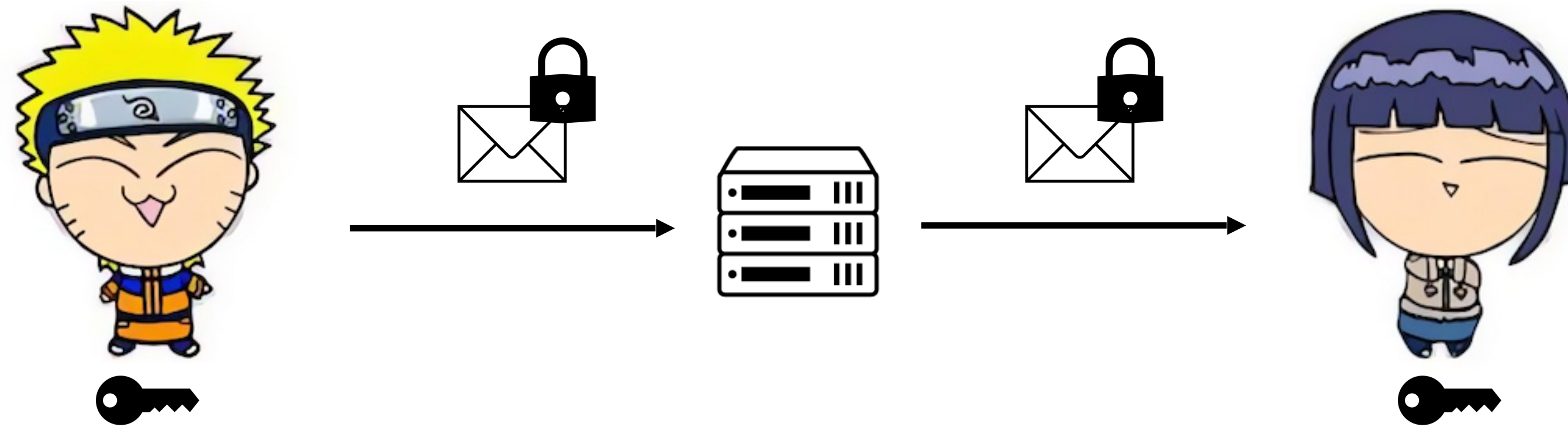


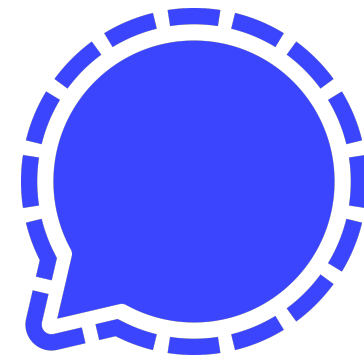
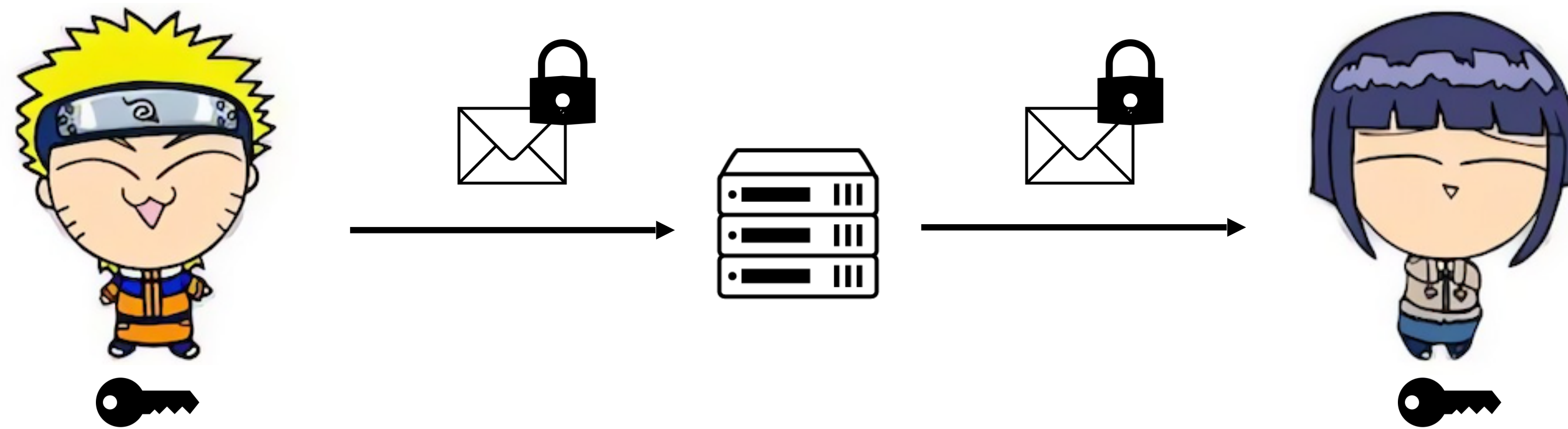
Analyzing Chat Encryption in Group Messaging Applications

