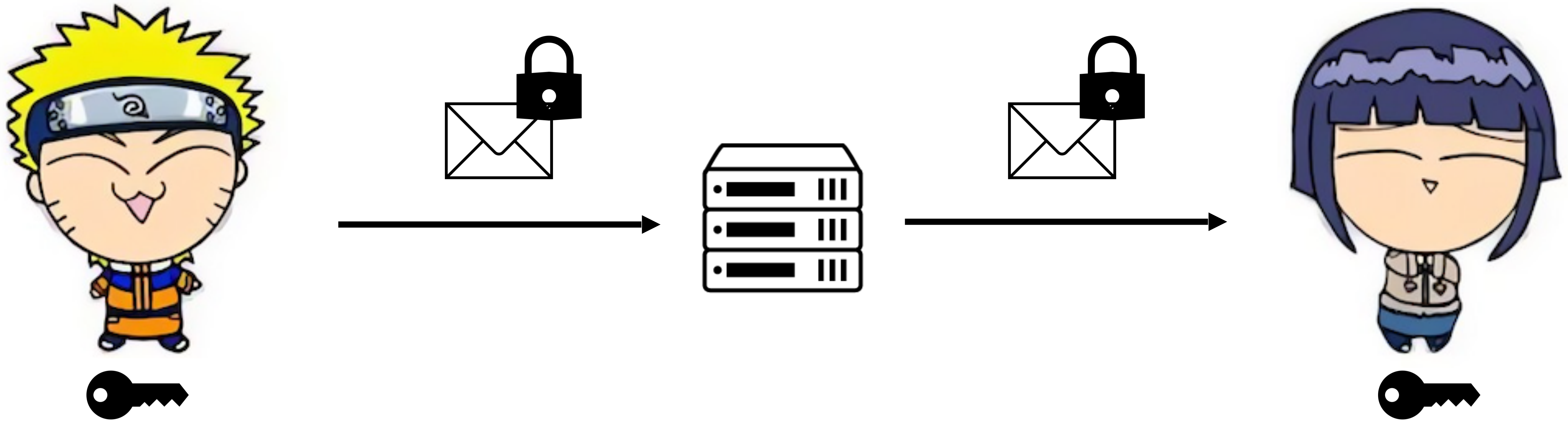
Analyzing Group Chat Encryption in MLS, Session, Signal, and Matrix

