Anamorphic Authenticated Key Exchange: Double Key Distribution under Surveillance

Weihao Wang, Shuai Han, Shengli Liu

Shanghai Jiao Tong University

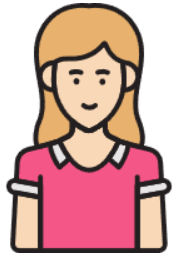
Asiacrypt 2024, Kolkata, India

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Recap: Anamorphic Cryptography

Target: To resist adversaries in **Coercive Environment**

For PKE



Alice
(pk_A, sk_A)

$c \leftarrow Enc(pk_B, m)$

m^*

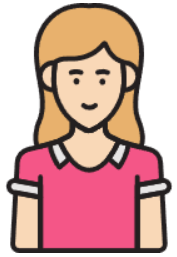
Bob
(pk_B, sk_B)

Anamorphic Encryption (AME)

Recap: Anamorphic Cryptography

Target: To resist adversaries in **Coercive Environment**

For PKE



Alice
(pk_A, sk_A)

$c \leftarrow Enc(pk_B, m)$

m^*

Bob
(pk_B, sk_B)

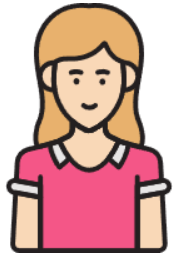
Anamorphic Encryption (AME)

Receiver-AME: Adversaries knowing the secret key of **the receiver**

Recap: Anamorphic Cryptography

Target: To resist adversaries in **Coercive Environment**

For PKE



Alice
(pk_A, sk_A)

$c \leftarrow Enc(pk_B, m)$

m^*

Bob
(pk_B, sk_B)

Anamorphic Encryption (AME)

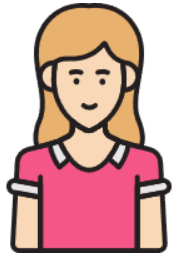
Receiver-AME: Adversaries knowing the secret key of **the receiver**

Sender-AME: Adversaries forcing **the sender** to send designated message

Recap: Anamorphic Cryptography

Target: To resist adversaries in **Coercive Environment**

For PKE



Alice
(pk_A, sk_A)

double key dk

$c \leftarrow Enc(pk_B, m)$

$c^* \leftarrow aEnc(pk_B, dk, m, m^*)$

$m^* \leftarrow aDec(dk, c^*)$

Bob
(pk_B, sk_B)

Seems like $c^* \approx c$

Anamorphic Encryption (AME)

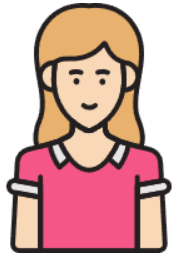
Receiver-AME: Adversaries knowing the secret key of **the receiver**

To establish advantage against adversaries, a **covert double key** dk is **pre-shared** in Receiver-AME setting

Recap: Anamorphic Cryptography

Target: To resist adversaries in **Coercive Environment**

For SIG



Alice

(pk_A, sk_A)

$\sigma \leftarrow Sign(sk_A, m)$

m^*

Bob

(pk_B, sk_B)

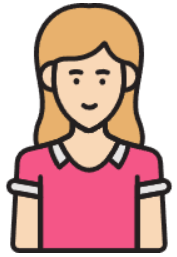
Anamorphic Signature

Adversaries knowing the signing key of the signer

Recap: Anamorphic Cryptography

Target: To resist adversaries in **Coercive Environment**

For SIG



Alice

(pk_A, sk_A)

double key dk

$\sigma \leftarrow Sign(sk_A, m)$

$\sigma^* \leftarrow aSign(sk_A, dk, m, m^*)$

$m^* \leftarrow aDec(dk, \sigma^*)$

Bob

(pk_B, sk_B)

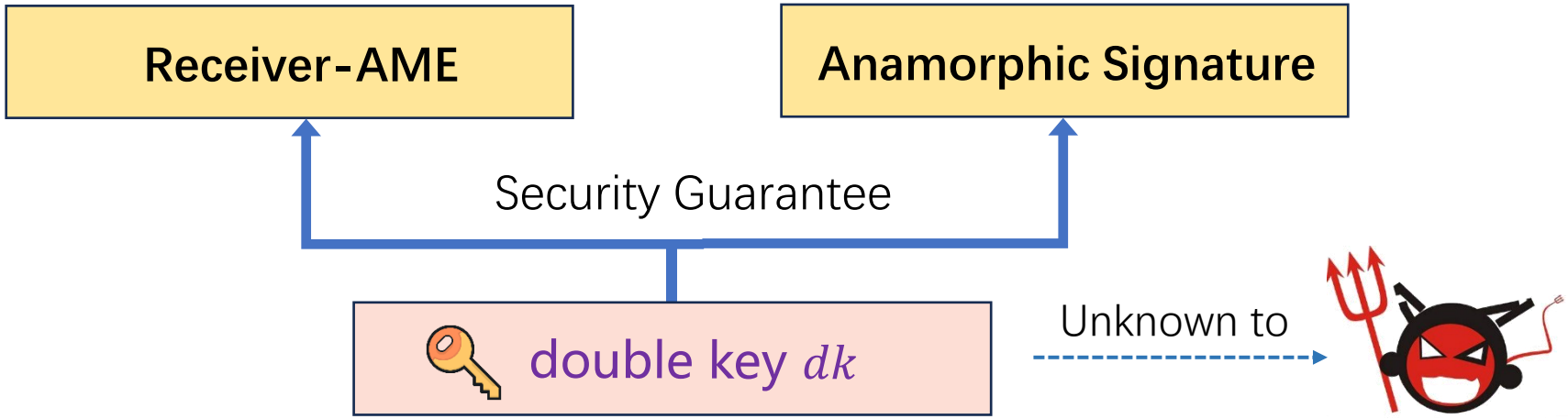
Seems like $\sigma^* \approx \sigma$

Anamorphic Signature

Adversaries knowing the signing key of the signer

Similarly, a **covert double key** dk is **pre-shared** in the setting of anamorphic signature

Double Key Supported Anamorphic Cryptography



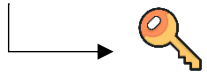
Core problem: the distribution of double key **under coercion**

Current Methods

Core problem: the distribution of double key **under coercion**

- [vH04] Steganography Key Exchange:
  Embed 1 bit of KE transcript into each ciphertext of PKE

- [HPRV19] Steganography Key Exchange with Secret Key Leakage:
  Embed a tiny piece of KE transcript into each ciphertext of PKE



Low Efficiency **Disordering Not Allowed** **Active Attack Not Allowed**

Current Methods

Core problem: the distribution of double key **under coercion**

- [vH04] Steganography Key Exchange:
  Embed 1 bit of KE transcript into each ciphertext of PKE

- [HPRV19] Steganography Key Exchange with Secret Key Leakage:
  Embed a tiny piece of KE transcript into each ciphertext of PKE

Low Efficiency | **Disordering Not Allowed** | **Active Attack Not Allowed**

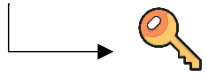
Achieve by KE? → Consider coercive environment → **Achieve by AKE?**

Current Methods

Core problem: the distribution of double key **under coercion**

- [vH04] Steganography Key Exchange:
  Embed 1 bit of KE transcript into each ciphertext of PKE

- [HPRV19] Steganography Key Exchange with Secret Key Leakage:
  Embed a tiny piece of KE transcript into each ciphertext of PKE