Joseph Jaeger*, Akshaya Kumar*, and Igors Stepanovs

E2EE/Secure Messaging



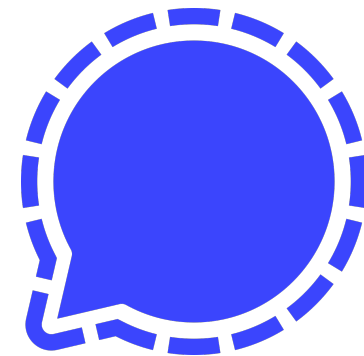
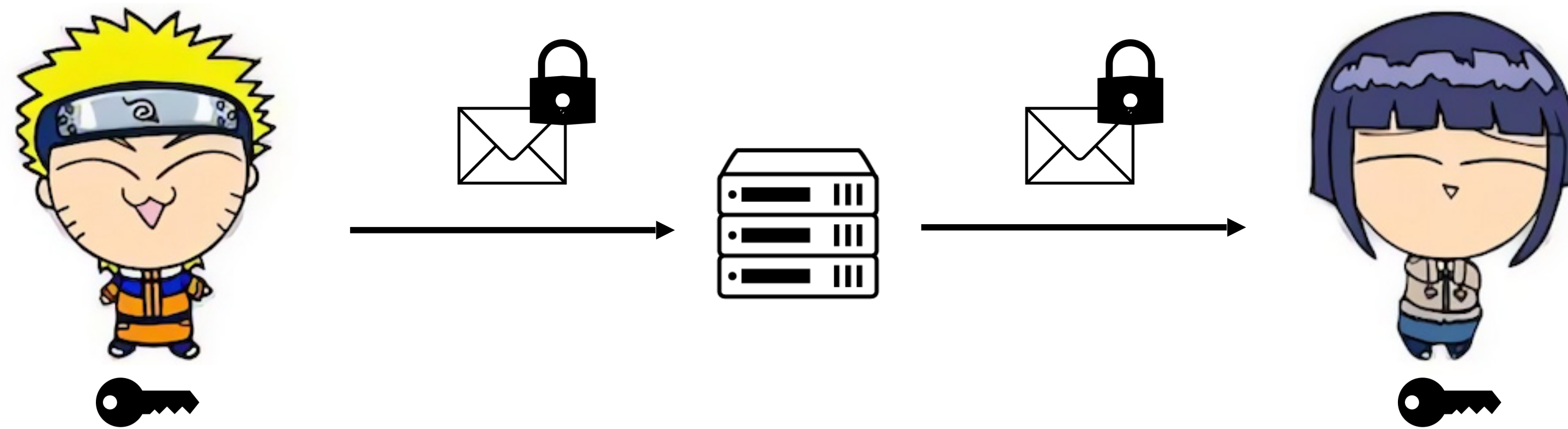
E2EE/Secure Messaging