Joseph Jaeger and Akshaya Kumar

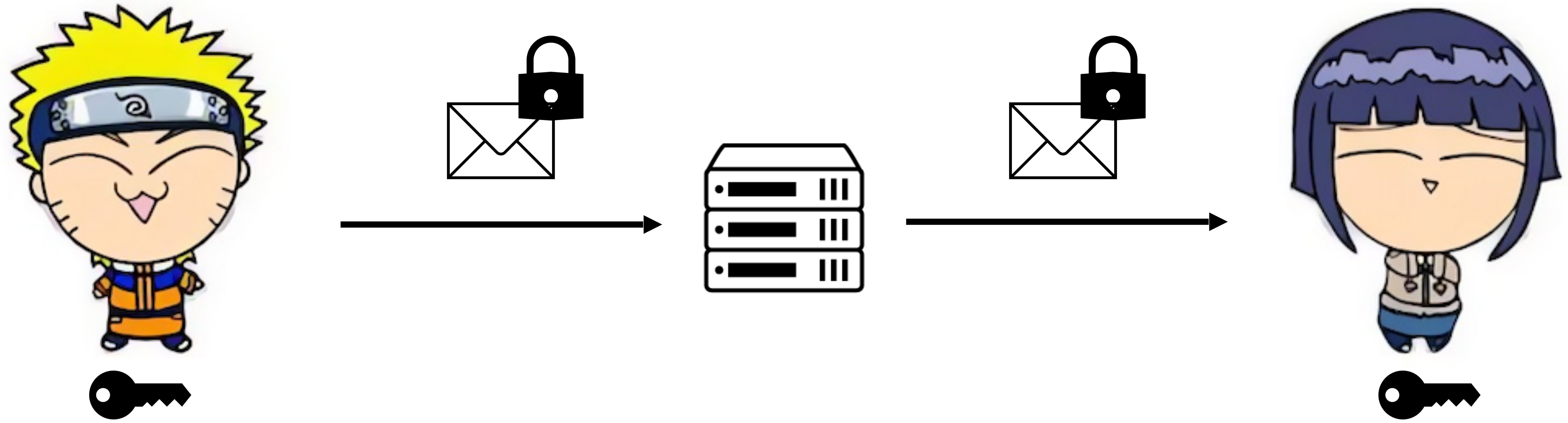
Eurocrypt 2025

Georgia Tech

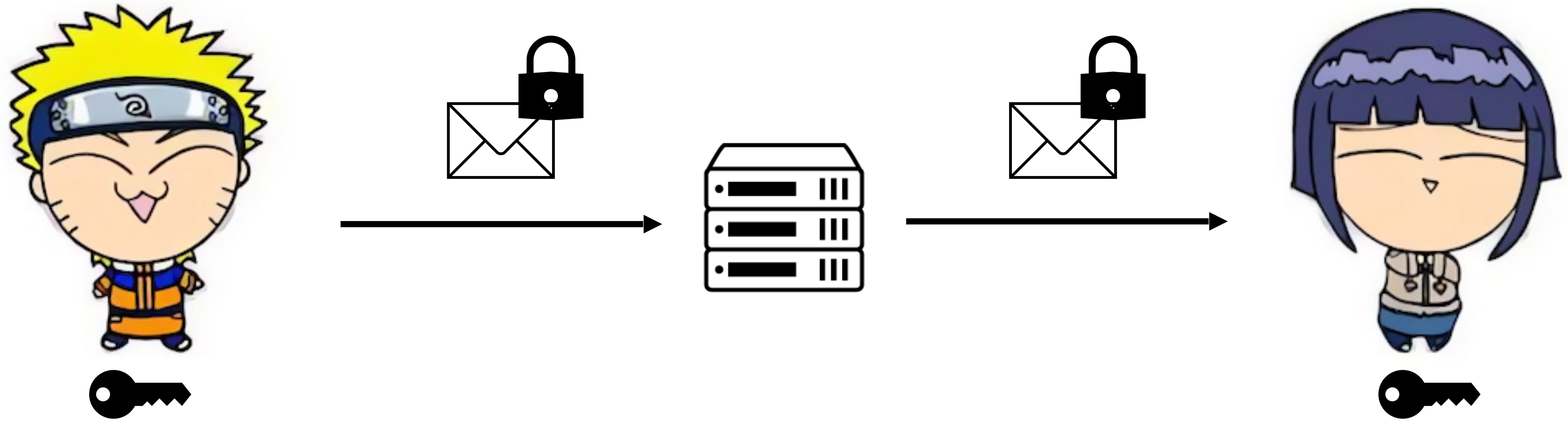
E2EE/Secure Messaging



E2EE/Secure Messaging



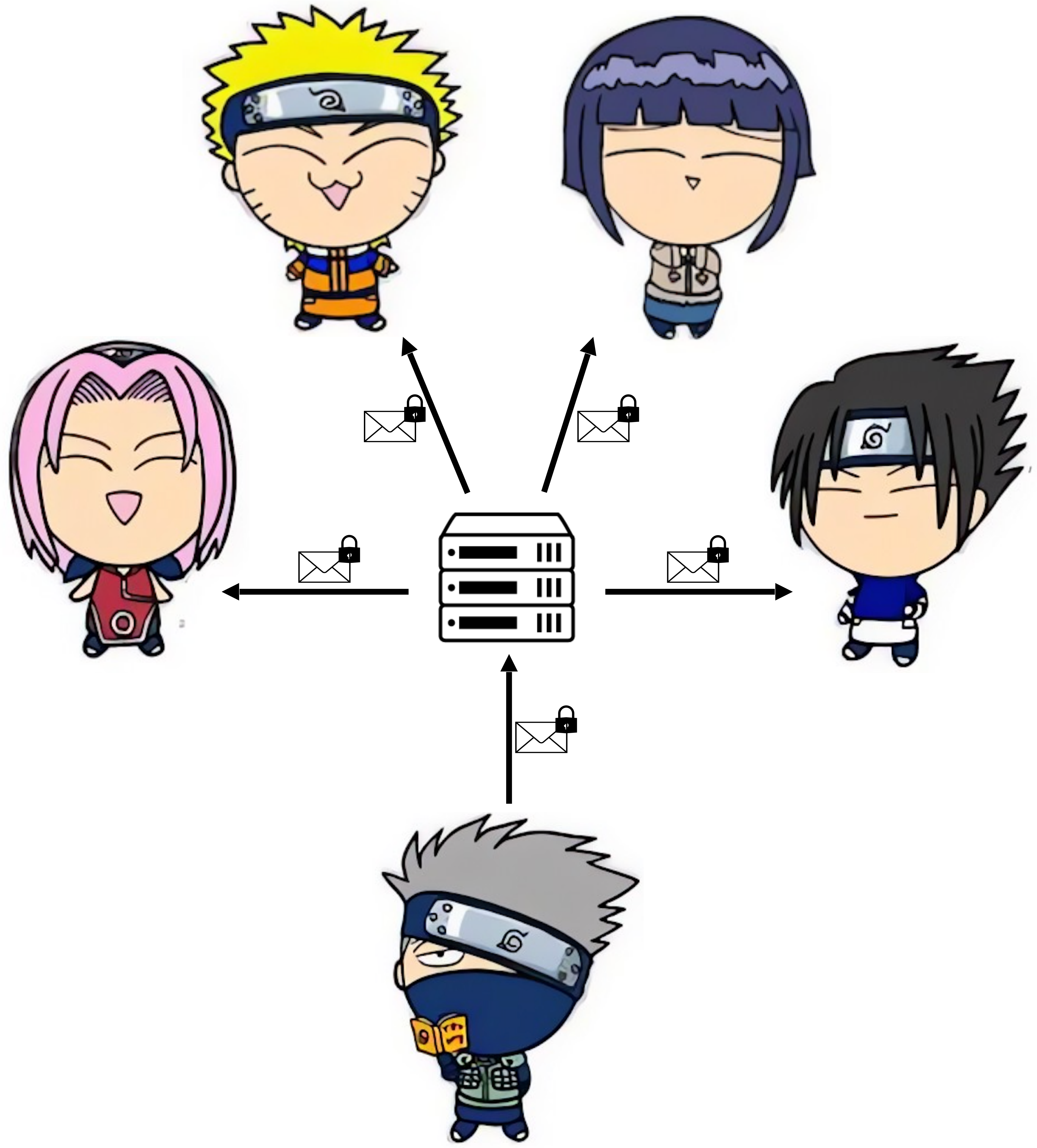
E2EE/Secure Messaging



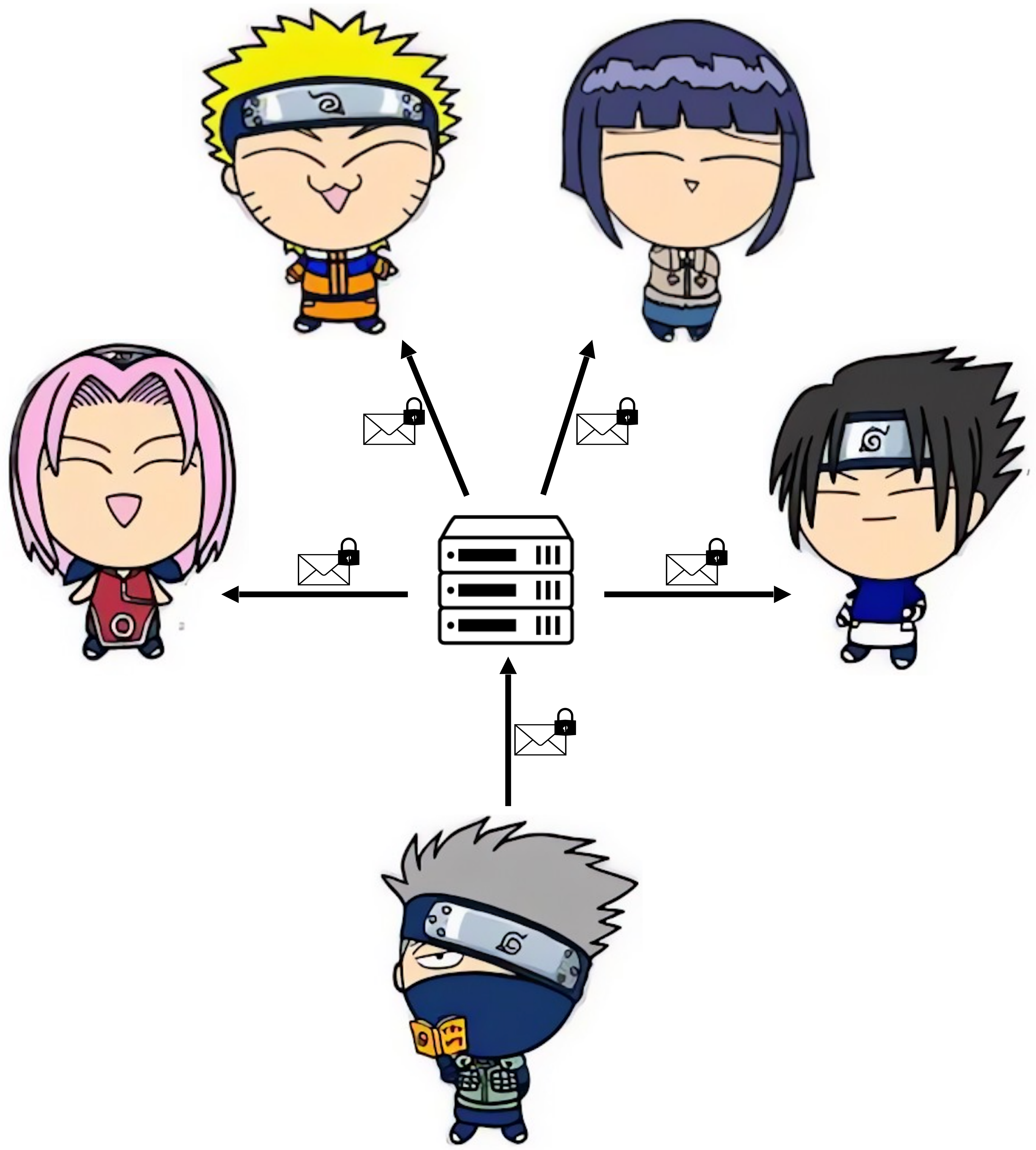
MLS

matrix

Secure Group Messaging

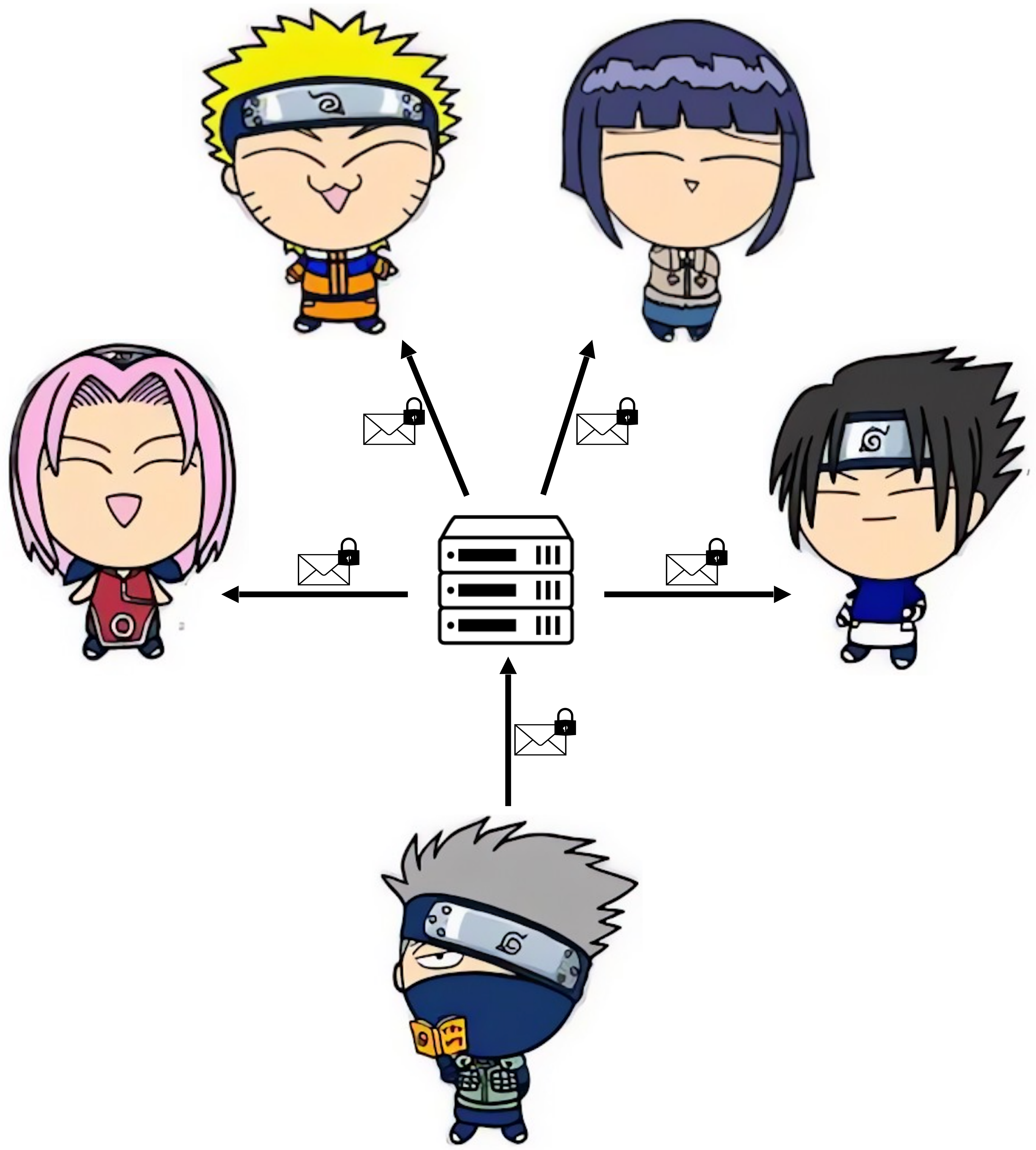


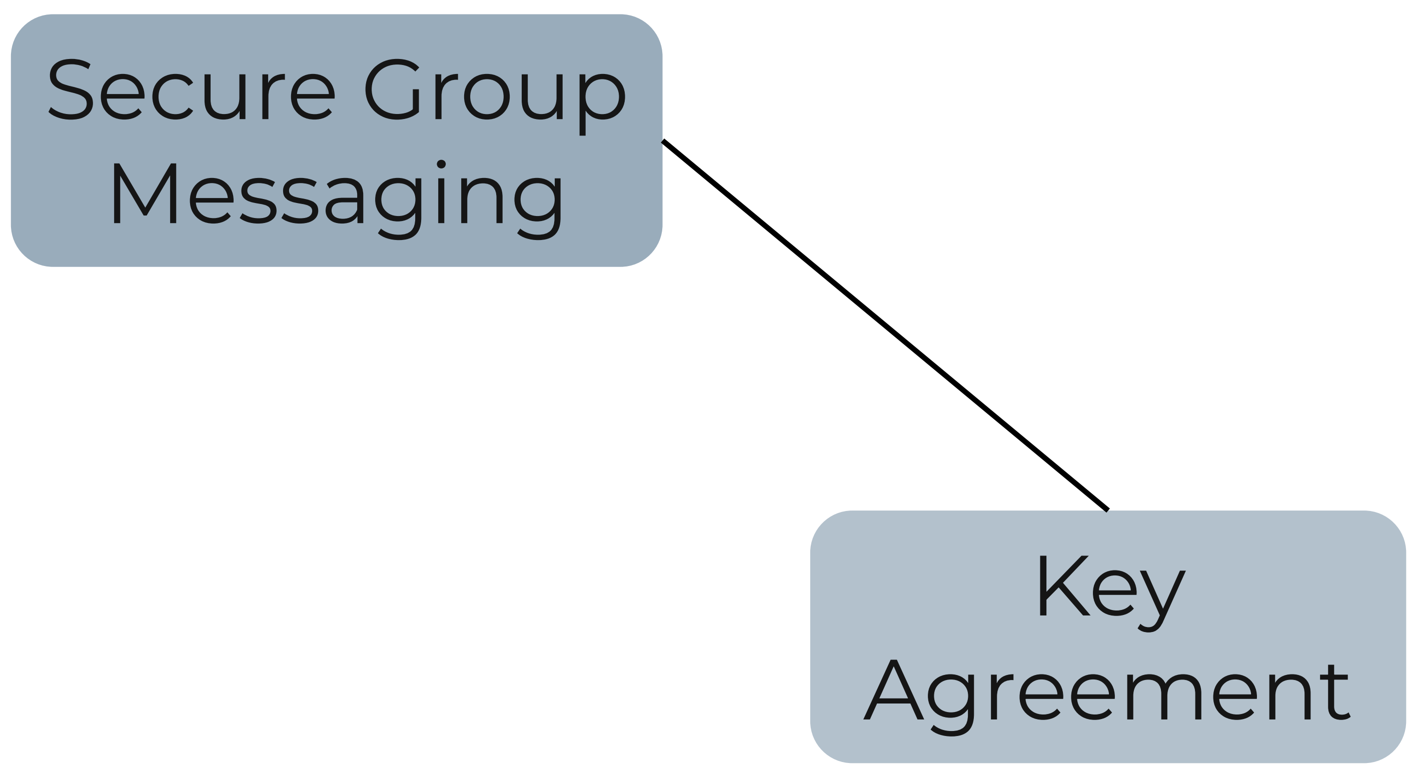
Secure Group Messaging



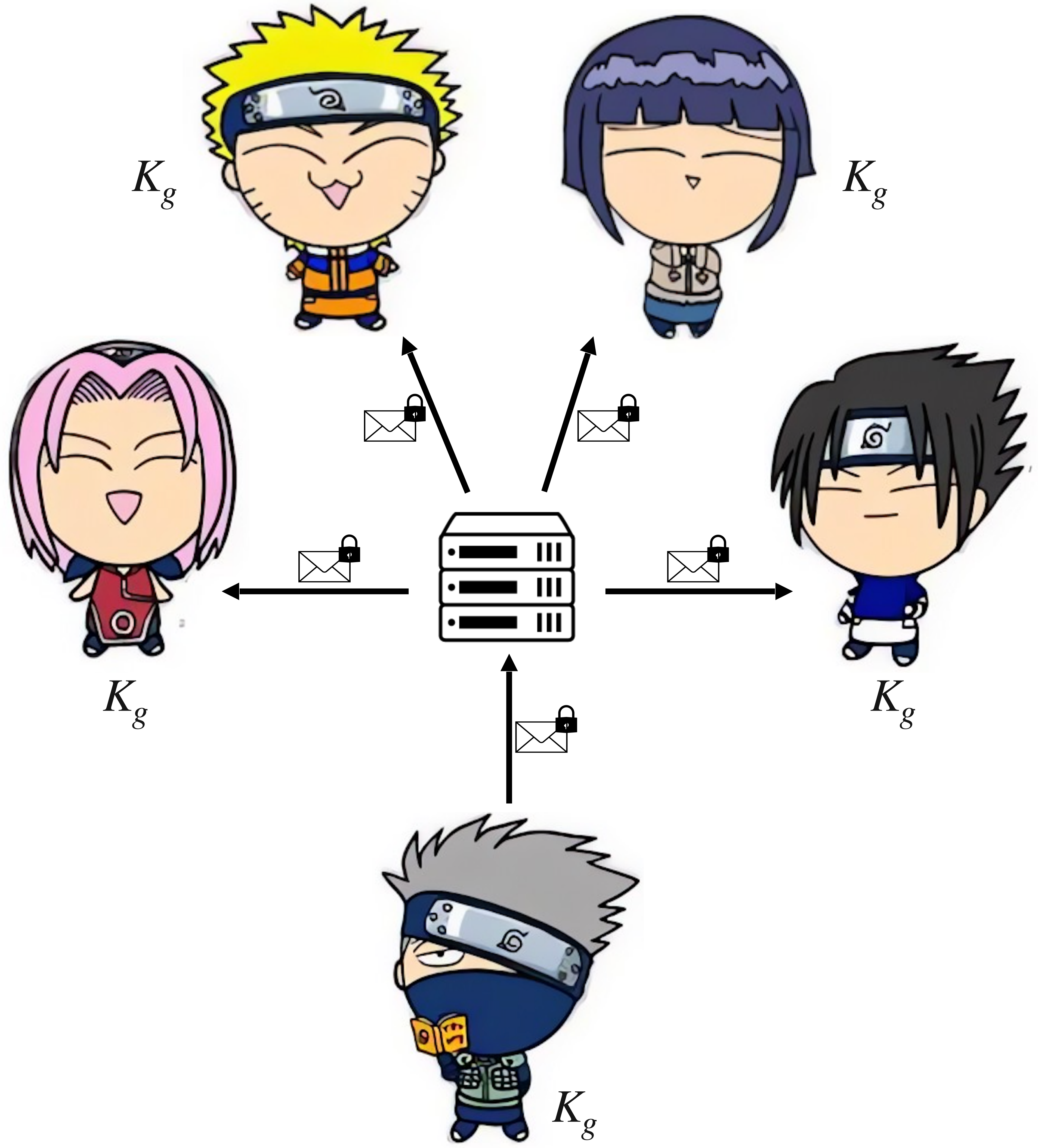
Secure Group Messaging

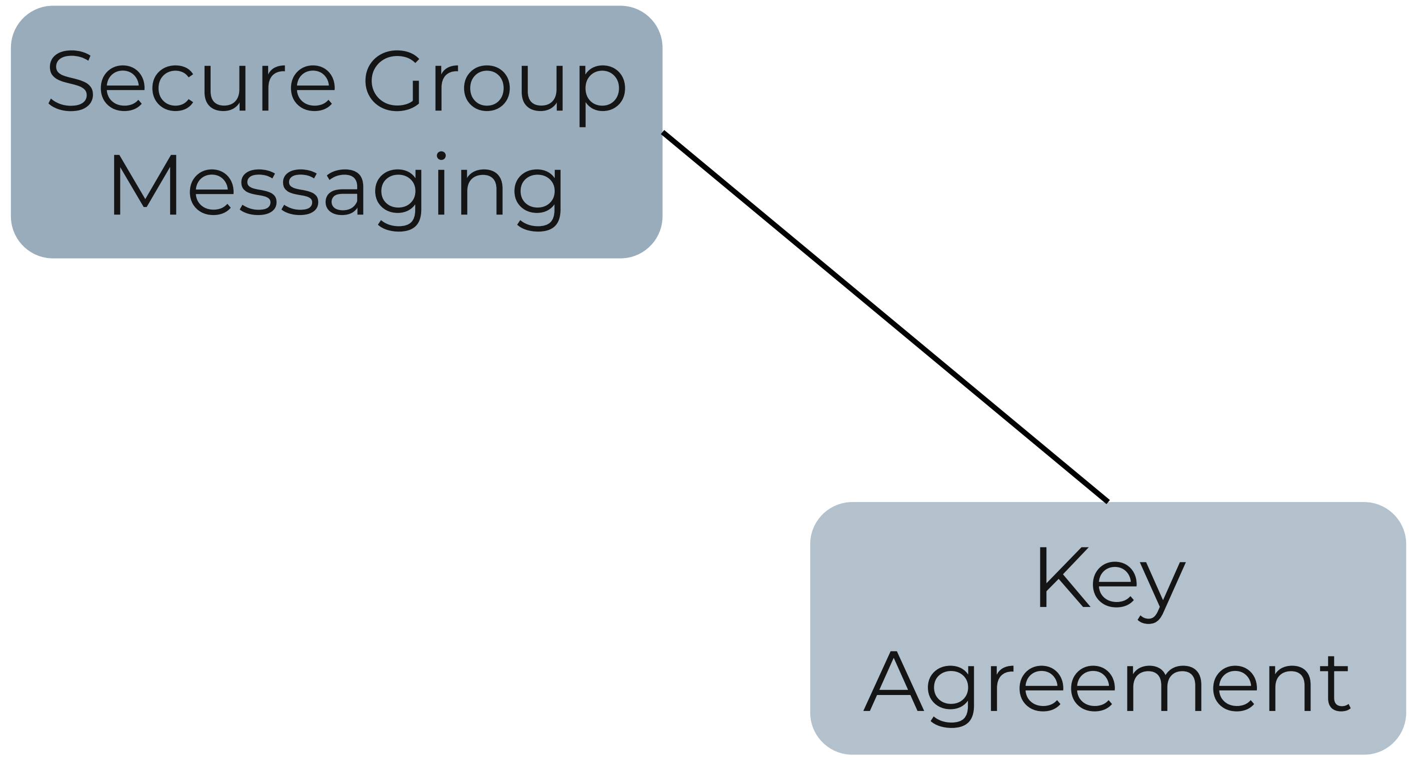
Secure Group Messaging



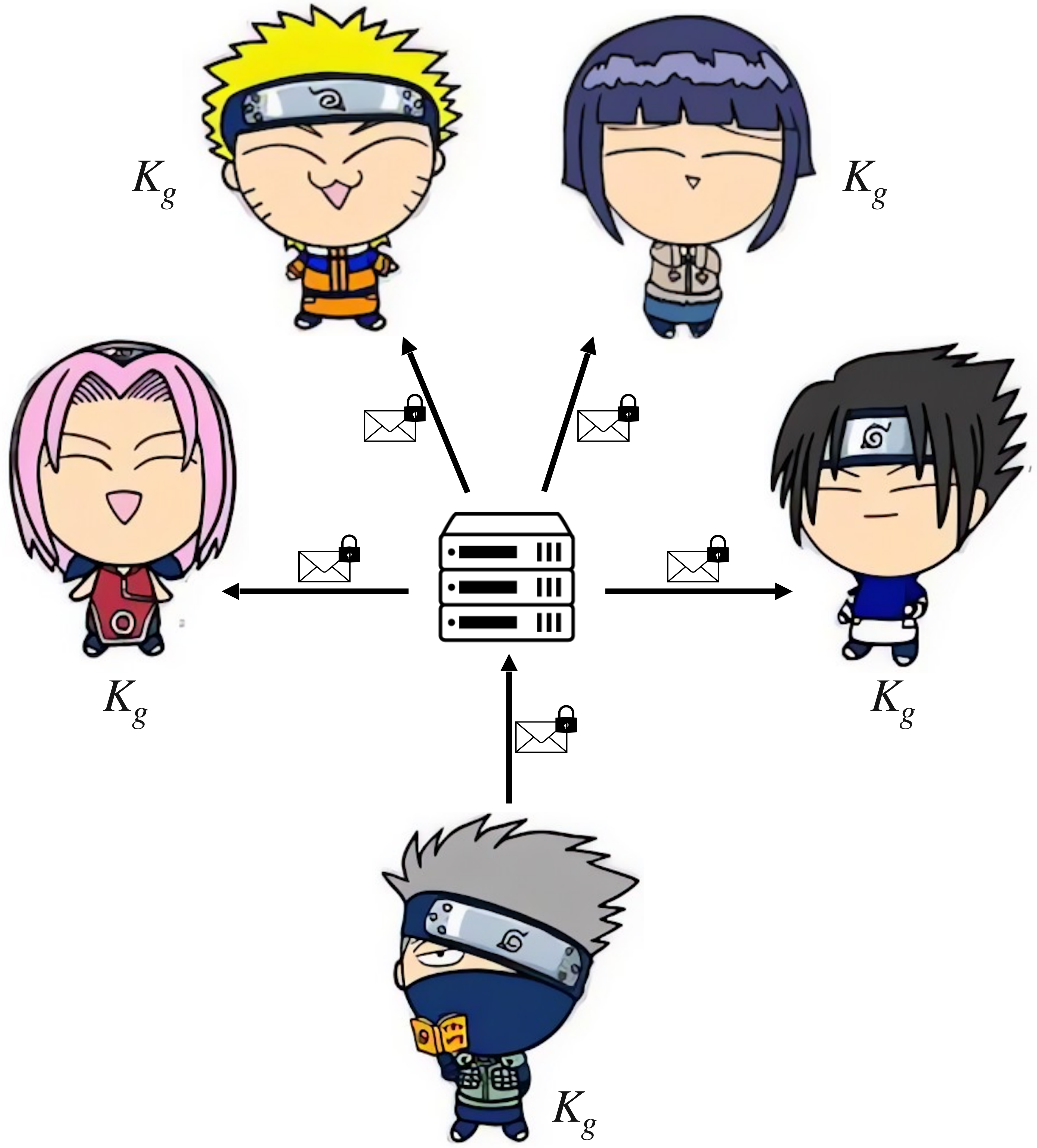


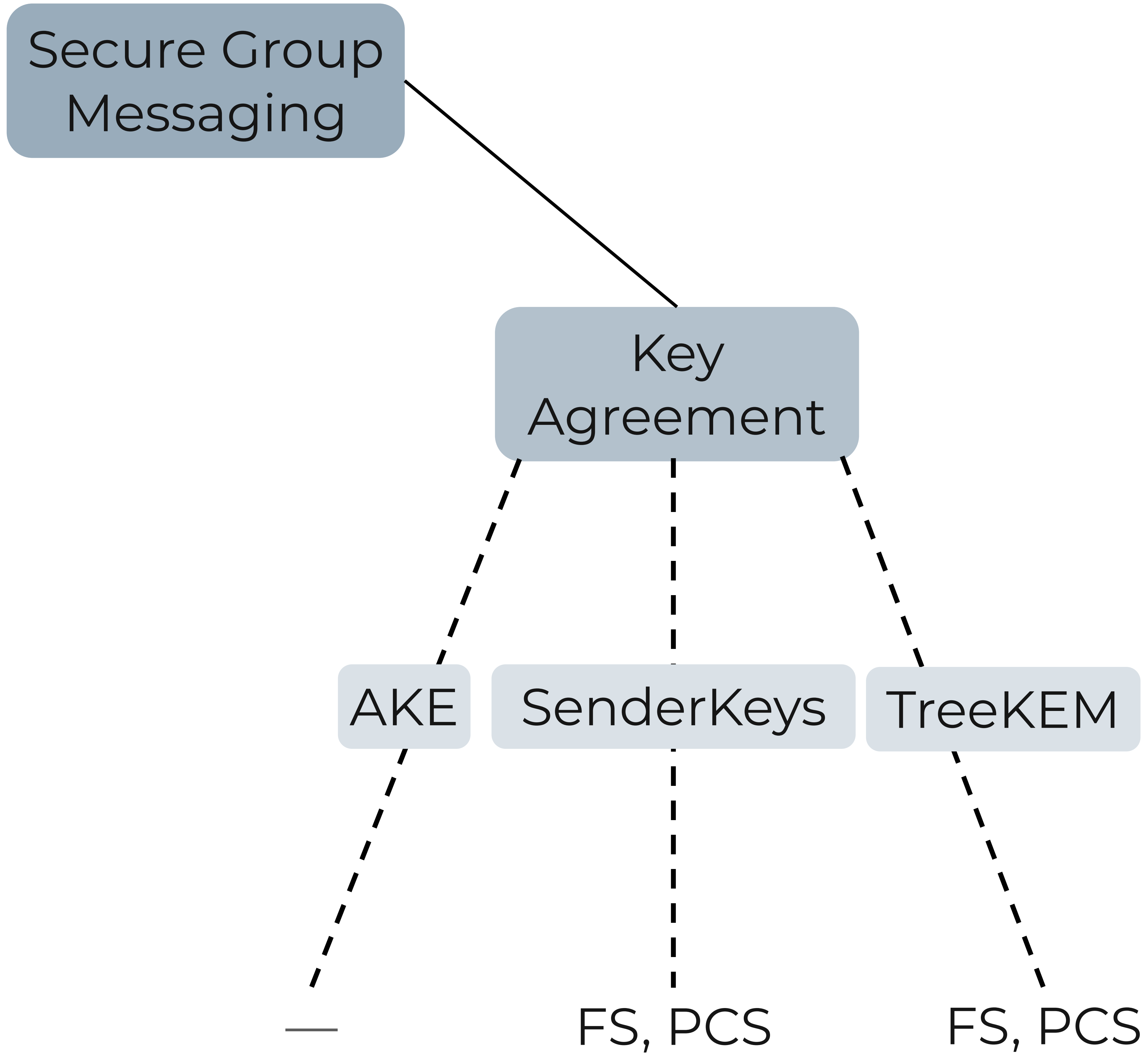
Secure Group Messaging



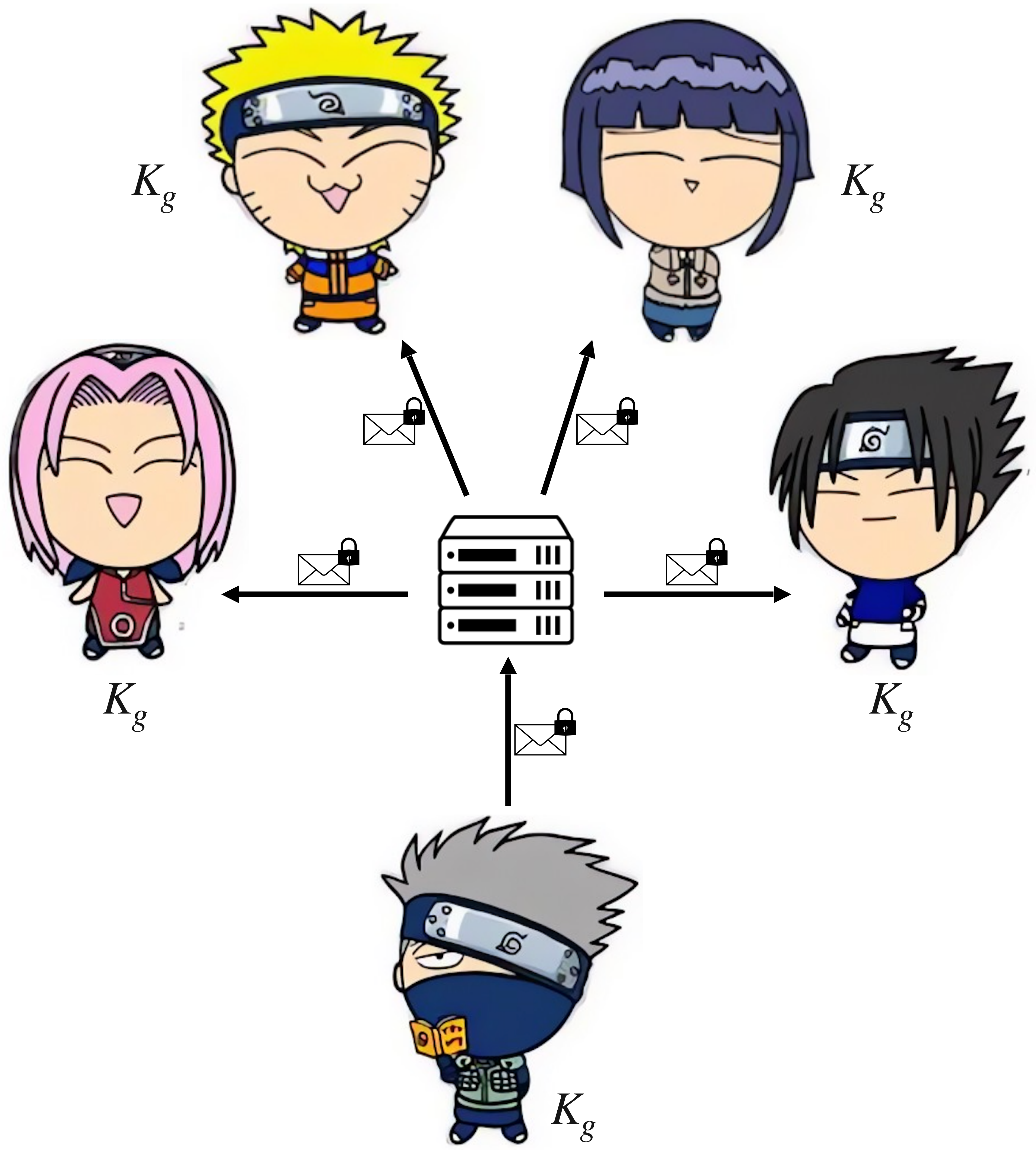


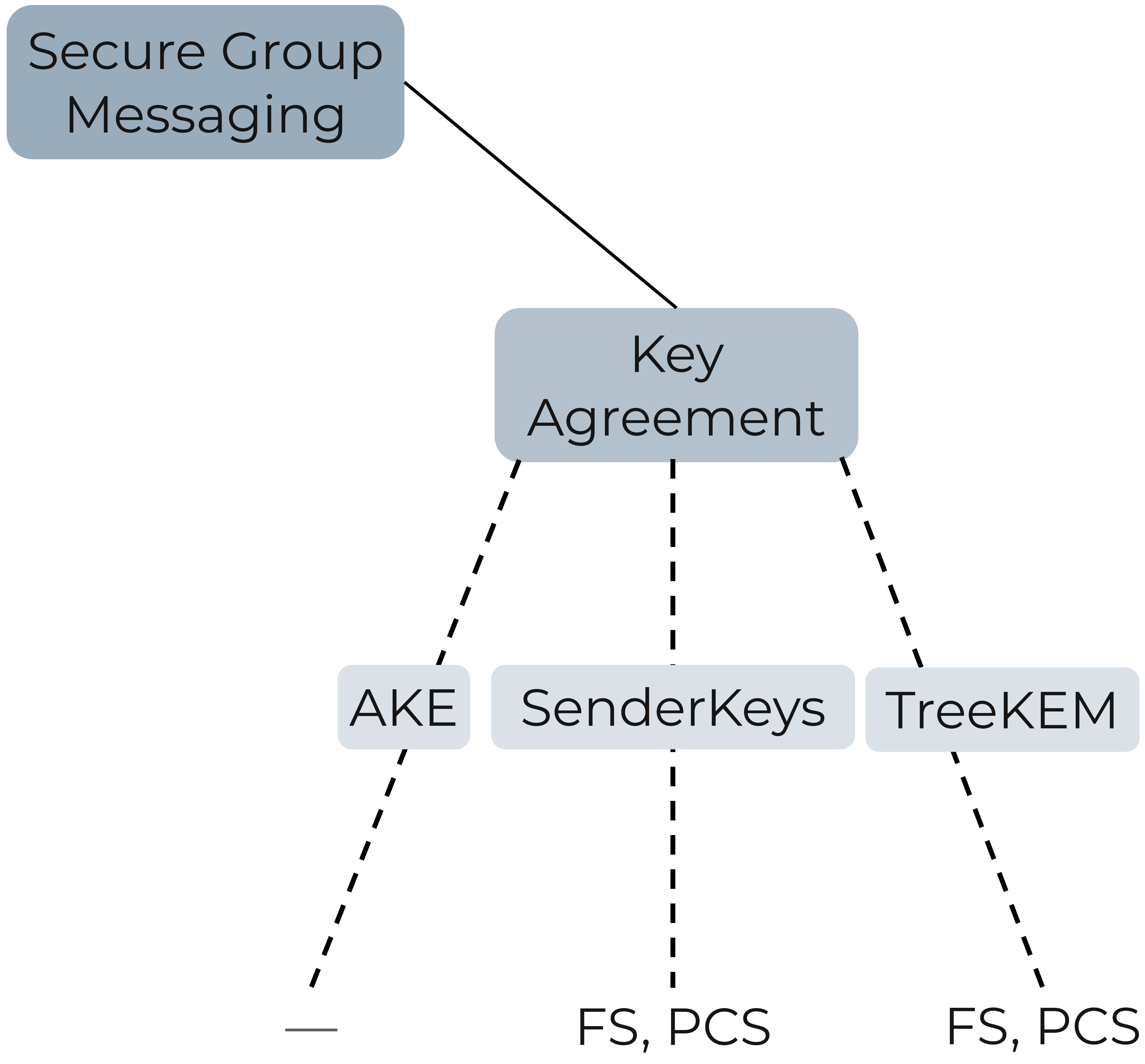
Secure Group Messaging





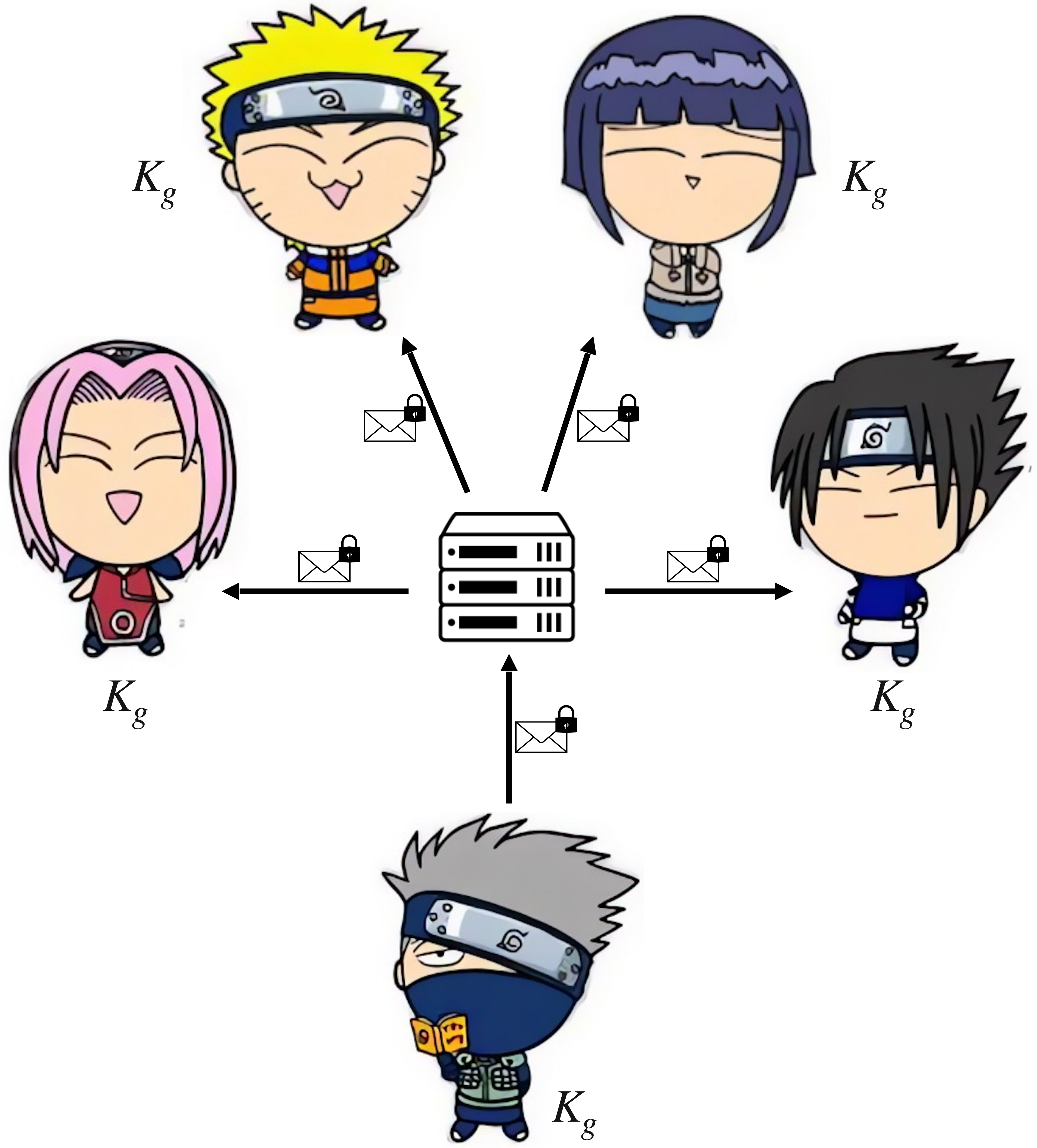
Secure Group Messaging

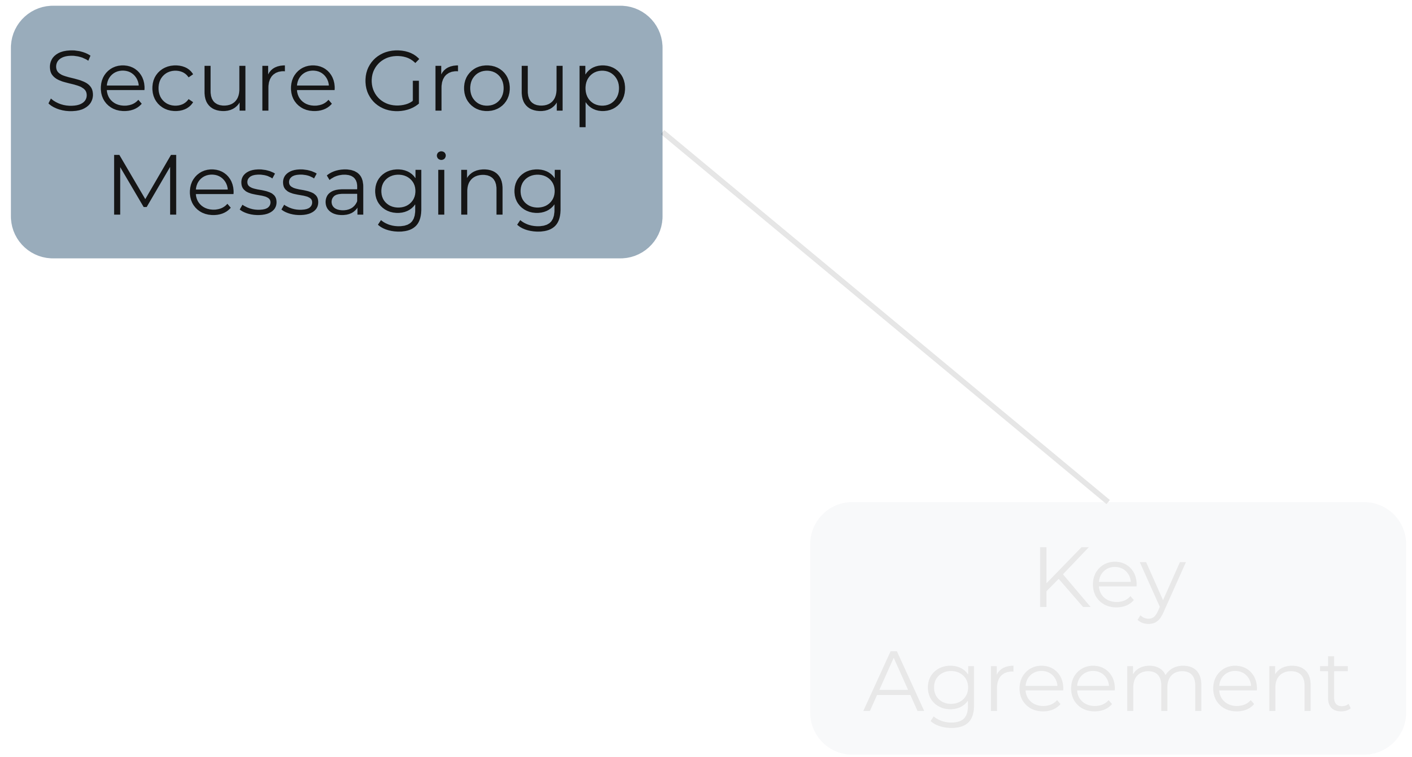




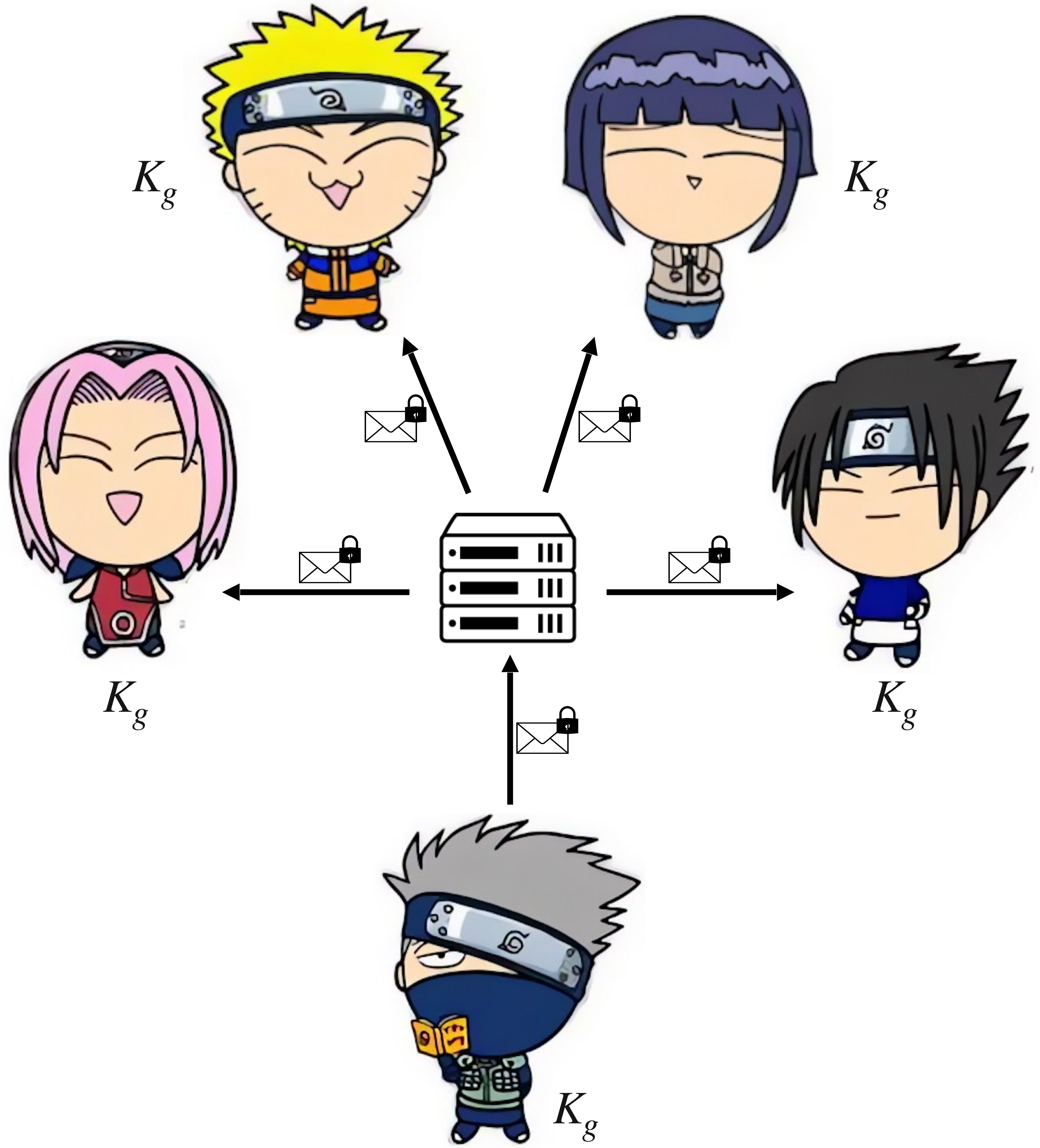
[BBR18, ACDT20, ACJM20, KPPW+21, ACDT21, HKPPW21, AHKM22, ACPP23, AWT23, BCG23, CLM23, ...]

Secure Group Messaging

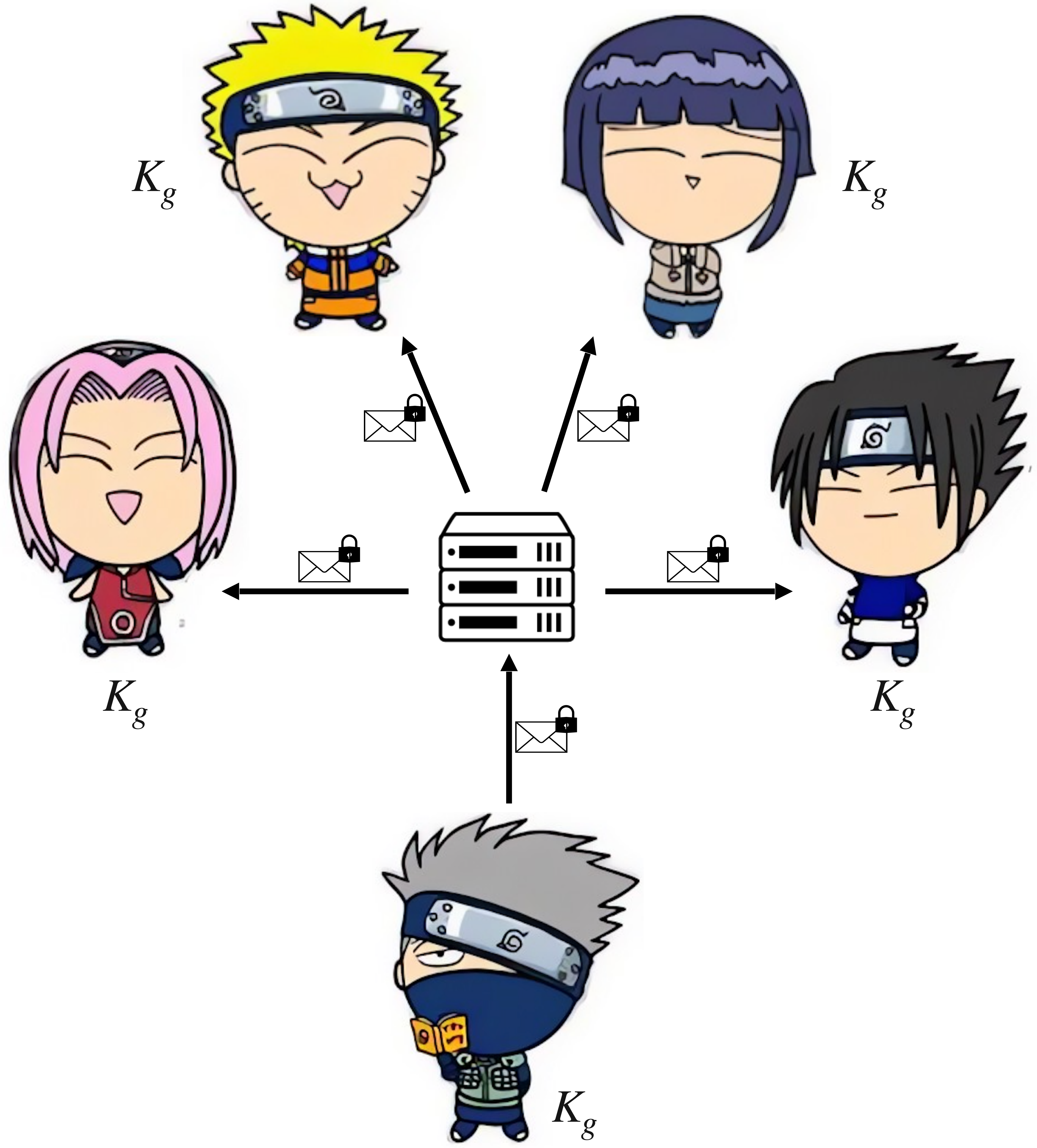


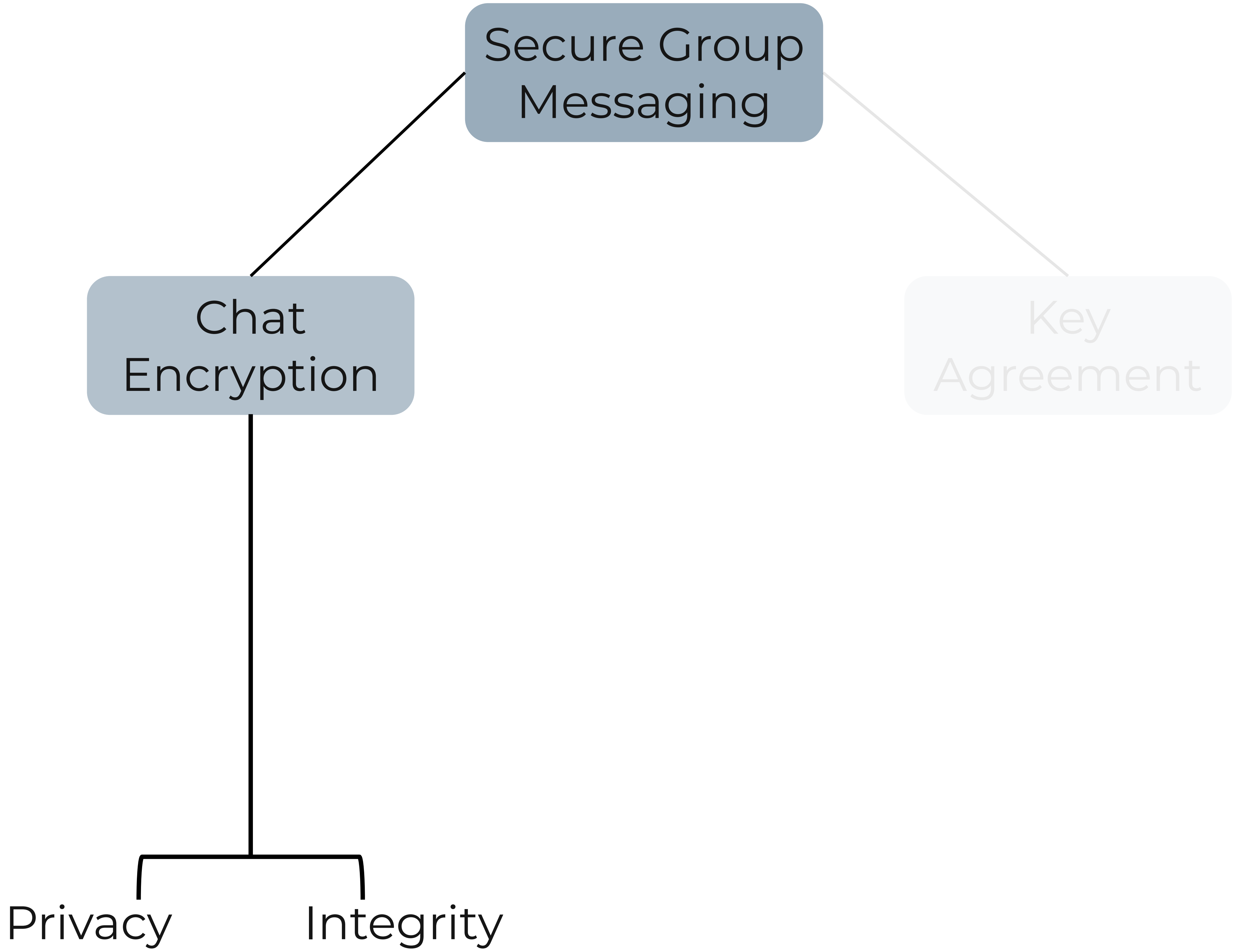


Secure Group Messaging

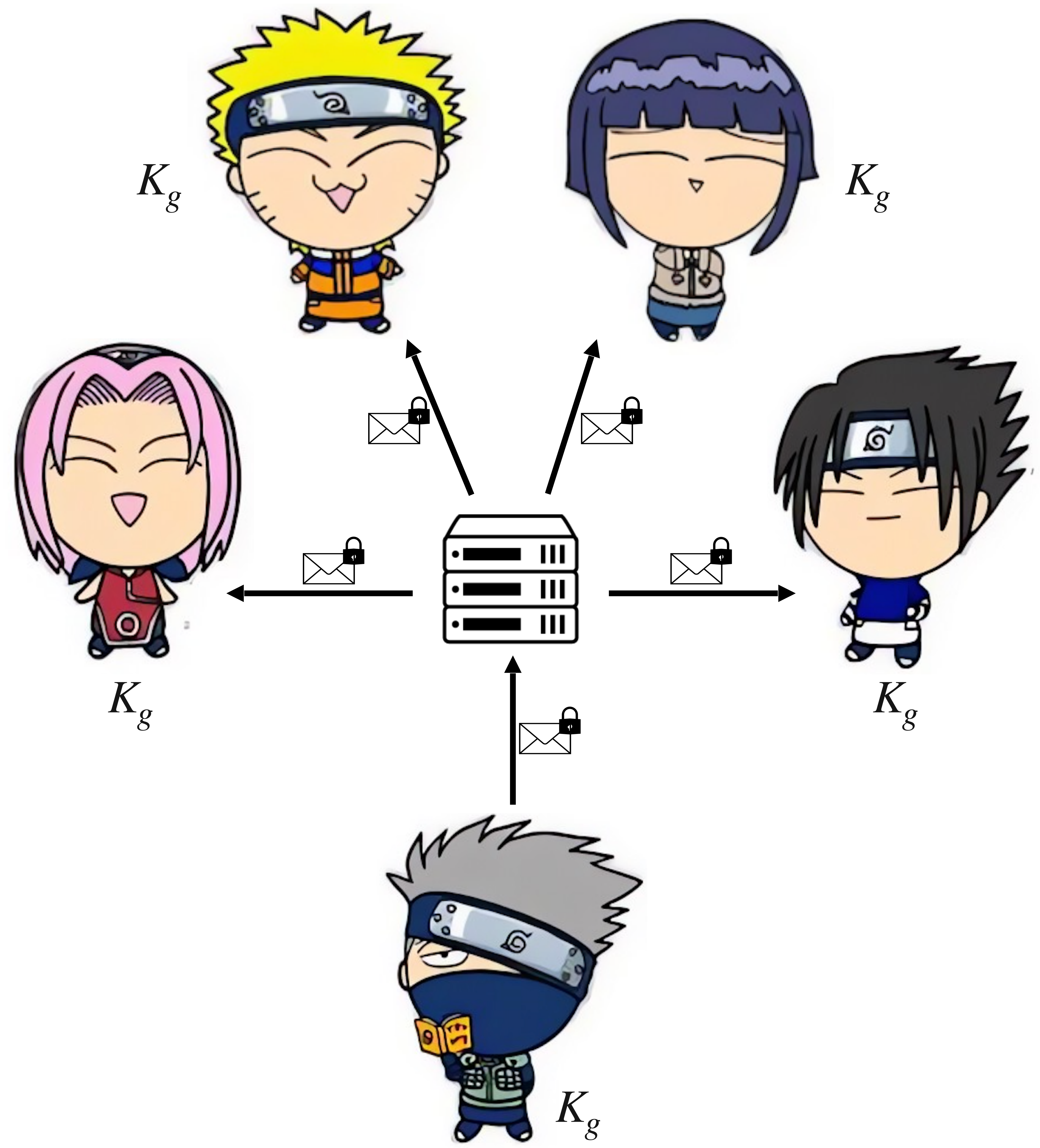


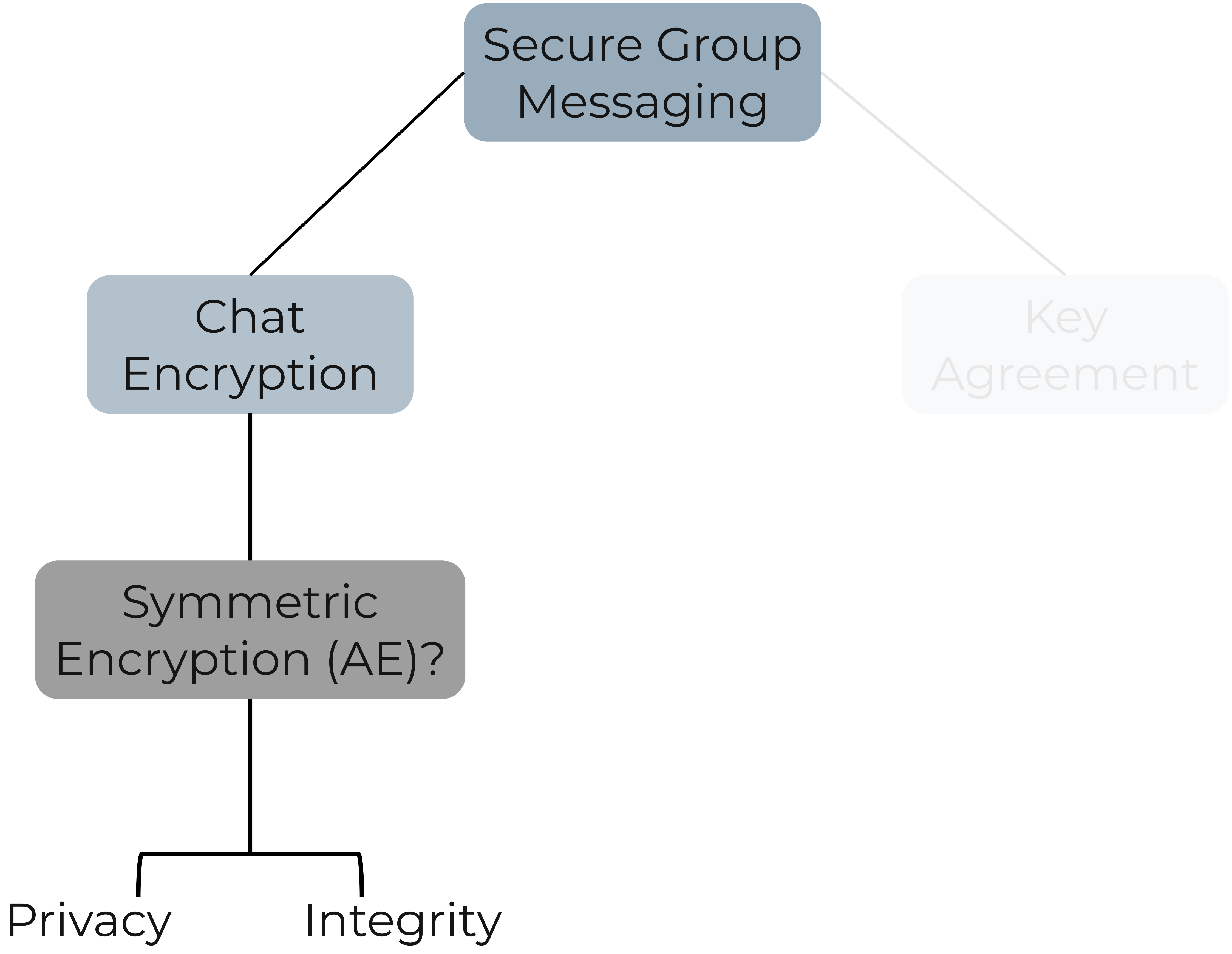
Secure Group Messaging



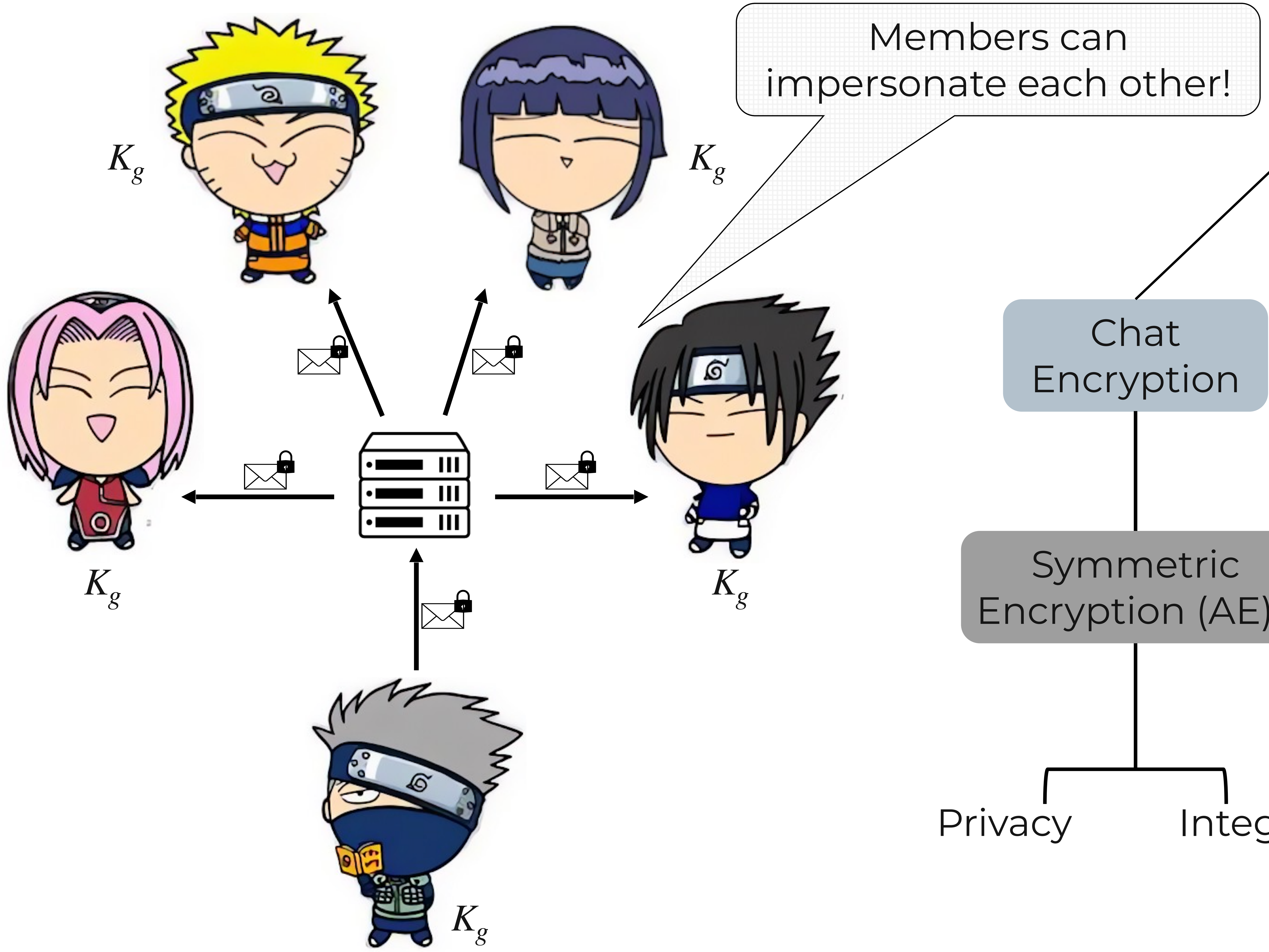


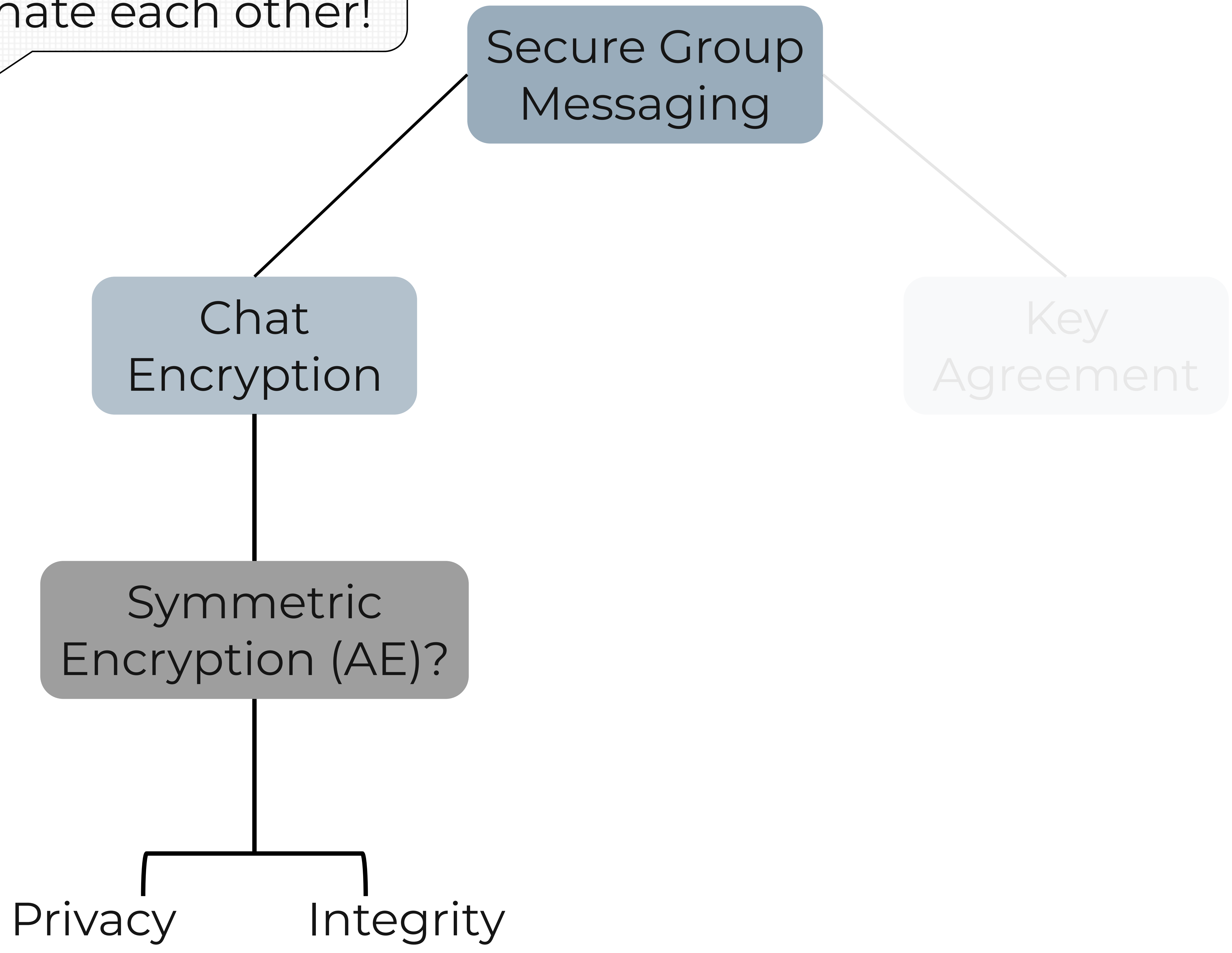
Secure Group Messaging



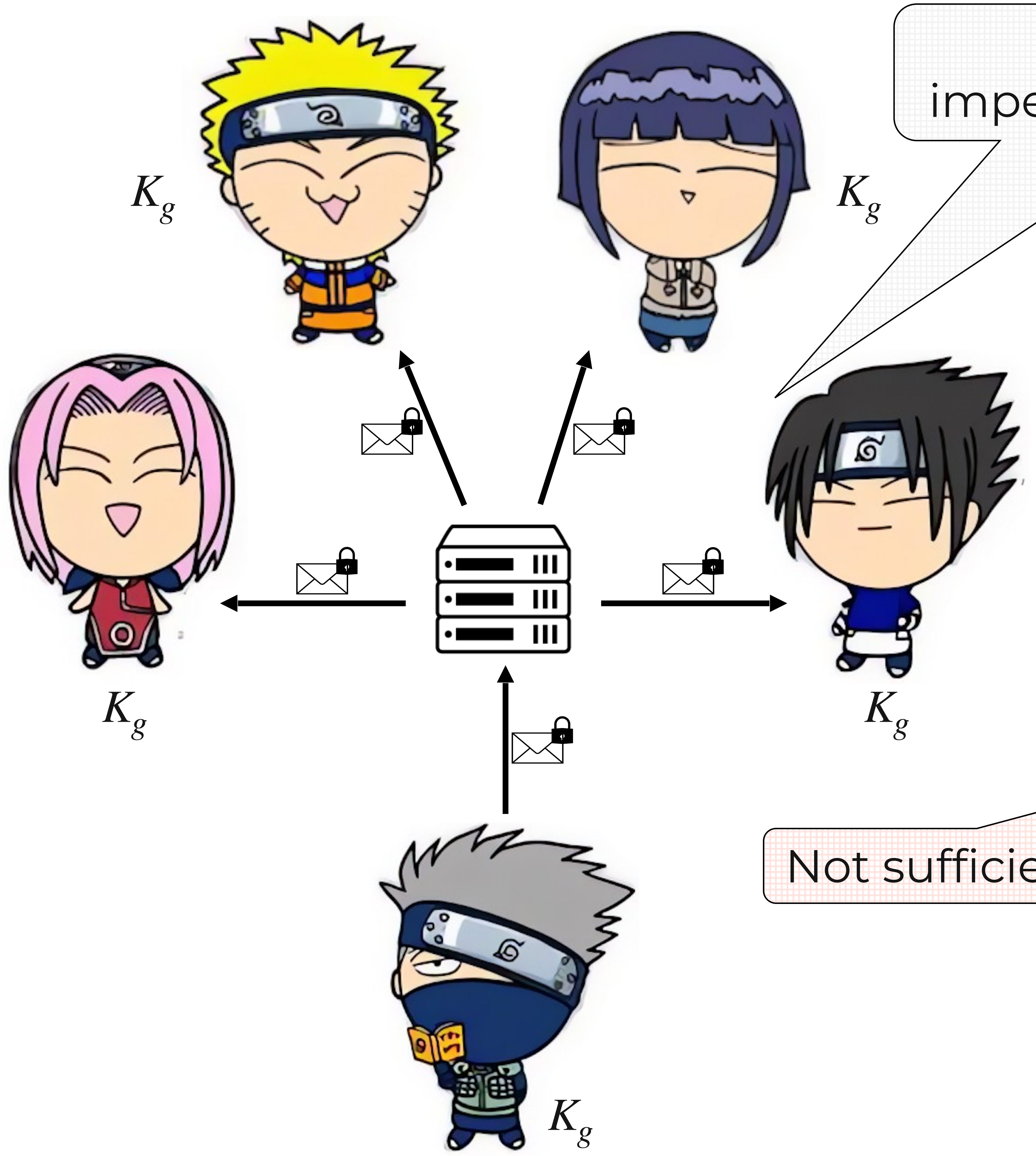


Secure Group Messaging



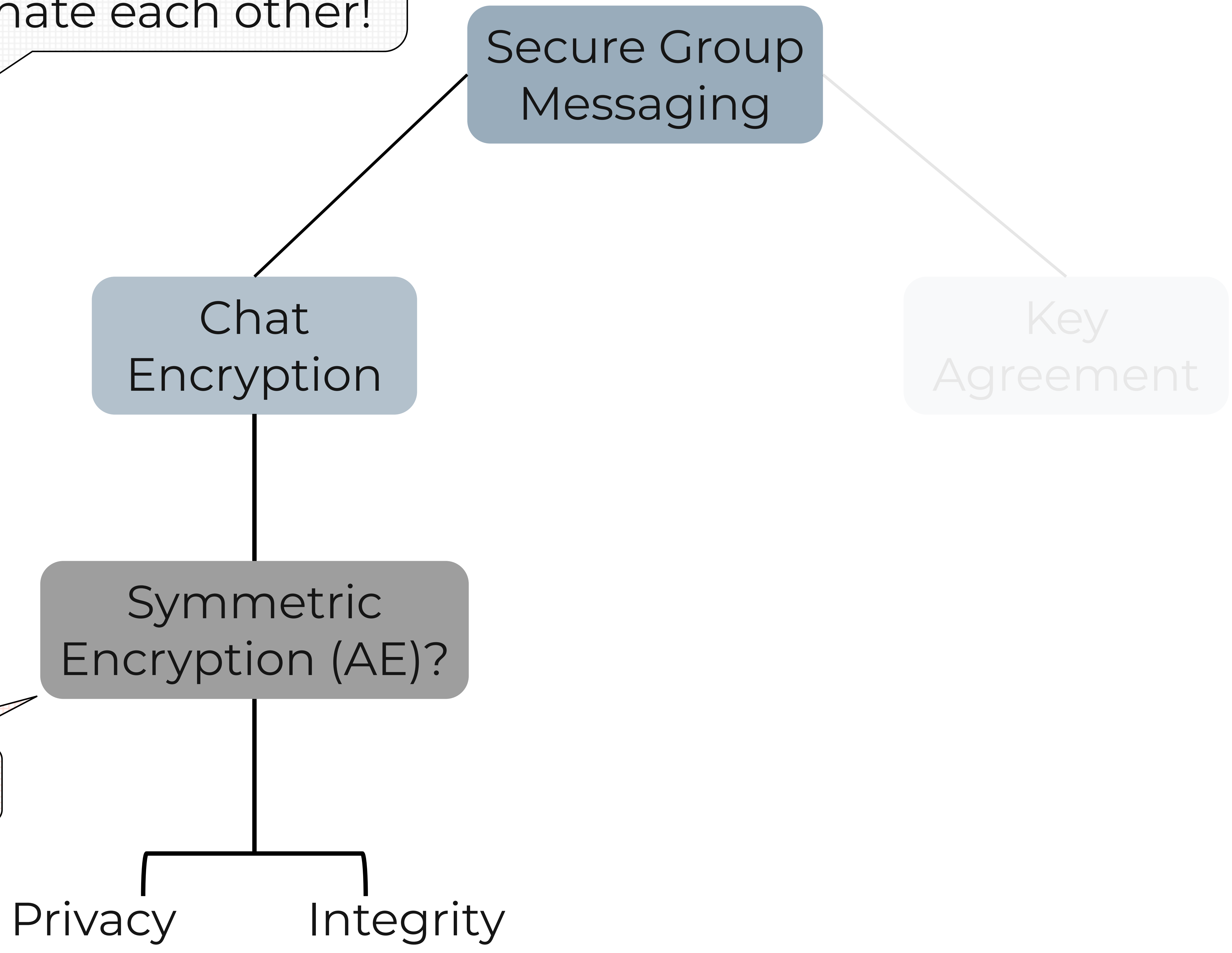


Secure Group Messaging



Members can impersonate each other!