Low Efficiency | **Disordering Not Allowed** | **Active Attack Not Allowed**

Achieve by KE? → Consider coercive environment → **Achieve by AKE?**

Our Method: Anamorphic Authenticated Key Exchange

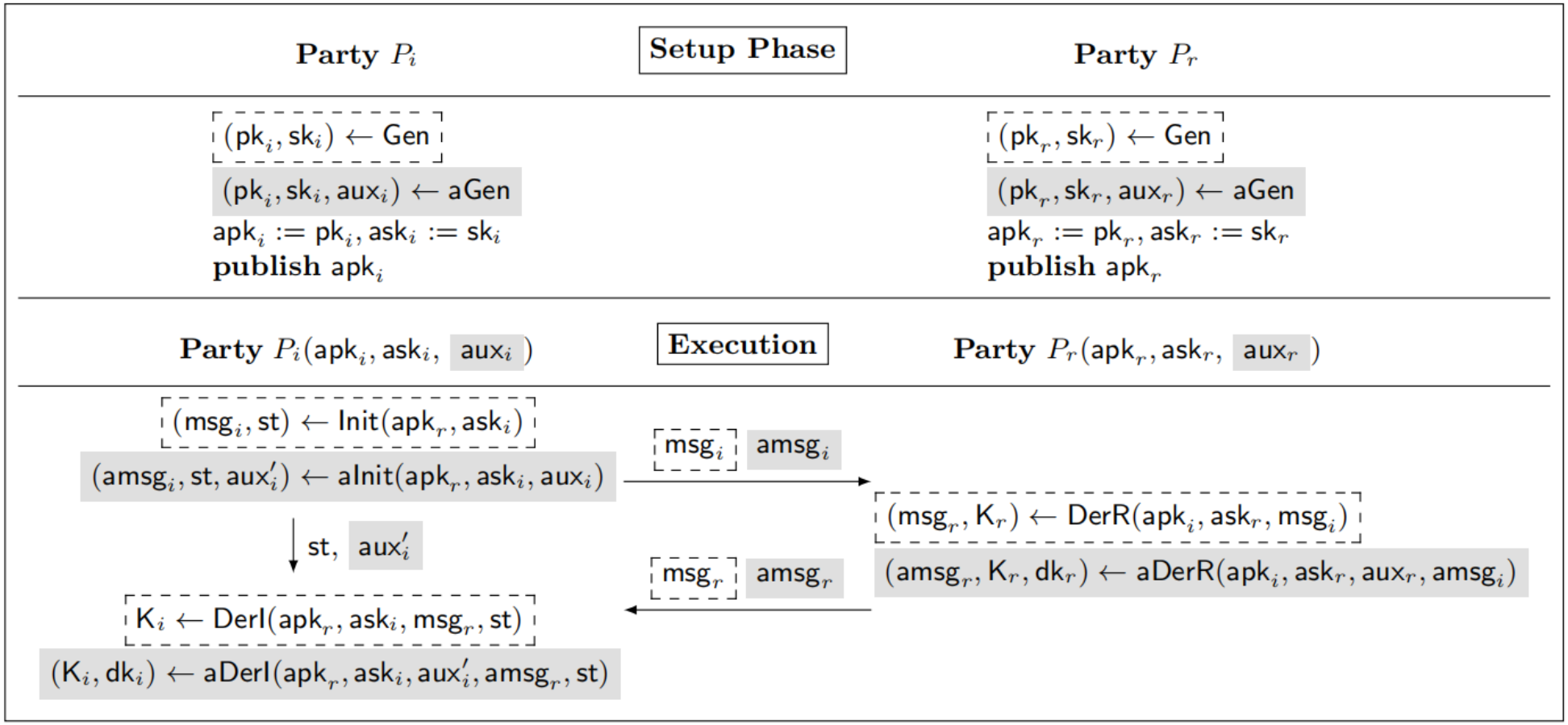
High Efficiency | **Achieving Robustness** | **Active Attack Allowed**

Contents

AM-AKE: Syntax

(Two-pass) AM-AKE = (Gen, Init, DerR, DerI) + (aGen, aInit, aDerR, aDerI)

Normal AKE Algorithms → Corresponding Anamorphic Version

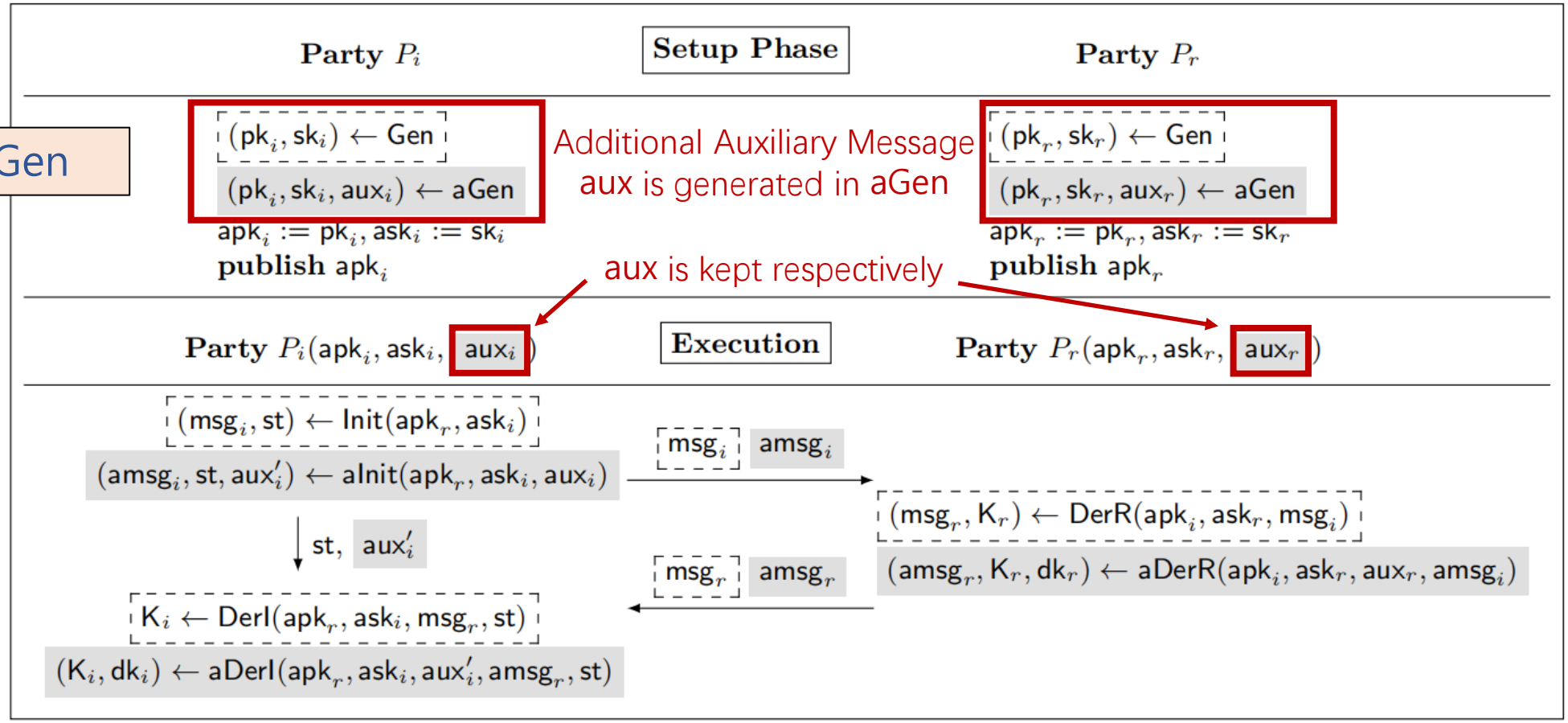


AM-AKE: Syntax

(Two-pass) AM-AKE = (Gen, Init, DerR, DerI) + (aGen, aInit, aDerR, aDerI)

Normal AKE Algorithms → Corresponding Anamorphic Version

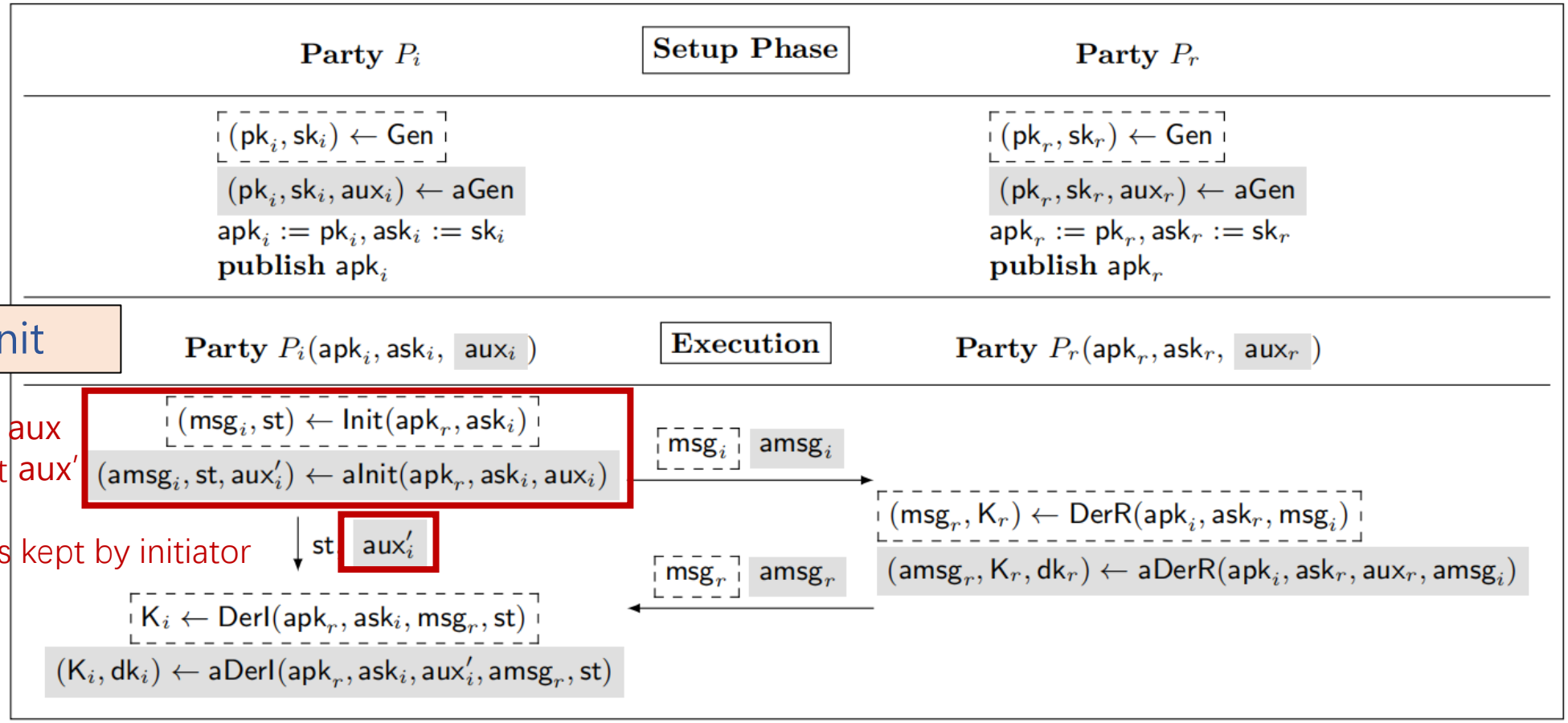
aGen VS Gen



AM-AKE: Syntax

(Two-pass) AM-AKE = (Gen, Init, DerR, DerI) + (aGen, aInit, aDerR, aDerI)

