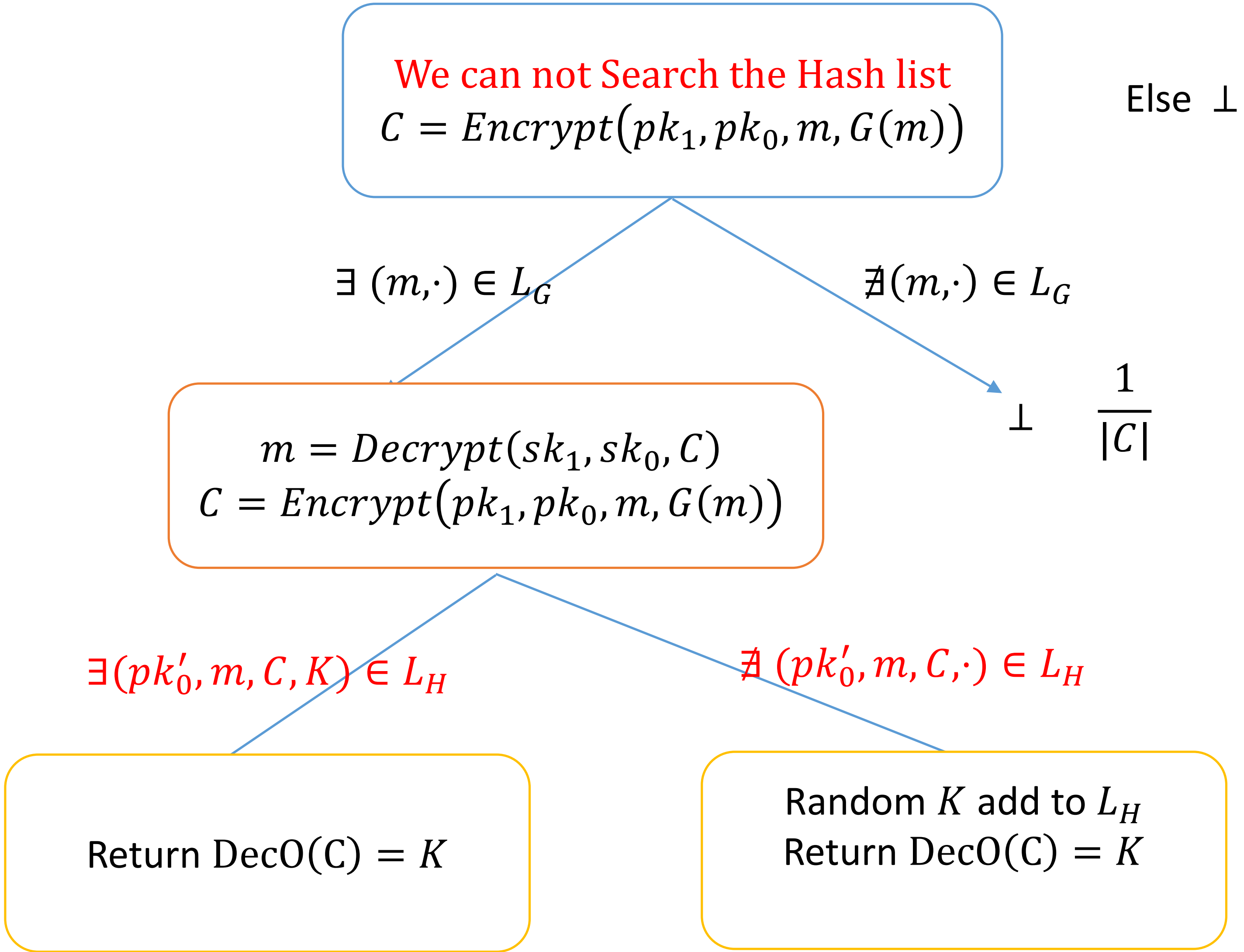
Random K add to L_H
Return DecO(C) $= K$

Challenges in Quantum RO

- We can not Search the Hash list

- Even m^* is in the hash H list the challenger may not detect

$$C^*; K^* = \mathrm{H}(m^*)$$



FO-QROM

Schemes	Inj. map.	PKE	Additional Hash	Security Bound	DecError
[TU16,HHK17]	-	IND/OW-CPA	len.-pre	$q\sqrt[4]{\varepsilon}$	✓
[SXY18]-1	✓	IND-CPA	×	$q\sqrt{\varepsilon}$	×
[SXY18]-2	✓	OW-CPA	no len.-pre,	$q\sqrt[4]{\varepsilon}$	×
[HKS+18]	✓	IND-CPA	×	$q\sqrt{\varepsilon}$	✓
[JZC+18]	✓	OW-CPA	×	$q\sqrt{\varepsilon}$	✓

Table 2. Comparison of existing PKE-to-KEM proofs in QROM. Inj. map. indicates the injective mapping with meet-in-the-end. len.-pre means the additional hash should be length preserving. q is the number of random oracle queries. ε is the the advantage of the reduced adversary against the OW/IND-CPA security of PKE.