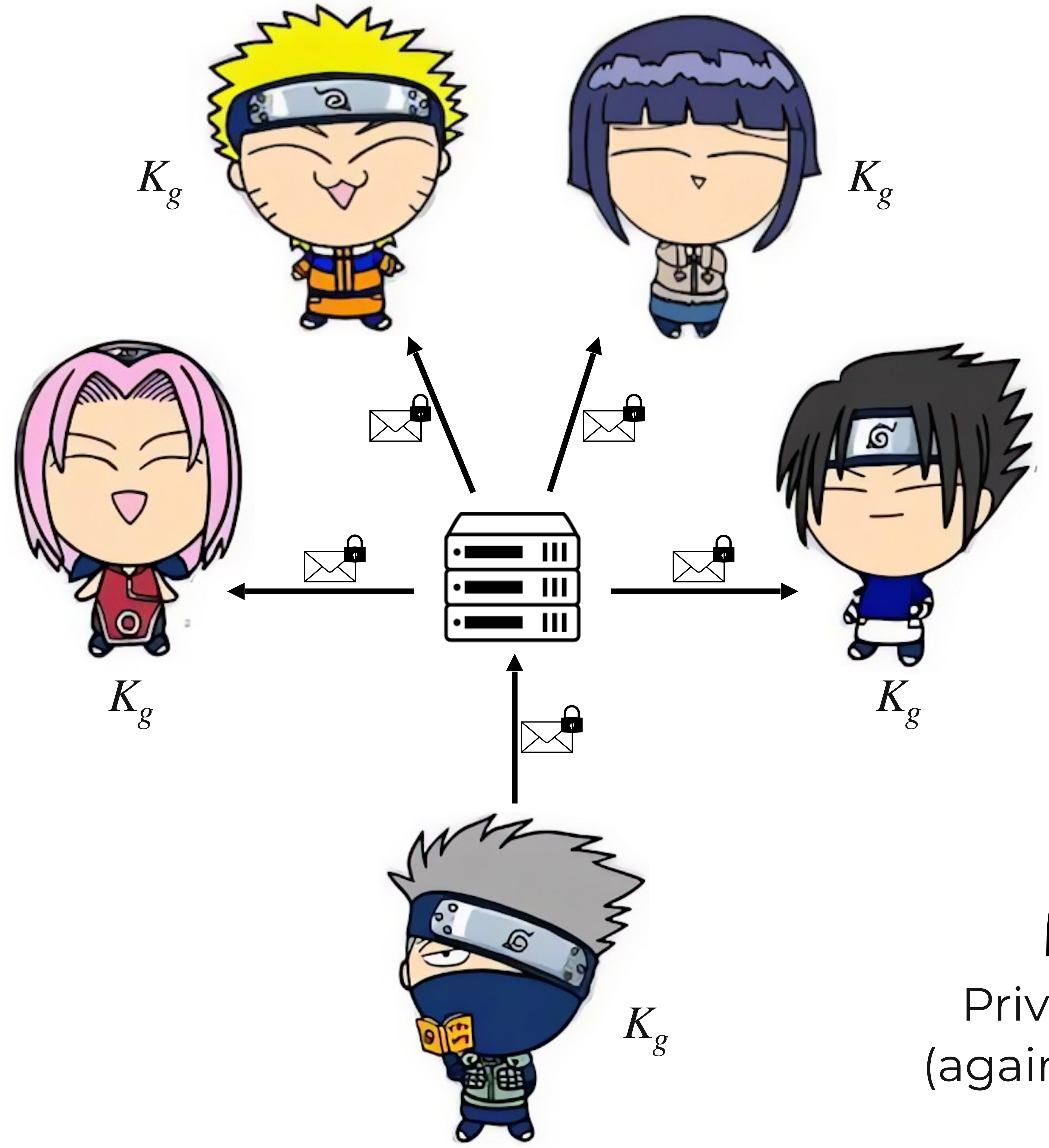
Not sufficient!

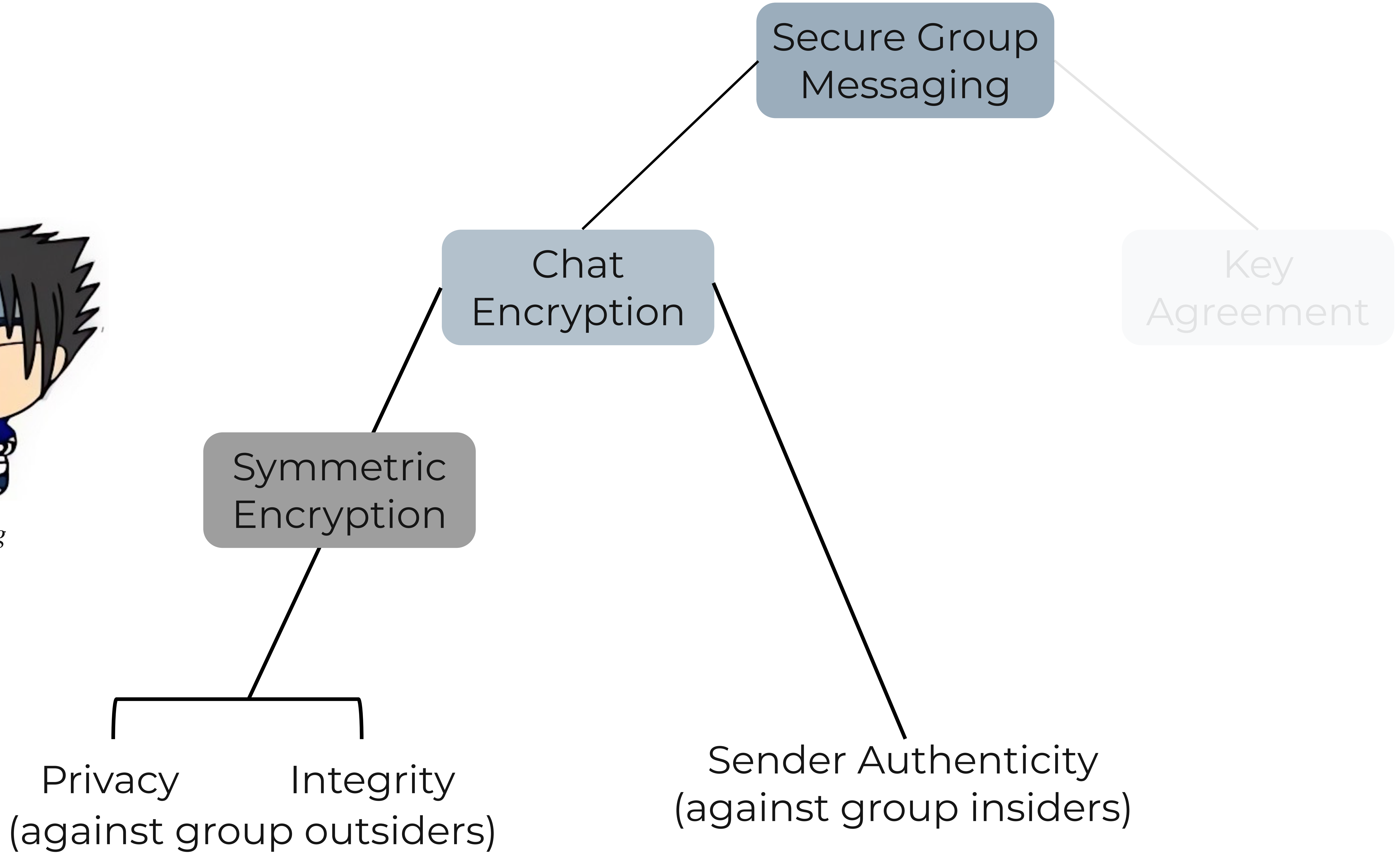


Secure Group Messaging

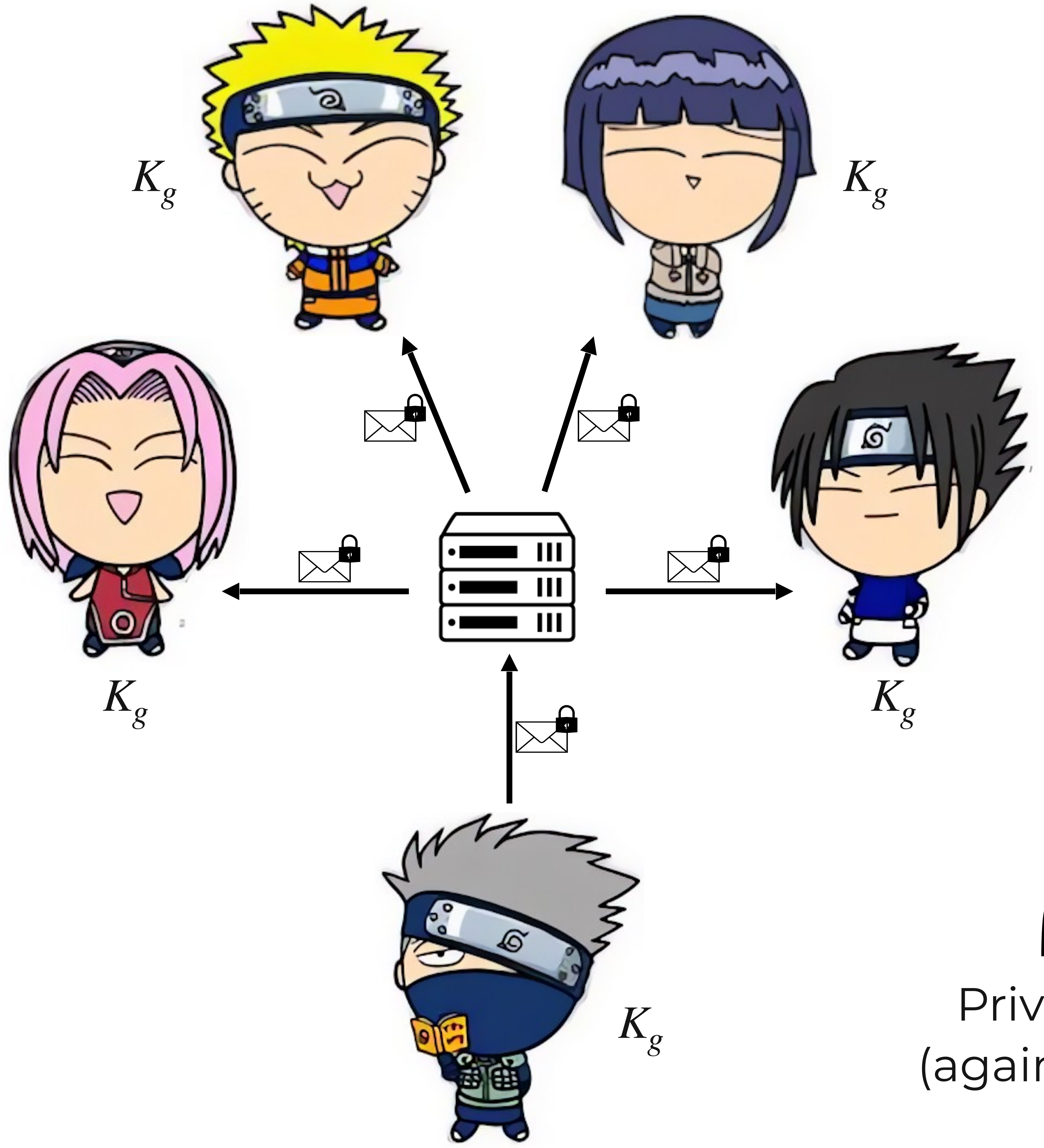
Need integrity against attackers **in the group** i.e. attackers that know K_g

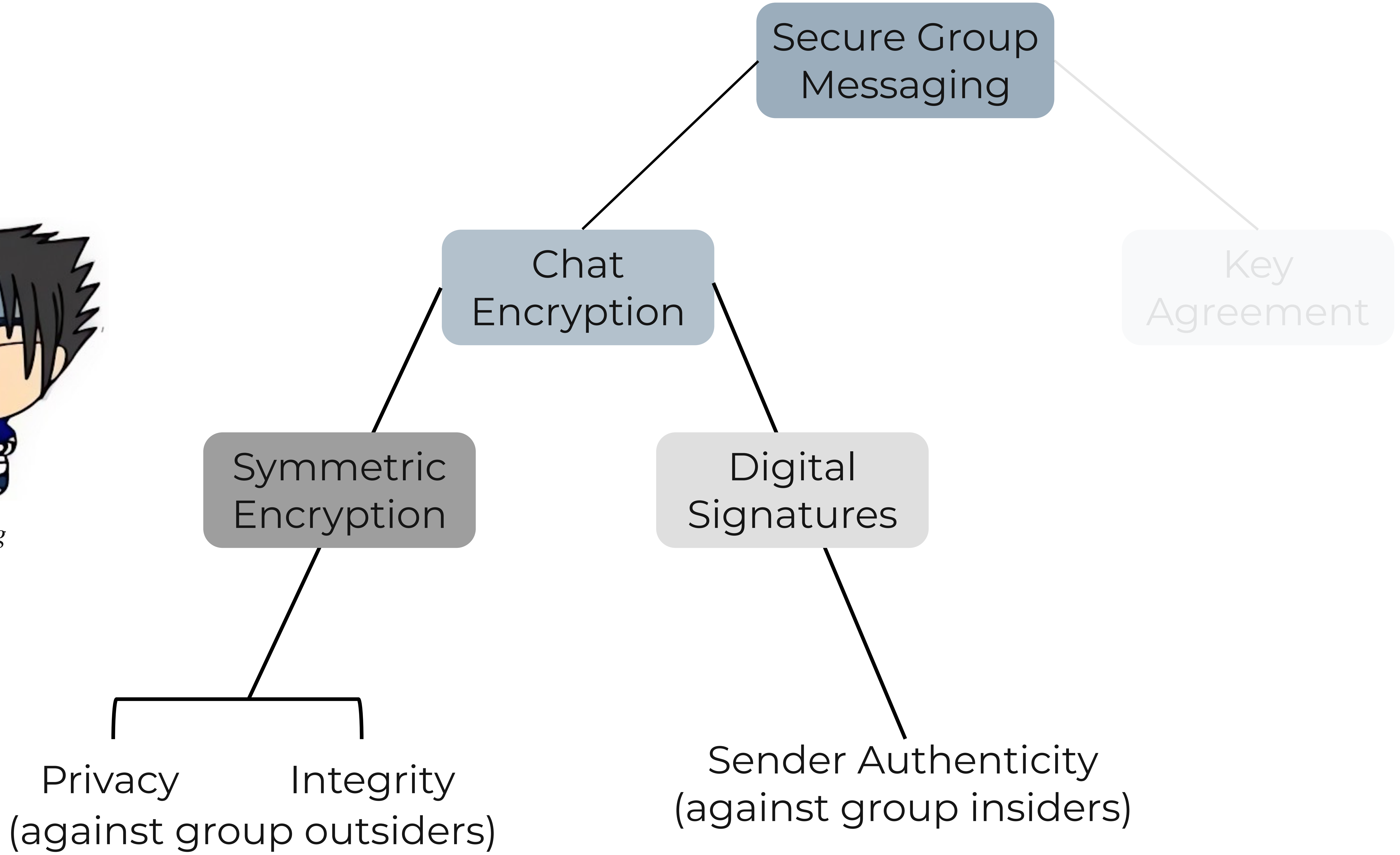
Secure Group Messaging



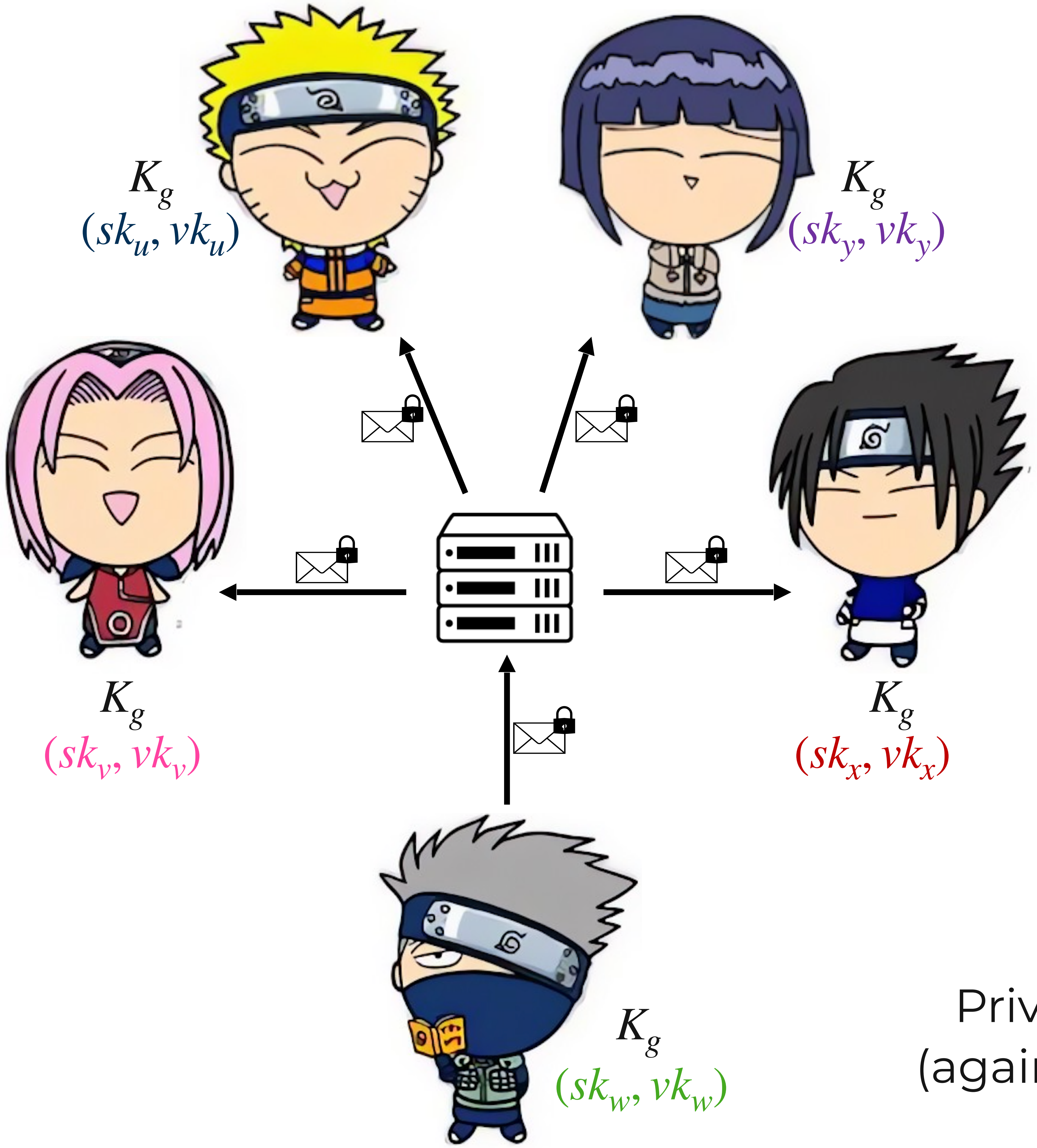


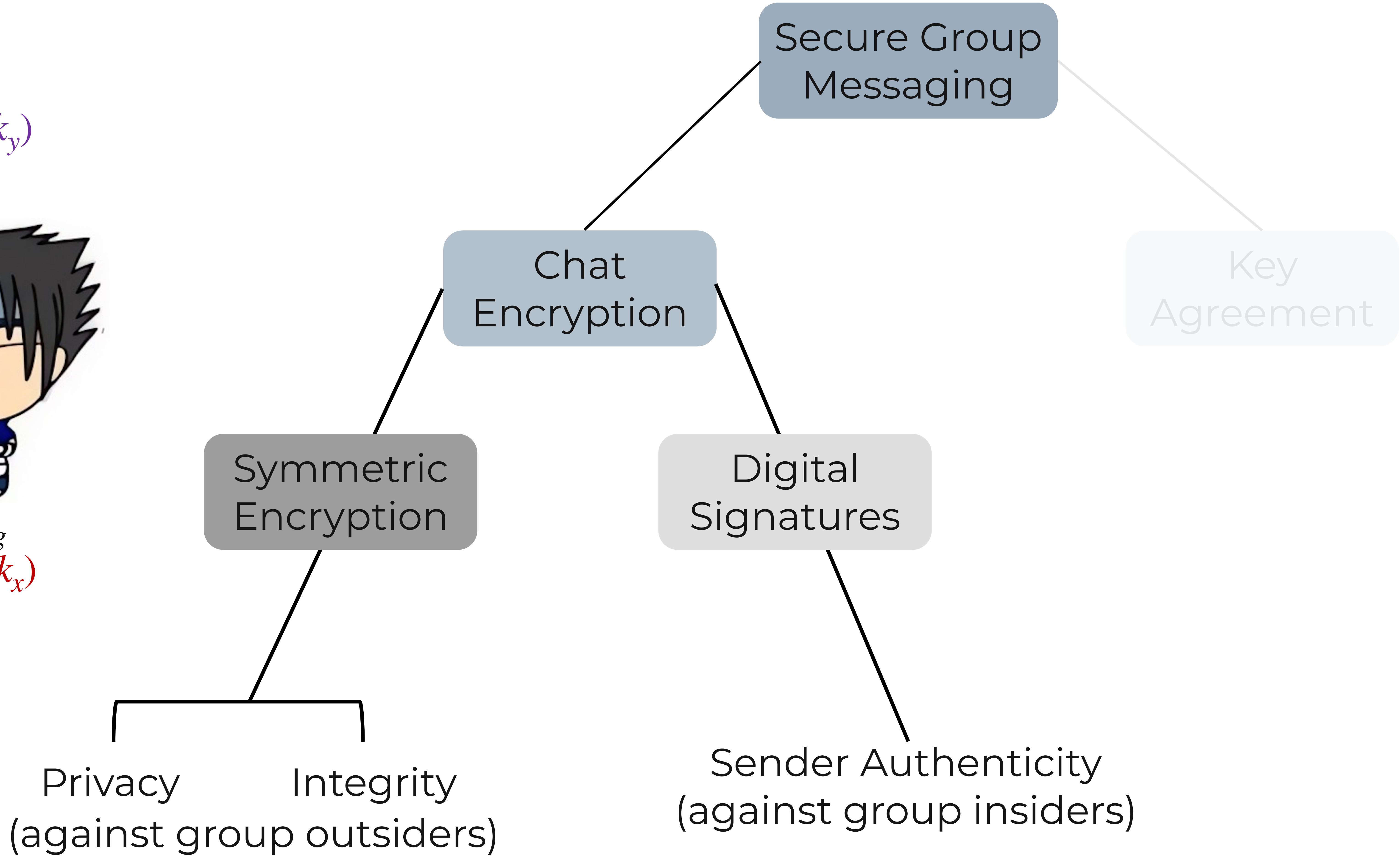
Secure Group Messaging



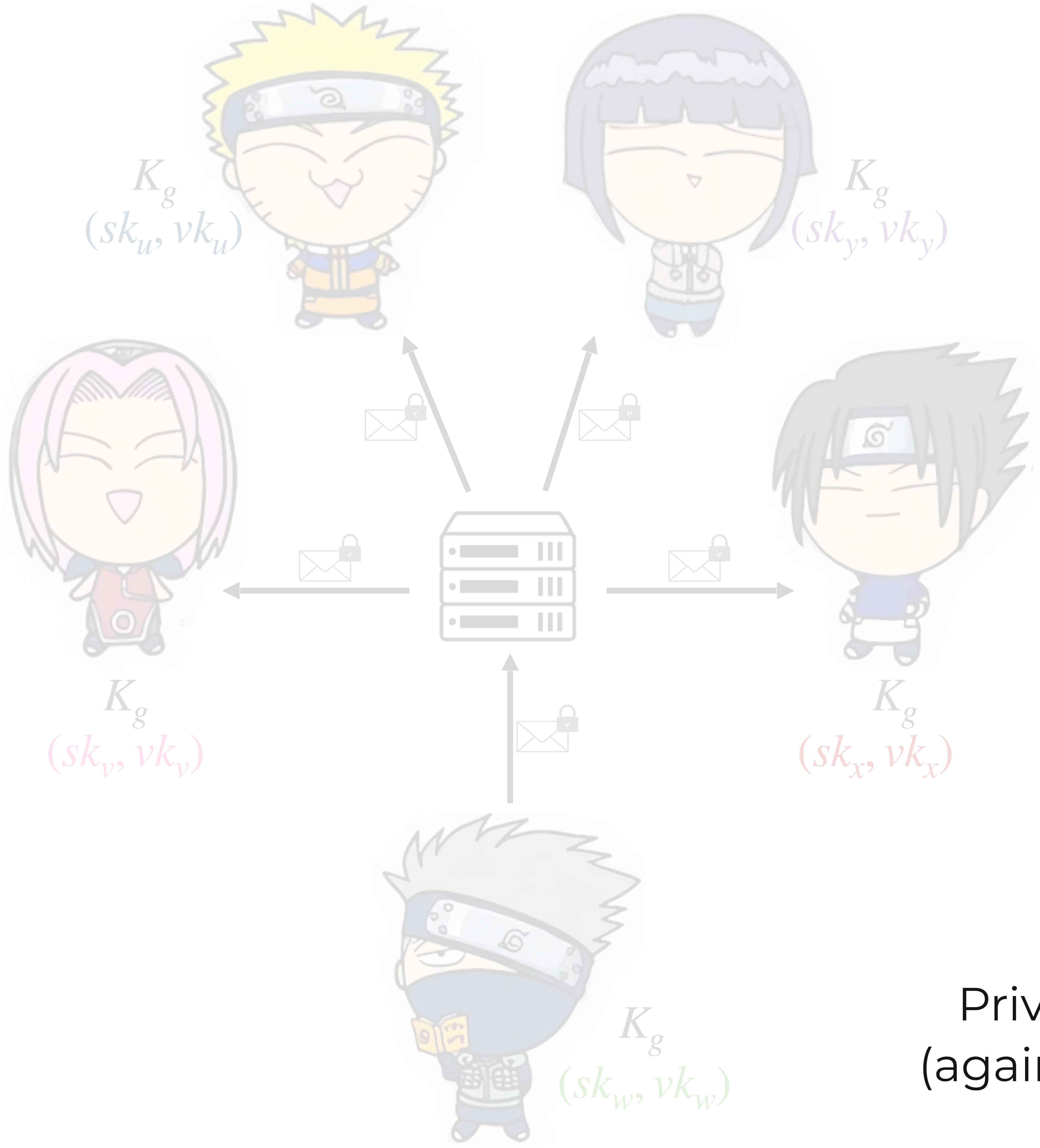


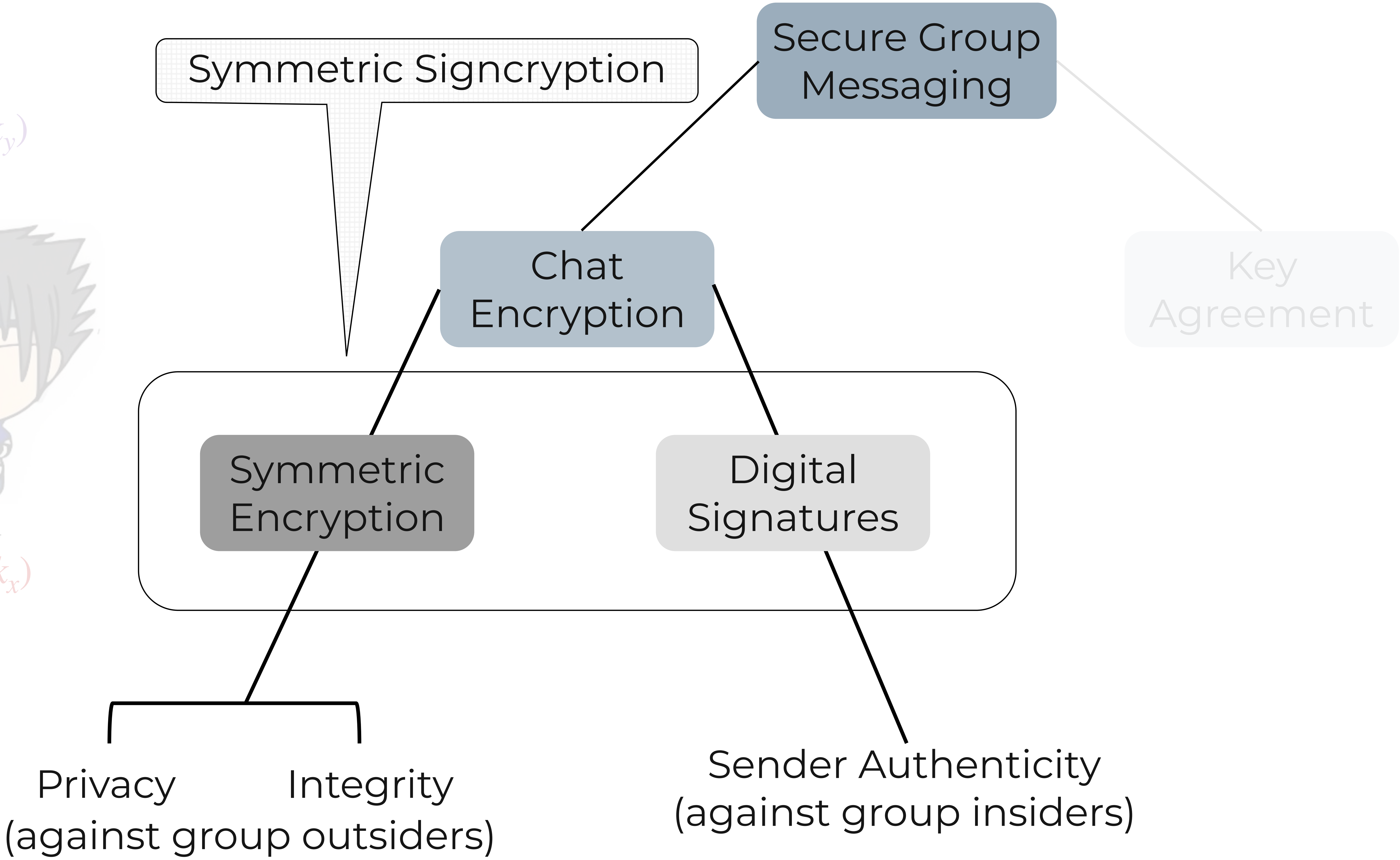
Secure Group Messaging





Secure Group Messaging





Methodology

Methodology

EC24

Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Methodology

EC24

Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Symmetric Signcryption Model

Methodology

EC24

Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Today!

Analyzing Group Chat Encryption in MLS, Session, Signal, and Matrix

Joseph Jaeger and Akshaya Kumar

Symmetric Signcryption Model

Methodology

EC24

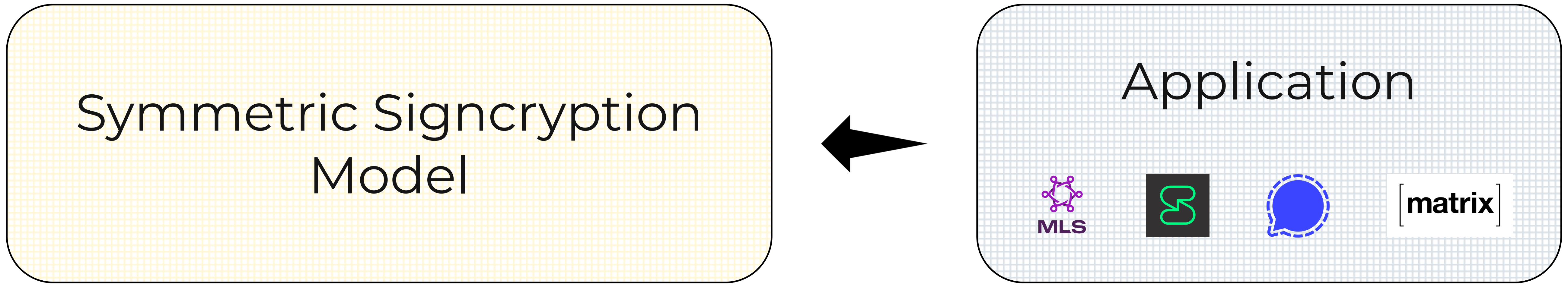
Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Today!

Analyzing Group Chat Encryption in MLS, Session, Signal, and Matrix

Joseph Jaeger and Akshaya Kumar



Methodology

EC24

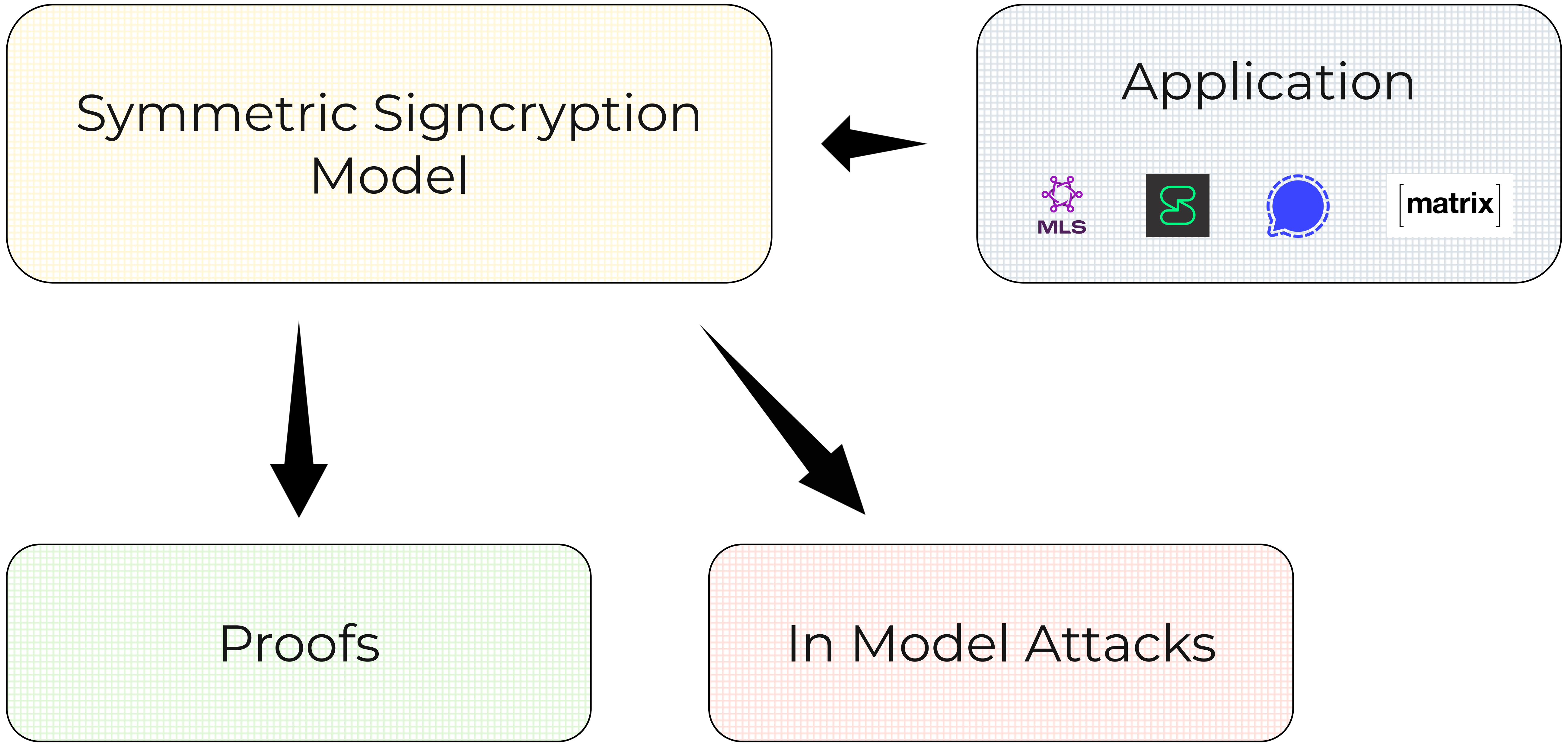
Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Today!

Analyzing Group Chat Encryption in MLS, Session, Signal, and Matrix

Joseph Jaeger and Akshaya Kumar



Methodology

EC24

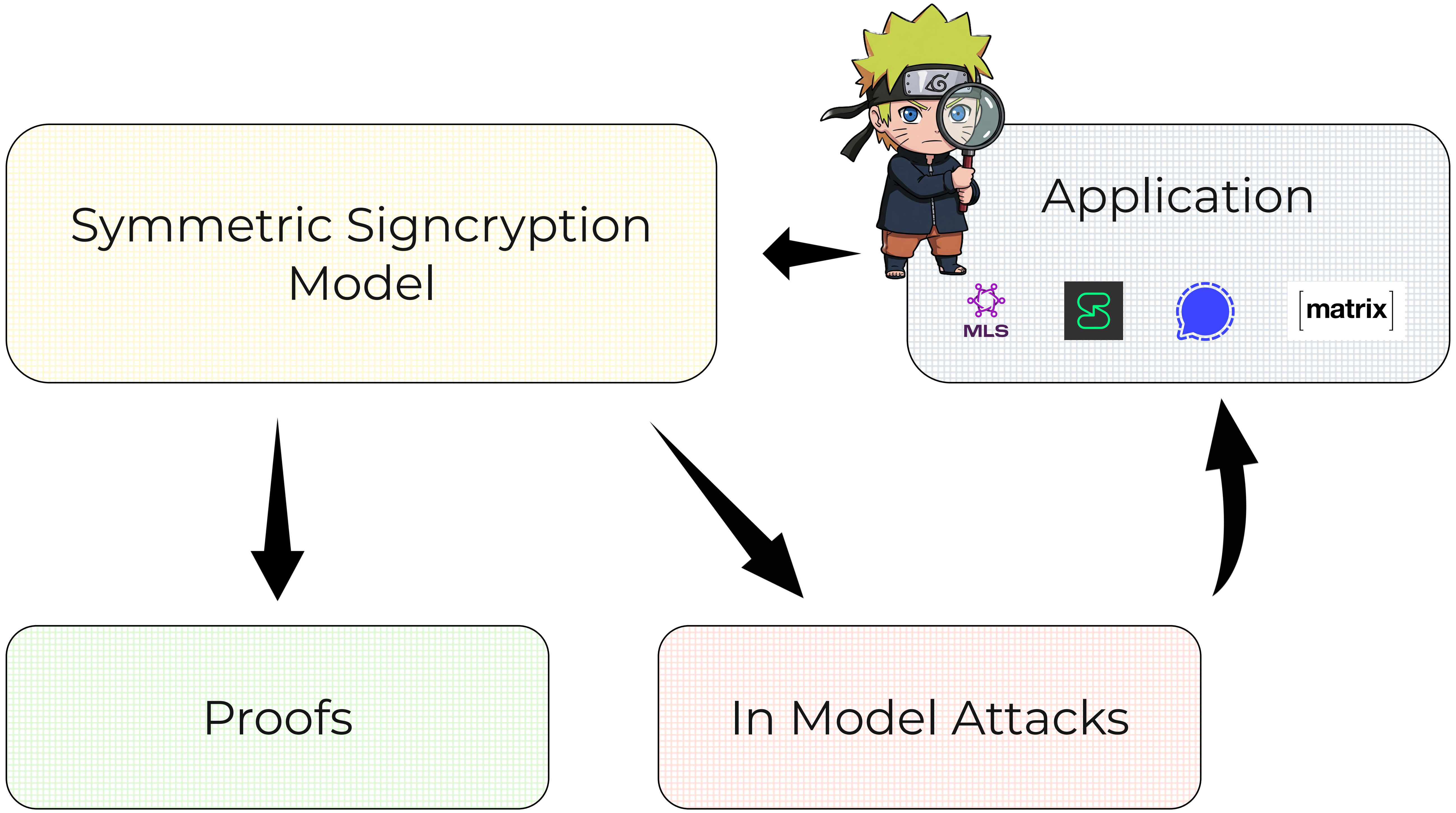
Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

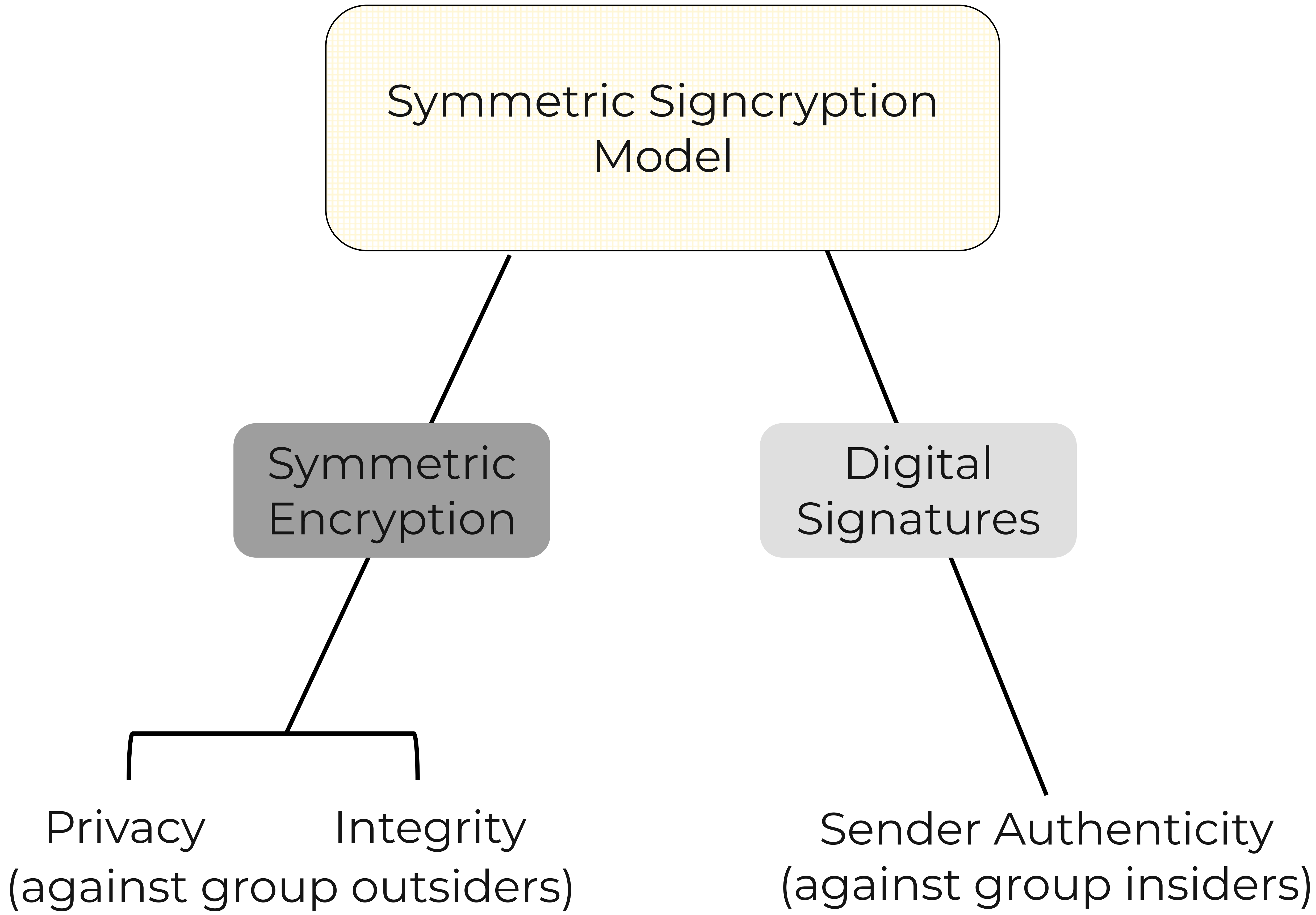
Today!