Normal AKE Algorithms → Corresponding Anamorphic Version



aInit VS Init

Additional Input aux

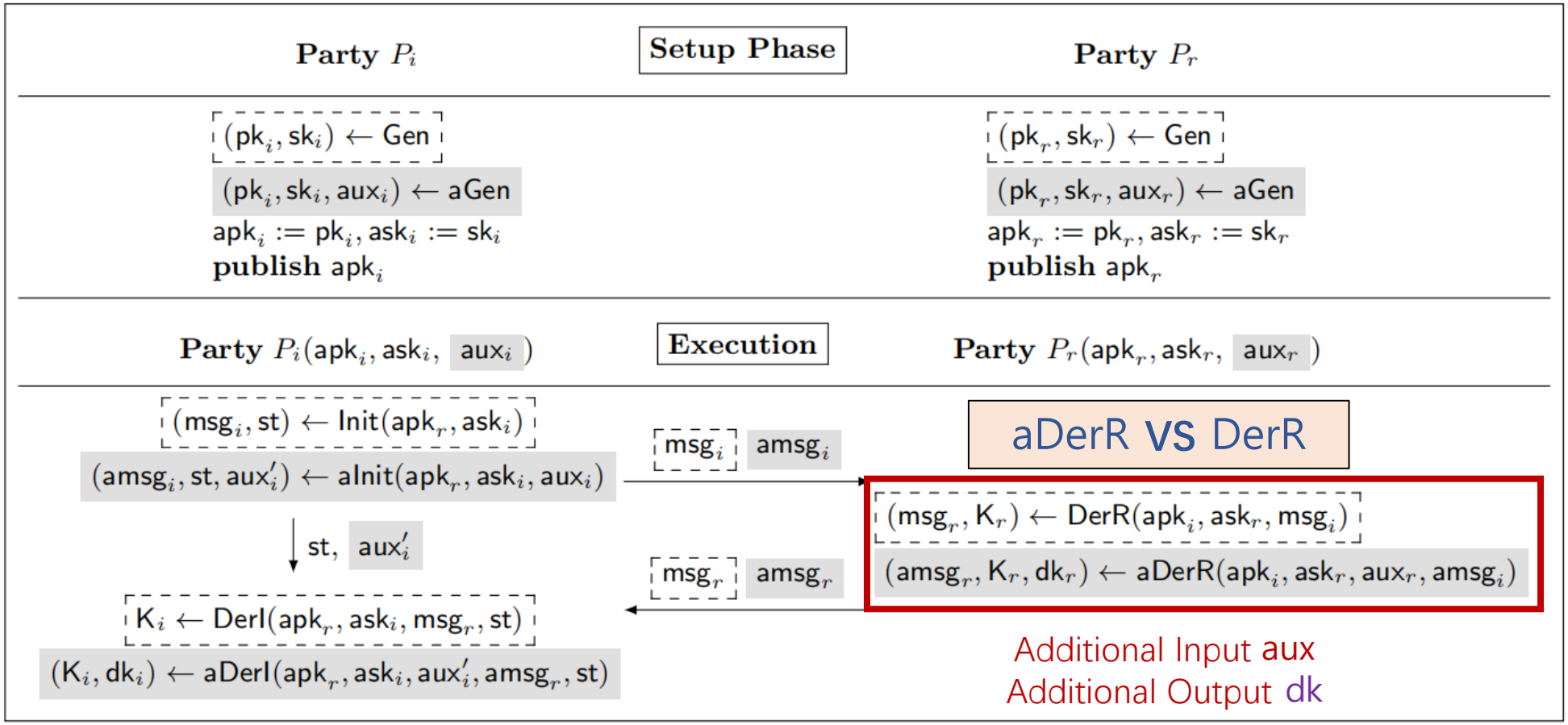
Additional Output aux'

aux' is kept by initiator

AM-AKE: Syntax

(Two-pass) AM-AKE = (Gen, Init, DerR, DerI) + (aGen, aInit, aDerR, aDerI)

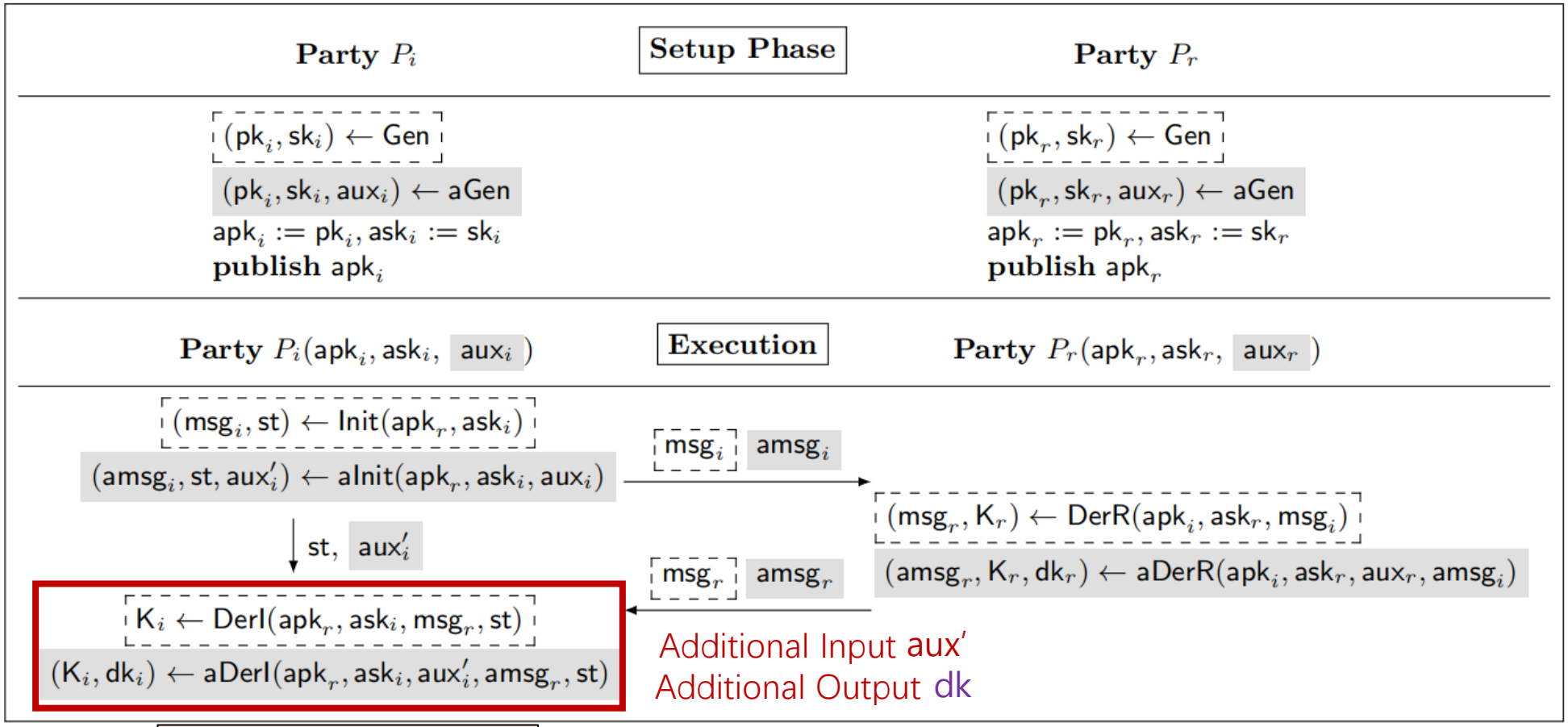
Normal AKE Algorithms → Corresponding Anamorphic Version



AM-AKE: Syntax

(Two-pass) AM-AKE = (Gen, Init, DerR, DerI) + (aGen, aInit, aDerR, aDerI)