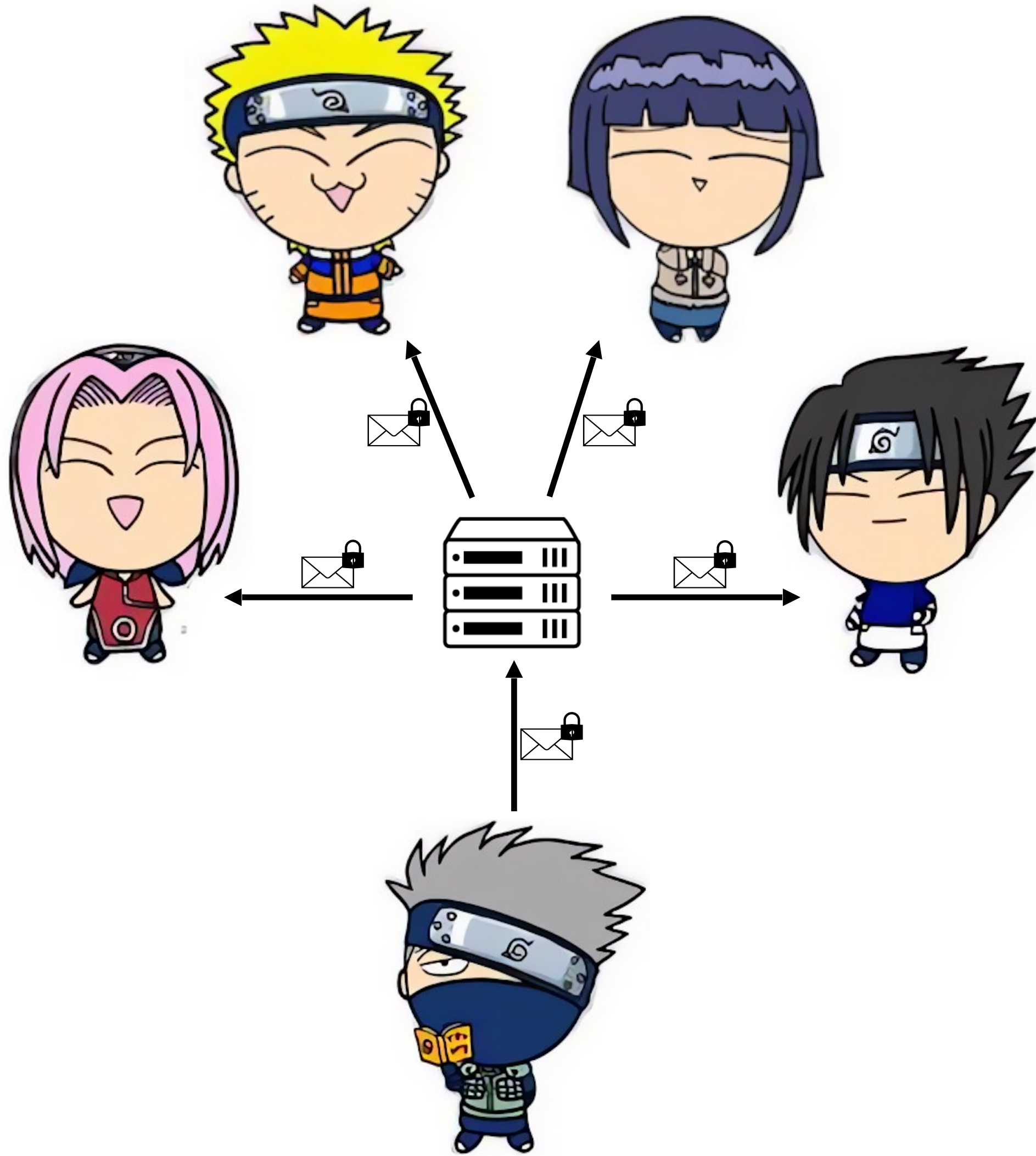
[matrix]