Analyzing Group Chat Encryption in MLS, Session, Signal, and Matrix

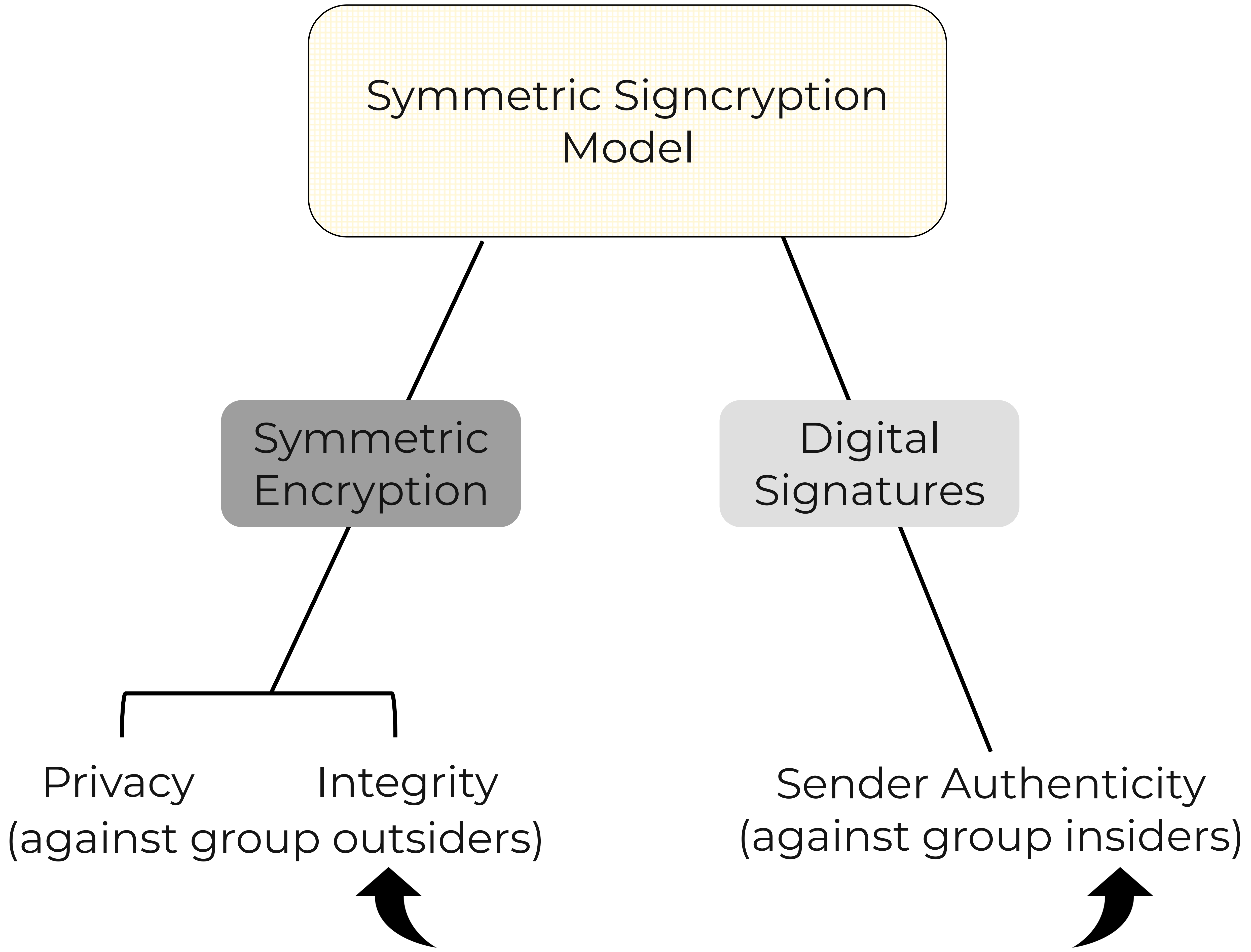
Joseph Jaeger and Akshaya Kumar



Insider and Outsider Attacks



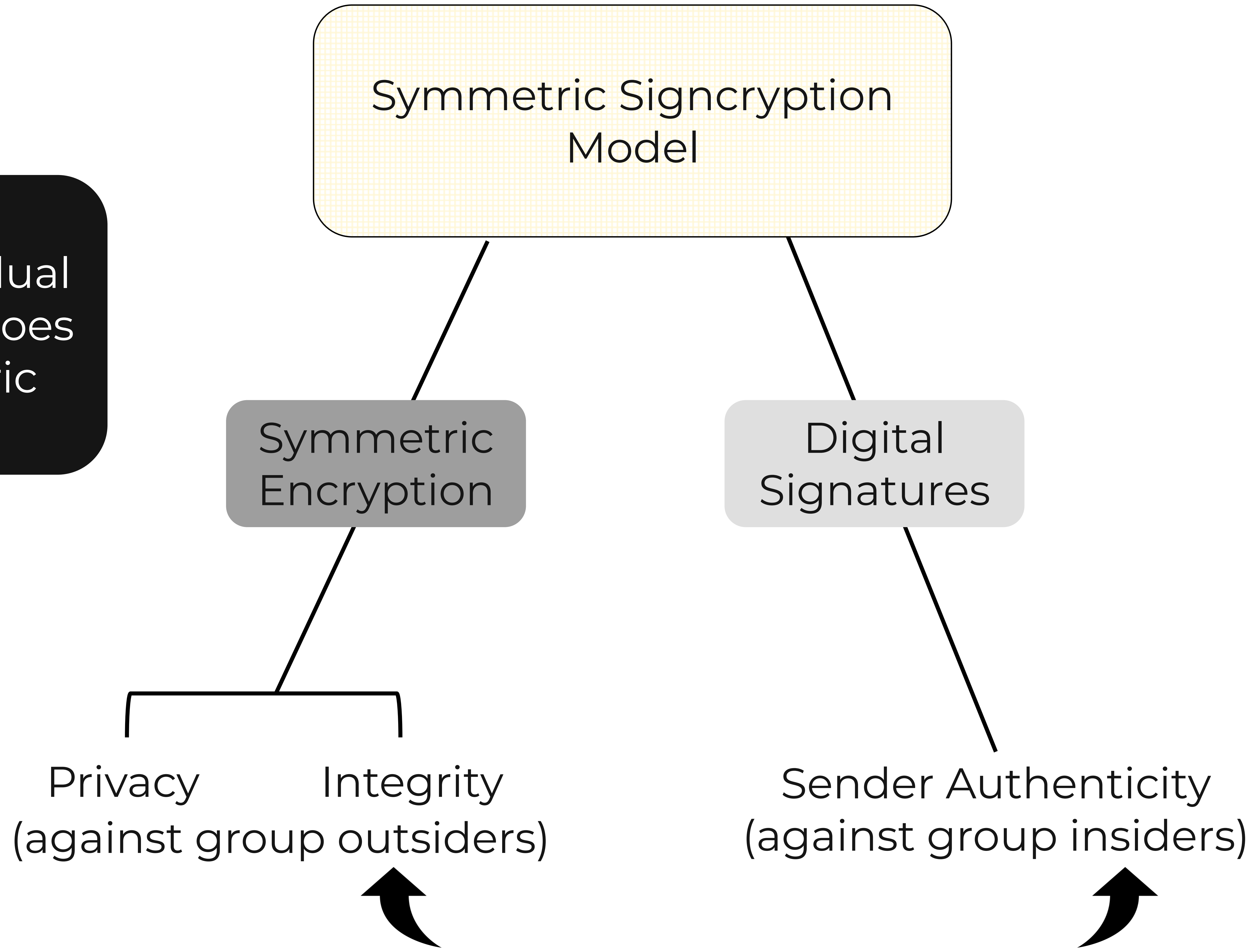
Insider and Outsider Attacks



Natural to think about security against insiders and outsiders

Insider and Outsider Attacks

Outsider
May compromise individual users' signing keys but does not know the symmetric group key



Natural to think about security against insiders and outsiders

Insider and Outsider Attacks

Symmetric Signcryption Model

Outsider
May compromise individual users' signing keys but does not know the symmetric group key

Insider
Is part of the group chat and knows the symmetric group key but not the signing key of anyone but themselves

Symmetric Encryption

Digital Signatures

Privacy Integrity
(against group outsiders)

Sender Authenticity
(against group insiders)

Natural to think about security against insiders and outsiders

Insider and Outsider Attacks

Symmetric Signcryption Model

Symmetric Encryption

Privacy Integrity
(against group outsiders)

Digital Signatures

Sender Authenticity
(against group insiders)

__Outsider__
May compromise individual users' signing keys but does not know the symmetric group key

Interesting attacks

__Insider__
Is part of the group chat and knows the symmetric group key but not the signing key of anyone but themselves

Natural to think about security against insiders and outsiders

Naive Constructions

Naive Constructions

Sign-then-Encrypt (StE)

Naive Constructions

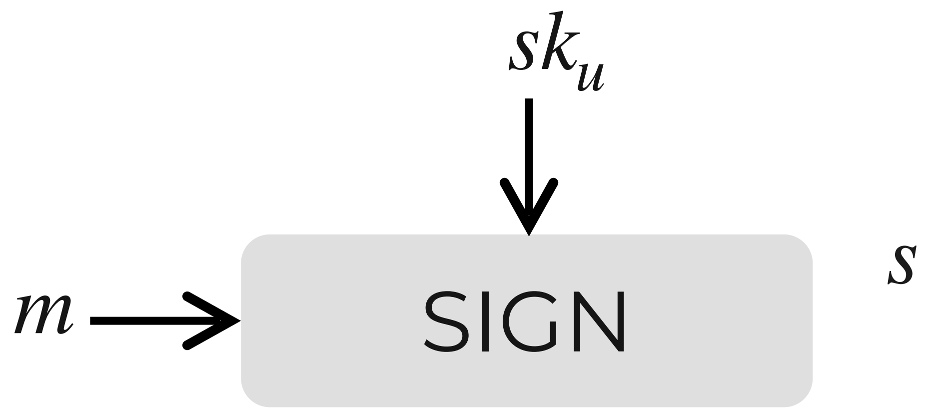
Sign-then-Encrypt (StE)

Encrypt-then-Sign (EtS)

Naive Constructions

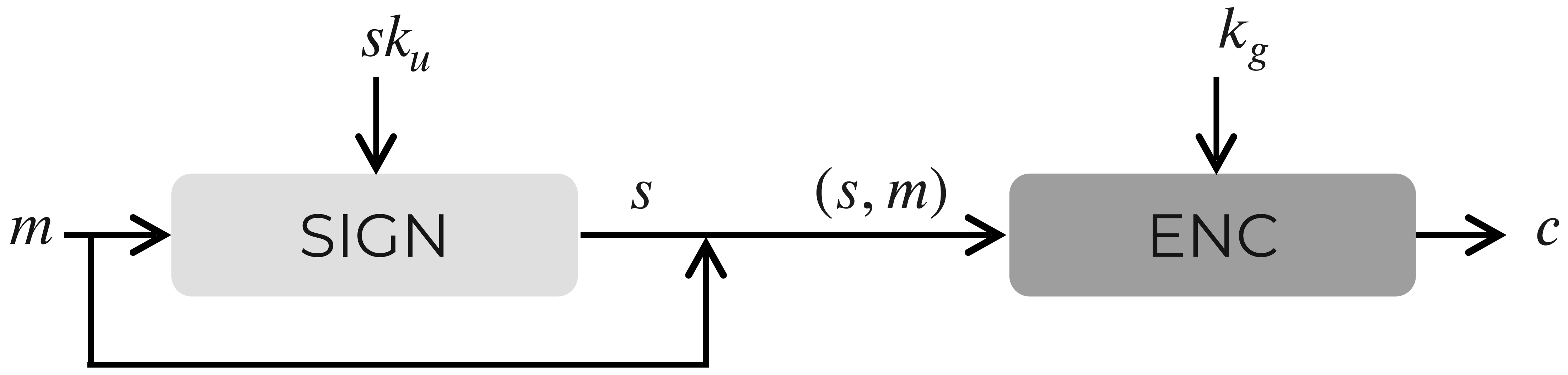
Sign-then-Encrypt (StE)

Encrypt-then-Sign (EtS)



Naive Constructions

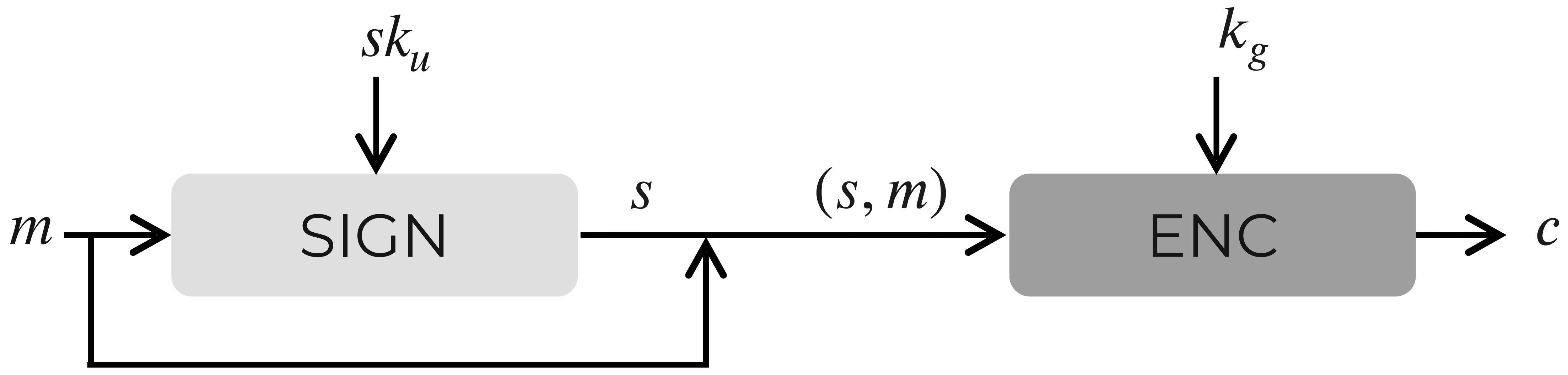
Sign-then-Encrypt (StE)



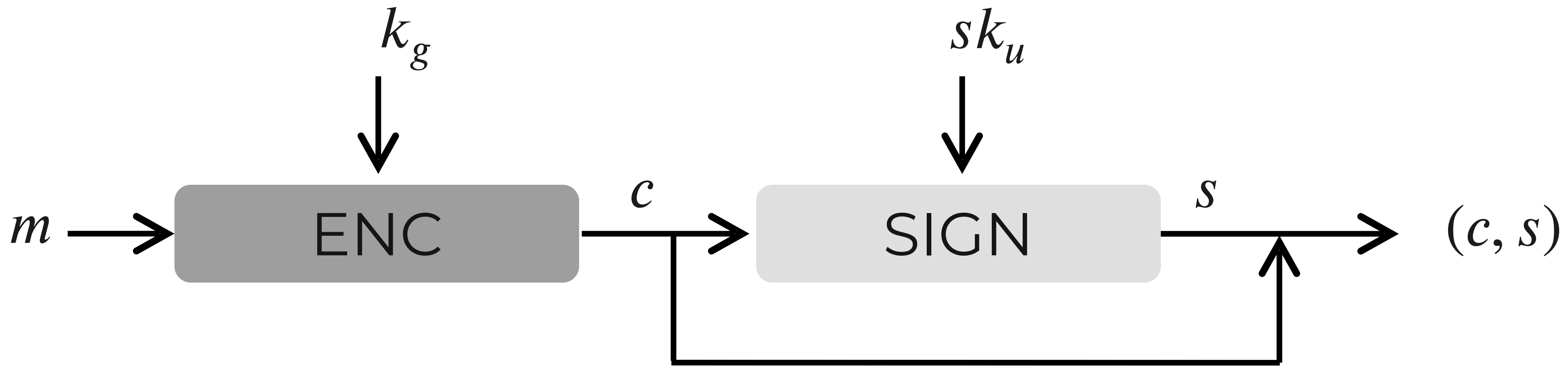
Encrypt-then-Sign (EtS)

Naive Constructions

Sign-then-Encrypt (StE)

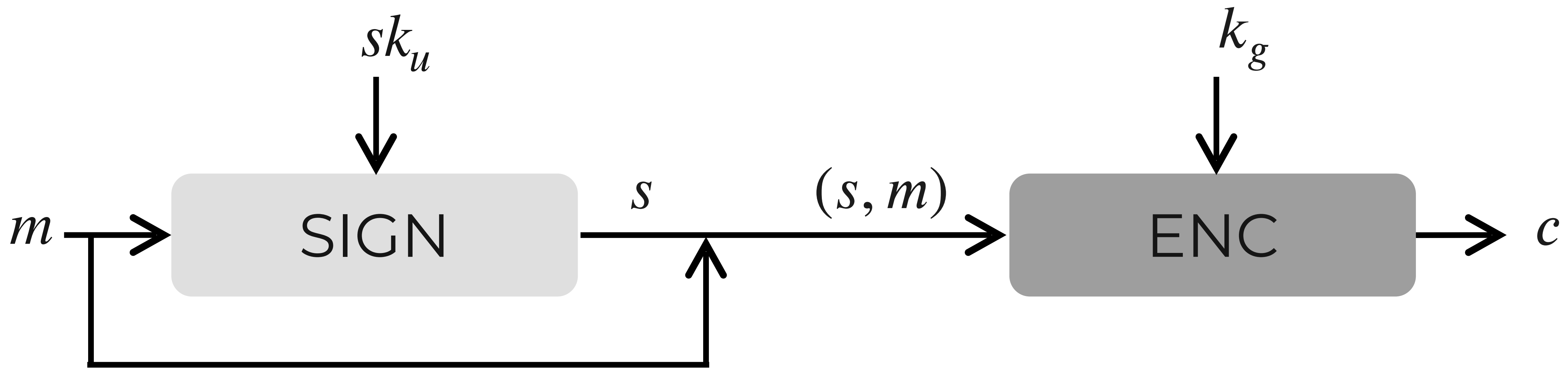


Encrypt-then-Sign (EtS)

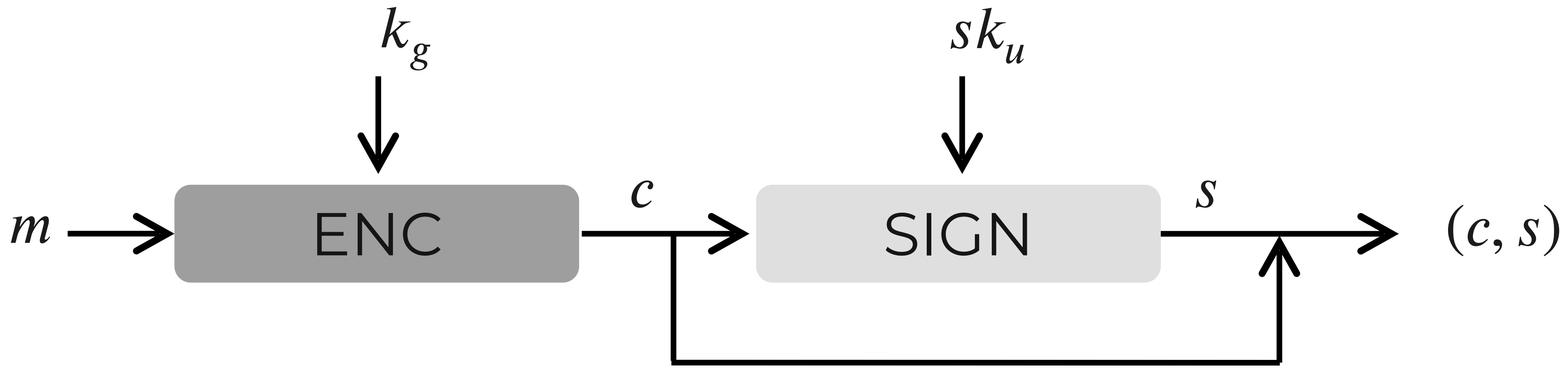


Naive Constructions

Sign-then-Encrypt (StE)

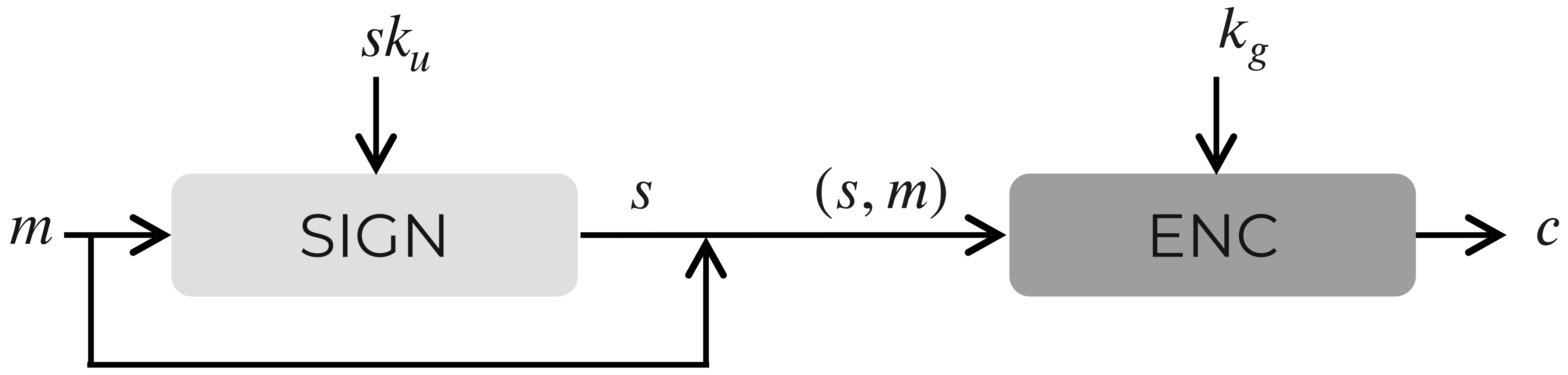


Encrypt-then-Sign (EtS)

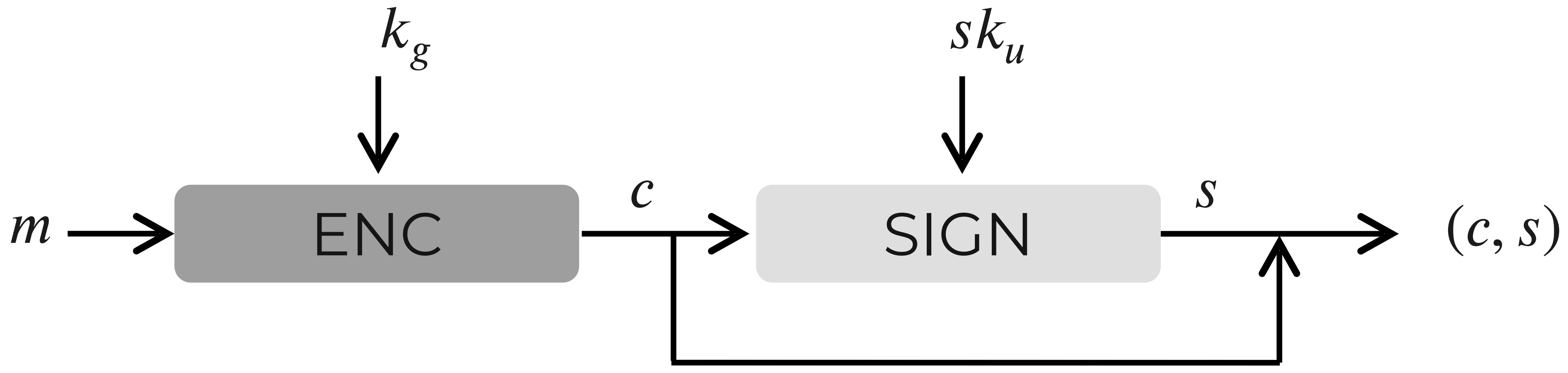


Naive Constructions

Sign-then-Encrypt (StE)

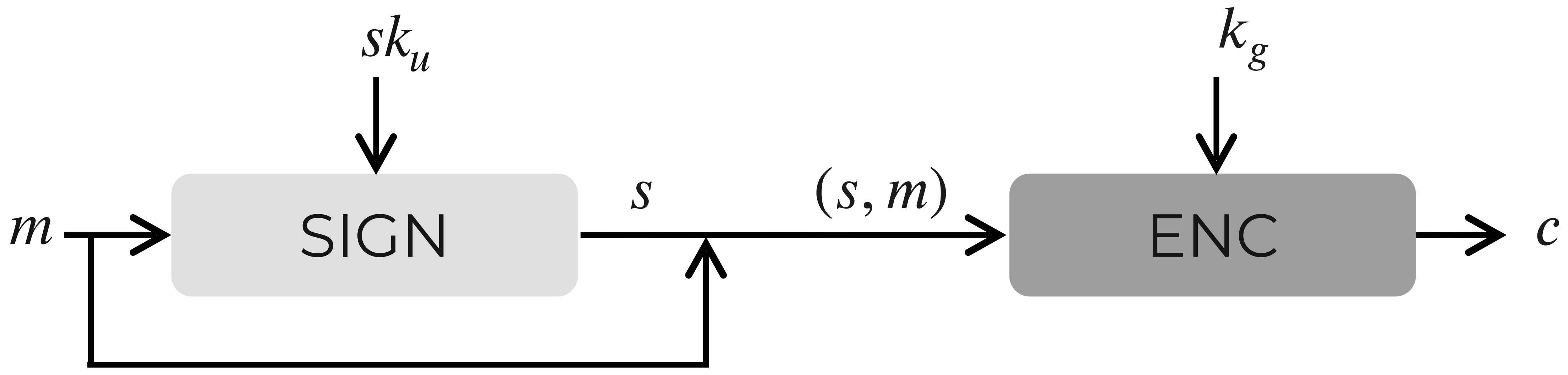


Encrypt-then-Sign (EtS)



Naive Constructions

Sign-then-Encrypt (StE)



Encrypt-then-Sign (EtS)