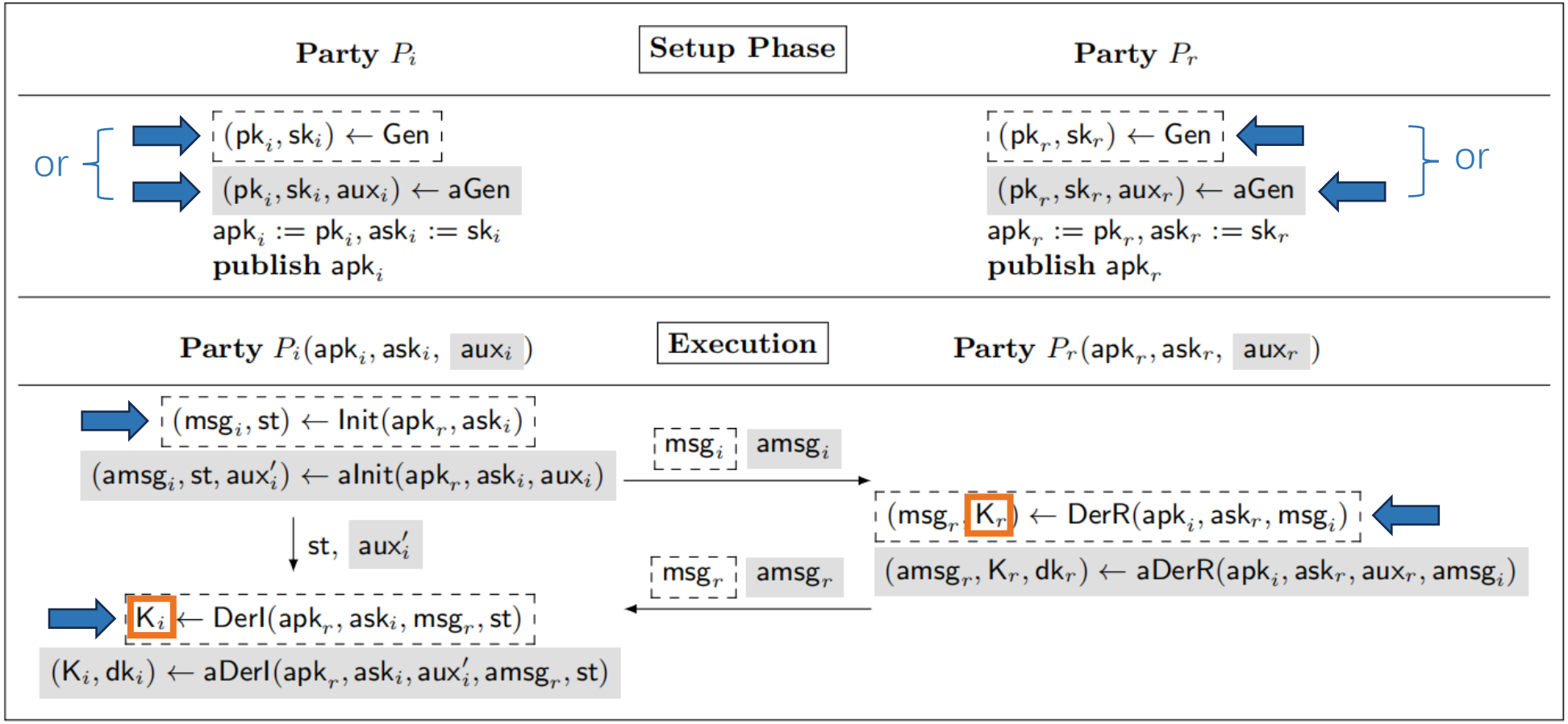
Normal AKE Algorithms → Corresponding Anamorphic Version



aDerI vs DerI

AM-AKE: Working Modes & Correctness

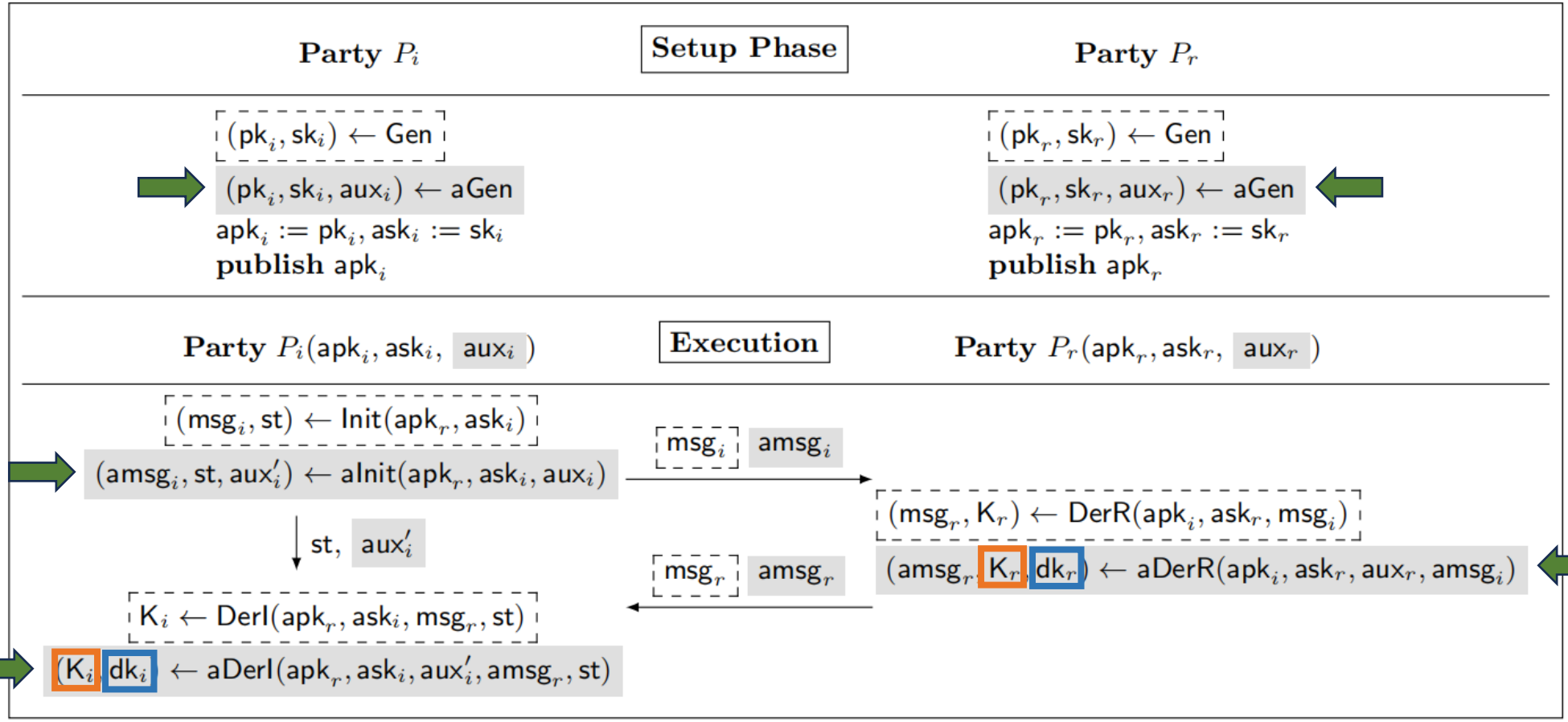
Normal Mode : Both P_i and P_r invoke normal algorithms **in execution phase**



Correctness Requirement : $K_i = K_r \neq \perp$

AM-AKE: Working Modes & Correctness

Anamorphic Mode : Both P_i and P_r invoke anamorphic algorithms

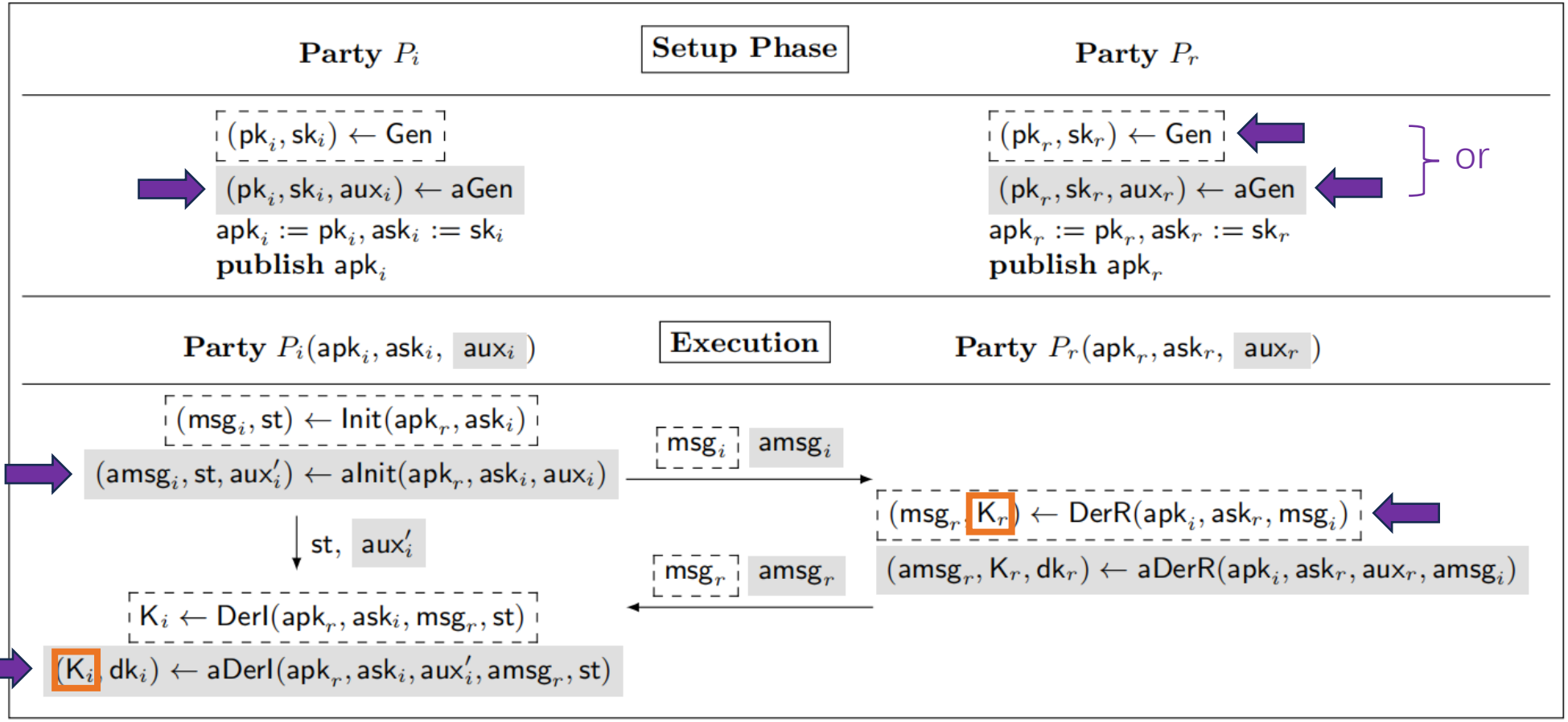


Correctness Requirement : $K_i = K_r \neq \perp$ & $dk_i = dk_r \neq \perp$

AM-AKE: Working Modes & Correctness

Half Mode : One invokes normal alg. & One invokes anamorphic alg. **in execution phase**