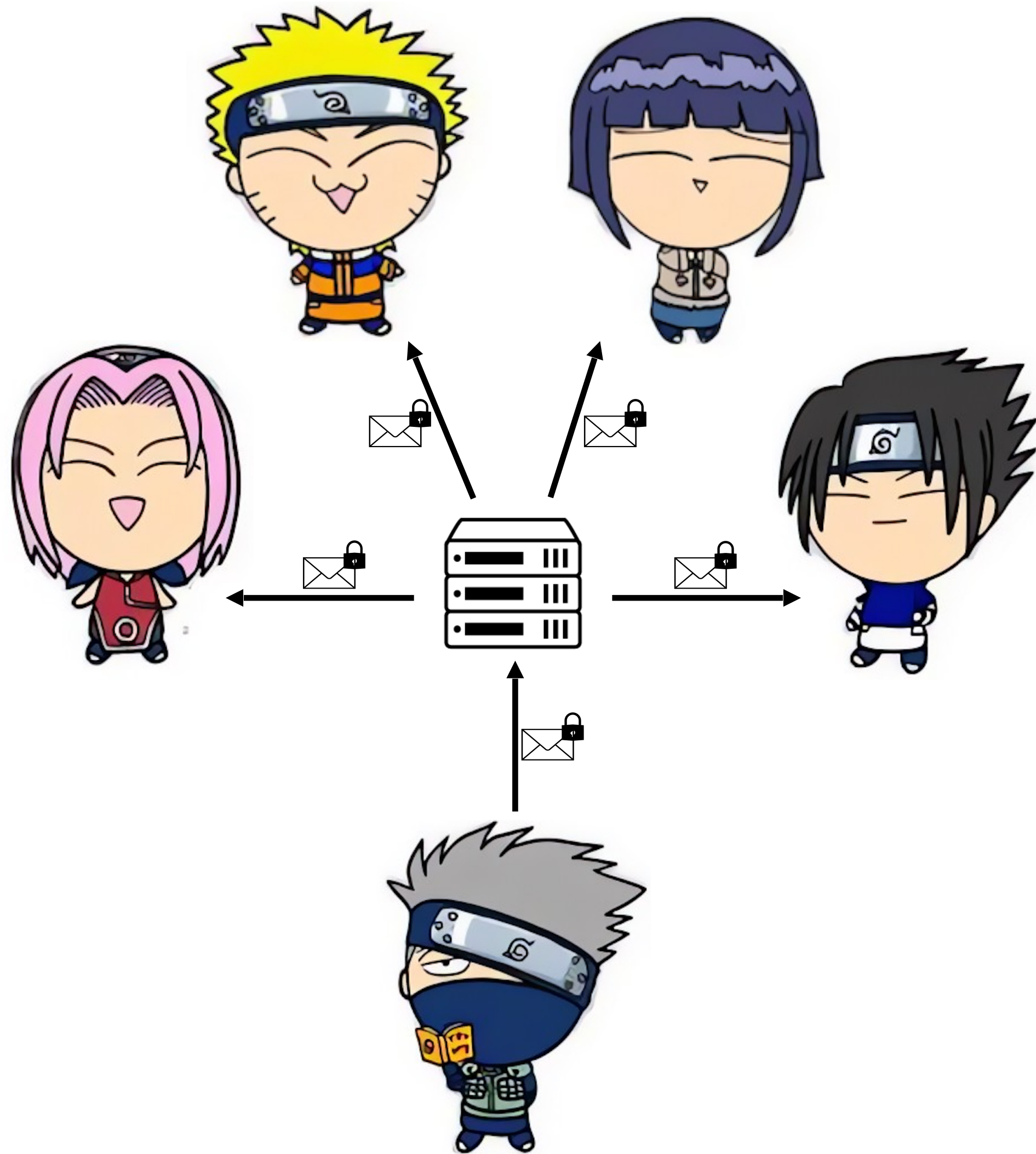
E2EE/Secure Messaging



Secure Group Messaging