MLS

matrix

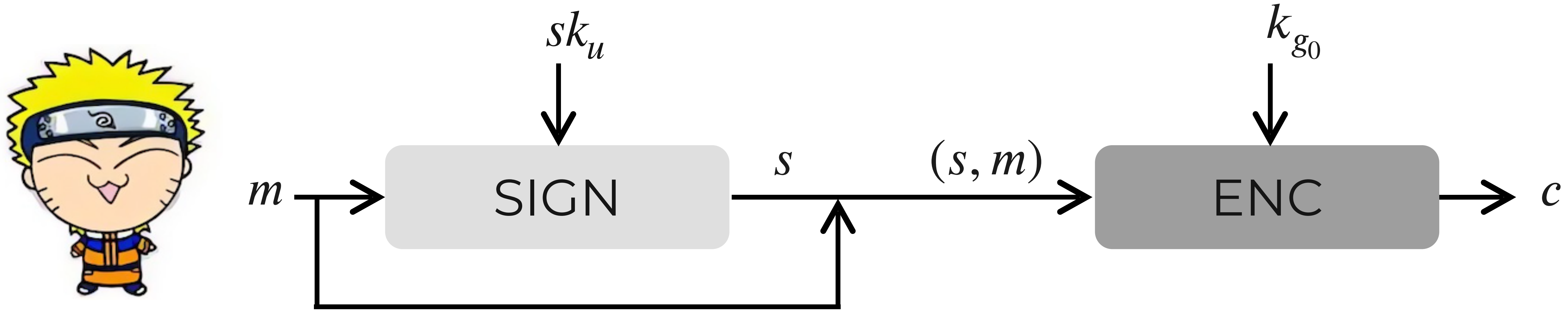
Context-Switching Attacks

Context-Switching Attacks

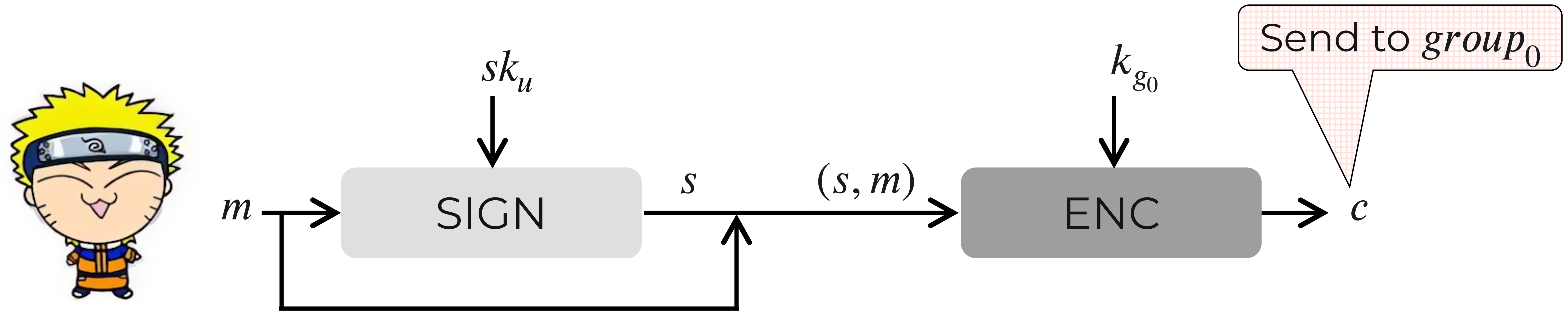
$\rightarrow group_0$

$\rightarrow group_1$

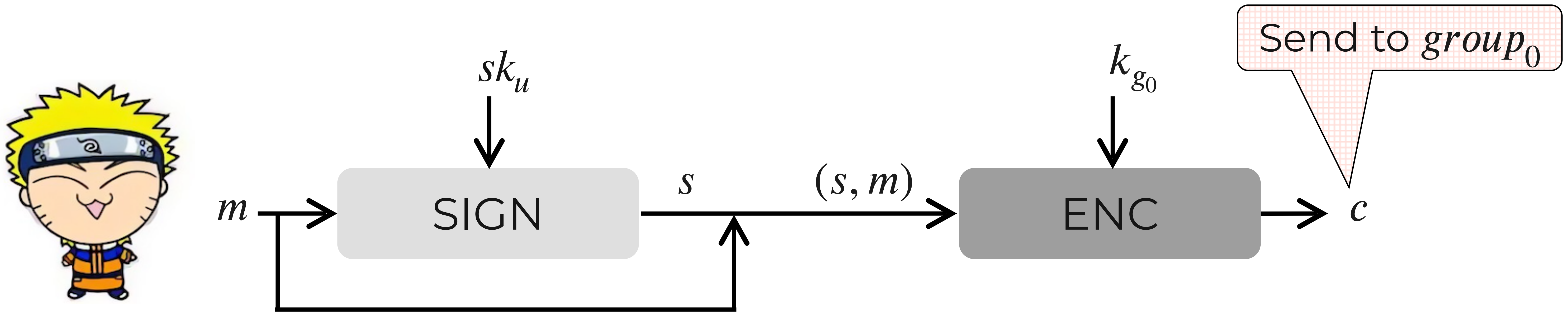
Context-Switching Attacks

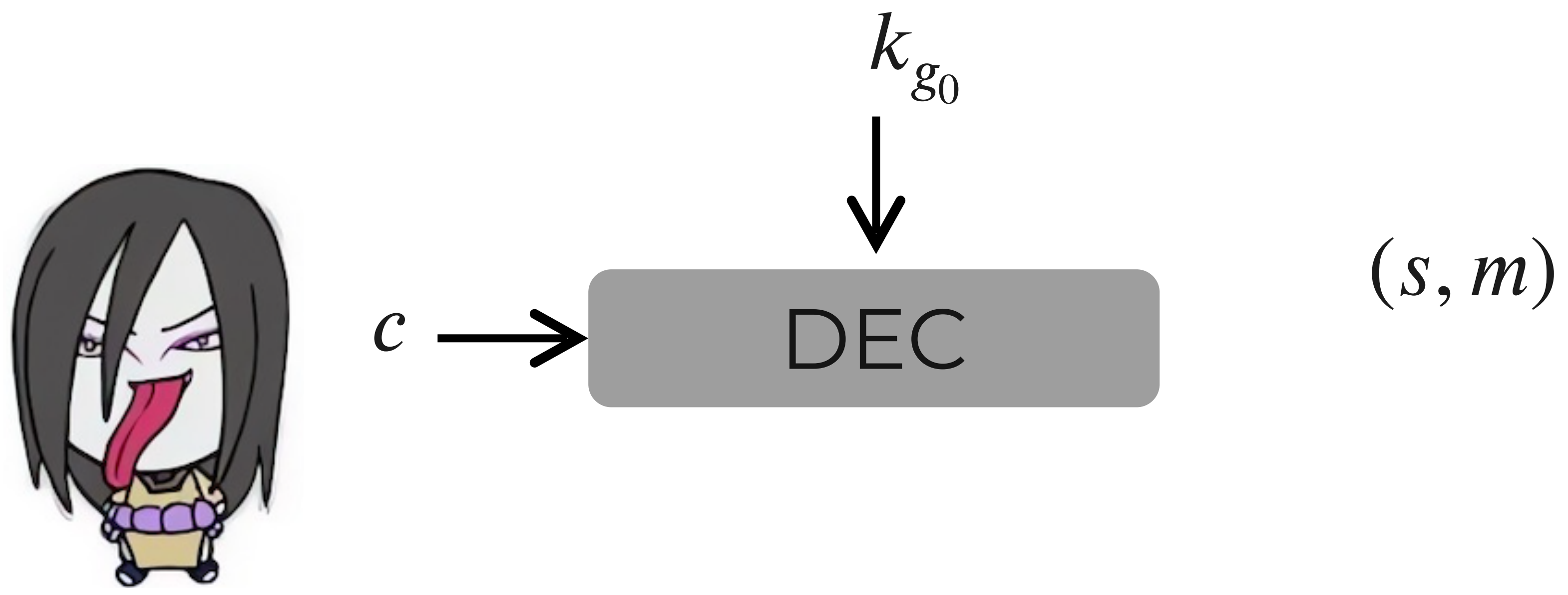


Context-Switching Attacks

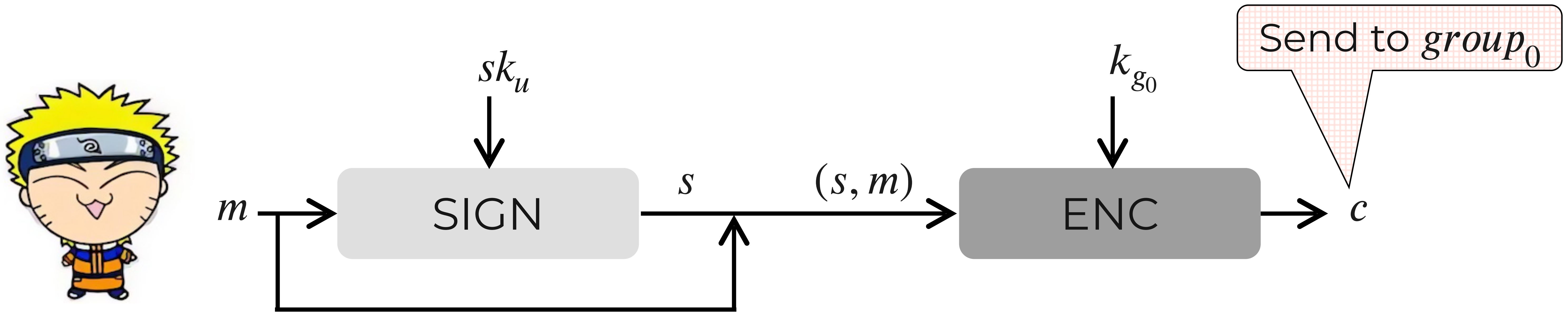


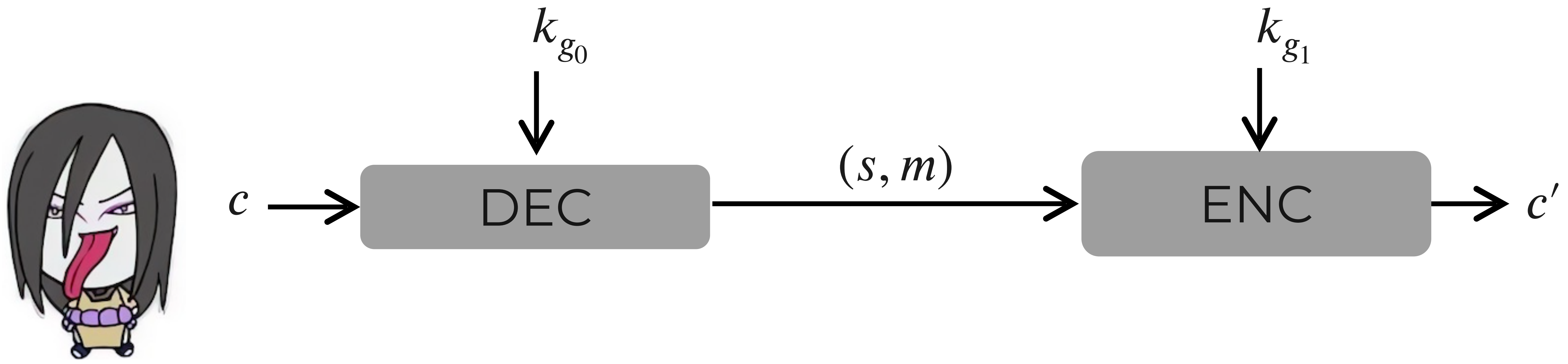
Context-Switching Attacks



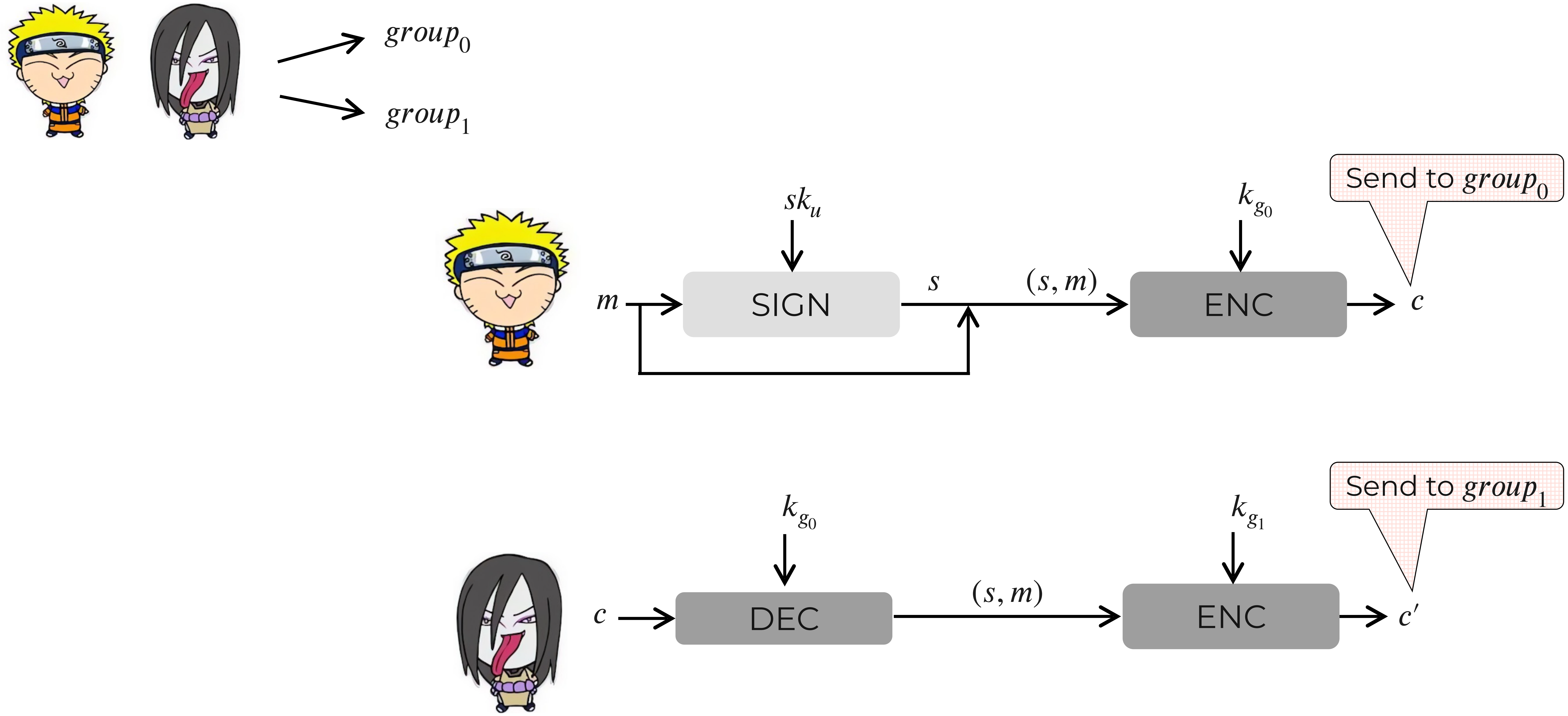


Context-Switching Attacks

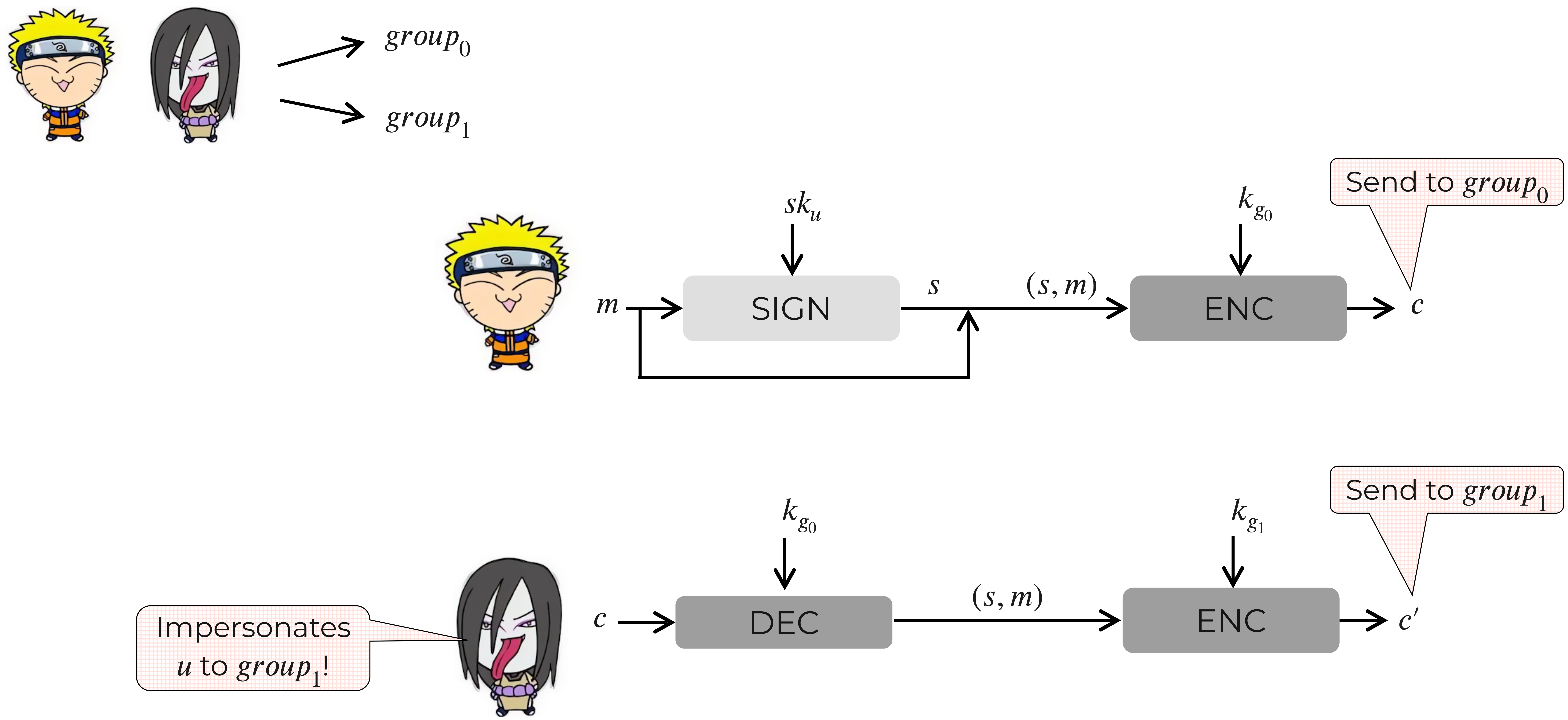




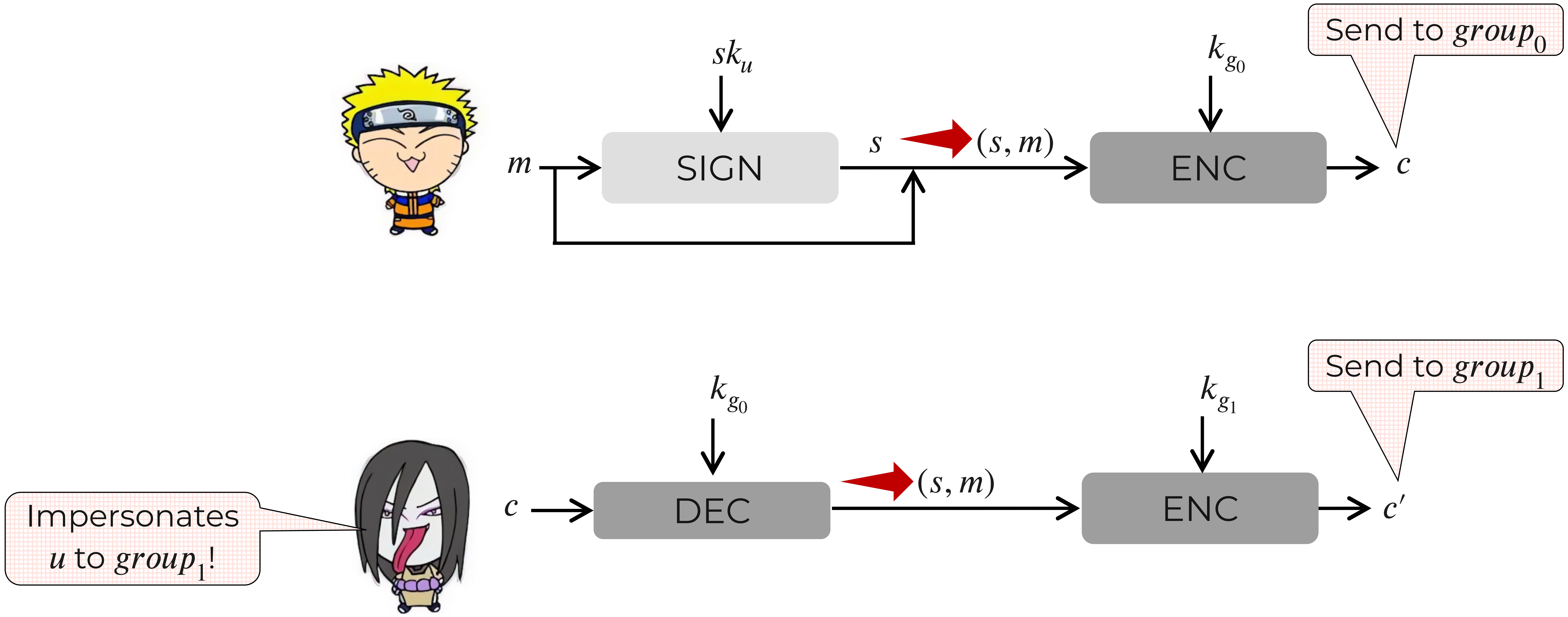
Context-Switching Attacks



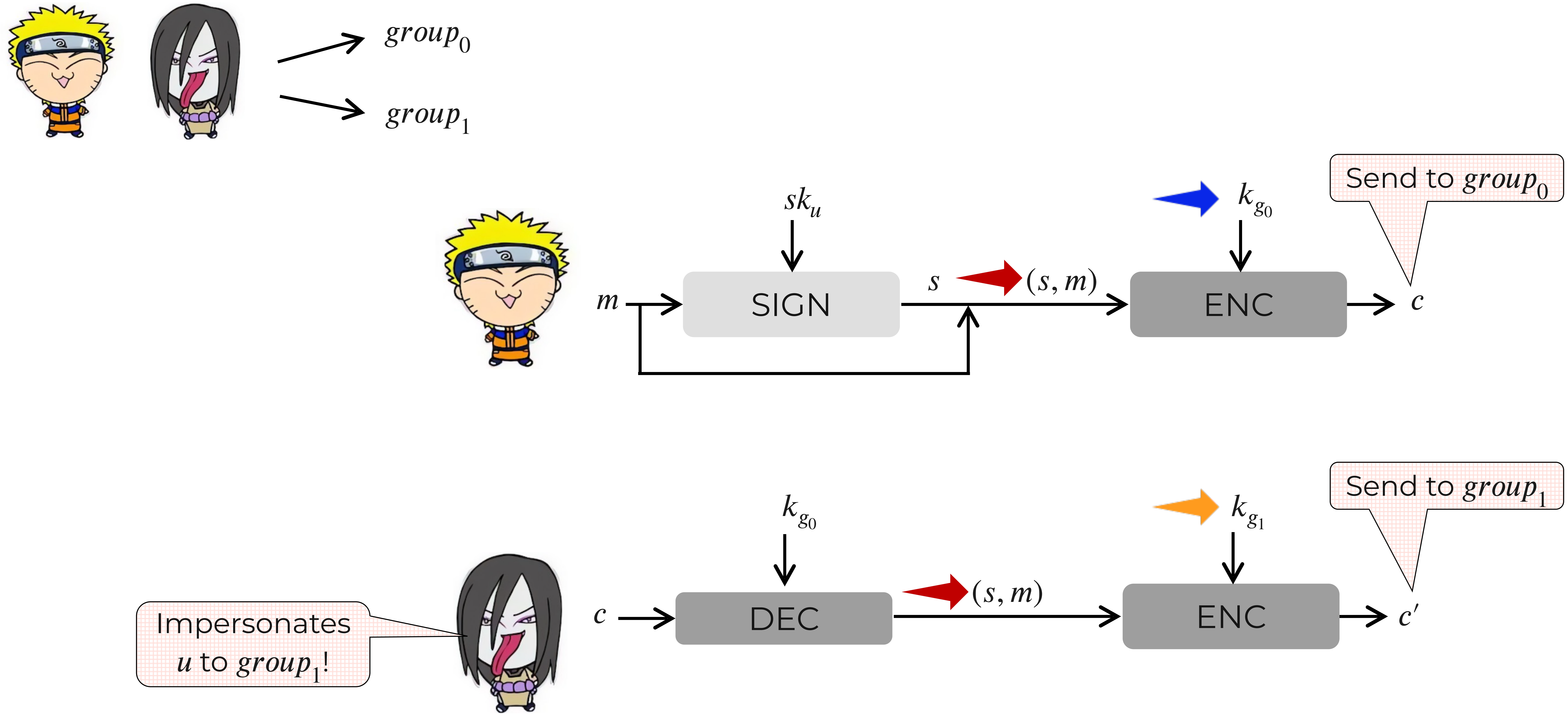
Context-Switching Attacks



Context-Switching Attacks

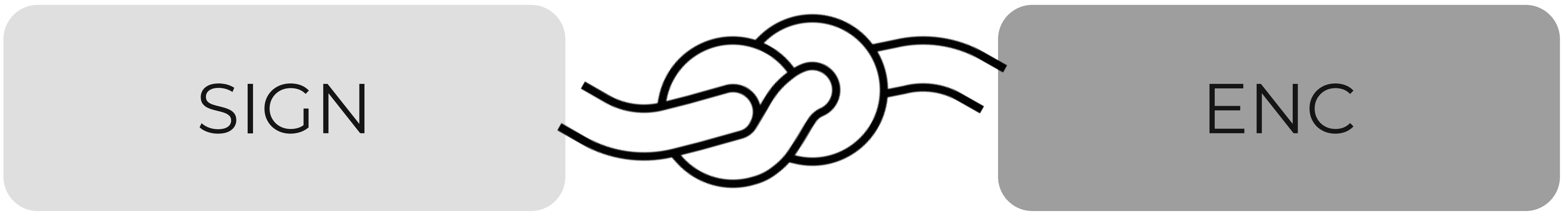


Context-Switching Attacks



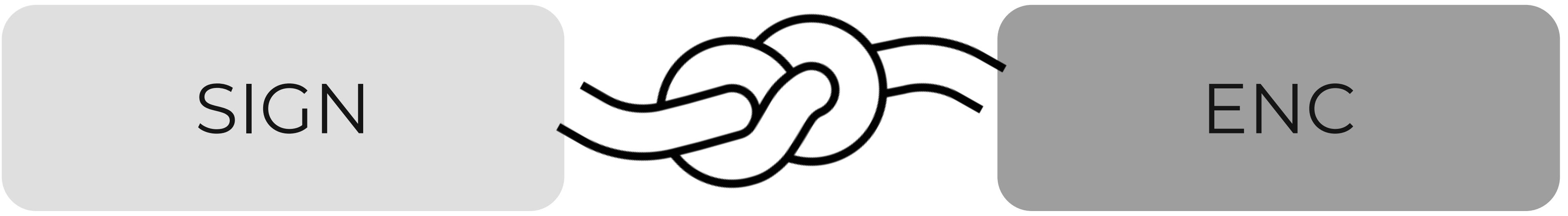
Context Binding

Context Binding



“bind”

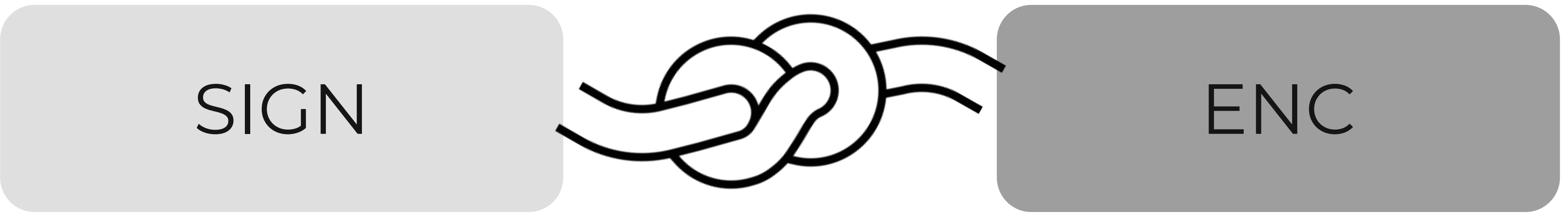
Context Binding



“bind”

SIGN
group_key_id
n
ad

Context Binding

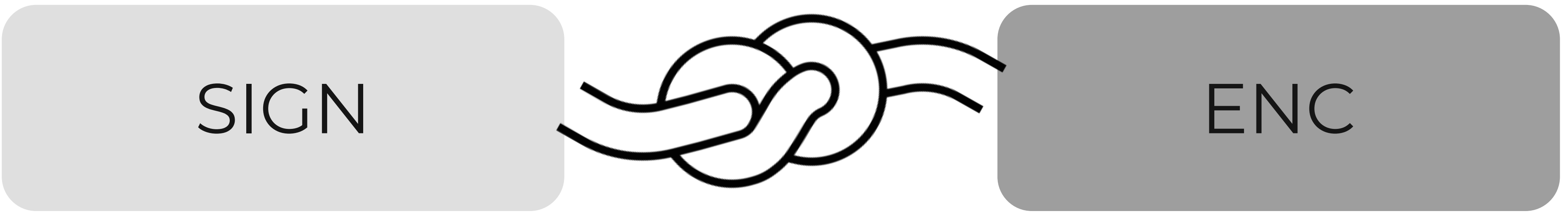


“bind”

SIGN
group_key_id
n
ad

Value that uniquely identifies the group’s encryption key

Context Binding



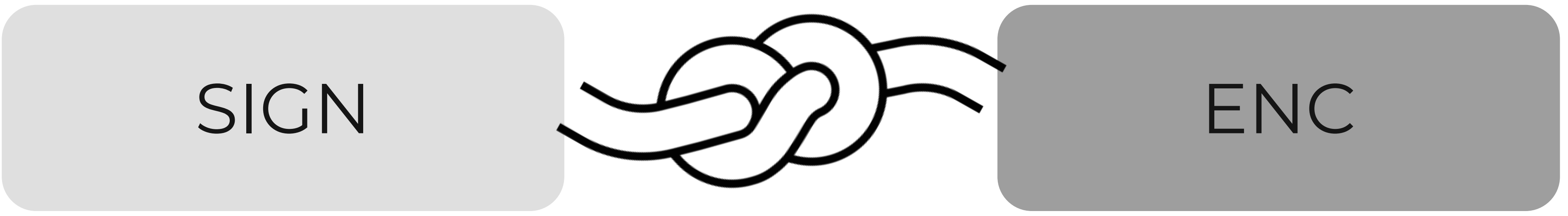
“bind”

Value that uniquely identifies the group’s encryption key

SIGN
group_key_id
n
ad

ENC
sender_id

Context Binding



“bind”

SIGN
group_key_id
n
ad

group_key_id: Value that uniquely identifies the group’s encryption key

ENC
sender_id

sender_id: Value that uniquely identifies the sender’s signing key

Our Analysis: An Overview

Our Analysis: An Overview

MLS

matrix

Our Analysis: An Overview

MLS

matrix

Insider Replay

Insider Re-ordering

Our Analysis: An Overview

MLS

matrix

Insider Replay

Insider Re-ordering

No context binding

Our Analysis: An Overview

MLS

Insider Replay

Insider Re-ordering

No context binding

Insider Replay

Outsider Replay

Outsider Forgery*

matrix

* stolen signing key

Our Analysis: An Overview

MLS

matrix

Insider Replay

Insider Re-ordering

Insider Replay

Outsider Replay

Outsider Forgery*

No context binding

No context binding

* stolen signing key

Our Analysis: An Overview

Insider Replay
Insider Re-ordering

Insider Replay
Outsider Replay
Outsider Forgery*

Outsider Forgery†

No context binding

No context binding

* stolen signing key † discovered by [BCG23]

Our Analysis: An Overview

MLS

Insider Replay
Insider Re-ordering

No context binding

Insider Replay
Outsider Replay
Outsider Forgery*

No context binding

Outsider Forgery†

Unauthenticated Symmetric Encryption

matrix

* stolen signing key † discovered by [BCG23]

Our Analysis: An Overview

MLS	Session	Signal	matrix
Insider Replay Insider Re-ordering	Insider Replay Outsider Replay Outsider Forgery*	Outsider Forgery†	In-model Insider Replay
No context binding	No context binding	Unauthenticated Symmetric Encryption	No context binding

* stolen signing key † discovered by [BCG23]

Our Analysis: An Overview

Insider Replay
Insider Re-ordering

No context binding

Insider Replay
Outsider Replay
Outsider Forgery*

No context binding

Outsider Forgery†

Unauthenticated Symmetric Encryption

In-model Insider Replay

No context binding

* stolen signing key † discovered by [BCG23]