Case 1

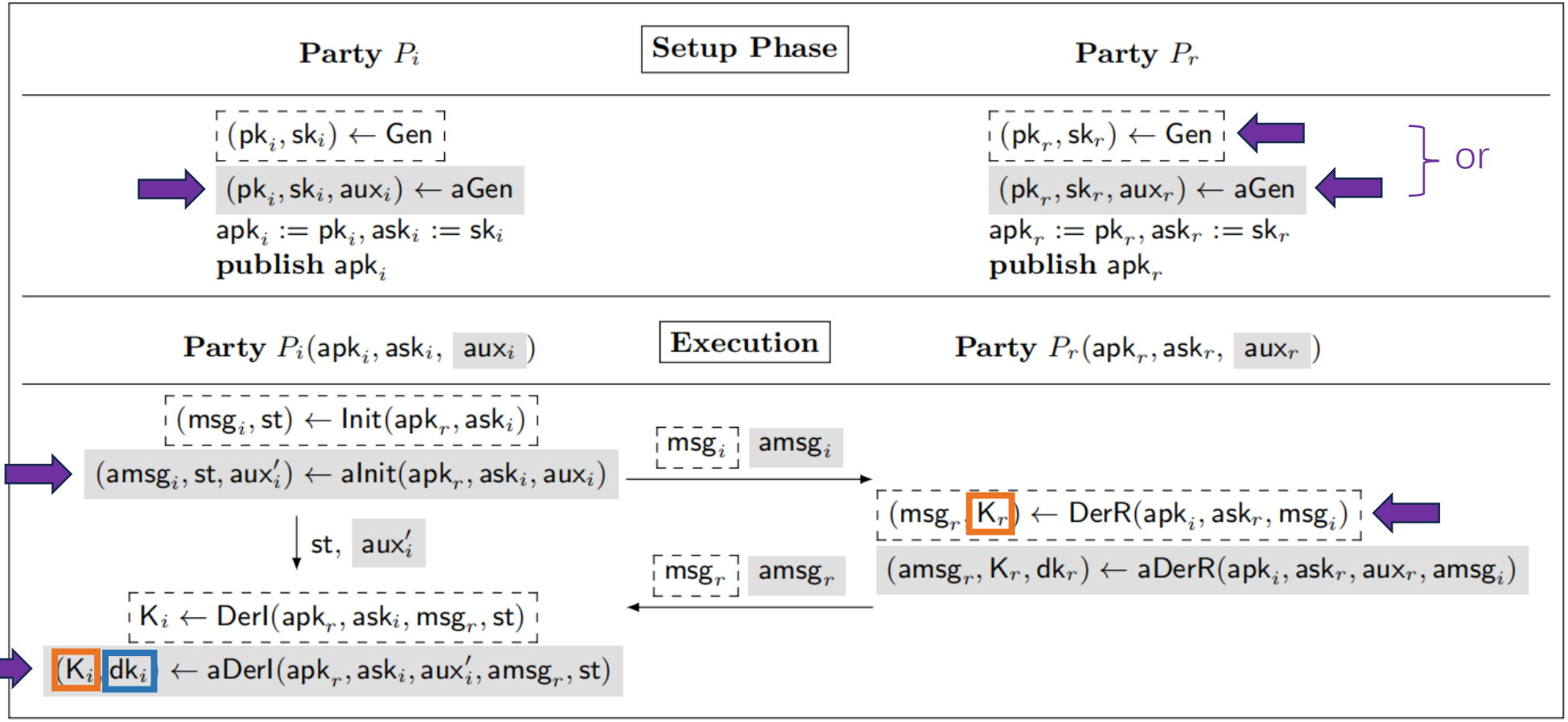


Correctness Requirement : $K_i = K_r \neq \perp$

AM-AKE: Working Modes & Correctness & Robustness

Half Mode : One invokes normal alg. & One invokes anamorphic alg. **in execution phase**

Case 1



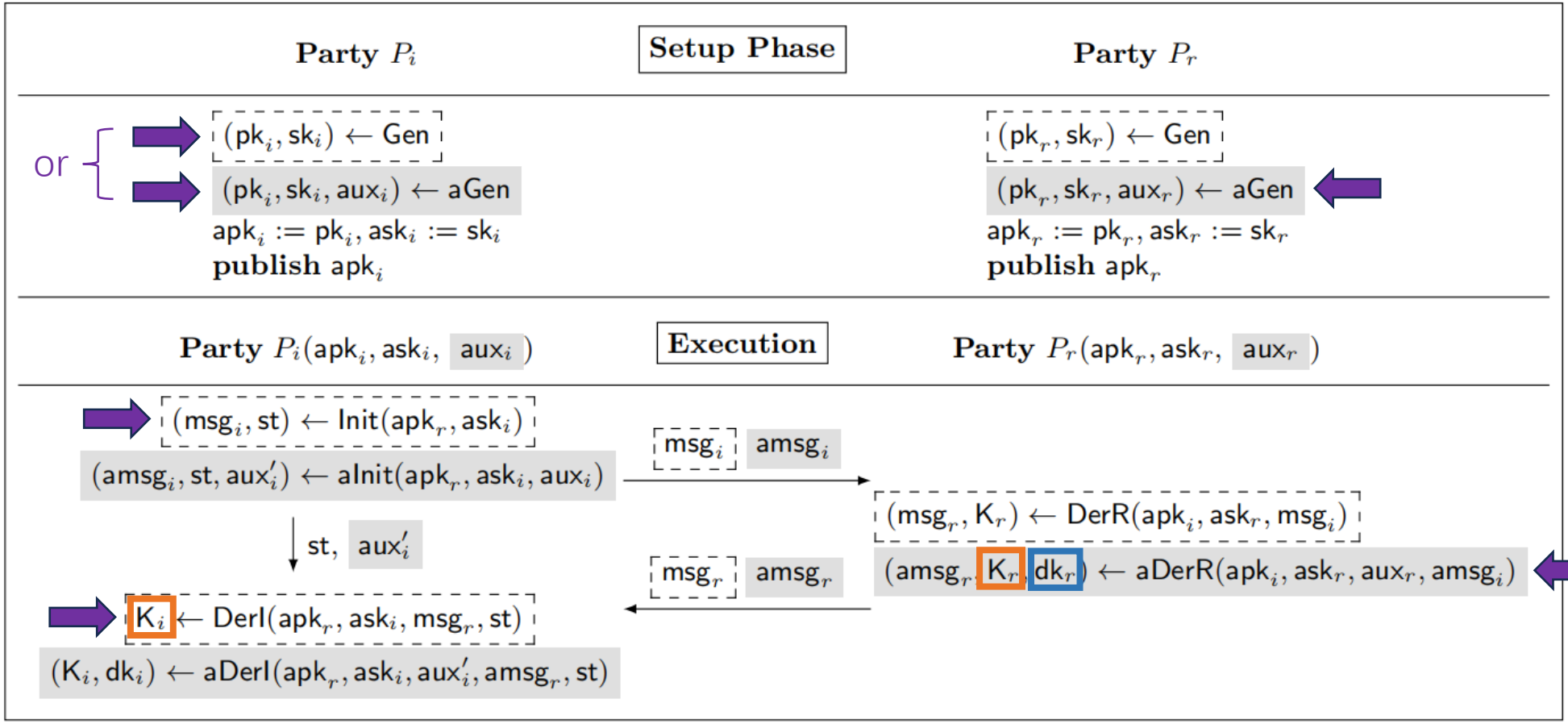
Correctness Requirement : $K_i = K_r \neq \perp$

Initiator-Robustness : $dk_i = \perp$

AM-AKE: Working Modes & Correctness & Robustness

Half Mode : One invokes normal alg. & One invokes anamorphic alg. **in execution phase**

Case 2



Correctness Requirement : $K_i = K_r \neq \perp$

Responder-Robustness : $dk_r = \perp$

AM-AKE: Security

In coercive environment, what can be known by the adversary?

- Long-term secret key
- Internal state of initiator
- Session key

In each session, the adversary can invoke $\mathsf{Der}I$ and check if the result equals to

AM-AKE: Security

In coercive environment, what can be known by the adversary?

- Long-term secret key
- Internal state of initiator
- Session key

In each session, the adversary can invoke Derl and check if the result equals to

What cannot be known by the adversary?

- **Working Modes of AM-AKE** —Indistinguishability→ IND-WM **Security**
- **Information of Double Key** —Pseudo-Randomness→ PR-DK **Security**

Contents

Plain AM-AKE

Problem: Sometimes it's hard to find a trapdoor for the secret key

Plain AM-AKE:

In aGen, (pk, sk) is generated **before** aux

Plain AM-AKE & Why Impossible?

Problem: Sometimes it's hard to find a trapdoor for the secret key

Plain AM-AKE:

In aGen, (pk, sk) is generated **before** aux

What does it lead to?