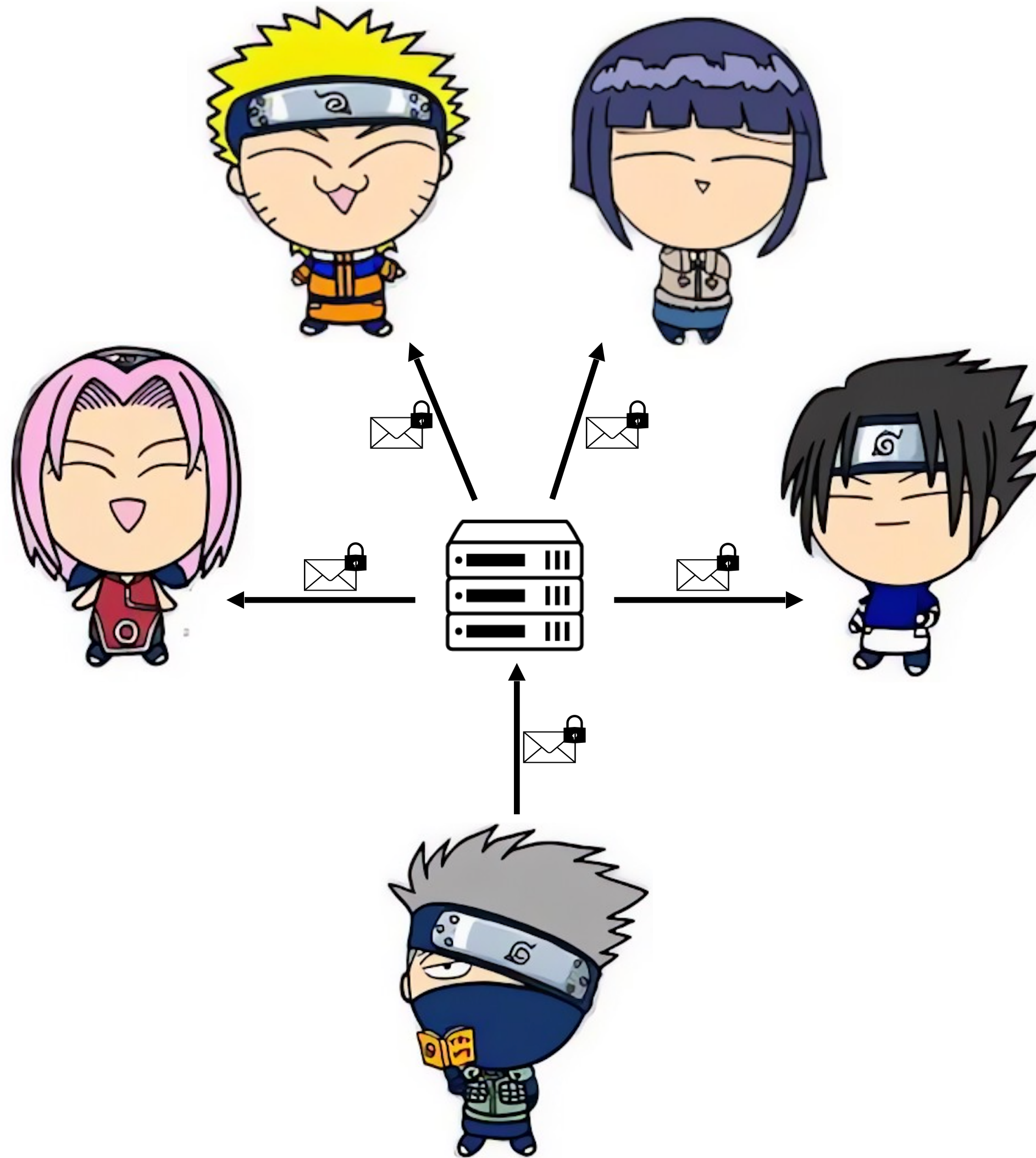


Secure Group Messaging



Secure Group
Messaging