Case Study I: MLS

MLS

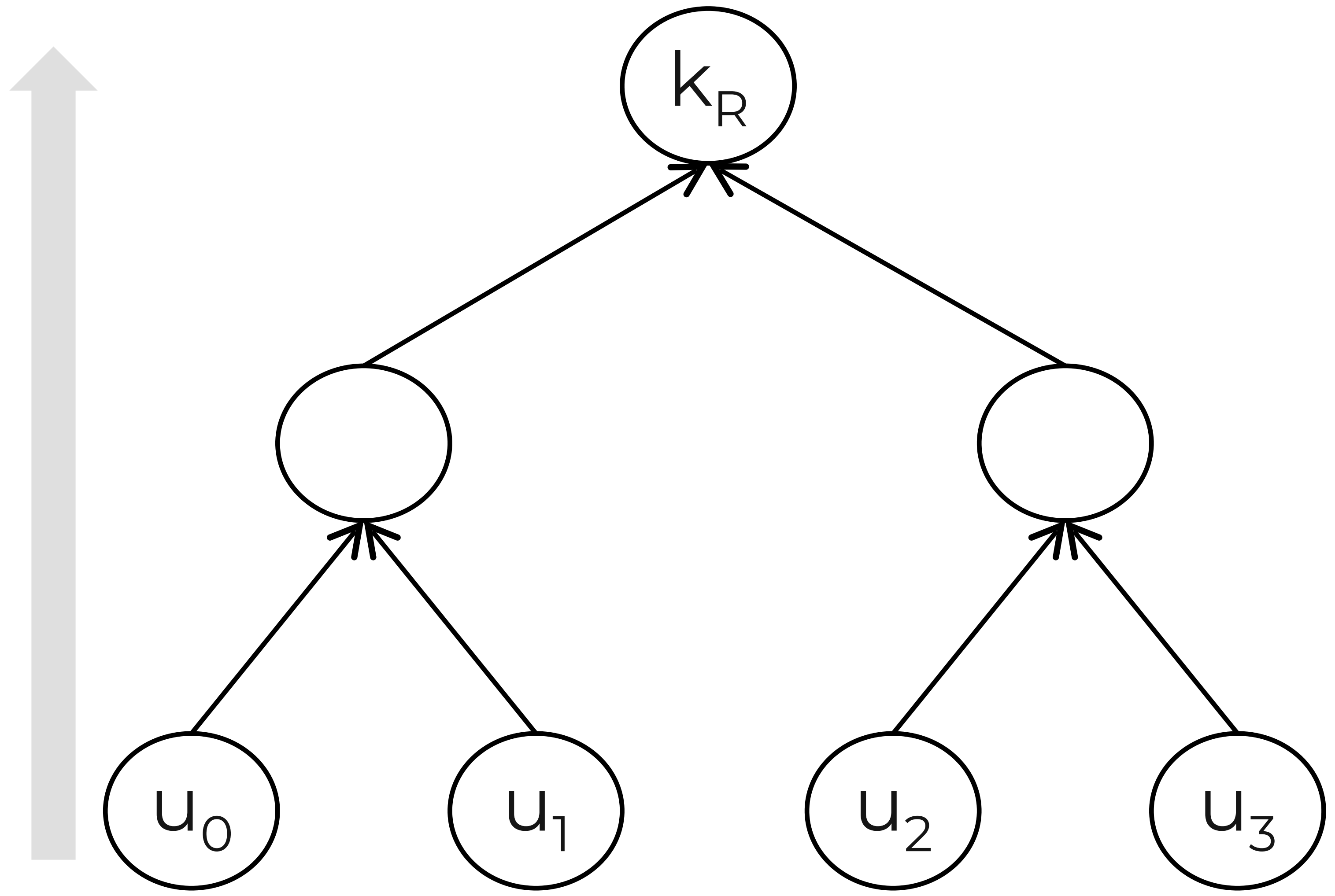
Encryption Key Derivation in MLS

Encryption Key Derivation in MLS

Ratchet tree

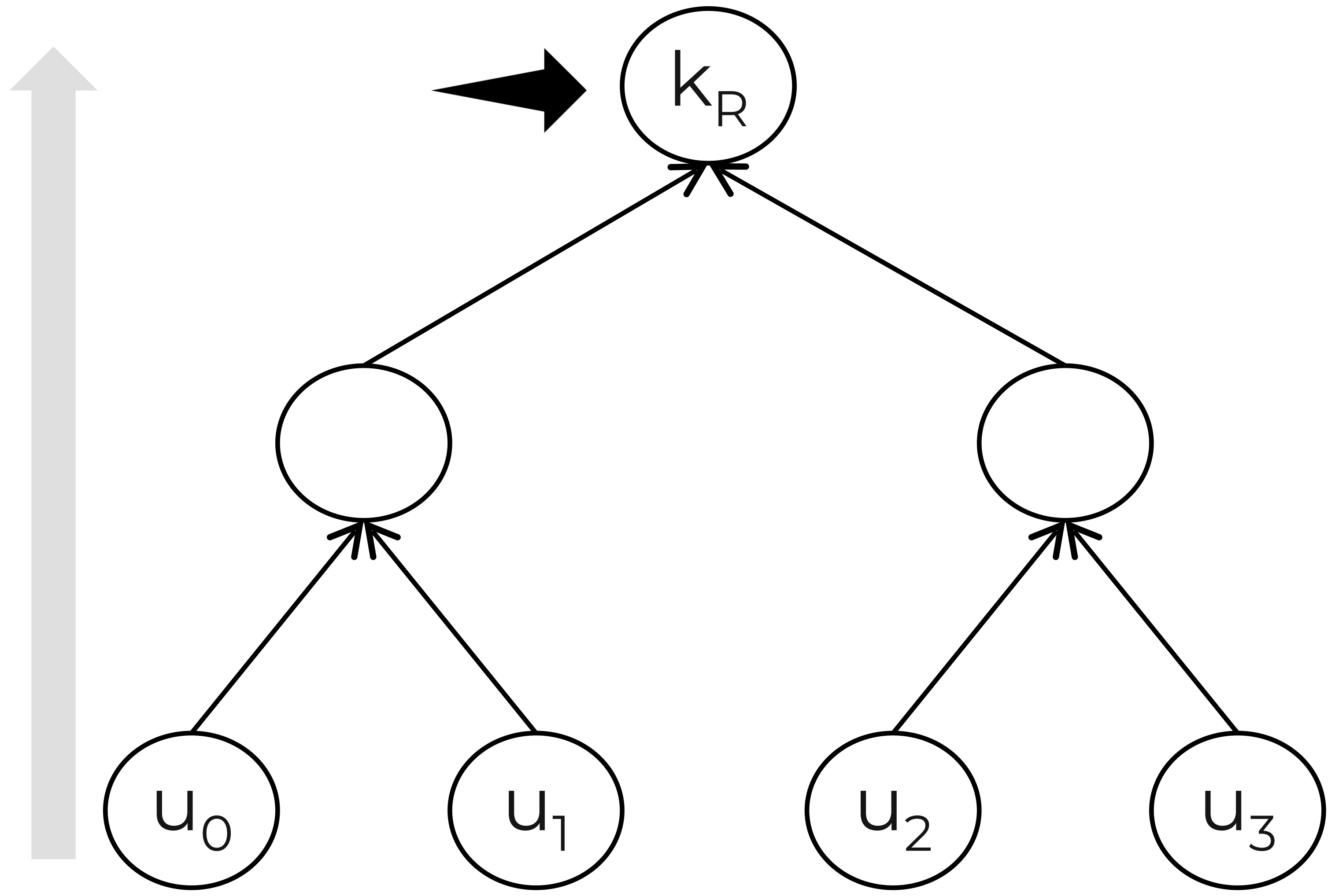
Encryption Key Derivation in MLS

Ratchet tree



Encryption Key Derivation in MLS

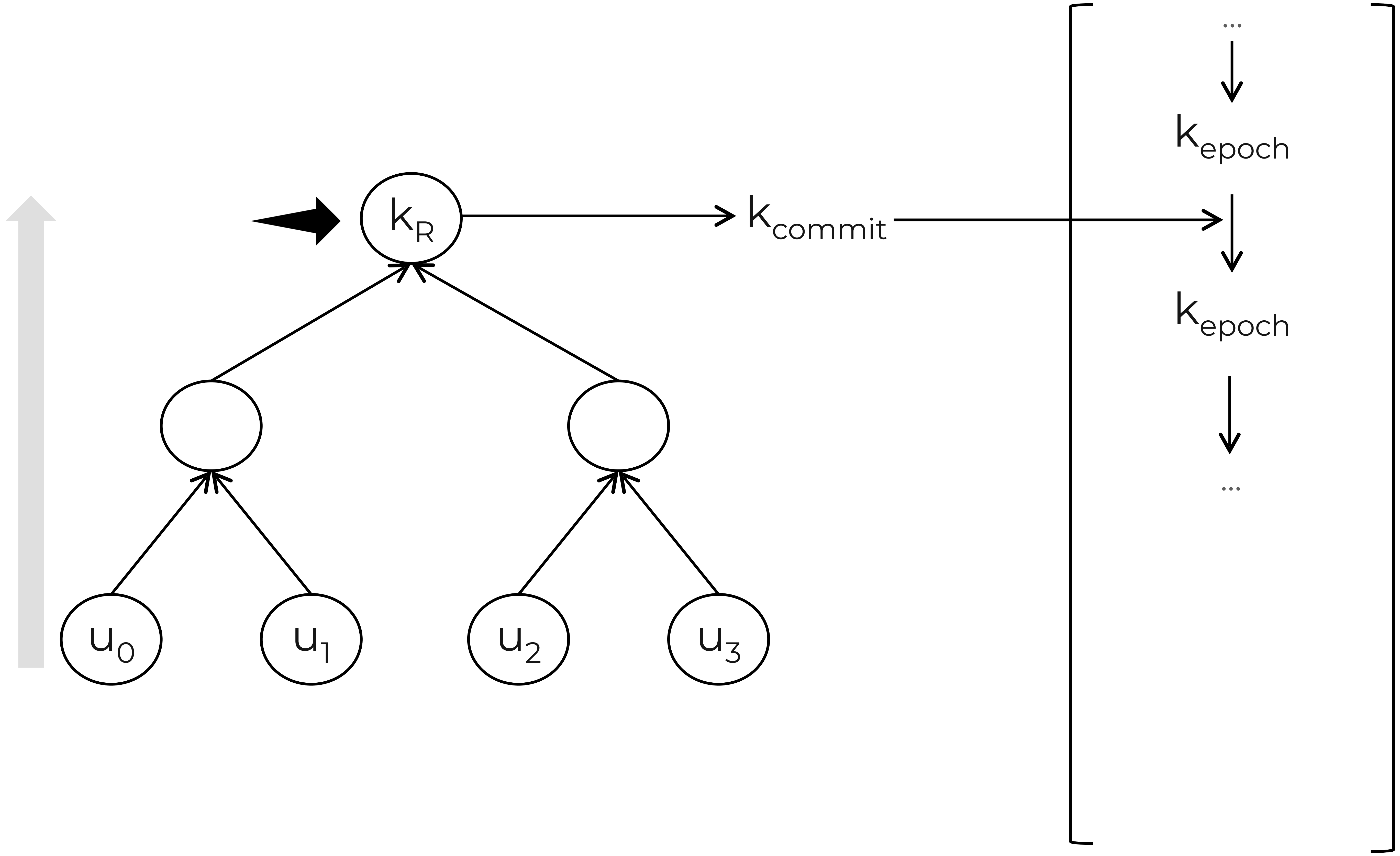
Ratchet tree



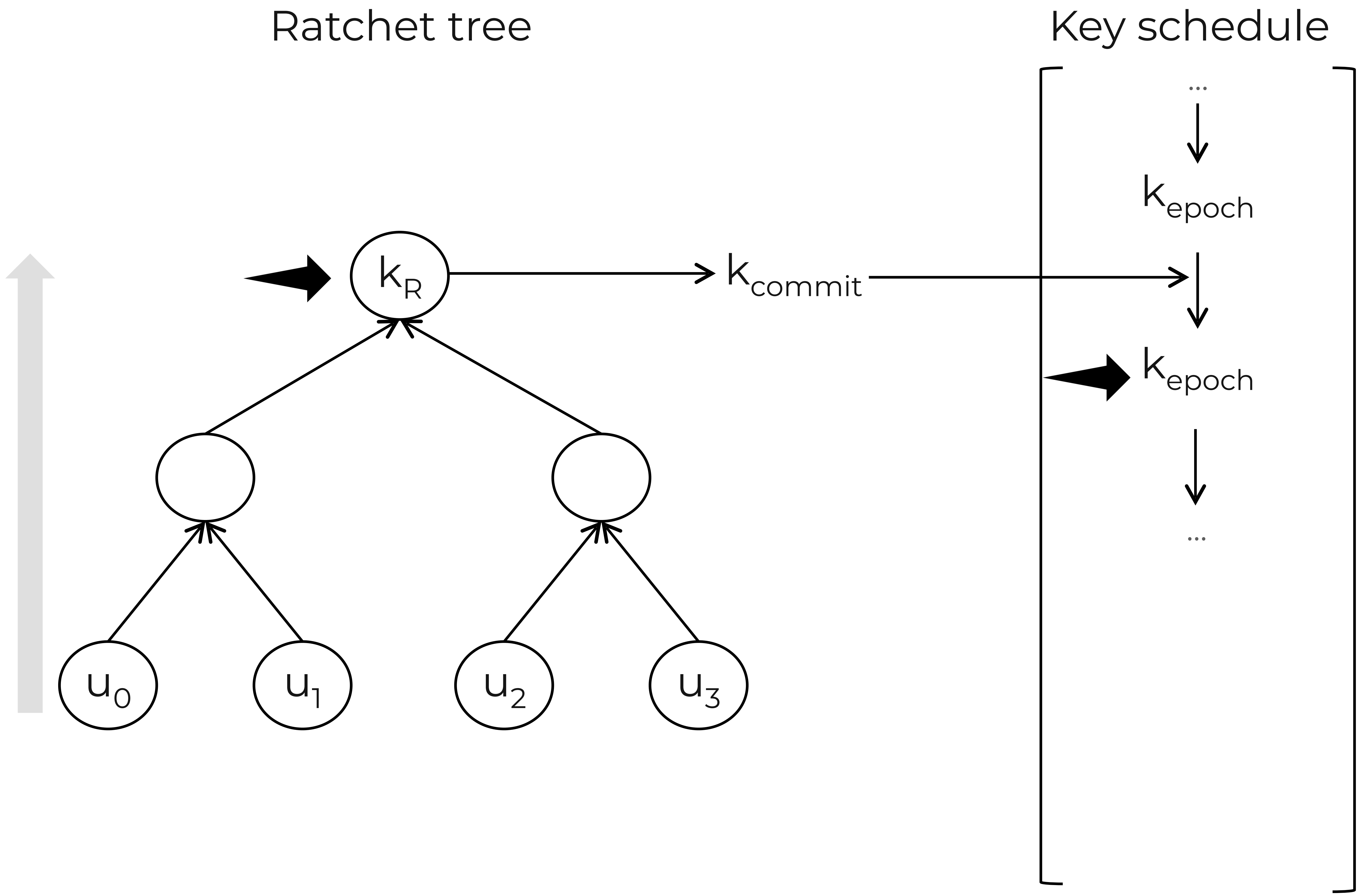
Encryption Key Derivation in MLS

Ratchet tree

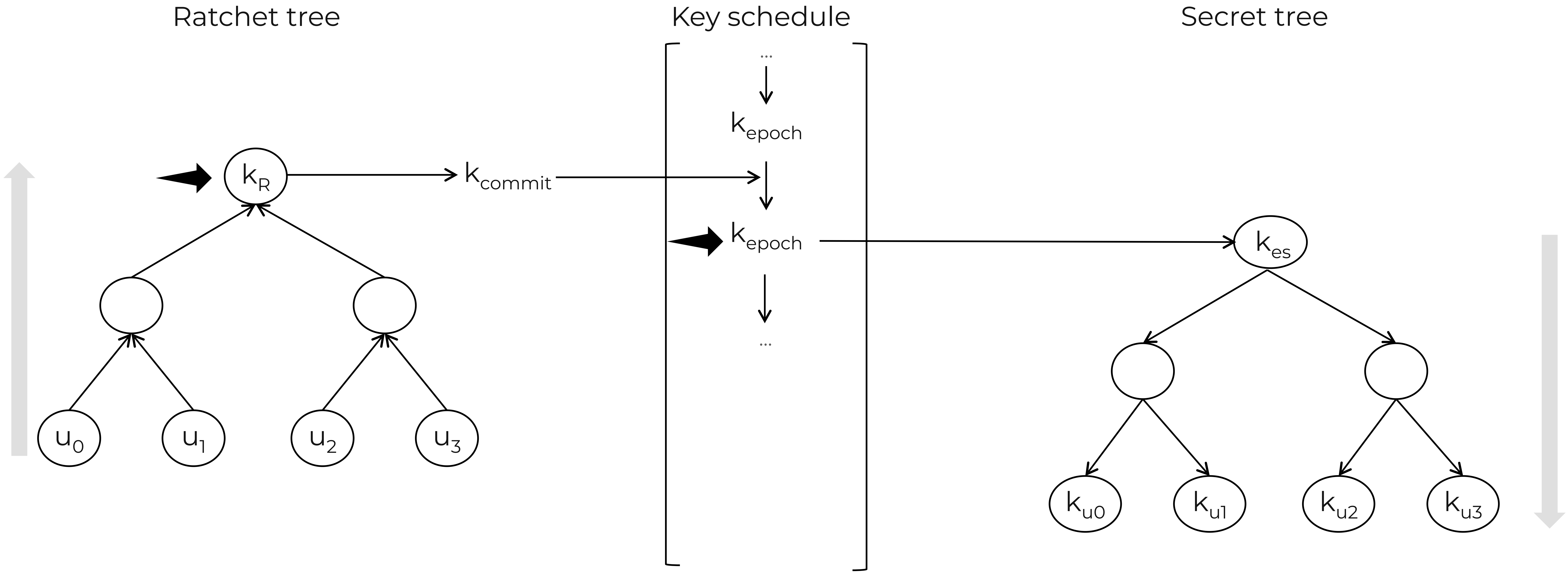
Key schedule



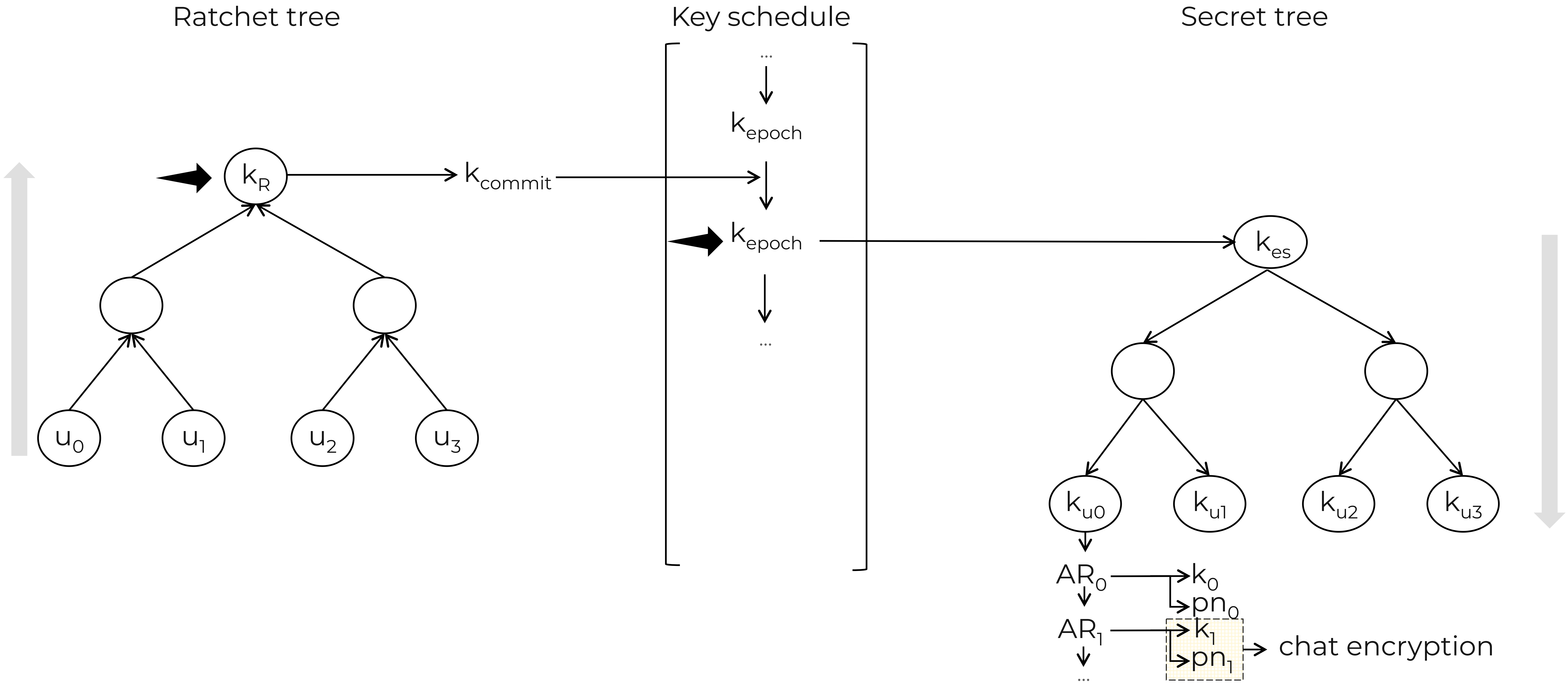
Encryption Key Derivation in MLS



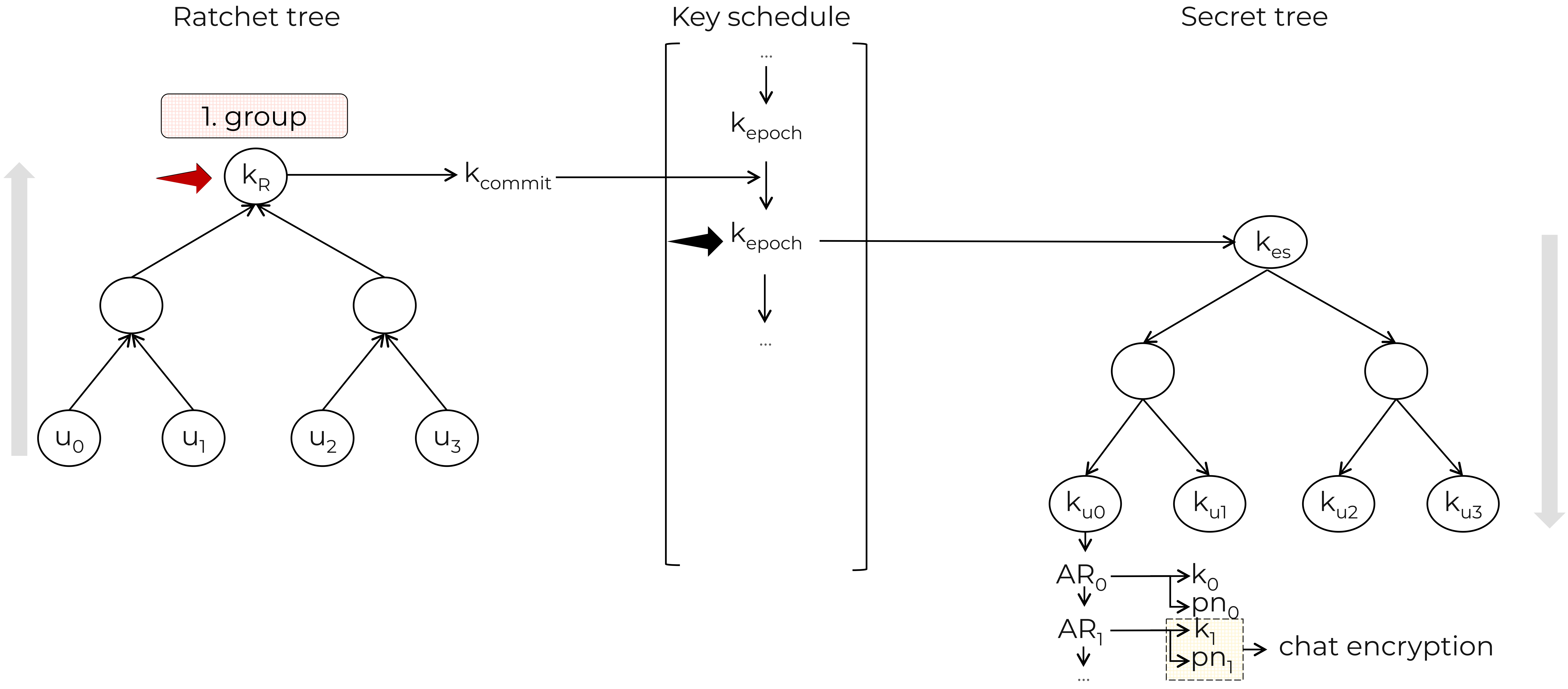
Encryption Key Derivation in MLS



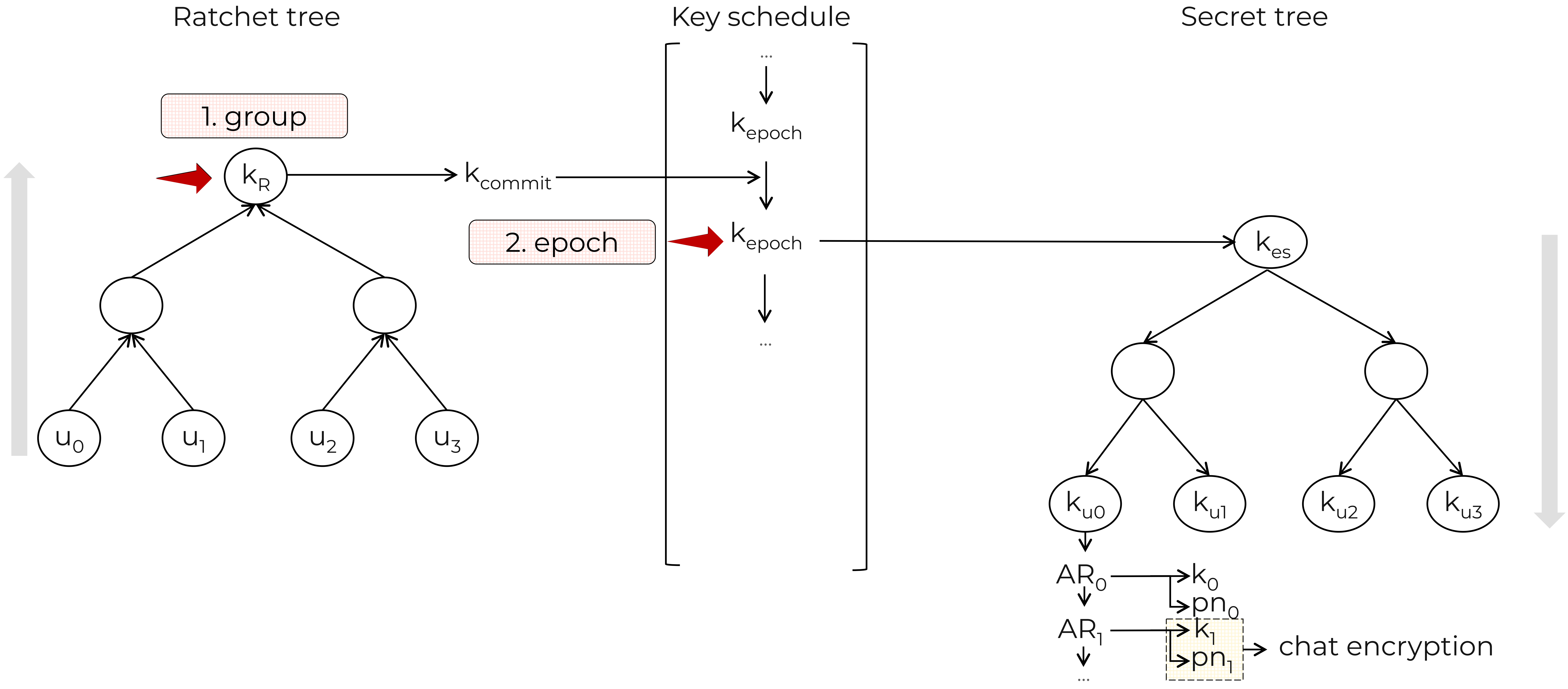
Encryption Key Derivation in MLS



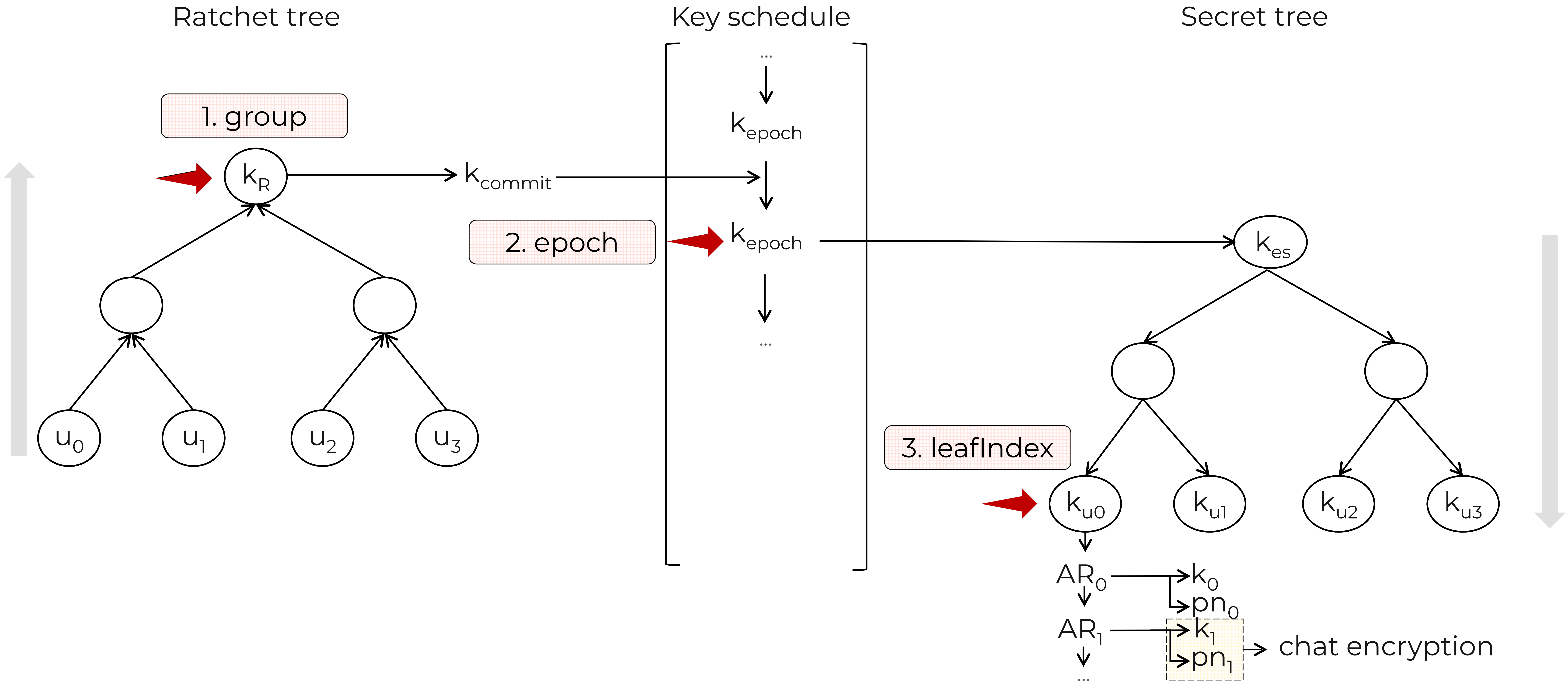
Encryption Key Derivation in MLS



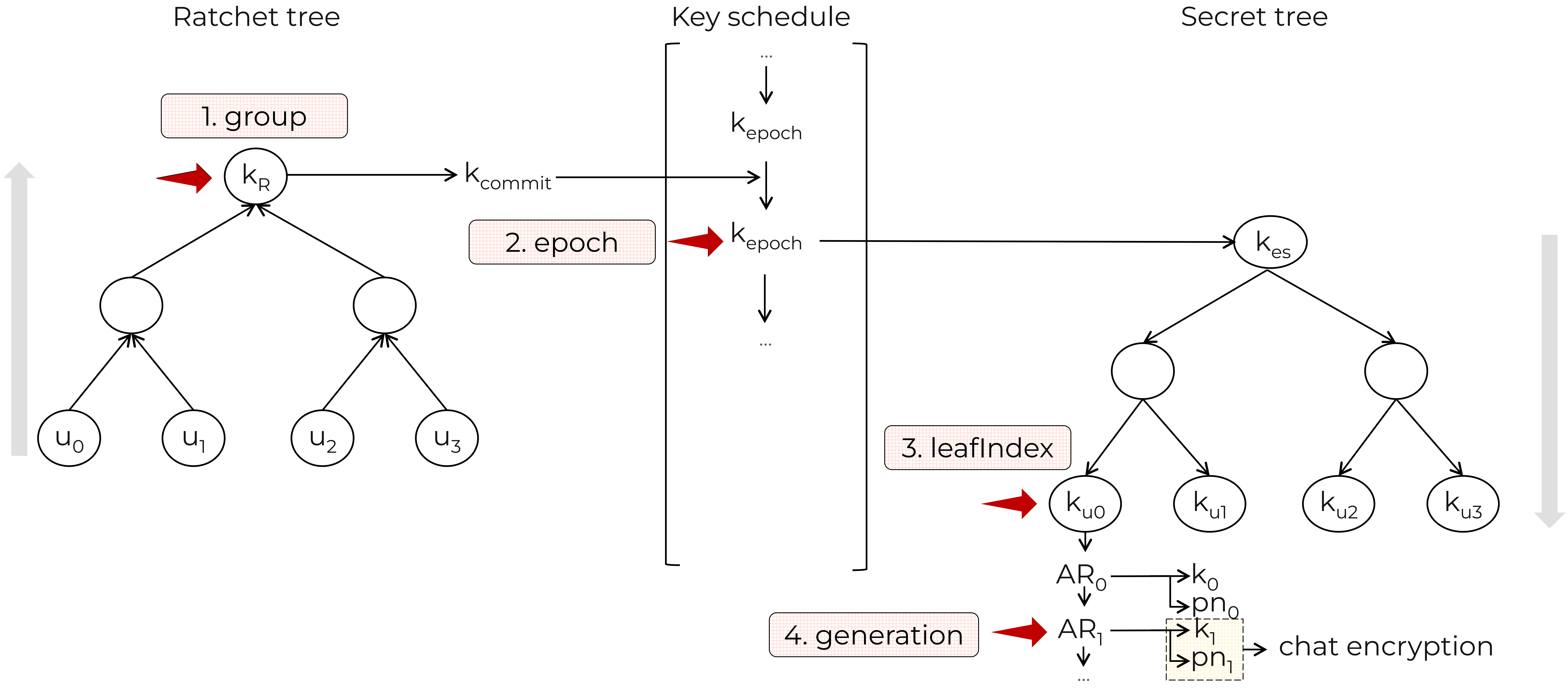
Encryption Key Derivation in MLS



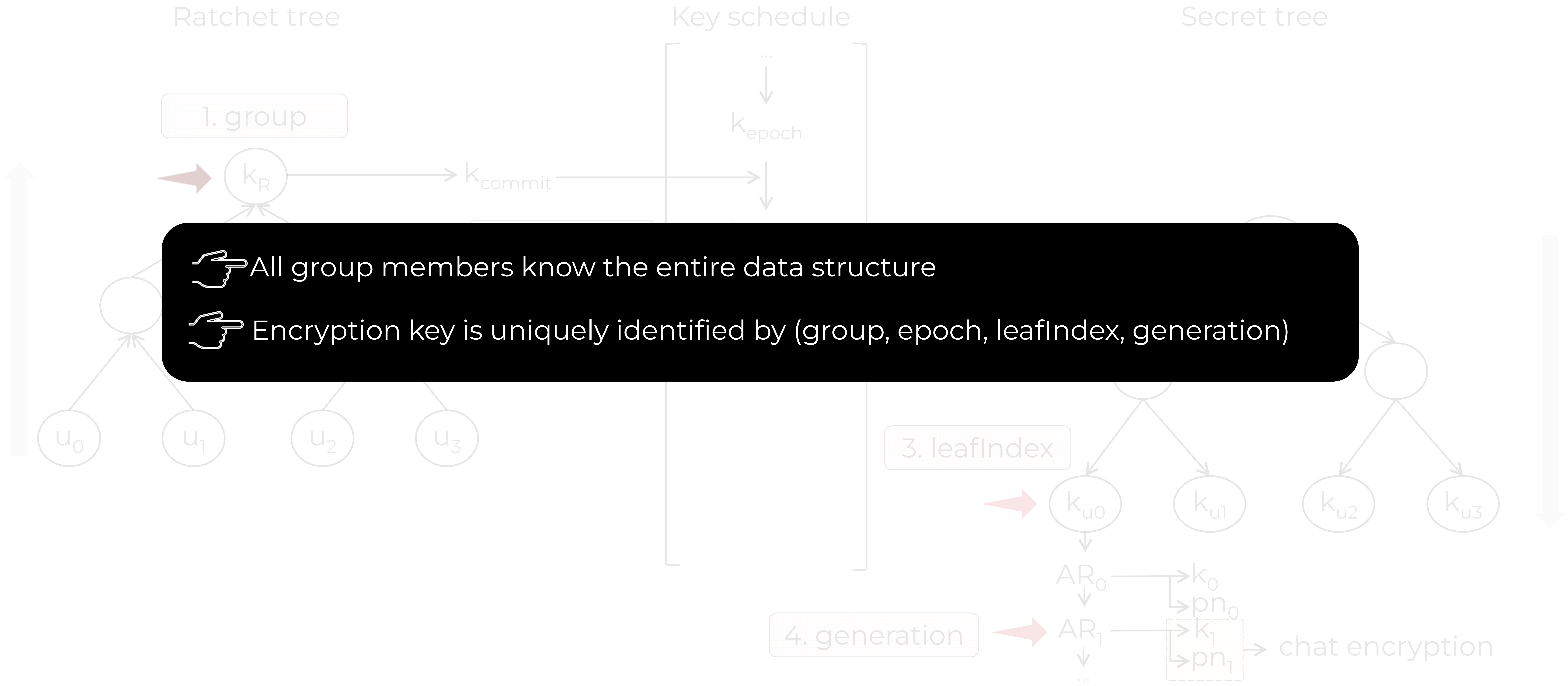
Encryption Key Derivation in MLS



Encryption Key Derivation in MLS



Encryption Key Derivation in MLS



- All group members know the entire data structure
- Encryption key is uniquely identified by (group, epoch, leafIndex, generation)

Chat Encryption in MLS

Chat Encryption in MLS

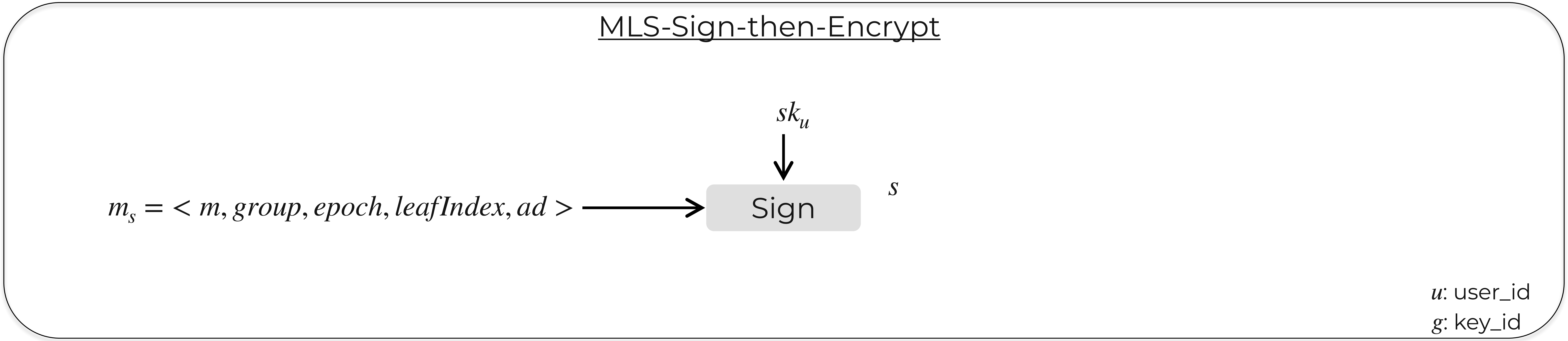
Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion

MLS-Sign-then-Encrypt

u: user_id
g: key_id

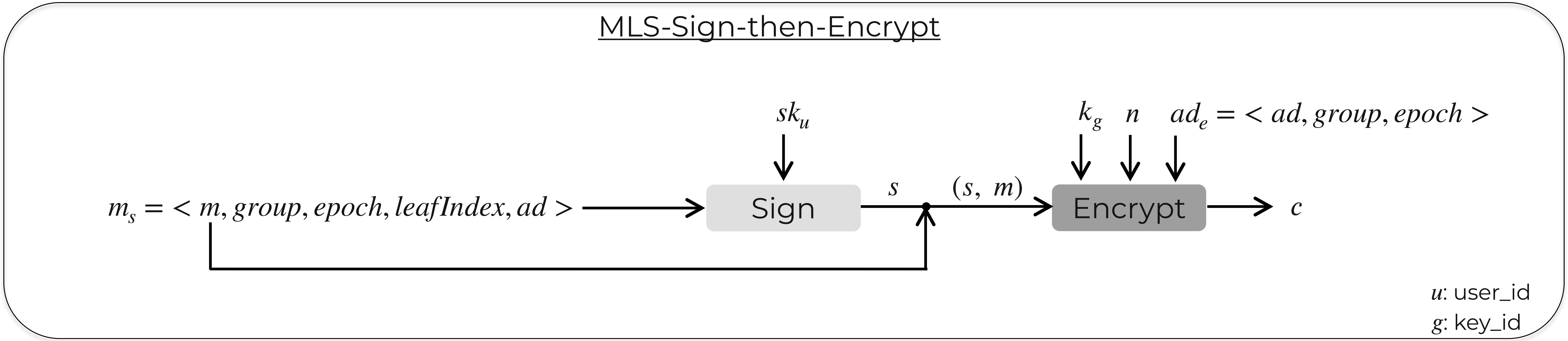
Chat Encryption in MLS

Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion



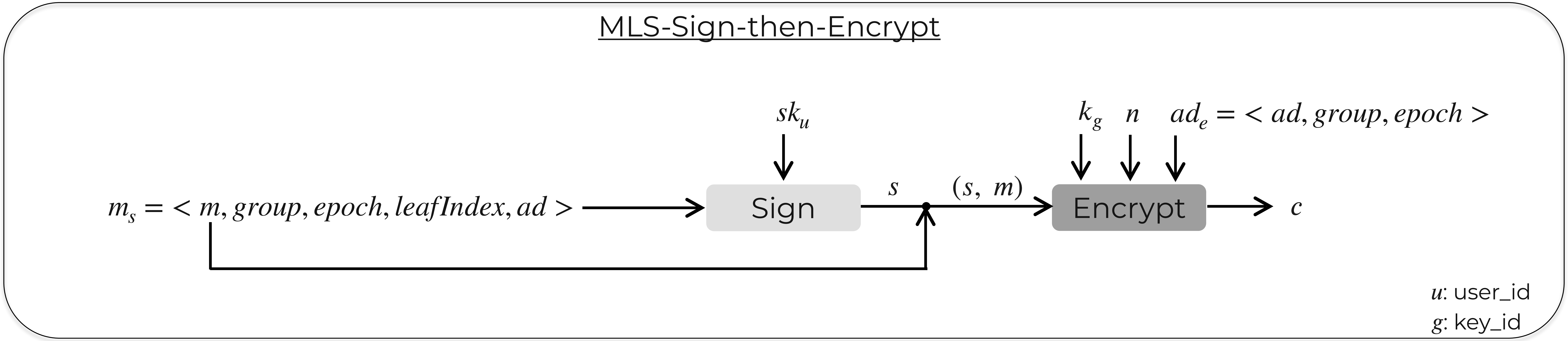
Chat Encryption in MLS

Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion



Chat Encryption in MLS

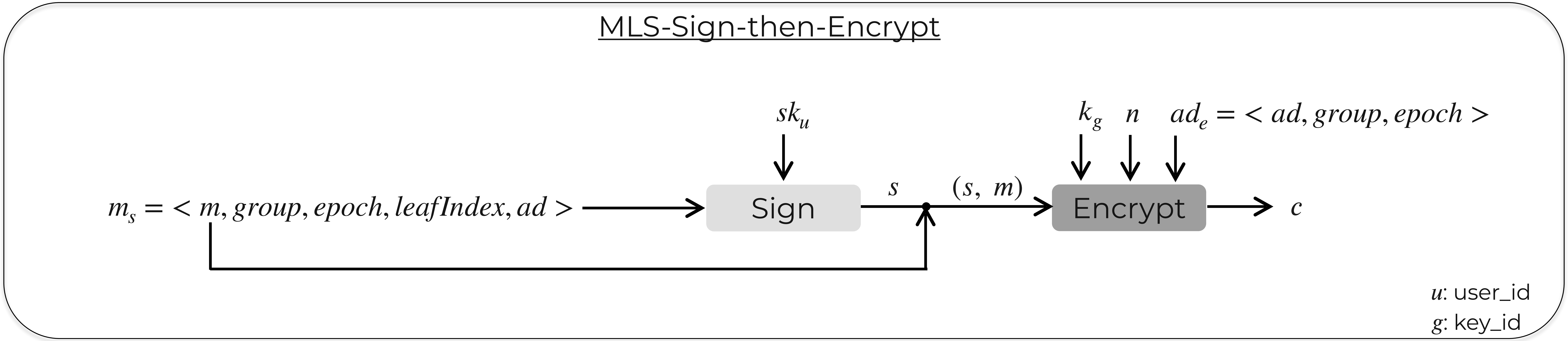
Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion



Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt $(s,\ m)$ using a different k and replay message to group

Chat Encryption in MLS

Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion

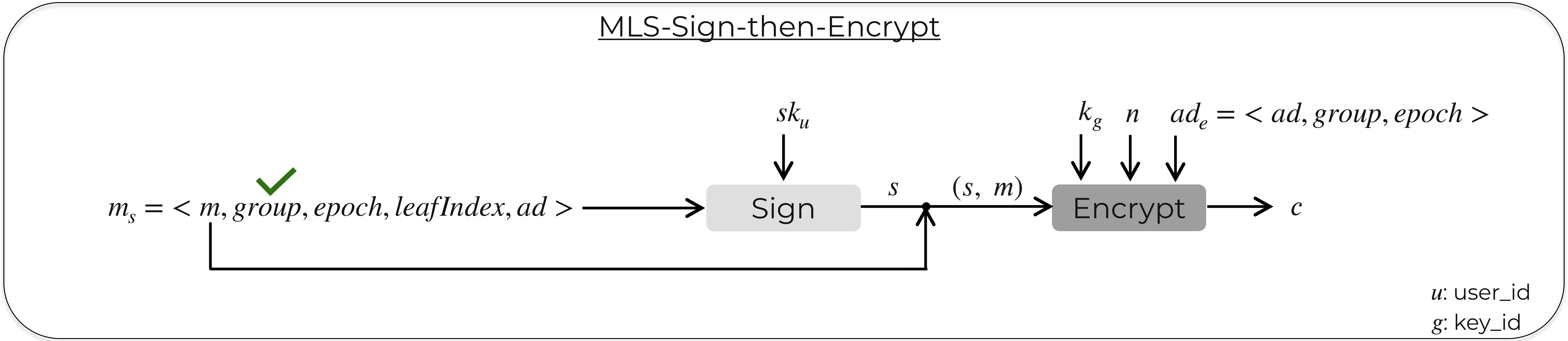


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt $(s,\ m)$ using a different k and replay message to group

Recall: key identifier $g = (group,\ epoch,\ leafIndex,\ generation)$

Chat Encryption in MLS

Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion

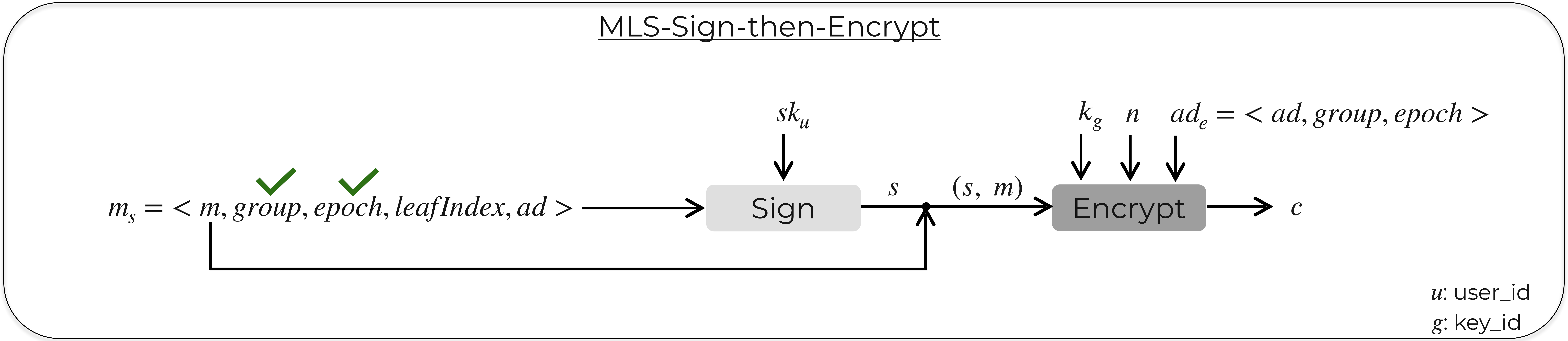


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt $(s,\ m)$ using a different k and replay message to group

Recall: key identifier $g = (group,\ epoch,\ leafIndex,\ generation)$

Chat Encryption in MLS

Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion

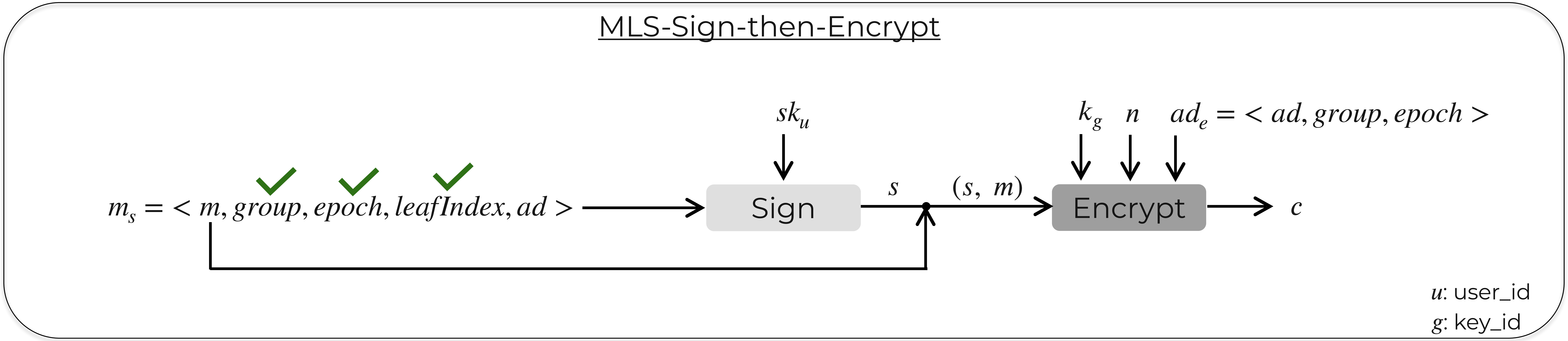


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt $(s,\ m)$ using a different k and replay message to group

Recall: key identifier $g = (group,\ epoch,\ leafIndex,\ generation)$

Chat Encryption in MLS

Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion

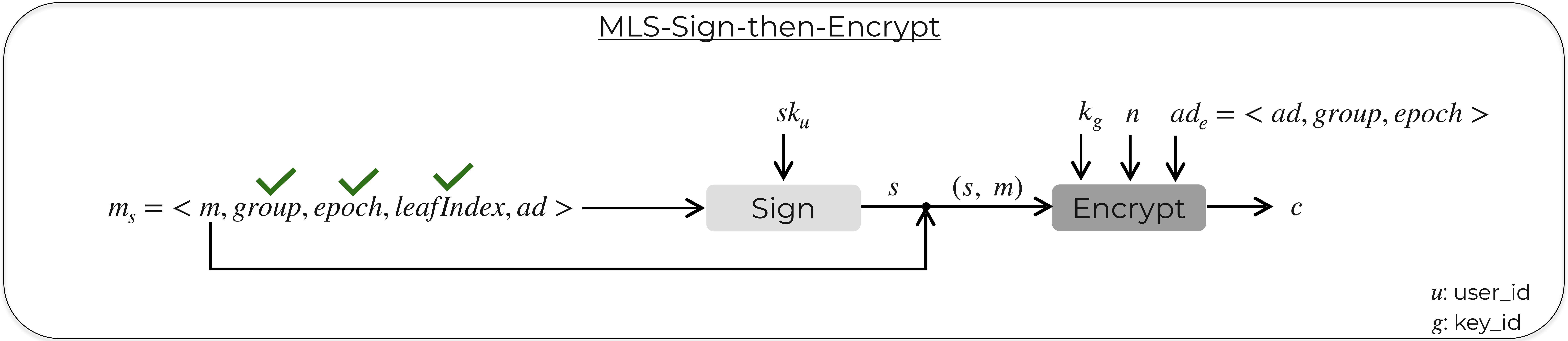


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt $(s,\ m)$ using a different k and replay message to group

Recall: key identifier $g = (group,\ epoch,\ leafIndex,\ generation)$

Chat Encryption in MLS

Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion

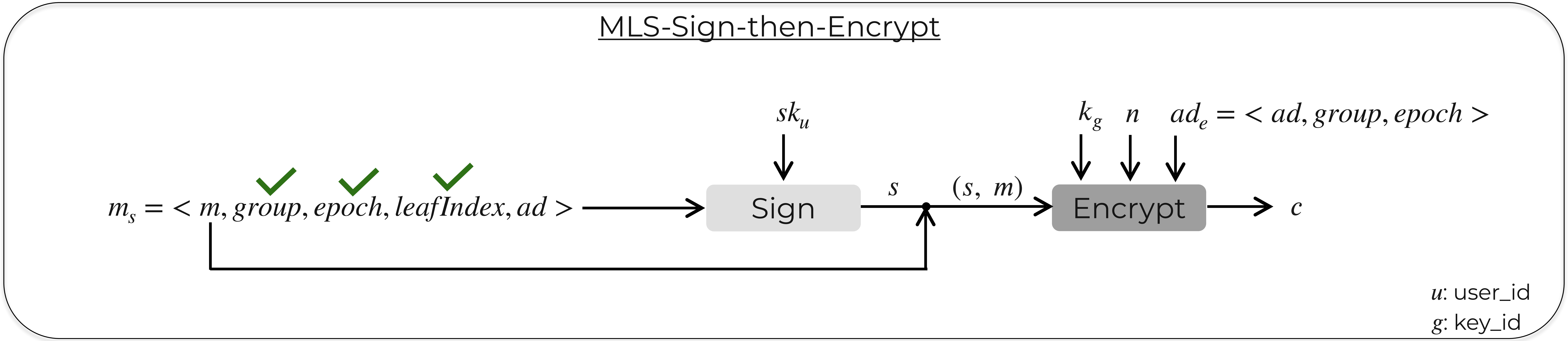


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt $(s,\ m)$ using a different k and replay message to group

Recall: key identifier $g = (group,\ epoch,\ leafIndex,\ generation)$

Chat Encryption in MLS

Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion



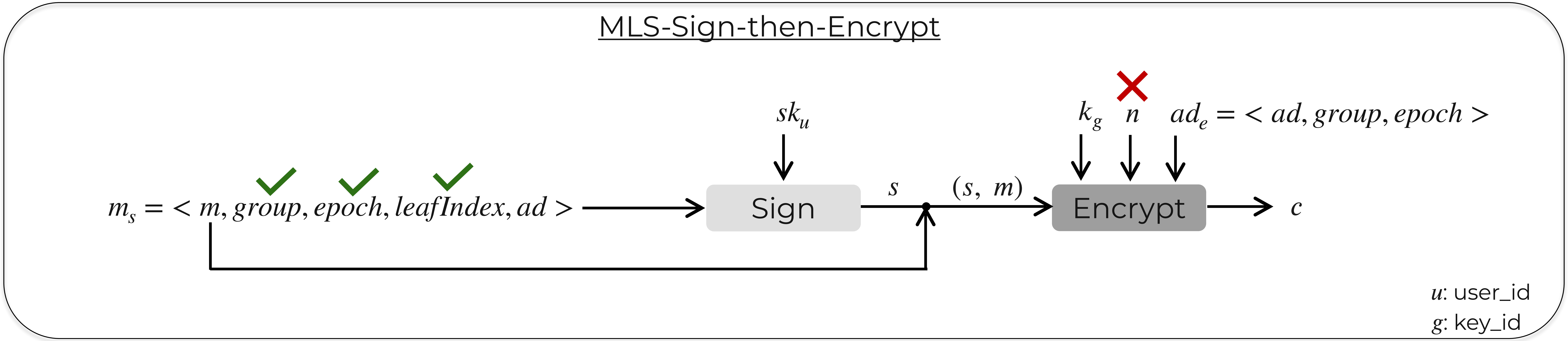
Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt $(s,\ m)$ using a different k and replay message to group

Recall: key identifier $g = (group,\ epoch,\ leafIndex,\ generation)$

SIGN
group_key_id

Chat Encryption in MLS

Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion

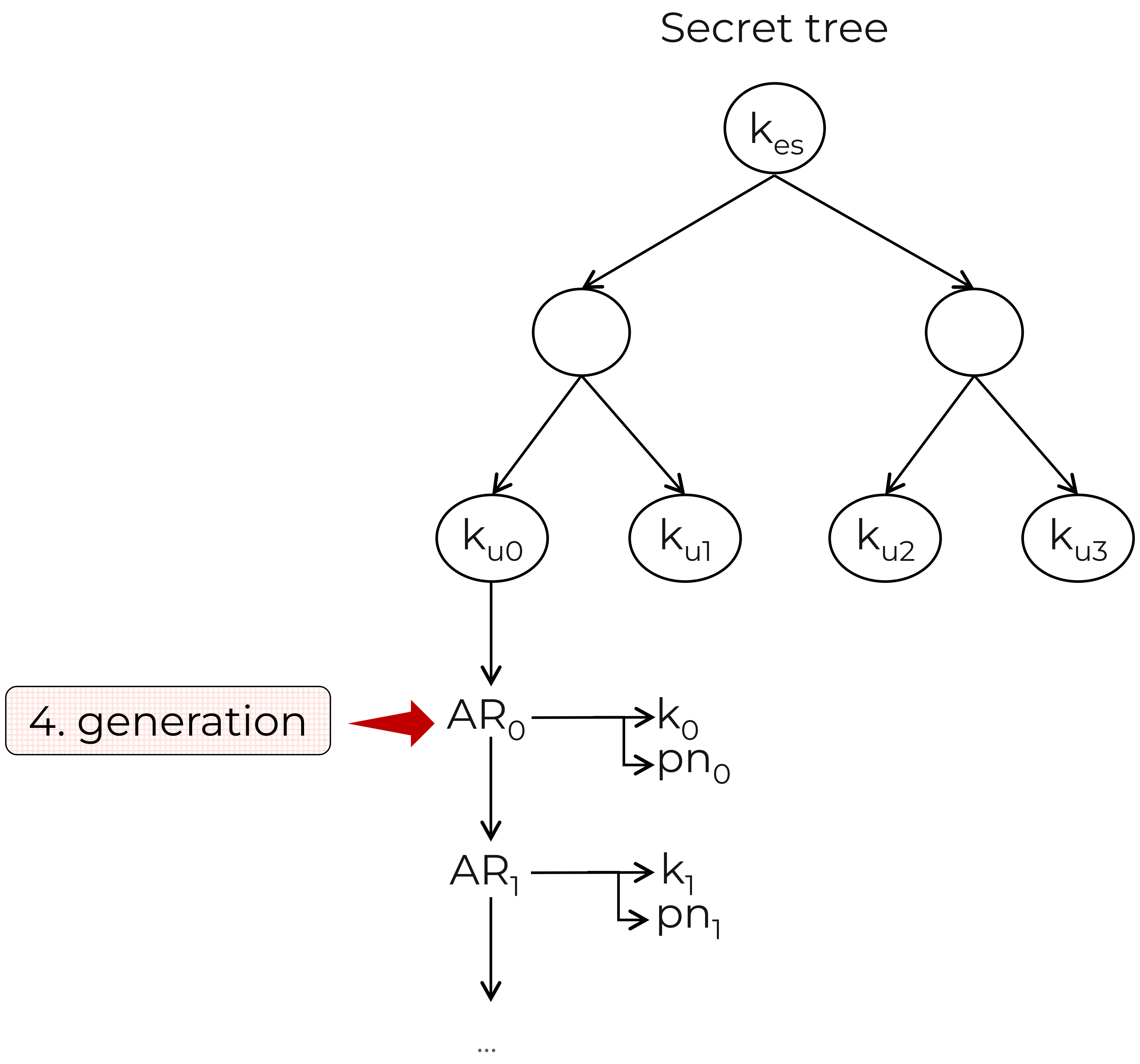


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt $(s,\ m)$ using a different k and replay message to group

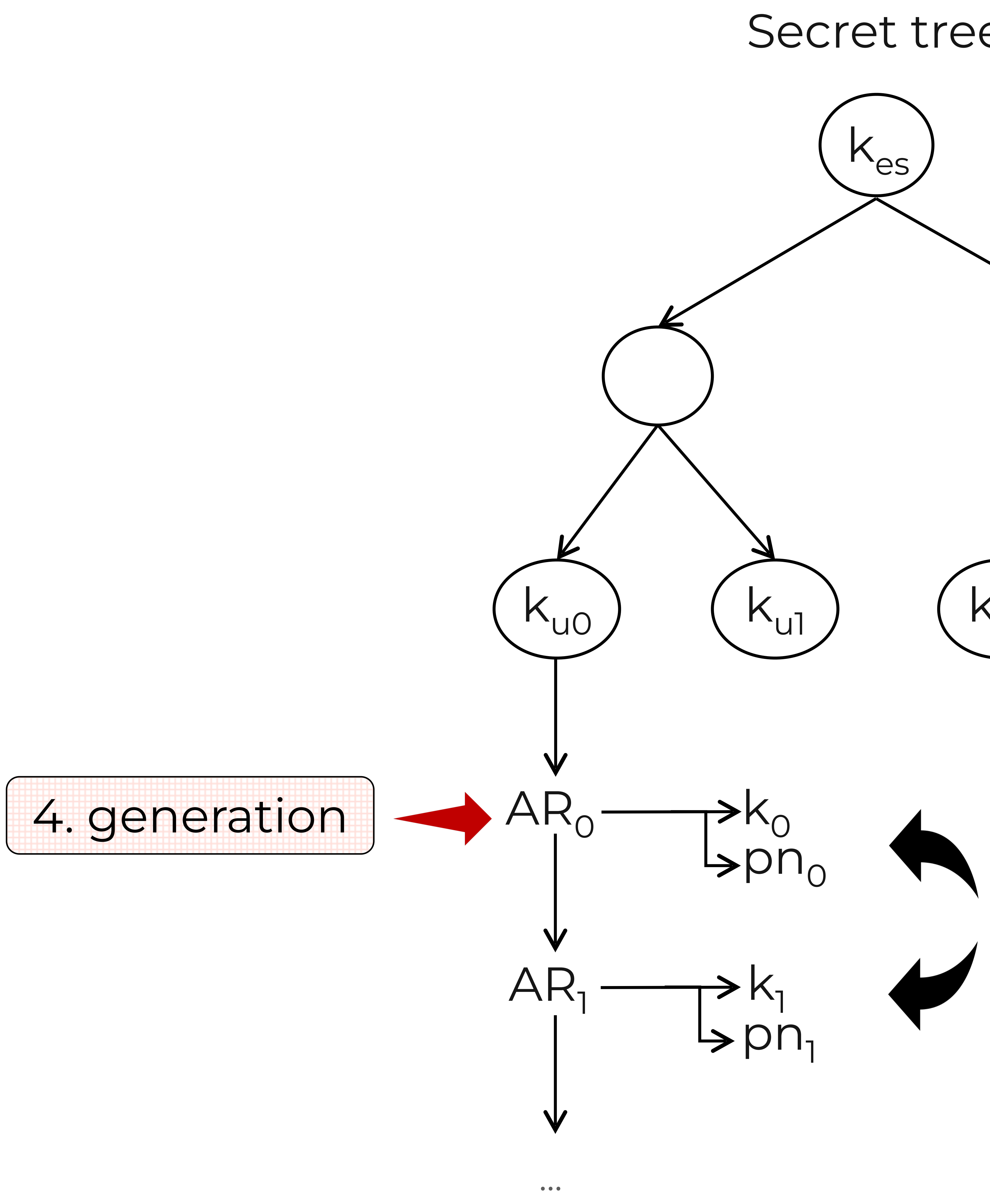
Recall: key identifier $g = (group,\ epoch,\ leafIndex,\ generation)$

SIGN
group_key_id

Encryption Key Derivation in MLS



Encryption Key Derivation in MLS



malicious insider can hop between generations!

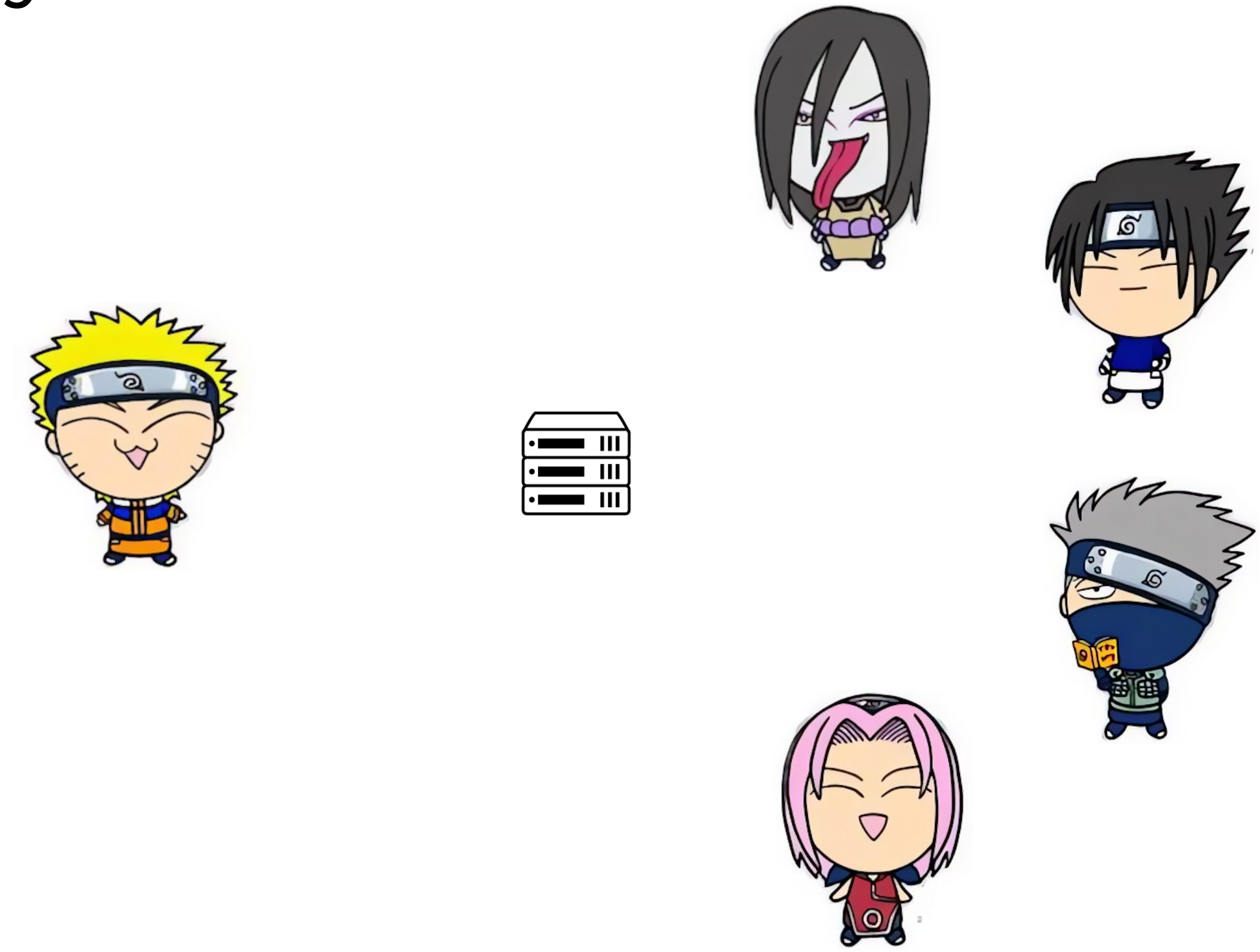
1. Insider Replay Attack

SIGN
group_key_id ❌

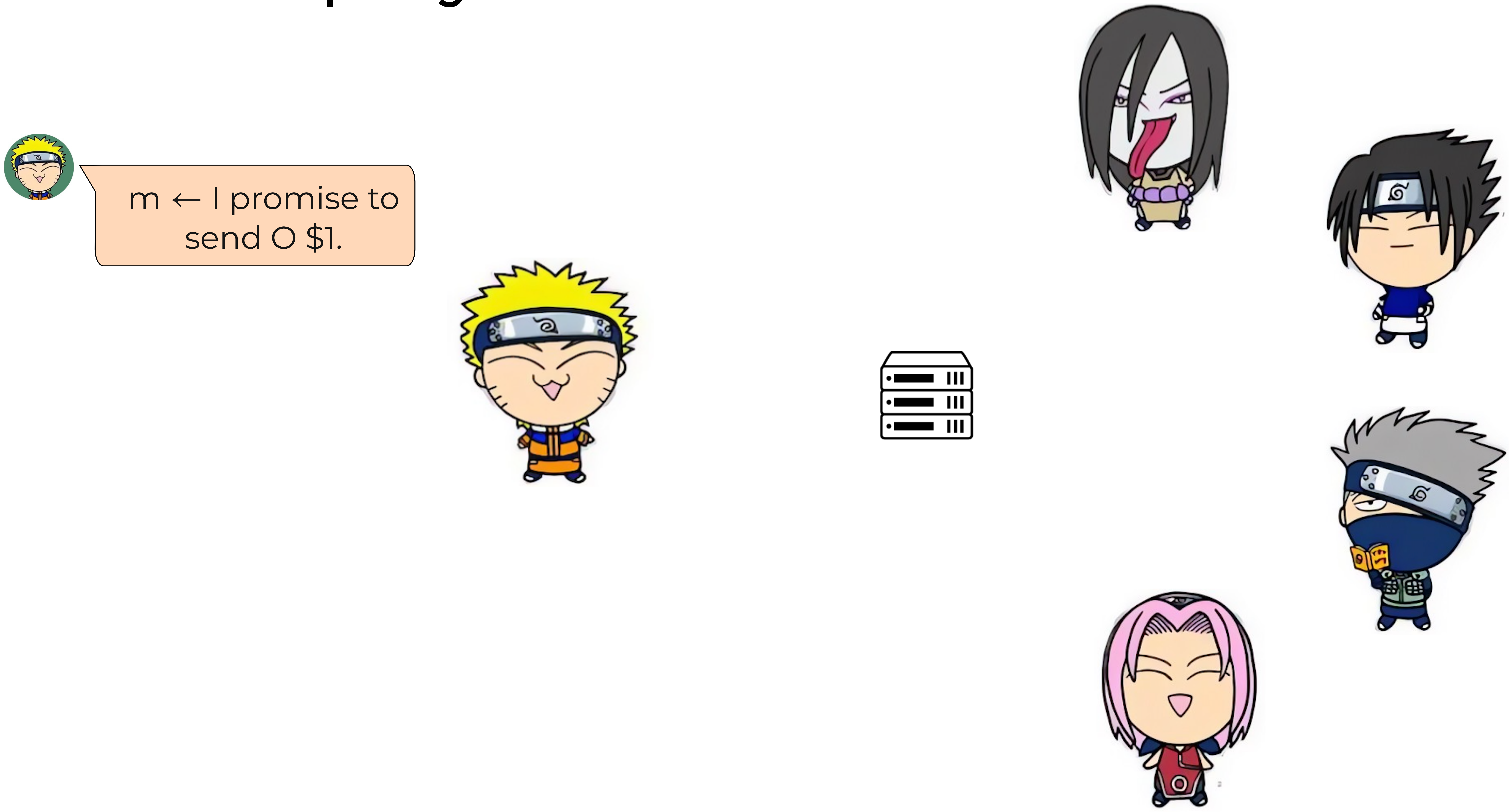
Insider Replay Attack

Insider Replay Attack

Insider Replay Attack



Insider Replay Attack



Insider Replay Attack

m ← I promise to send O $1.