No effective trapdoor is generated

⇒ Parties have **no advantage** against adversary

⇒ Adversary can perfectly **impersonate any party** and conduct active attack!

Impossibility Results

It's impossible for a plain two-pass AM-AKE to achieve:

- **responder-robustness**
- both **initiator-robustness** and **IND-WM** security
- **PR-DK** security

How to Circumvent ? → Relaxed Security

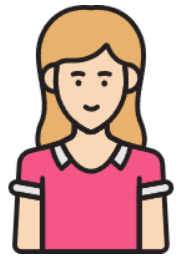
To bypass the impossibility results, we define **relaxed security** for plain AM-AKE

→ Active Attack is **Disallowed**

I cannot impersonate parties in relaxed model

Generic Construction of Plain AM-AKE

Core Idea: Covertly embed a KEM into AKE



$msg_i \leftarrow \mathrm{Init}(pk_r, sk_i)$

$msg_r \leftarrow \mathrm{DerR}(pk_i, sk_r)$

Initiator
(pk_i, sk_i)

AKE

Responder
(pk_r, sk_r)

KEM

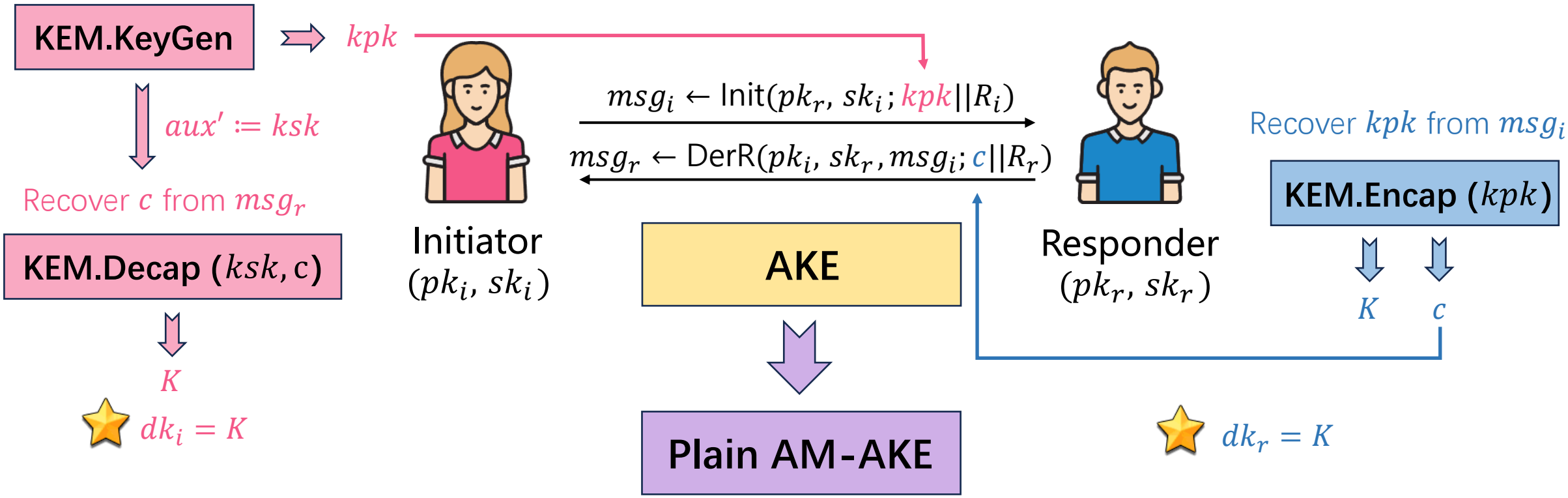
KEM.KeyGen → kpk ksk

KEM.Encap (kpk) → $dk_r := K$ c

KEM.Decap (ksk, c) → $dk_i := K$

Generic Construction of Plain AM-AKE

Core Idea: Covertly embed a KEM into AKE



Correctness: Init and DerR should be **partially randomness-recoverable**

Security: KEM should be **fully pseudorandom**, i.e., $(kpk,\ c,\ K) \approx (\$,\ \$,\ \$)$

Contents

Generic Construction of AM-AKE

Core Idea:

① Extract an identical secret s before communication

② Use s as PRF seed to get dk during communication

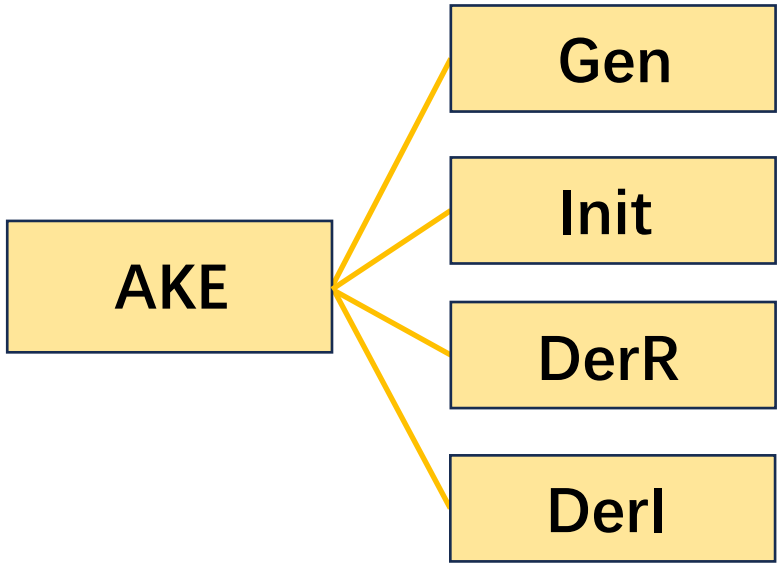
Generic Construction of AM-AKE

Core Idea:

① Extract an identical secret s before communication

② Use s as PRF seed to get dk during communication

How to achieve? ⇒ **Requiring new properties for underlying AKE**



Secret Extractability

3-Separability

3-Separability

Generic Construction of AM-AKE

Core Idea:

① Extract an identical secret s before communication

② Use s as PRF seed to get dk during communication

How to achieve? ⟹ **Requiring new properties for underlying AKE**

AKE

Gen — **Secret Extractability**

- There exists **SimGen** ≈ **Gen,** but outputs an additional msk serving as trapdoor
- For all (pk_i, sk_i, msk_i), $(pk_r, sk_r, msk_r) \leftarrow$ SimGen:

$$\text{Extract}(msk_i,\ pk_r) = s = \text{Extract}(msk_r,\ pk_i)$$

- s is pseudorandom to  even in the presence of sk_i and sk_r

In this way, **aGen** is set to **SimGen**, where $aux := msk$

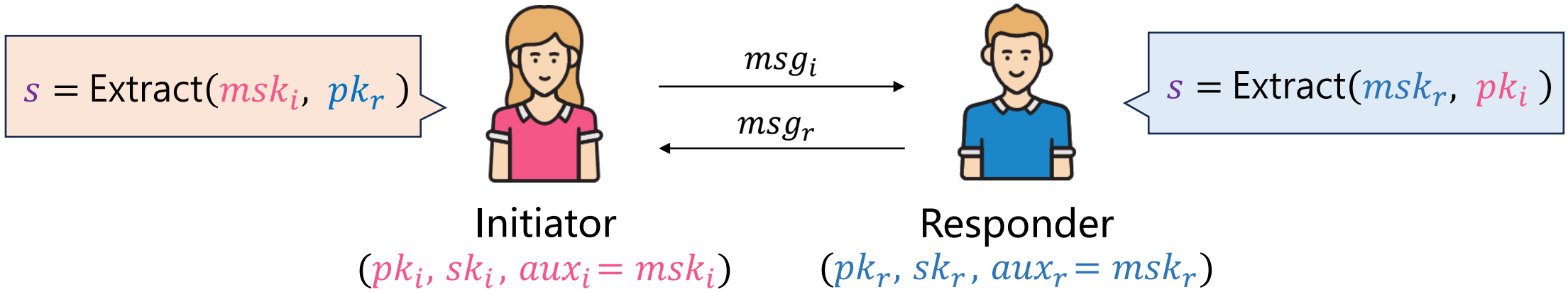
Generic Construction of AM-AKE

Core Idea:

① Extract an identical secret s before communication

② Use s as PRF seed to get dk during communication

How to achieve? ⟹ **Requiring new properties for underlying AKE**



$$dk = PRF(s, (msg_i, msg_r))$$

Generic Construction of AM-AKE

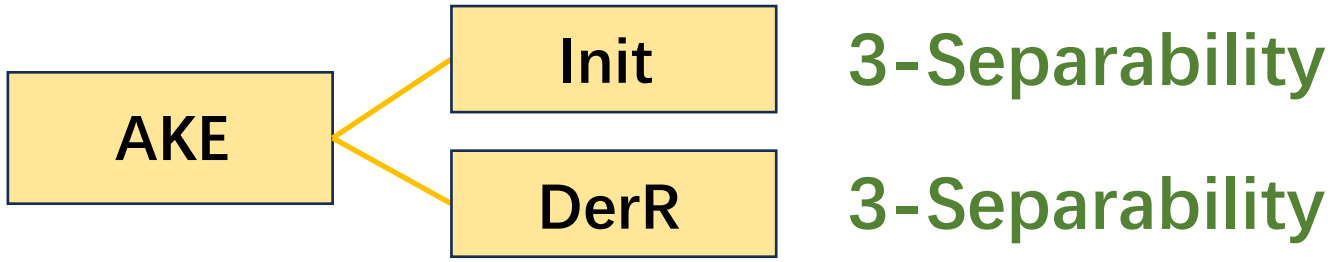
Core Idea:

① Extract an identical secret s before communication

② Use s as PRF seed to get dk during communication

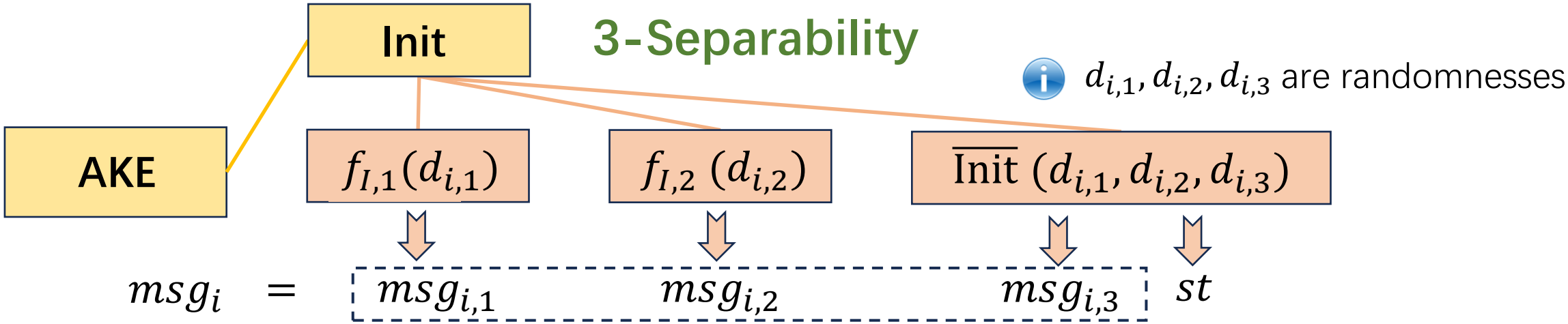
How to achieve? ⟹ **Requiring new properties for underlying AKE**

What about Robustness?



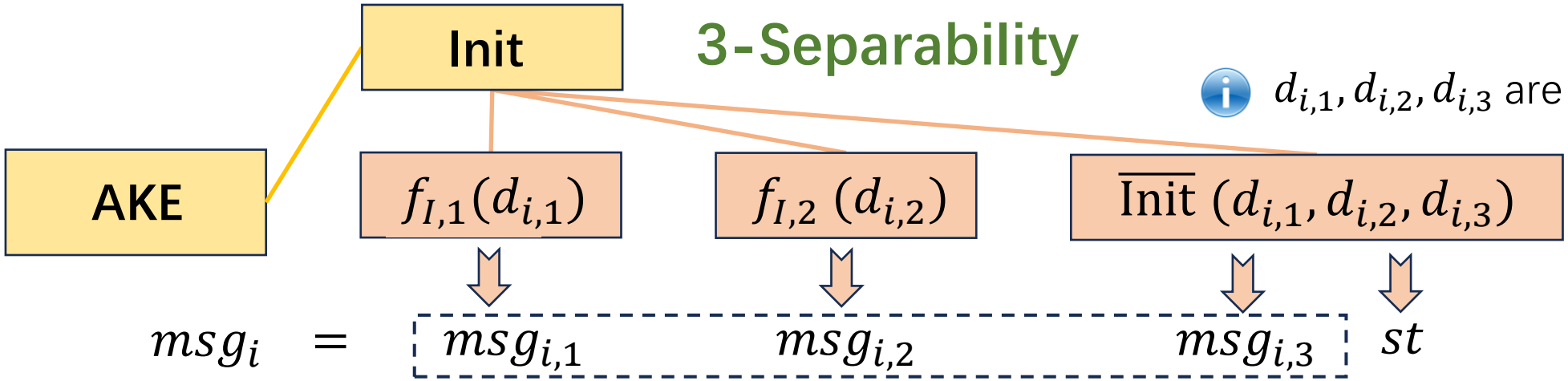
Generic Construction of AM-AKE

Take **Responder-Robustness** as Example



Generic Construction of AM-AKE

Take **Responder-Robustness** as Example



How to achieve Responder-Robustness? ⇒ Set $d_{i,2} = PRF(s, msg_{i,1})$

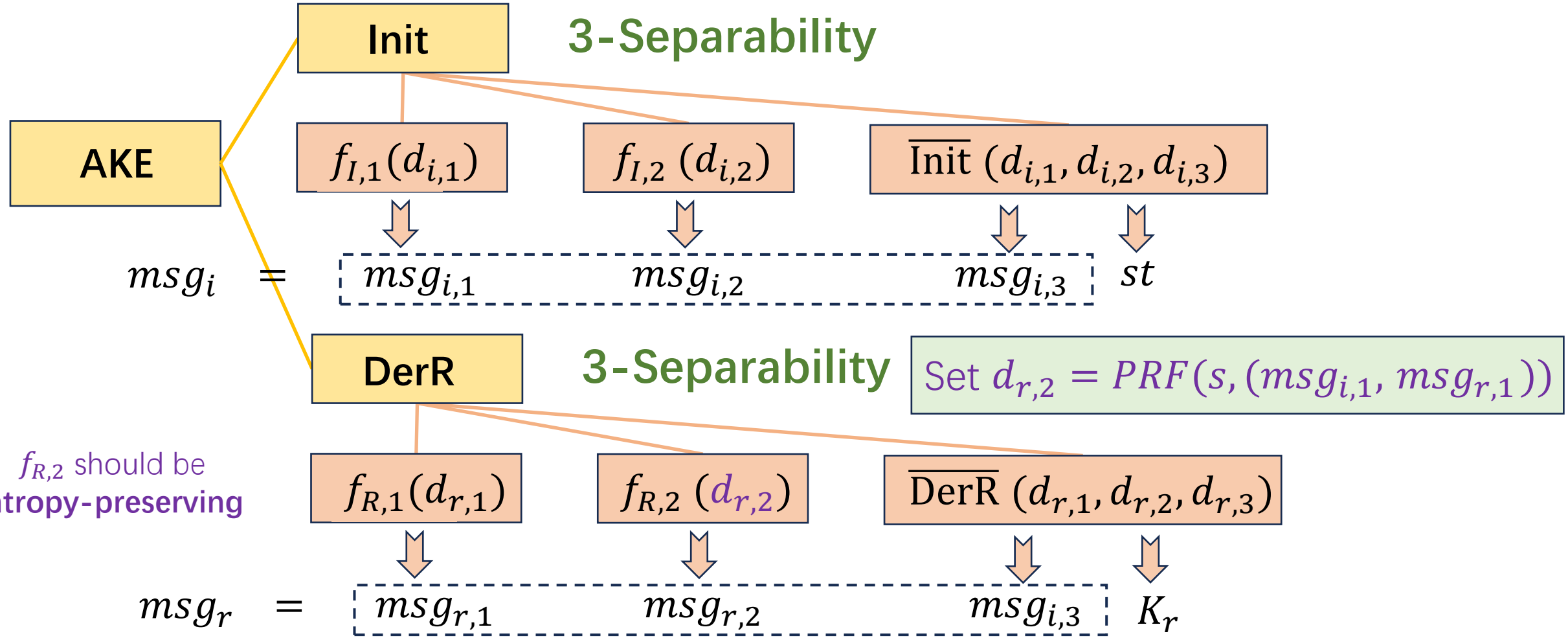
I can compute $f_{I,2}\left(PRF\left(s, msg_{i,1}\right)\right)$ and checks whether it equals to $msg_{i,2}$

Responder

$f_{I,2}$ should be **entropy-preserving**: Different inputs always lead to different outputs