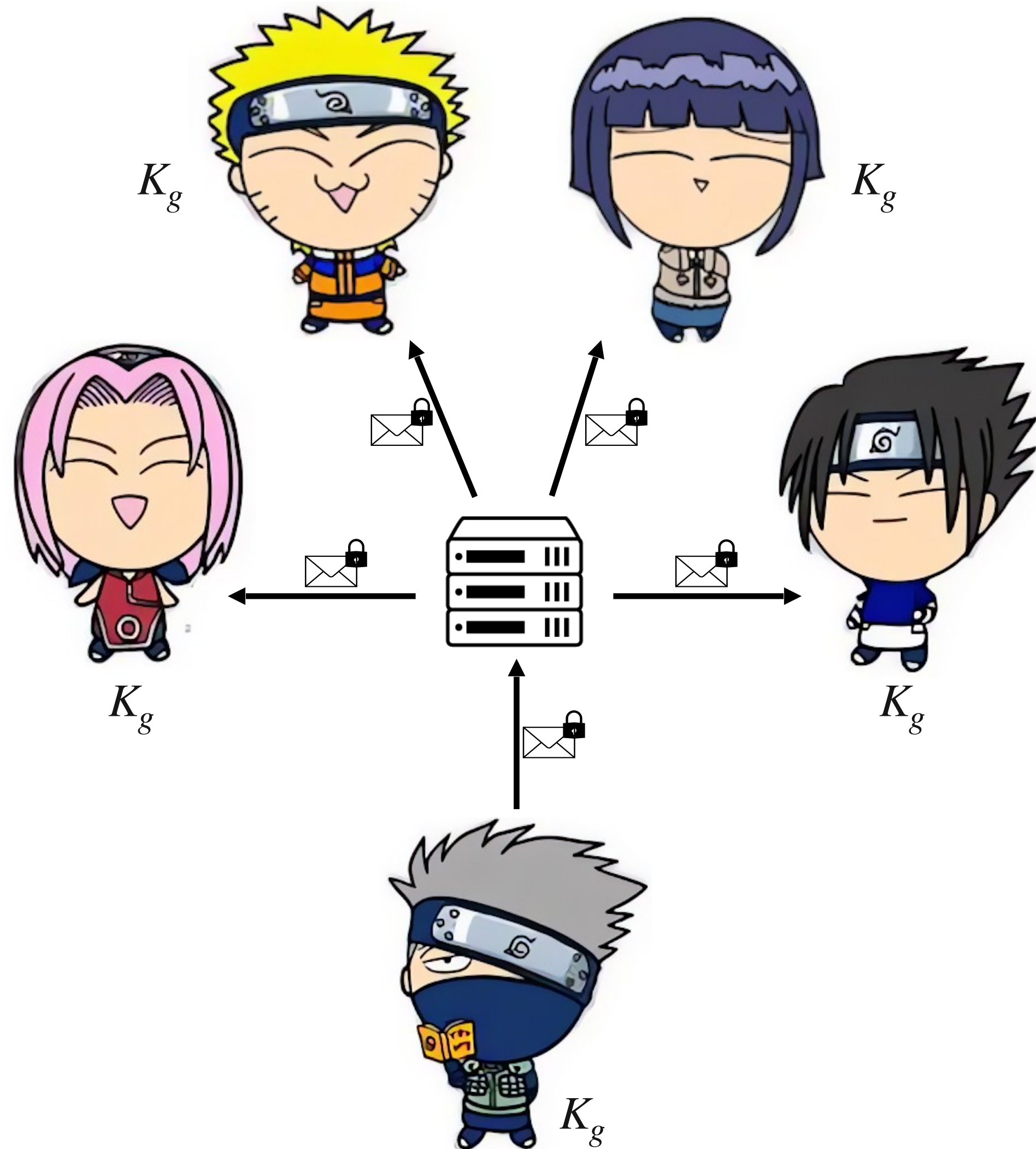
Secure Group Messaging



Secure Group
Messaging

Key
Agreement