MLS-Sign-then-Encrypt

$s \leftarrow \text{Sign}(sk_u,\ m_s)$

$c \leftarrow \text{Enc}(k_{g_0},\ n,\ s||m,\ ad_e)$

u : user_id
g_0 : key_id

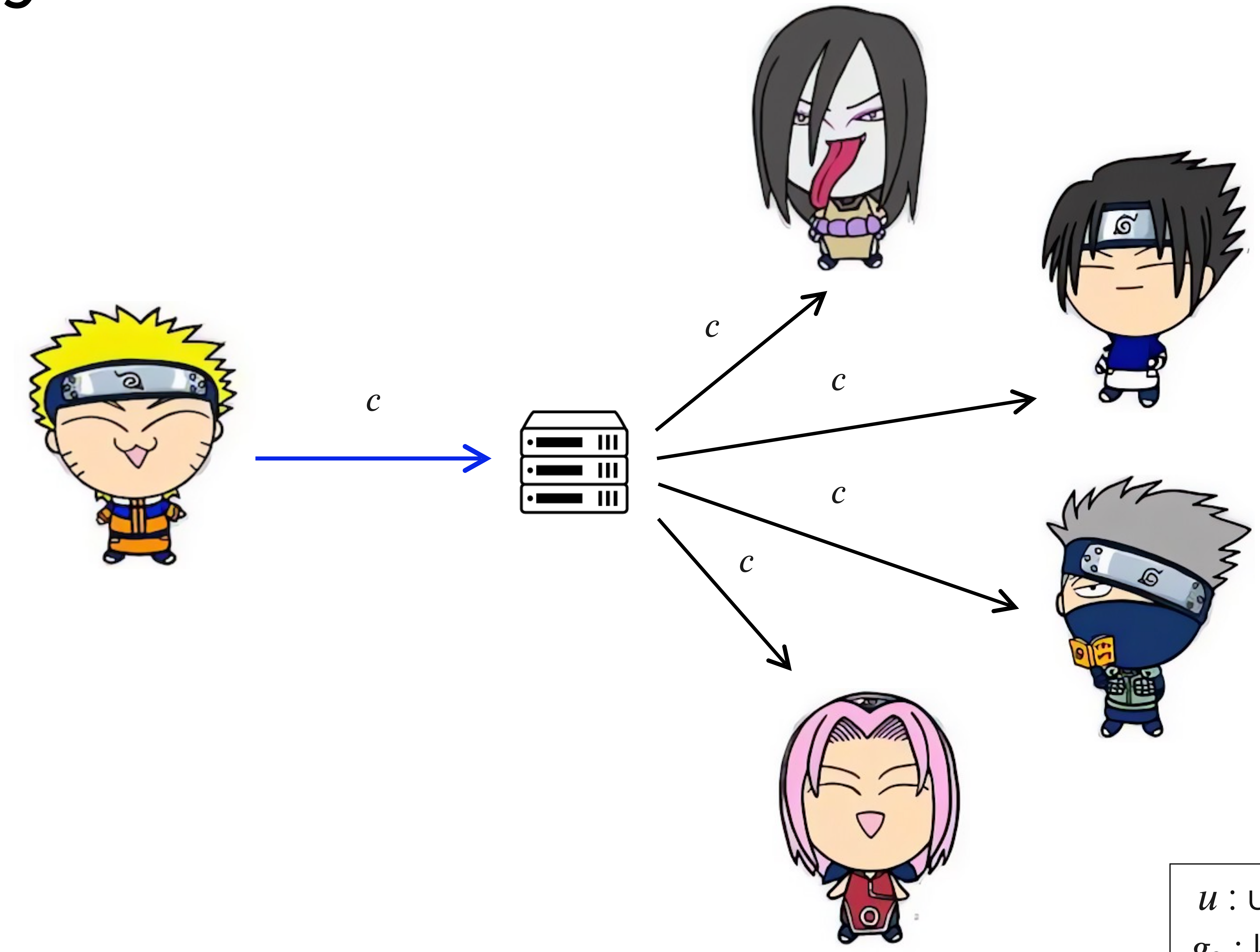
Insider Replay Attack

m ← I promise to send O $1.

MLS-Sign-then-Encrypt

$s \leftarrow \text{Sign}(sk_u,\ m_s)$

$c \leftarrow \text{Enc}(k_{g_0},\ n,\ s || m,\ ad_e)$



u : user_id
g_0 : key_id

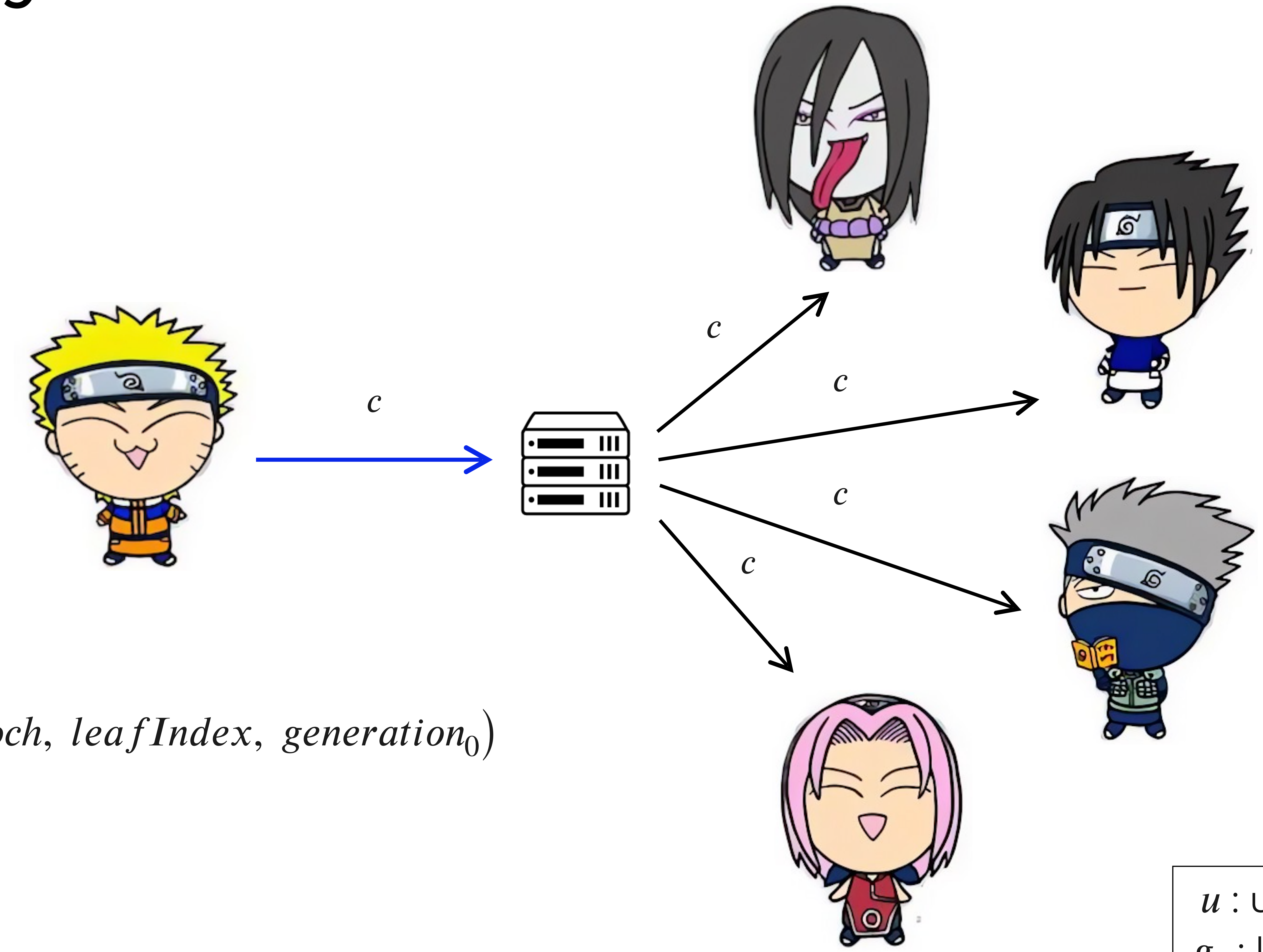
Insider Replay Attack

m ← I promise to send O $1.

<u>MLS-Sign-then-Encrypt</u>

$s \leftarrow$ Sign$(sk_u,\ m_s)$

$c \leftarrow$ Enc$(k_{g_0},\ n,\ s\,||\,m,\ ad_e)$

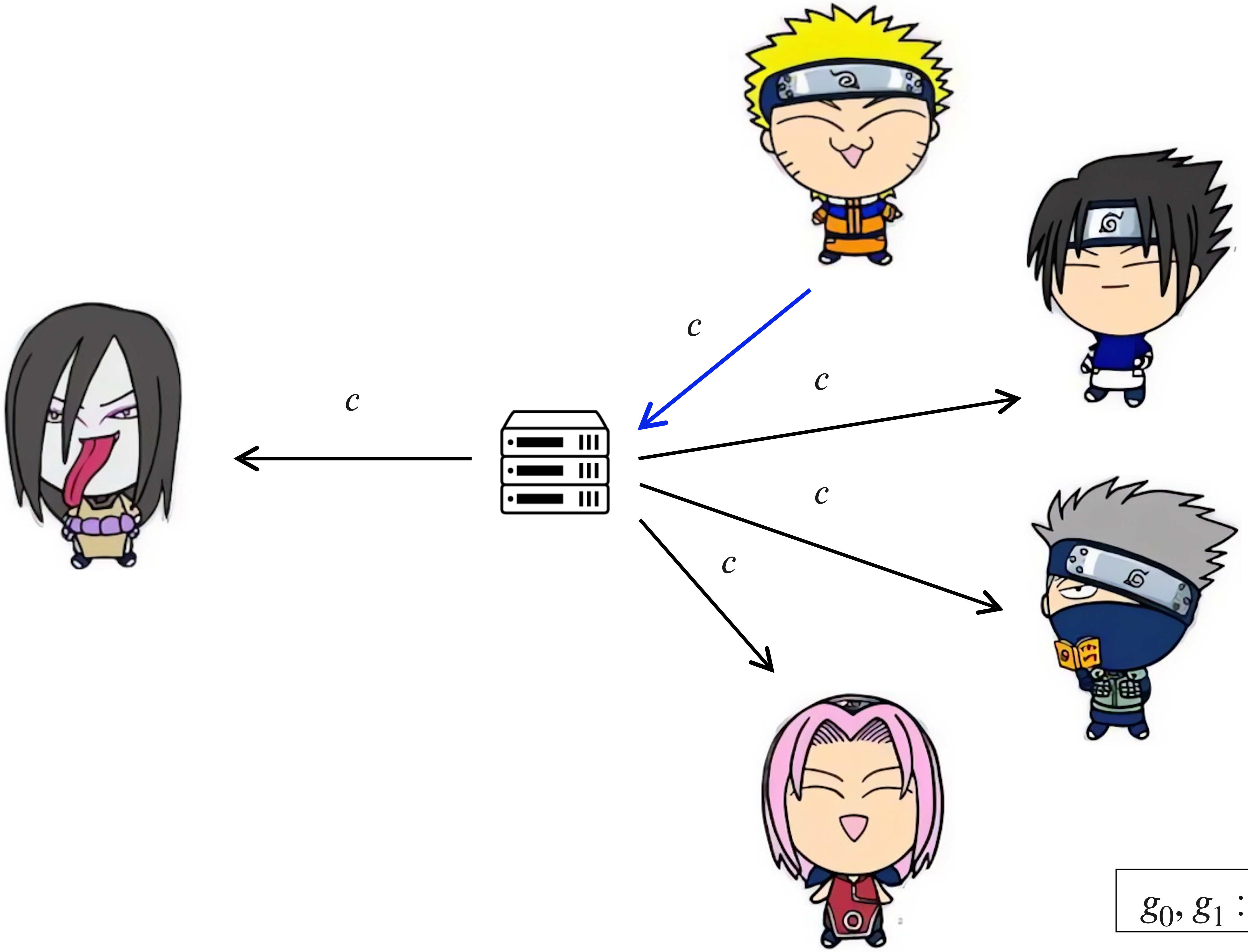


Let key identifier $g_0 = \left(group,\ epoch,\ leafIndex,\ generation_0\right)$

u : user_id
g_0 : key_id

Insider Replay Attack

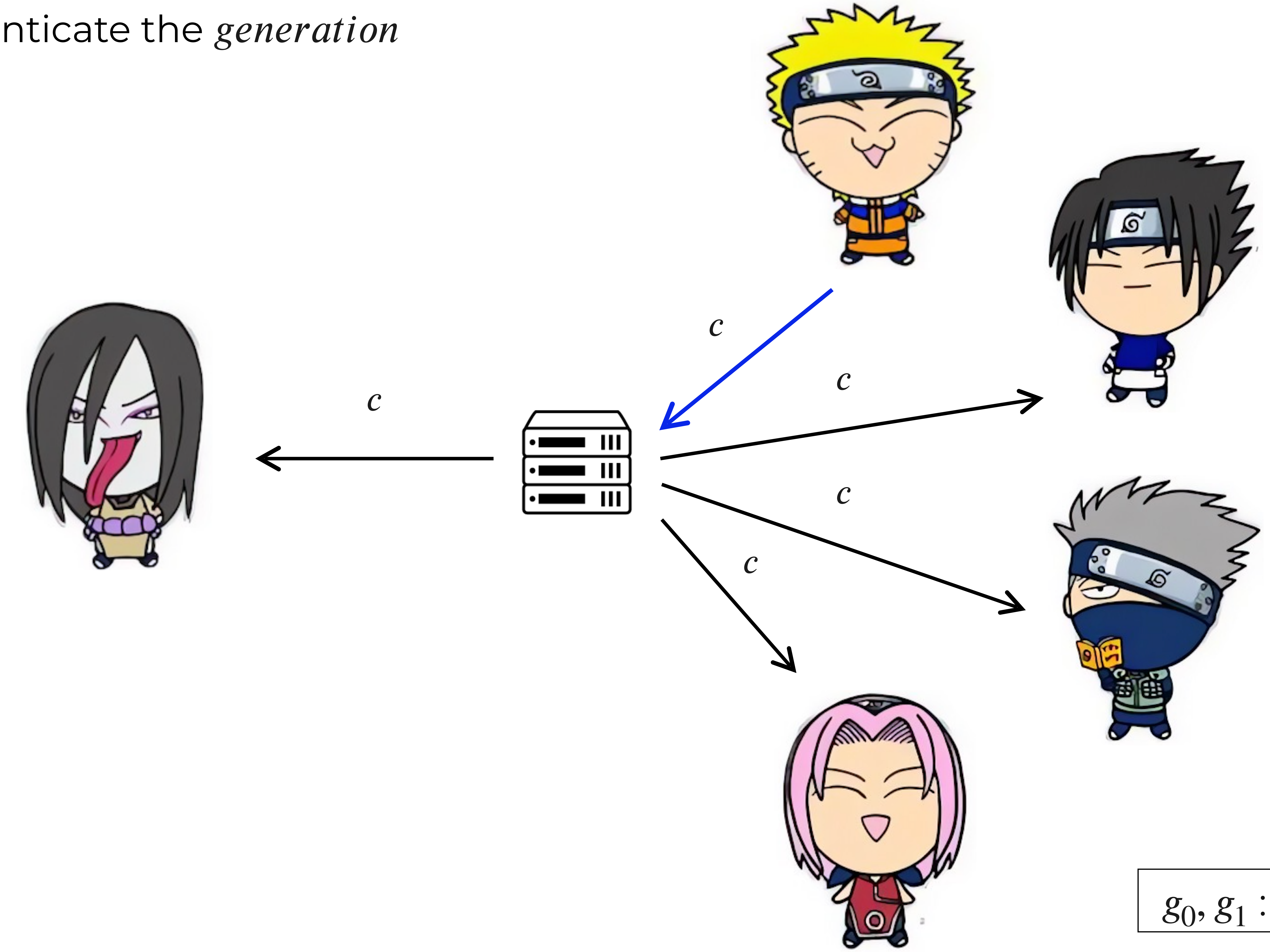
Insider Replay Attack



g_0, g_1 : key_id

Insider Replay Attack

Recall: signature s does not authenticate the *generation*



g_0, g_1 : key_id

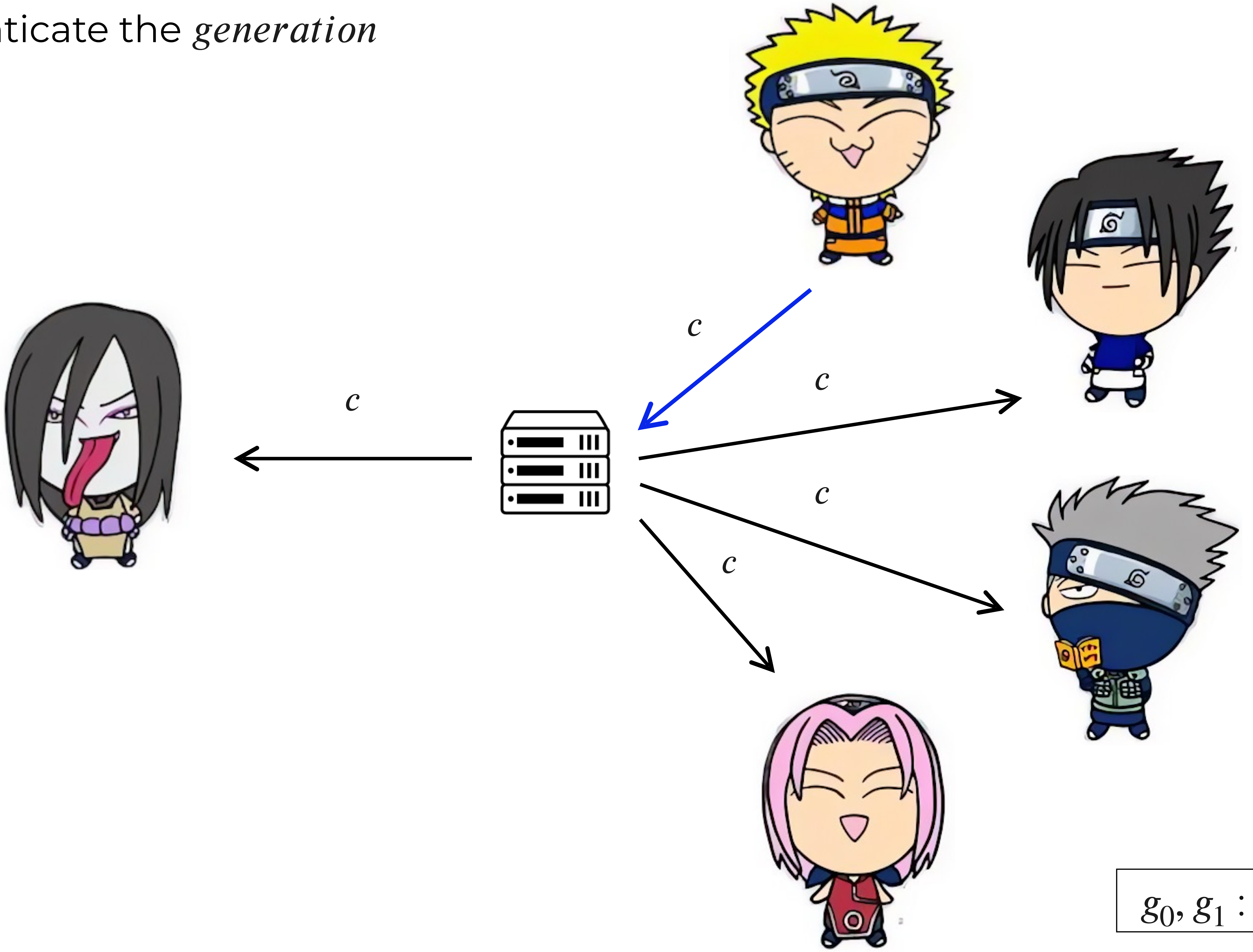
Insider Replay Attack

Recall: signature s does not authenticate the *generation*

<u>Decrypt-then-Re-Encrypt</u>

$s||m \leftarrow \text{Dec}(k_{g_0}, n, c, ad_e)$

$c' \leftarrow \text{Enc}(k_{g_1}, n, s||m, ad_e)$



g_0, g_1 : key_id

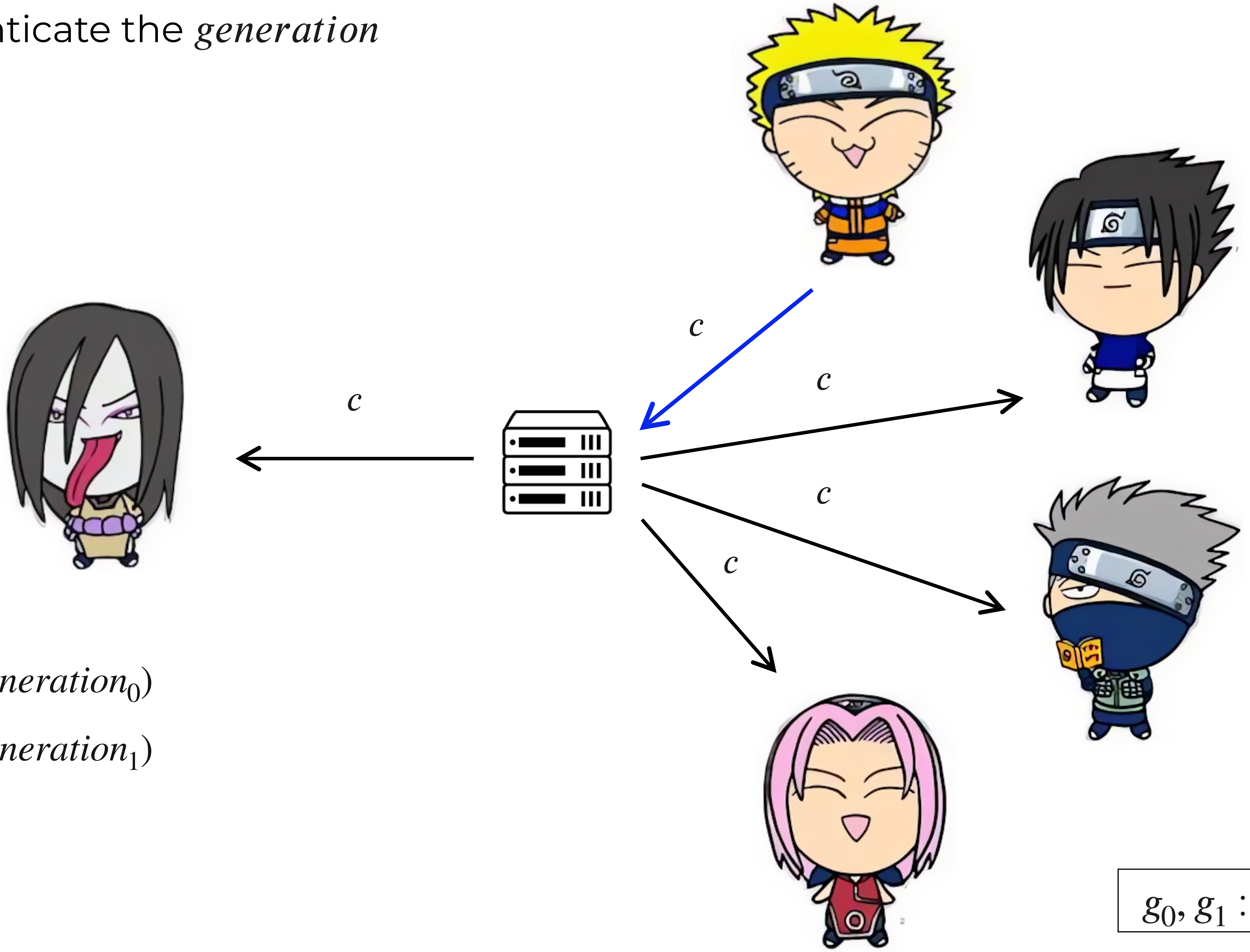
Insider Replay Attack

Recall: signature s does not authenticate the *generation*

<u>Decrypt-then-Re-Encrypt</u>

$s||m \leftarrow \text{Dec}(k_{g_0},\ n,\ c,\ ad_e)$

$c' \leftarrow \text{Enc}(k_{g_1},\ n,\ s||m,\ ad_e)$



$g_0 = (group, epoch, leafIndex, generation_0)$

$g_1 = (group, epoch, leafIndex, generation_1)$

$(generation_1 > generation_0)$

g_0, g_1 : key_id

Insider Replay Attack

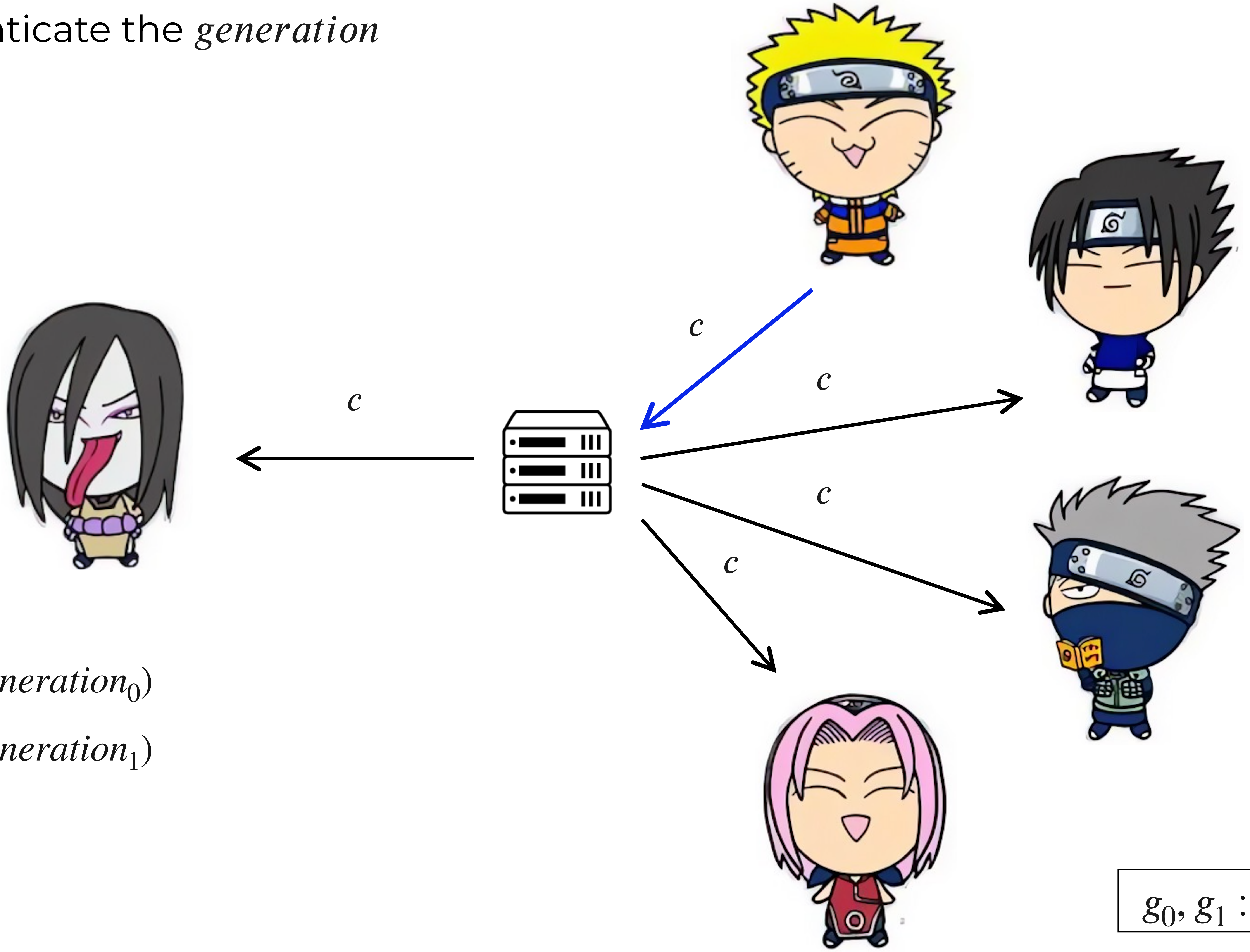
Recall: signature s does not authenticate the *generation*

<u>Decrypt-then-Re-Encrypt</u>

$s||m \leftarrow \text{Dec}(k_{g_0},\ n,\ c,\ ad_e)$

$c' \leftarrow \text{Enc}(k_{g_1},\ n,\ s||m,\ ad_e)$

Save for later



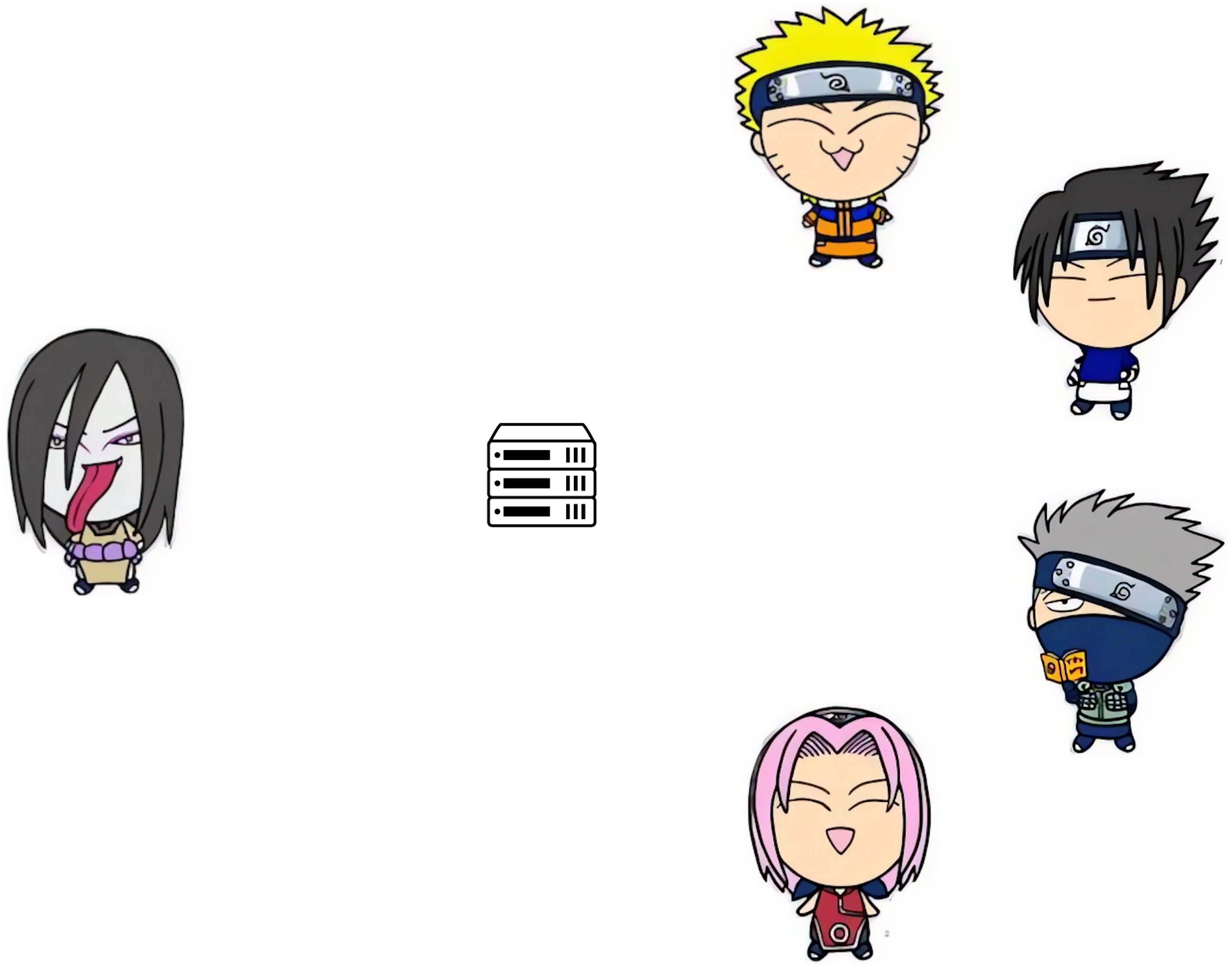
$g_0 = (group, epoch, leafIndex, generation_0)$

$g_1 = (group, epoch, leafIndex, generation_1)$

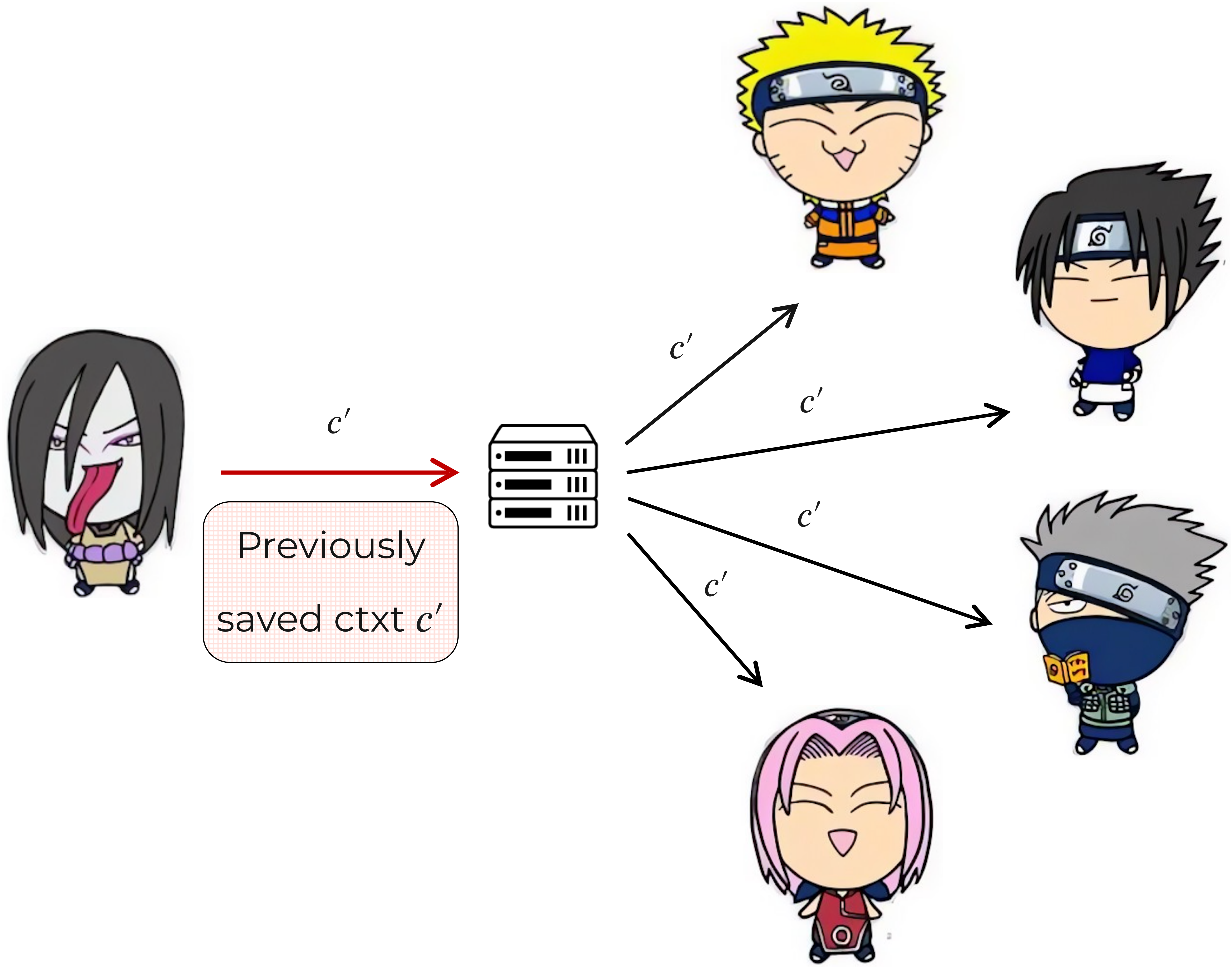
$(generation_1 > generation_0)$

g_0, g_1 : key_id

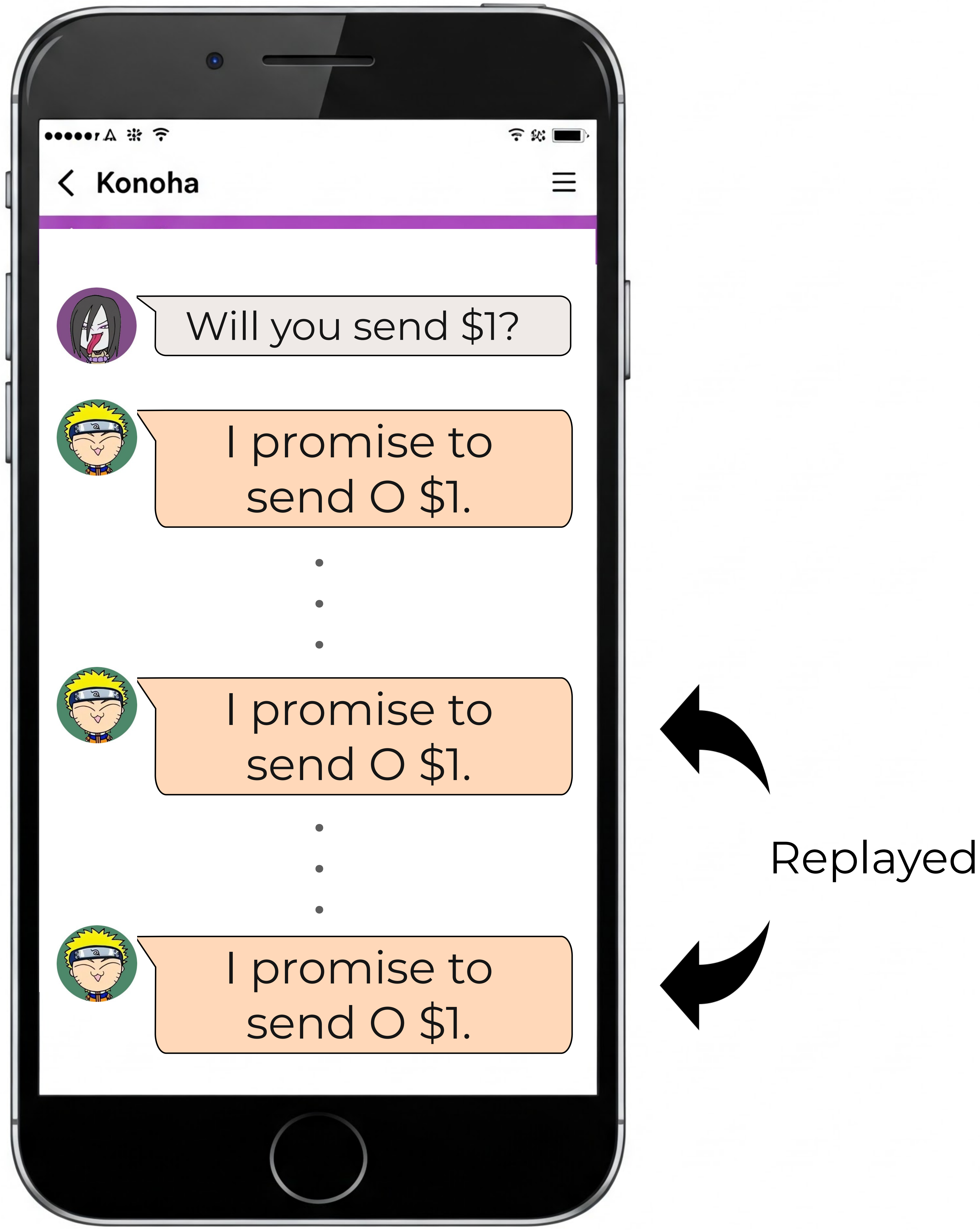
Insider Replay Attack



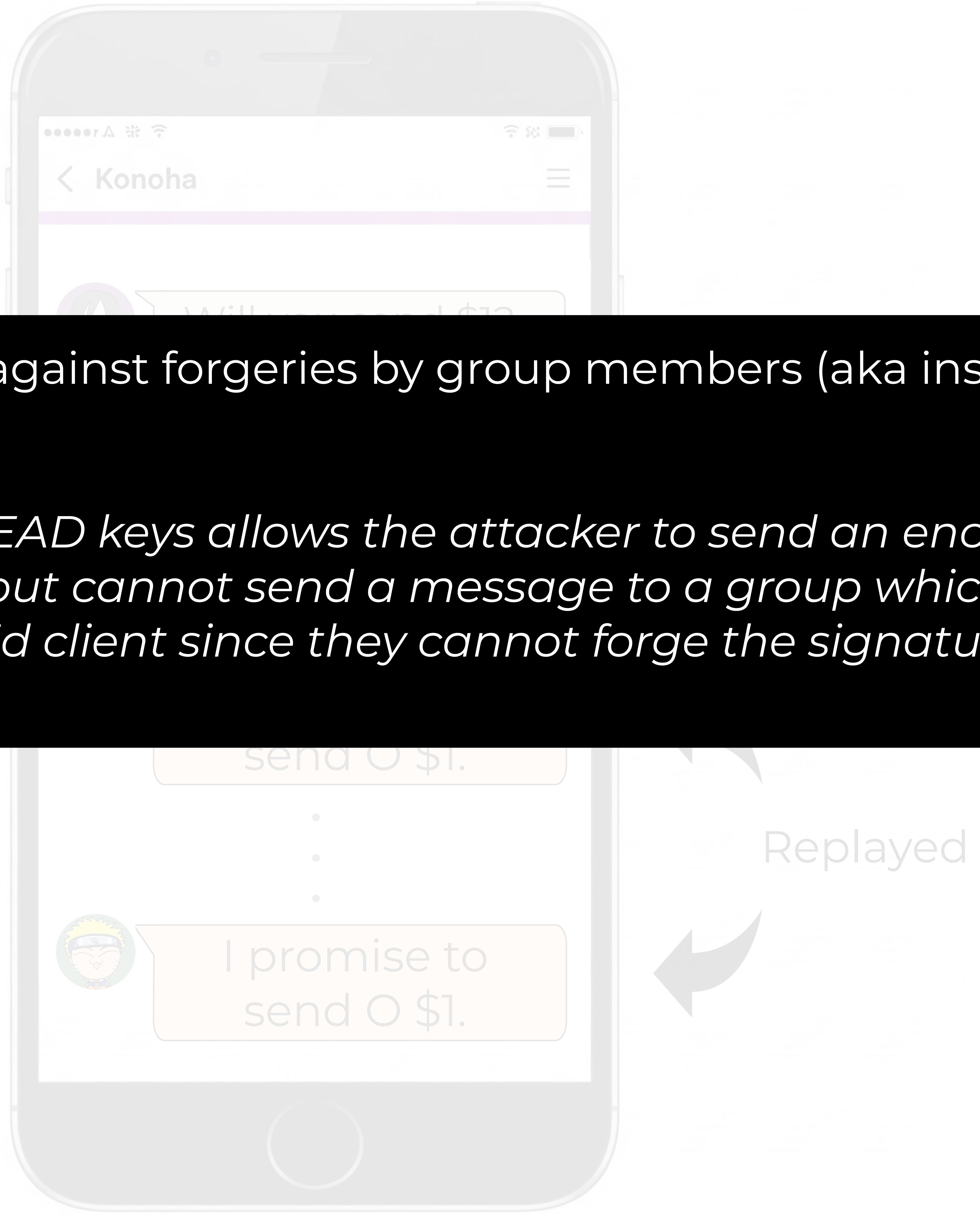
Insider Replay Attack



Insider Replay Attack



Insider Replay Attack



MLS aims to protect against forgeries by group members (aka insiders)

"[Knowledge] of the AEAD keys allows the attacker to send an encrypted message using that key, but cannot send a message to a group which appears to be from any valid client since they cannot forge the signature."