Generic Construction of AM-AKE

Initiator-Robustness is achieved in similar way



IND-WM security of AM-AKE

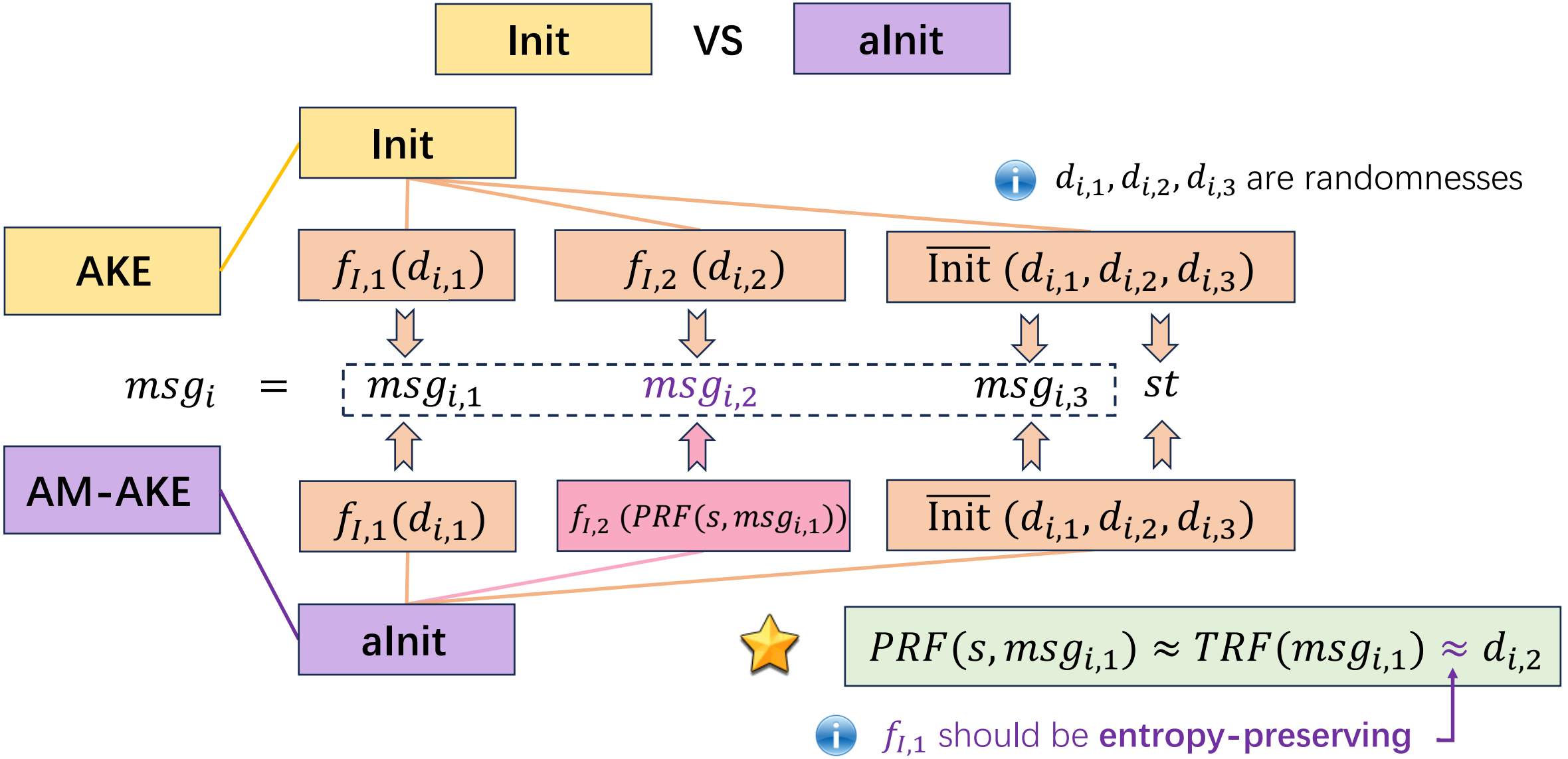
Does it achieve **strong IND-WM security?**

Gen VS **aGen**

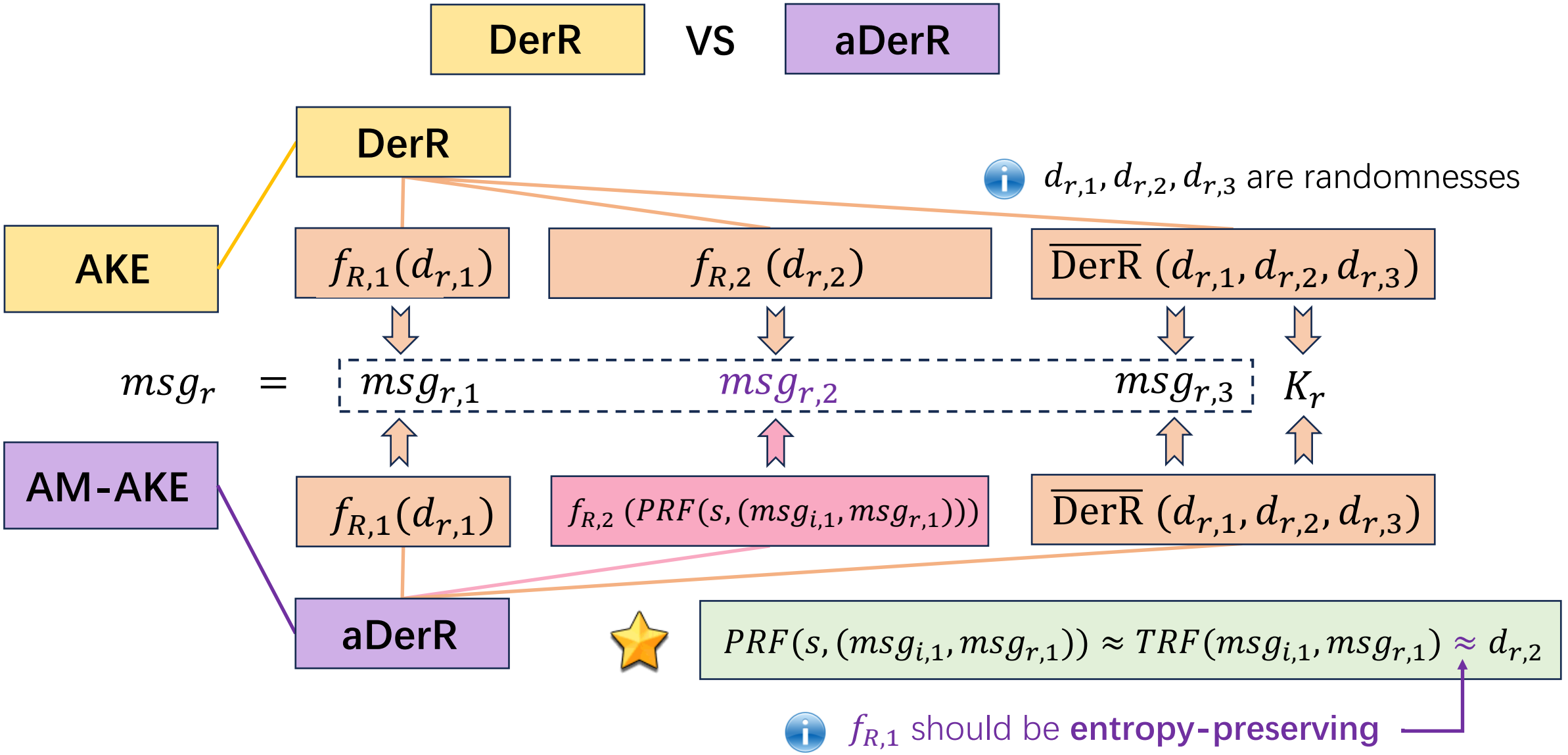
Secret Extractability of Gen directly guarantees the indistinguishability

- There exists **SimGen** ≈ **Gen,** but outputs an additional msk serving as trapdoor
- **aGen** is set to **SimGen**, where $aux := msk$

IND-WM security of AM-AKE



IND-WM security of AM-AKE



IND-WM security of AM-AKE

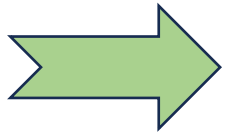
Finally,

Gen ≈ aGen

Init ≈ aInit

DerR ≈ aDerR

DerI = aDerI



AKE

≈

AM-AKE

PR-DK security of AM-AKE

- For **Passive Attack**: $dk = PRF(s, (msg_i, msg_r)) \approx TRF(msg_i, msg_r) \approx \$$
- For **Active Attack**: In each session,  can only control **one side!**
  - If initiator is controlled to send some designated m:

$$dk = PRF(s, (m, msg_r)) \approx TRF(m, msg_r) \approx \$$$

Freshness of $msg_{r,1} = f_{R,1}(d_{r,1})$

  - If responder is controlled to send some designated m:

$$dk = PRF(s, (msg_i, m)) \approx TRF(msg_i, m) \approx \$$$

Freshness of $msg_{i,1} = f_{I,1}(d_{i,1})$

Instantiation of AM-AKE

AKE is instantiated by SIG+KEM paradigm / 3KEM paradigm. Take SIG+KEM as example

- **Secret Extractability** of AKE.Gen ⟹ **Secret Extractability** of SIG.Gen

$$sk_{SIG} = g_2^x, \qquad pk_{SIG} = e(g_1, g_2)^x$$

  - For SimGen (= AKE.Gen), $aux := msk_{SIG} = x$
  - For $(e(g_1, g_2)^x, g_2^x, x)$, $(e(g_1, g_2)^y, g_2^y, y) \leftarrow$ SimGen:

$$\text{Extract}(x, e(g_1, g_2)^y) = \text{Extract}(y, e(g_1, g_2)^x) = e(g_1, g_2)^{xy} = s$$

  - s is pseudorandom to adversary knowing sk_{SIG} (i.e., g_2^x and g_2^y)

    Otherwise it solves DDH problem on group G_2!

- **3-Separability** of AKE.Init/DerR ⟹ **2-Separability** of SIG.Sign — EUF-CMA can be proved

  SIG.Sign($sk_{SIG} = g_2^x, m$): choose $r \leftarrow \$$, compute $\sigma = \left(\sigma_1 = g_1^r, \sigma_2 = g_2^{xH(m,\sigma_1)+r}\right)$

- **Entropy-preserving** of subfunc. ⟹ **Entropy-preserving** of $f(r) = g_1^r$, etc. — Easy to prove

Our Contribution

- **Definition** of a new primitive **Anamorphic Authenticated Key Exchange**.
  - Define its syntax, working modes, robustness and security models
  - Solve the problem of **double key distribution under coercion**
- **Impossibility Results** of Plain AM-AKE
- **Generic Construction** of Plain AM-AKE with initiator-robustness and relaxed security
- **Generic Construction** of AM-AKE with **full robustness** and **strong security**
- **Instantiations** from SIG+KEM /3KEM Paradigms

Thanks! Questions?

ePrint: ia.cr/2024/1438