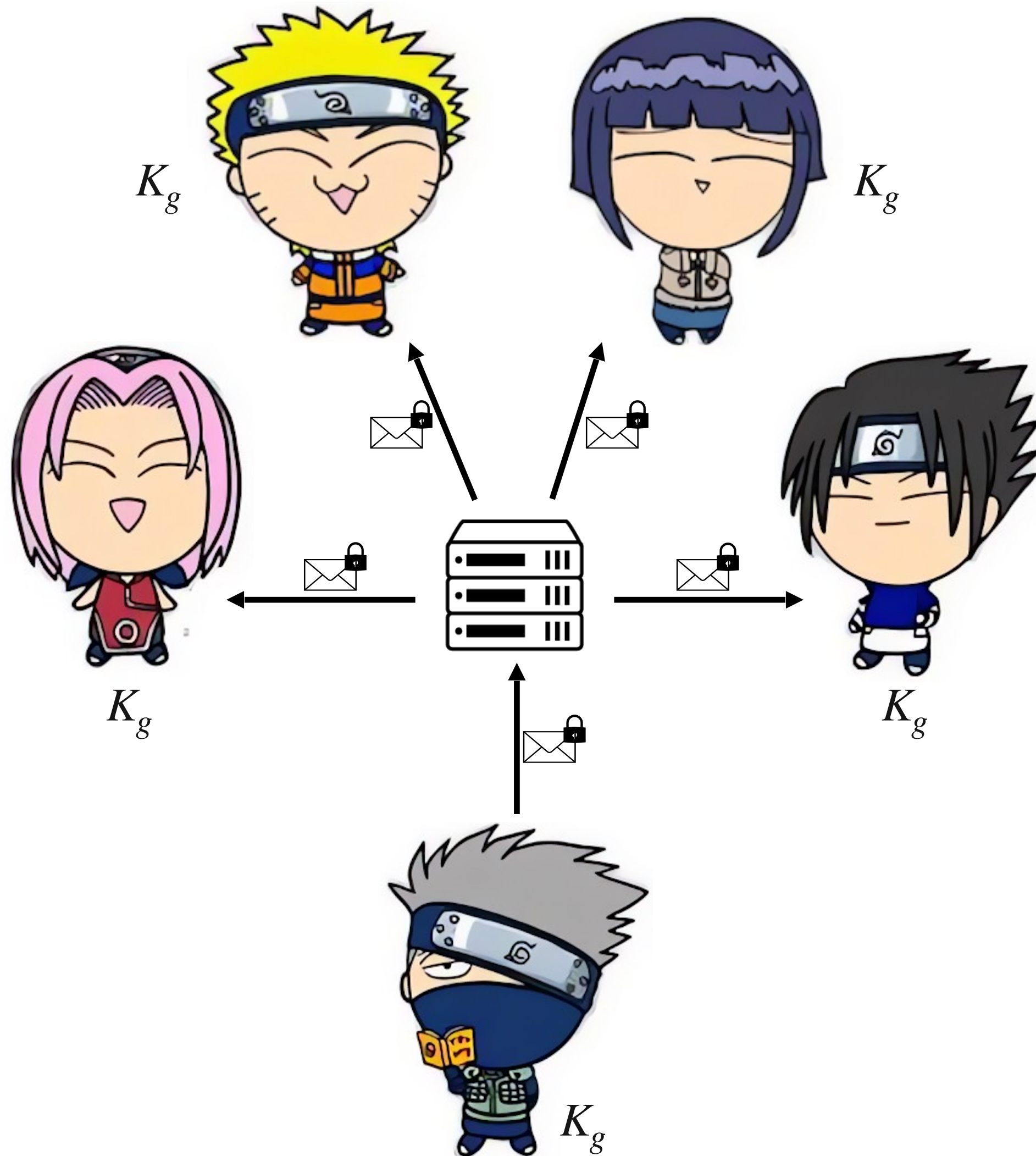
Secure Group Messaging



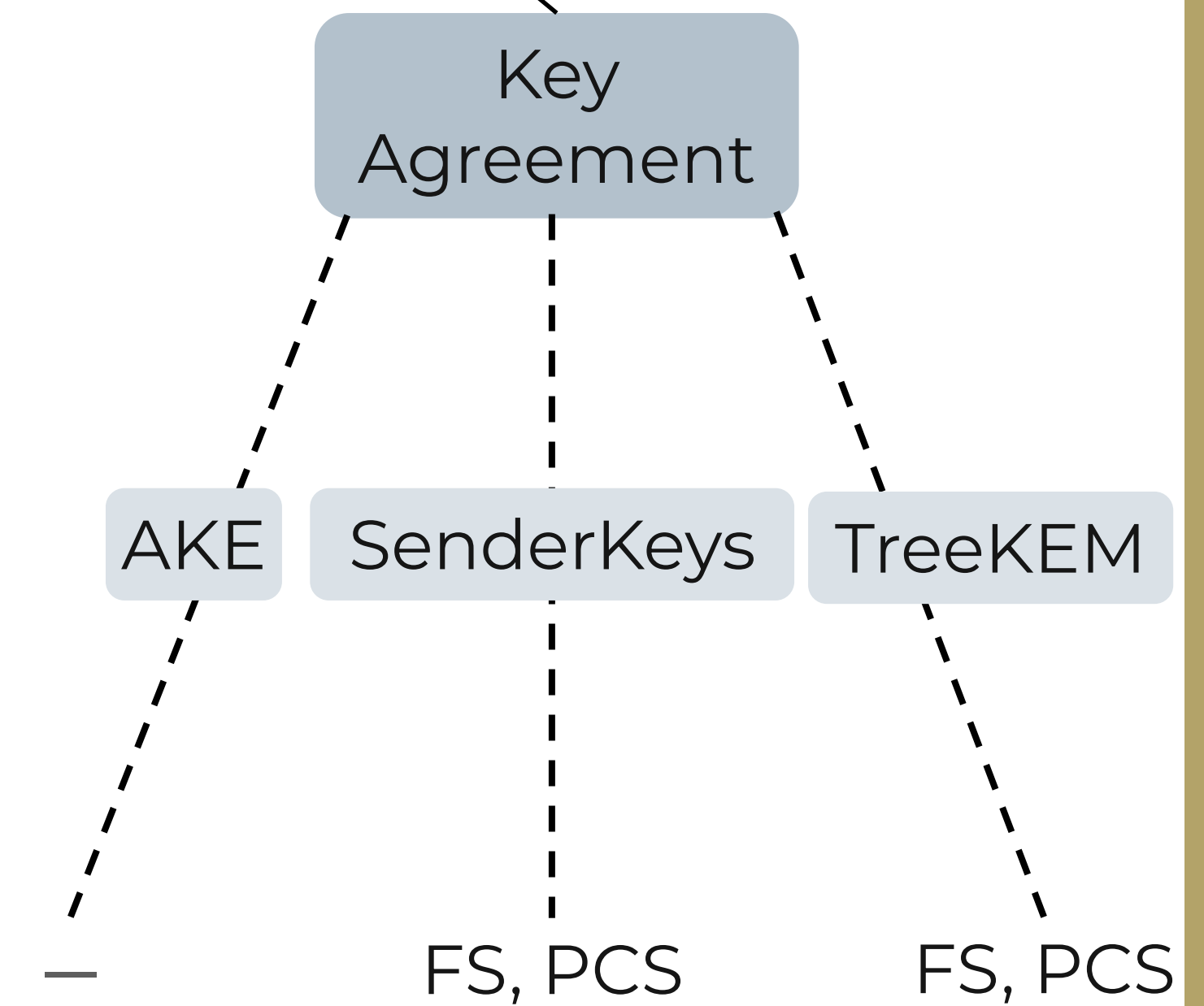