Mitigation and Disclosure

Mitigation and Disclosure

Attack results from lack of binding between signature and generation; mitigation is to bind them

Mitigation and Disclosure

Attack results from lack of binding between signature and generation; mitigation is to bind them

$$m_s = < m, group, epoch, leafIndex, ad >$$

Mitigation and Disclosure

Attack results from lack of binding between signature and generation; mitigation is to bind them

MLS already signs ad; could just include the generation in ad

$$m_s = < m, group, epoch, leafIndex, ad >$$

Mitigation and Disclosure

Attack results from lack of binding between signature and generation; mitigation is to bind them

MLS already signs ad; could just include the generation in ad

$$m_s = < m, group, epoch, leafIndex, ad >$$

☞ Disclosed our findings to the MLS WG by posting to the mailing list

Mitigation and Disclosure

Attack results from lack of binding between signature and generation; mitigation is to bind them

MLS already signs ad; could just include the generation in ad

$$m_s = < m, group, epoch, leafIndex, ad >$$

☞Disclosed our findings to the MLS WG by posting to the mailing list

☞Turn around time very quick ~couple hours, acknowledgement of findings

Mitigation and Disclosure

Attack results from lack of binding between signature and generation; mitigation is to bind them

MLS already signs ad; could just include the generation in ad

$$m_s = < m, group, epoch, leafIndex, ad >$$

☞ Disclosed our findings to the MLS WG by posting to the mailing list

☞ Turn around time very quick ~couple hours, acknowledgement of findings

☞ Presented to the WG at IETF 122 to discuss whether spec wants to address replays

Mitigation and Disclosure

Attack results from lack of binding between signature and generation; mitigation is to bind them

No Protection against Replay by Insiders

MLS does not protect against one group member replaying a PrivateMessage sent by another group member within the same epoch that the message was originally sent. Similarly, MLS does not protect against the replay (by a group member or otherwise) of a PublicMessage within the same epoch that the message was originally sent. Applications for whom replay is an important risk should apply mitigations at the application layer, as discussed below.

In addition to the risks discussed in {{symmetric-key-compromise}}, an attacker with access to the Ratchet Secrets for an endpoint can replay PrivateMessage objects sent by other members of the group by taking the signed content of the message and re-encrypting it with a new generation of the original sender's ratchet. If the other members of the group interpret a message with a new generation as a fresh message, then this message will appear fresh. (This is possible because the message signature does not cover the `generation` field of the message.) Messages sent as PublicMessage objects similarly lack replay protections. There is no message counter comparable to the `generation` field in PrivateMessage.

Applications can detect replay by including a unique identifier for the message (e.g., a counter) in either the message payload or the `authenticated_data` field, both of which are included in the signatures for PublicMessage and PrivateMessage.

- Disc

- Turn around time very quick ~couple hours, acknowledgement of findings

- Presented to the WG at IETF 122 to discuss whether spec wants to address replays

Interlude

Interlude

Session 5C: Messaging and Privacy — CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

Modular Design of Secure Group Messaging Protocols and the Security of MLS

Joël Alwen
AWS Wickr
alwenjo@amazon.com

Sandro Coretti
IOHK
sandro.coretti@iohk.io

Yevgeniy Dodis
New York University
dodis@cs.nyu.edu

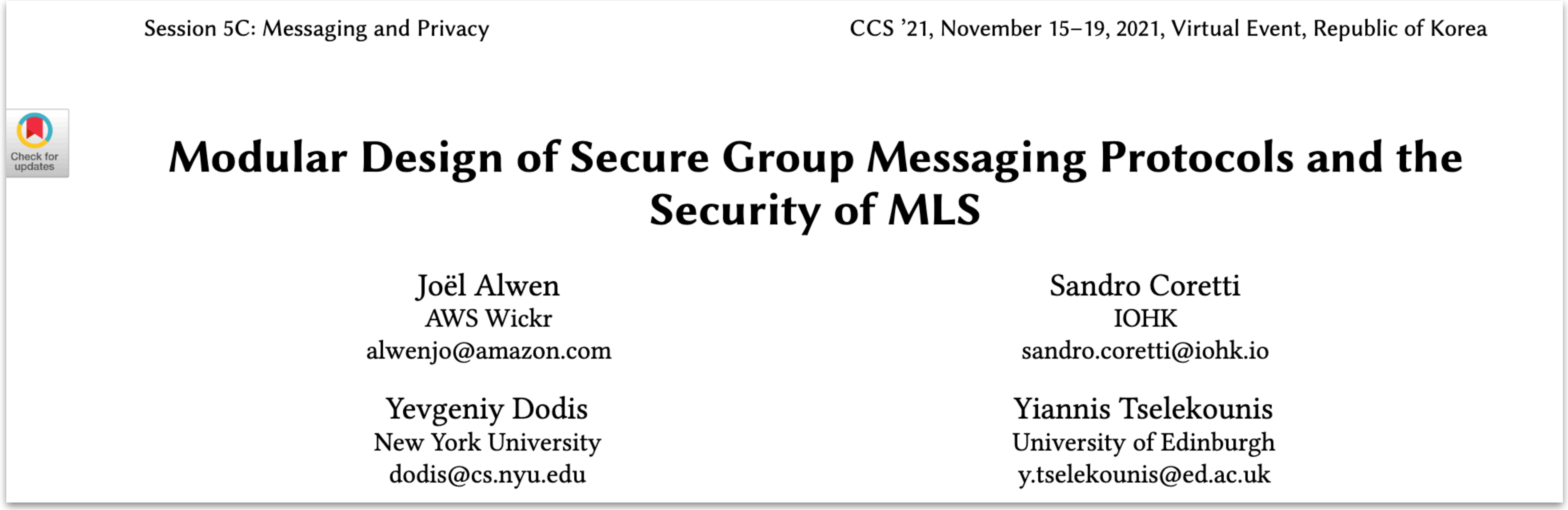
Yiannis Tselekounis
University of Edinburgh
y.tselekounis@ed.ac.uk

Modular Design of the Messaging Layer Security (MLS) Protocol

Master Thesis

Tijana Klimovic

Interlude


Session 5C: Messaging and Privacy

CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

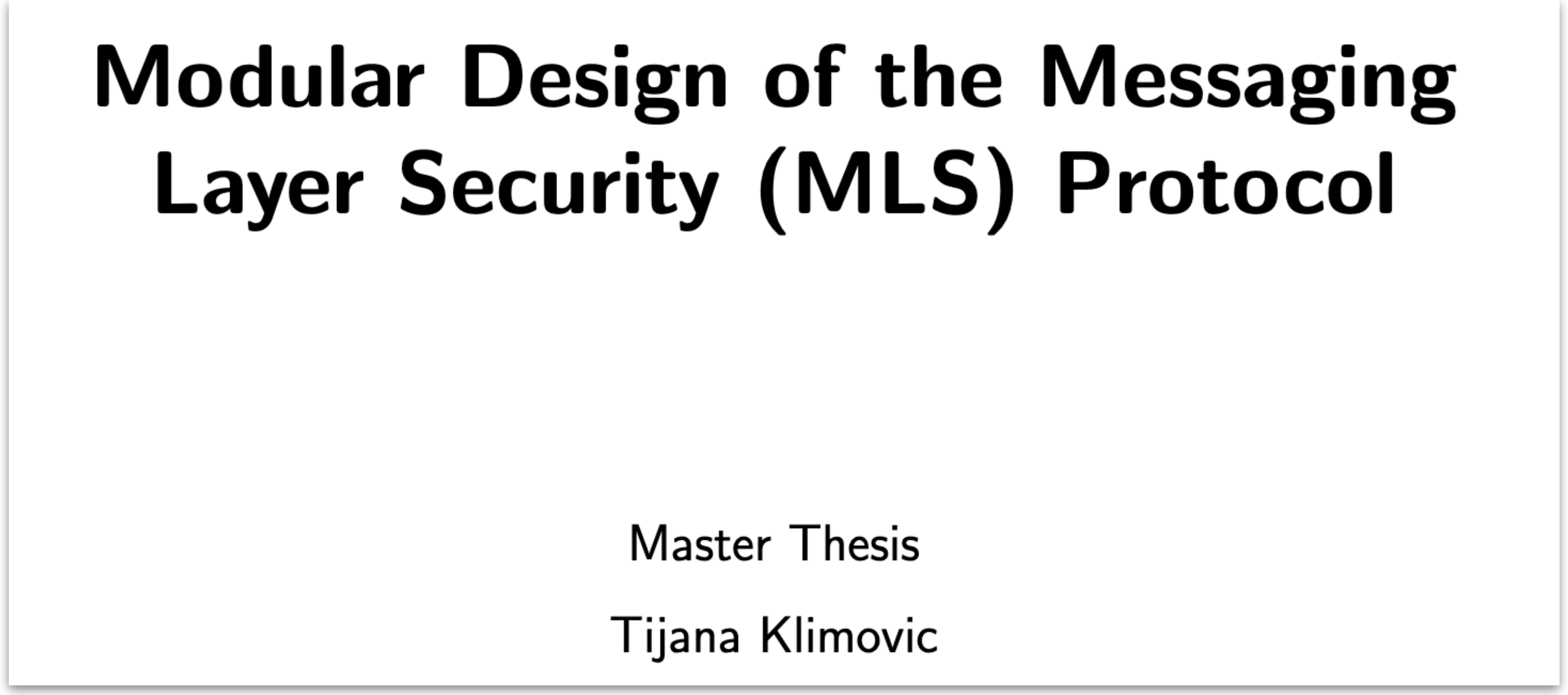
Modular Design of Secure Group Messaging Protocols and the Security of MLS

Joël Alwen
AWS Wickr
alwenjo@amazon.com

Sandro Coretti
IOHK
sandro.coretti@iohk.io

Yevgeniy Dodis
New York University
dodis@cs.nyu.edu

Yiannis Tselekounis
University of Edinburgh
y.tselekounis@ed.ac.uk


Modular Design of the Messaging Layer Security (MLS) Protocol

Master Thesis

Tijana Klimovic

Modeled but did not formally analyze MLS chat encryption

Interlude

Session 5C: Messaging and Privacy   CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

Check for updates

Modular Design of Secure Group Messaging Protocols and the Security of MLS

Joël Alwen
AWS Wickr
alwenjo@amazon.com

Sandro Coretti
IOHK
sandro.coretti@iohk.io

Yevgeniy Dodis
New York University
dodis@cs.nyu.edu

Yiannis Tselekounis
University of Edinburgh
y.tselekounis@ed.ac.uk

Modeled MLS chat encryption as Encrypt-then-Sign

Modular Design of the Messaging Layer Security (MLS) Protocol

Master Thesis

Tijana Klimovic

Modeled but did not formally analyze MLS chat encryption

Interlude

Session 5C: Messaging and Privacy   CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

Modular Design of Secure Group Messaging Protocols and the Security of MLS

Joël Alwen
AWS Wickr
alwenjo@amazon.com

Sandro Coretti
IOHK
sandro.coretti@iohk.io

Yevgeniy Dodis
New York University
dodis@cs.nyu.edu

Yiannis Tselekounis
University of Edinburgh
y.tselekounis@ed.ac.uk

Modeled MLS chat encryption as Encrypt-then-Sign

Signed message generation

Modular Design of the Messaging Layer Security (MLS) Protocol

Master Thesis

Tijana Klimovic

Modeled but did not formally analyze MLS chat encryption

Interlude

Session 5C: Messaging and Privacy — CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

Modular Design of Secure Group Messaging Protocols and the Security of MLS

Joël Alwen
AWS Wickr
alwenjo@amazon.com

Sandro Coretti
IOHK
sandro.coretti@iohk.io

Yevgeniy Dodis
New York University
dodis@cs.nyu.edu

Yiannis Tselekounis
University of Edinburgh
y.tselekounis@ed.ac.uk

Modeled MLS chat encryption as Encrypt-then-Sign

Signed message generation

Security game defines corruption as learning both symmetric and signing keys

Modular Design of the Messaging Layer Security (MLS) Protocol

Master Thesis

Tijana Klimovic

Modeled but did not formally analyze MLS chat encryption

Interlude

Session 5C: Messaging and Privacy — CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

Modular Design of Secure Group Messaging Protocols and the Security of MLS

Joël Alwen
AWS Wickr
alwenjo@amazon.com

Sandro Coretti
IOHK
sandro.coretti@iohk.io

Yevgeniy Dodis
New York University
dodis@cs.nyu.edu

Yiannis Tselekounis
University of Edinburgh
y.tselekounis@ed.ac.uk

Modular Design of the Messaging Layer Security (MLS) Protocol

Master Thesis

Tijana Klimovic

Modeled MLS chat encryption as Encrypt-then-Sign

Modeled but did not formally analyze MLS chat encryption

Signed message generation

```
struct {
    uint8  group[32];
    uint32 epoch;
    uint32 generation;
    uint32 sender;
    opaque content<0..2^32-1>;
} MLSSignatureContent;
```

Version 1

Security game defines corruption as learning both symmetric and signing keys

Interlude

Session 5C: Messaging and Privacy CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

Check for updates

Modular Design of Secure Group Messaging Protocols and the Security of MLS

Joël Alwen
AWS Wickr
alwenjo@amazon.com

Sandro Coretti
IOHK
sandro.coretti@iohk.io

Yevgeniy Dodis
New York University
dodis@cs.nyu.edu

Yiannis Tselekounis
University of Edinburgh
y.tselekounis@ed.ac.uk

Modular Design of the Messaging Layer Security (MLS) Protocol

Master Thesis

Tijana Klimovic

Modeled MLS chat encryption as Encrypt-then-Sign

Signed message generation

Security game defines corruption as learning both symmetric and signing keys

Modeled but did not formally analyze MLS chat encryption

```
struct {
    uint8  group[32];
    uint32 epoch;
    uint32 generation;
    uint32 sender;
    opaque content<0..2^32-1>;
} MLSSignatureContent;
```

Version 1

```
struct {
    opaque group_id<0..255>;
    uint32 epoch;
    uint32 sender;
    ContentType content_type;

    select (MLSPlaintext.content_type) {
        case handshake:
            GroupOperation operation;

        case application:
            opaque application_data<0..2^32-1>;
    }

    opaque signature<0..2^16-1>;
} MLSPlaintext;
```

Version 5

Case Study II: Session

Insider Replay Attack in Session

Insider Replay Attack in Session

MLS Insider Replay $\approx$ Session Insider Replay

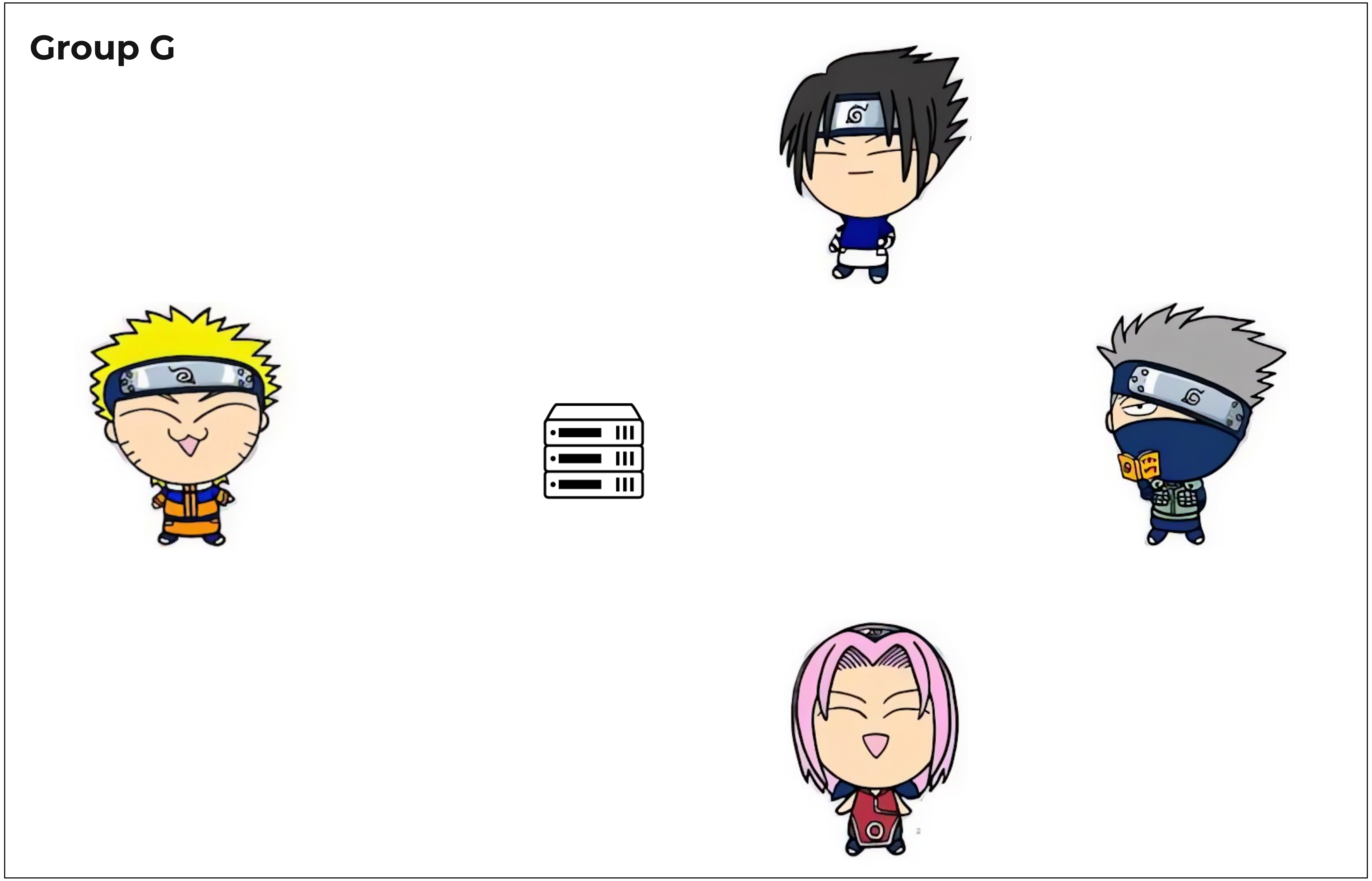
Insider Replay Attack in Session

MLS Insider Replay $\approx$ Session Insider Replay

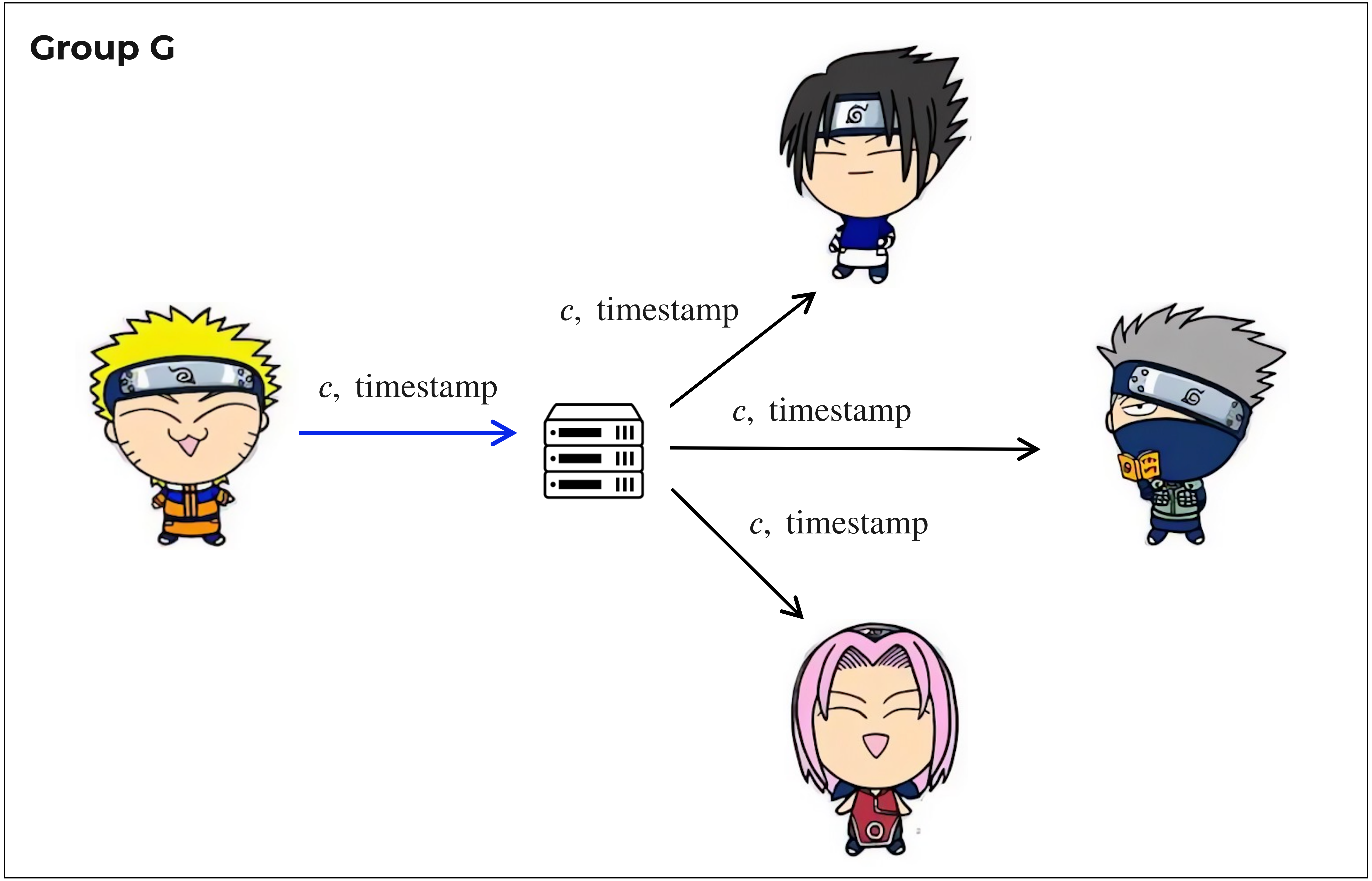
SIGN
group_key_id

Outsider Replay Attack in Session

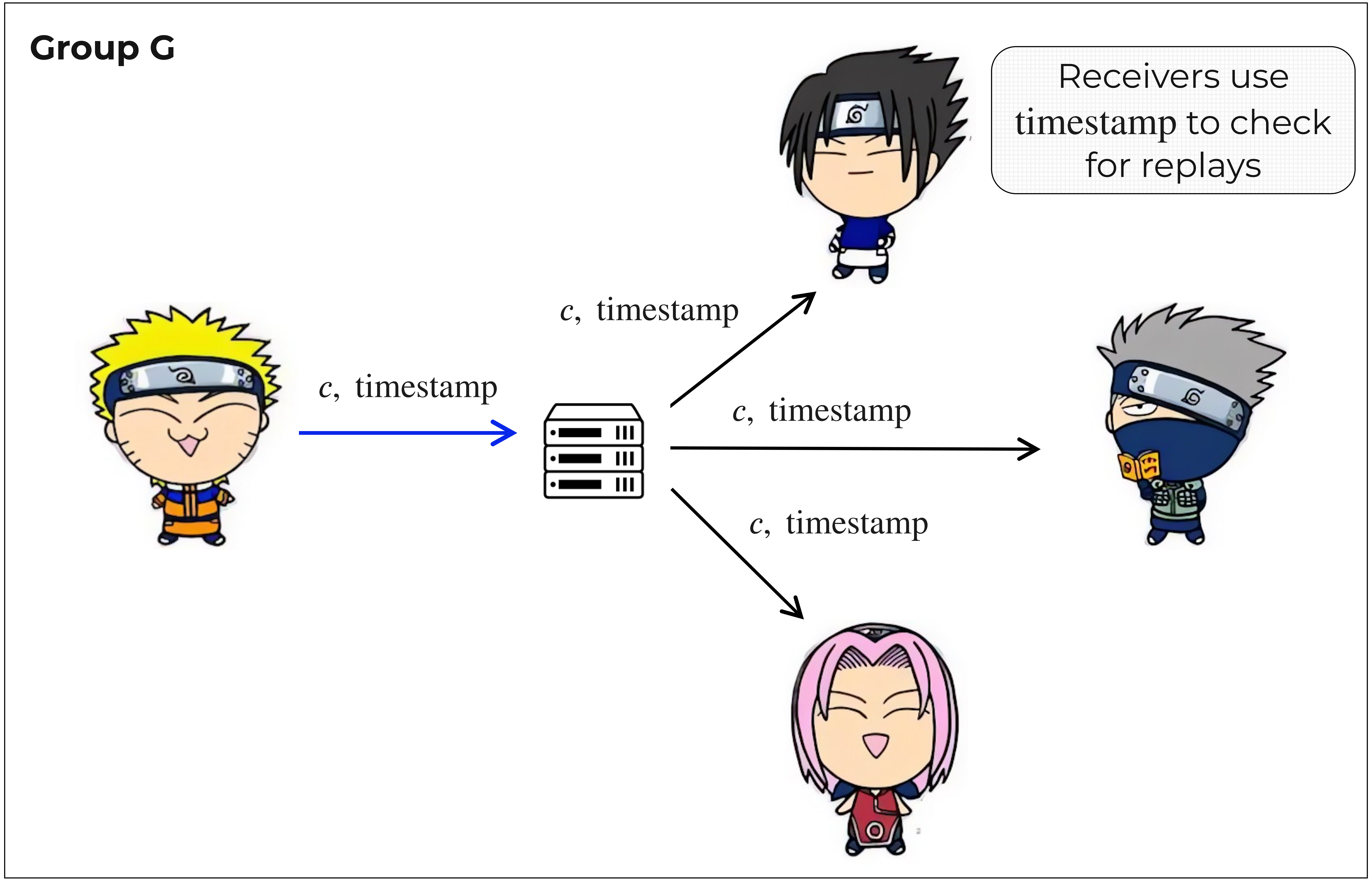
Outsider Replay Attack in Session



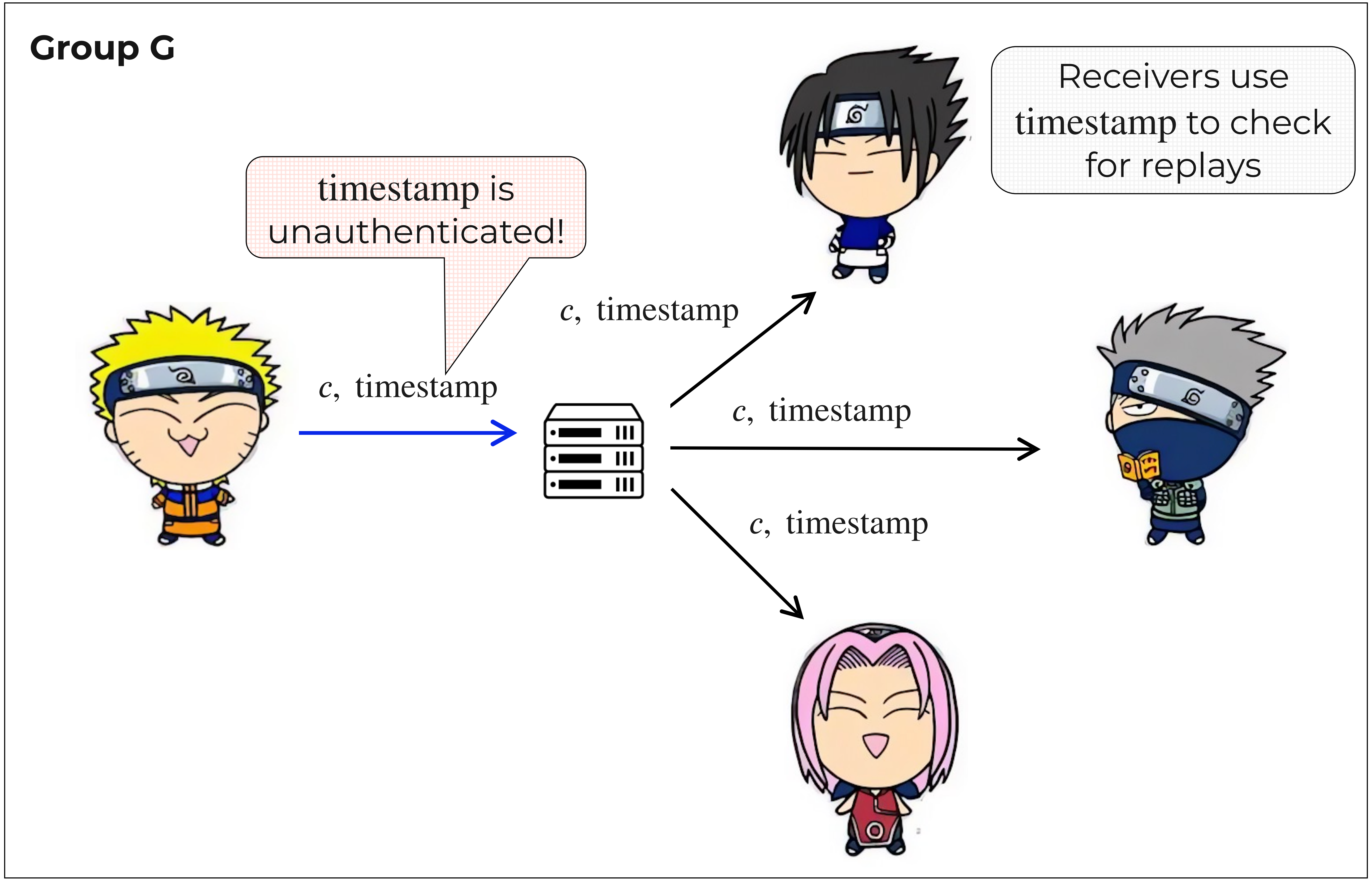
Outsider Replay Attack in Session



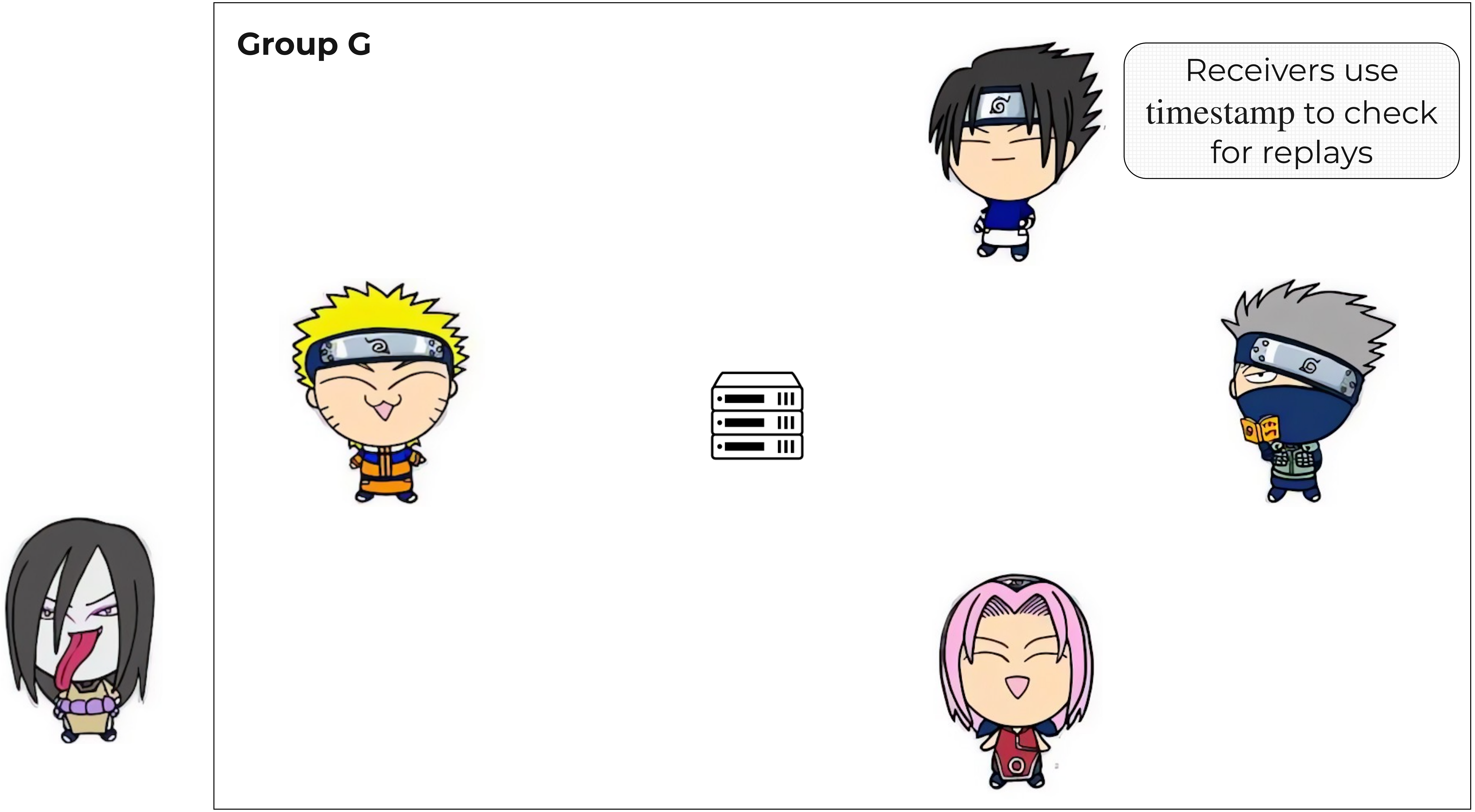
Outsider Replay Attack in Session



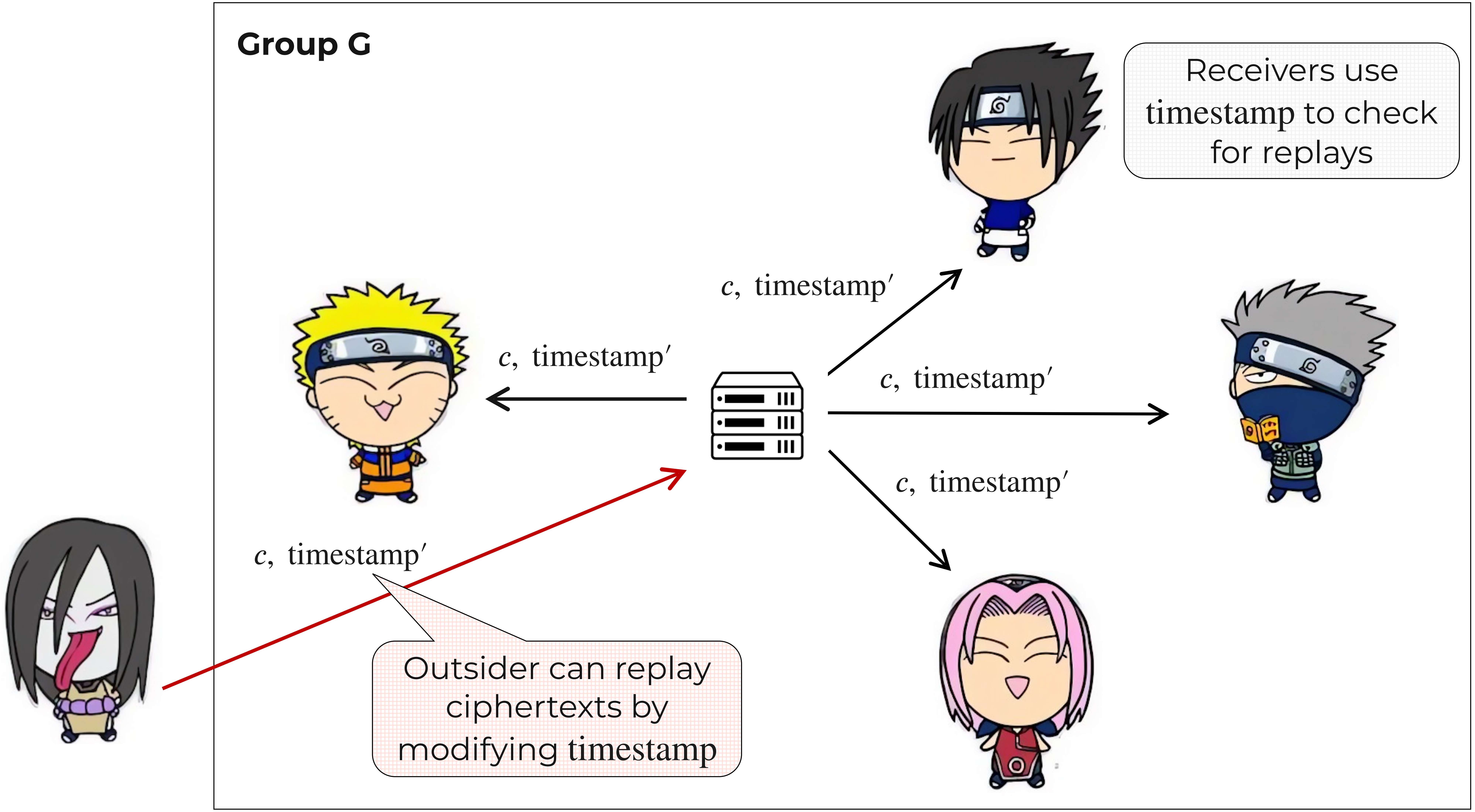
Outsider Replay Attack in Session



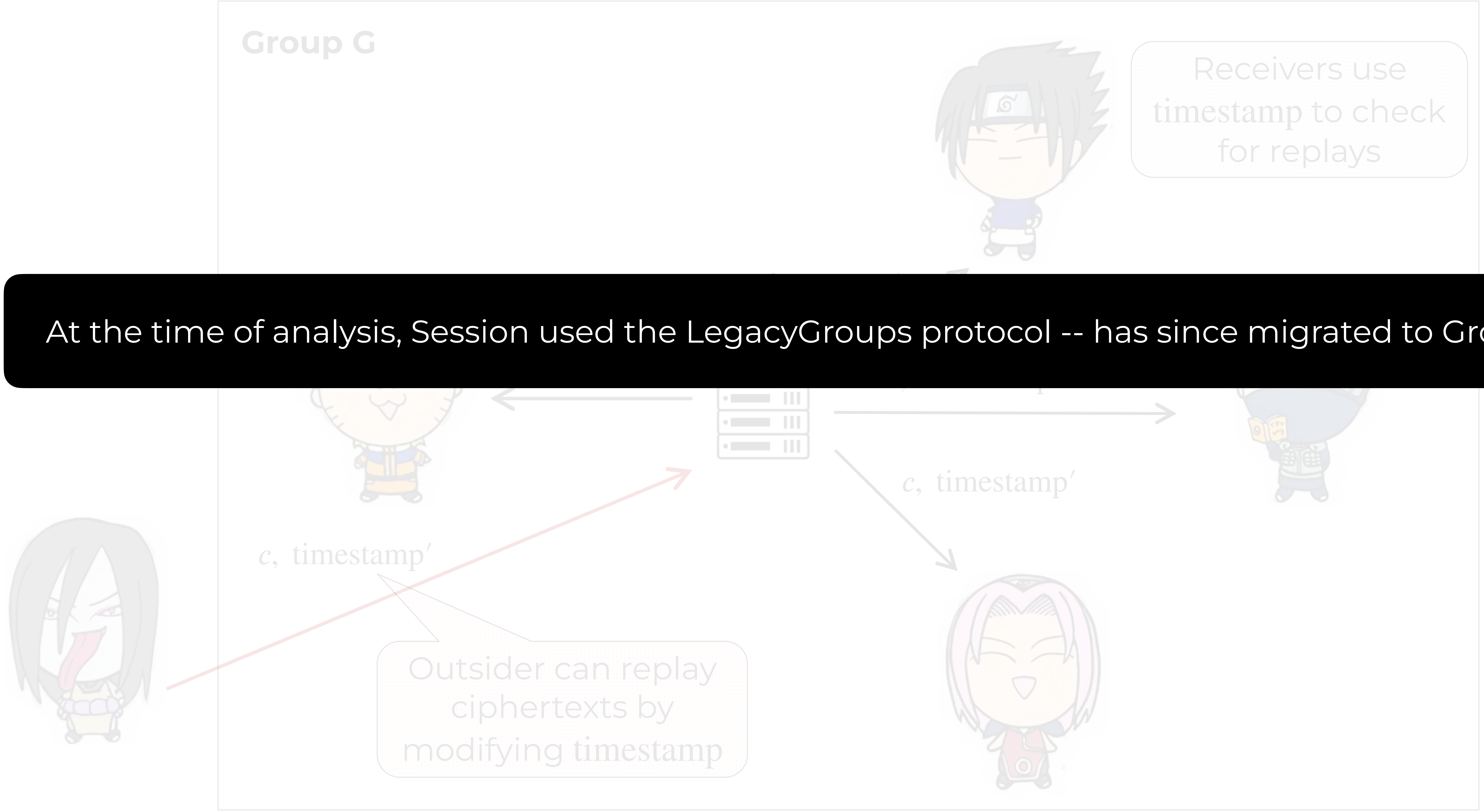
Outsider Replay Attack in Session



Outsider Replay Attack in Session



Outsider Replay Attack in Session



At the time of analysis, Session used the LegacyGroups protocol -- has since migrated to GroupsV2

Takeaways

Takeaways

- Message and sender authenticity are important in group messaging settings (sometimes even more than privacy)

Takeaways

- Message and sender authenticity are important in group messaging settings (sometimes even more than privacy)

- Combining encryption with MAC/signatures is non-trivial

Takeaways

- Message and sender authenticity are important in group messaging settings (sometimes even more than privacy)

- Combining encryption with MAC/signatures is non-trivial

- Formal definitions are useful to analyze real-world security

Takeaways

- Message and sender authenticity are important in group messaging settings (sometimes even more than privacy)

- Combining encryption with MAC/signatures is non-trivial

- Formal definitions are useful to analyze real-world security

Details in the paper! eprint.iacr.org/2025/554