Secure Group
Messaging

Key
Agreement

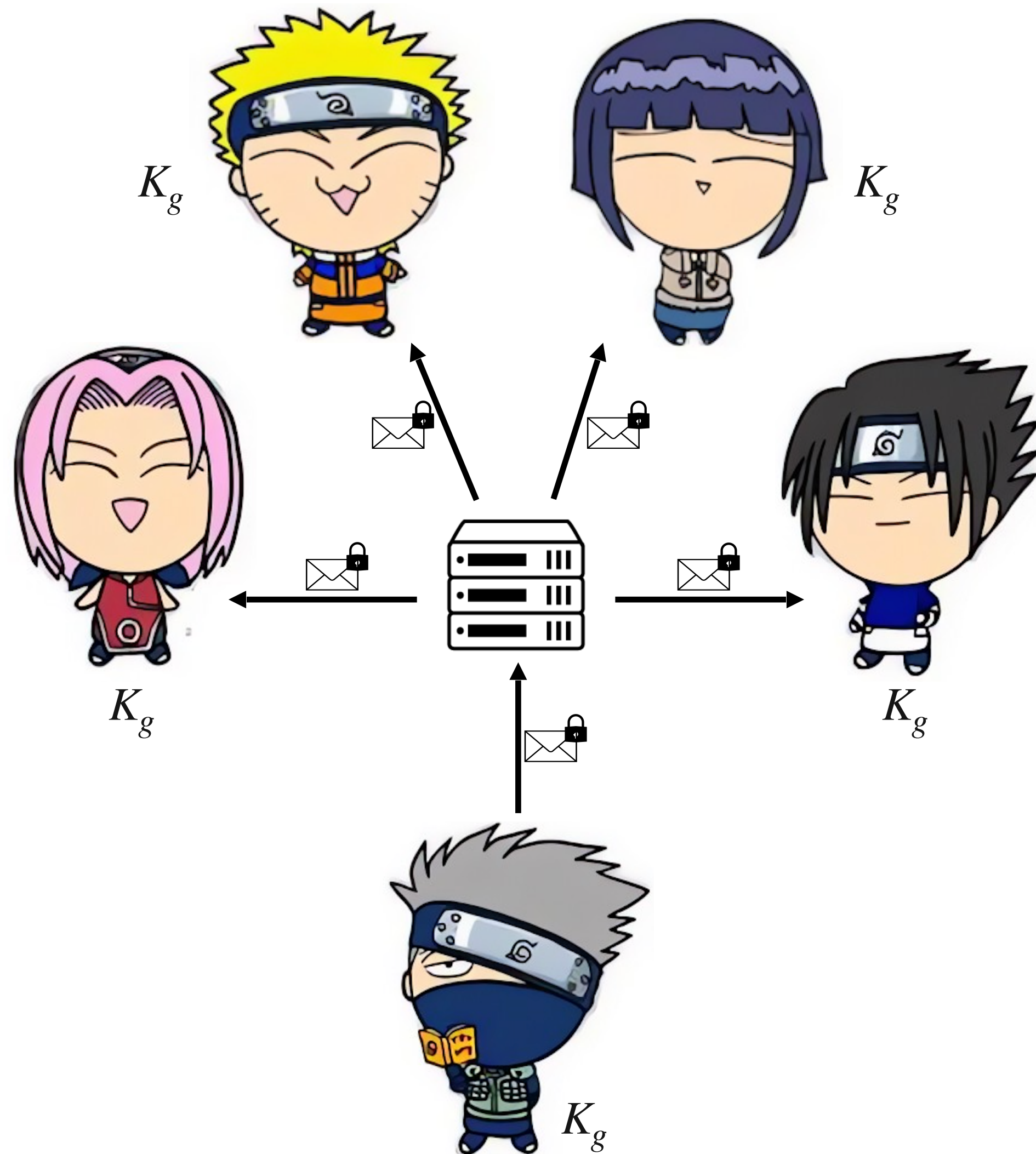
Secure Group Messaging



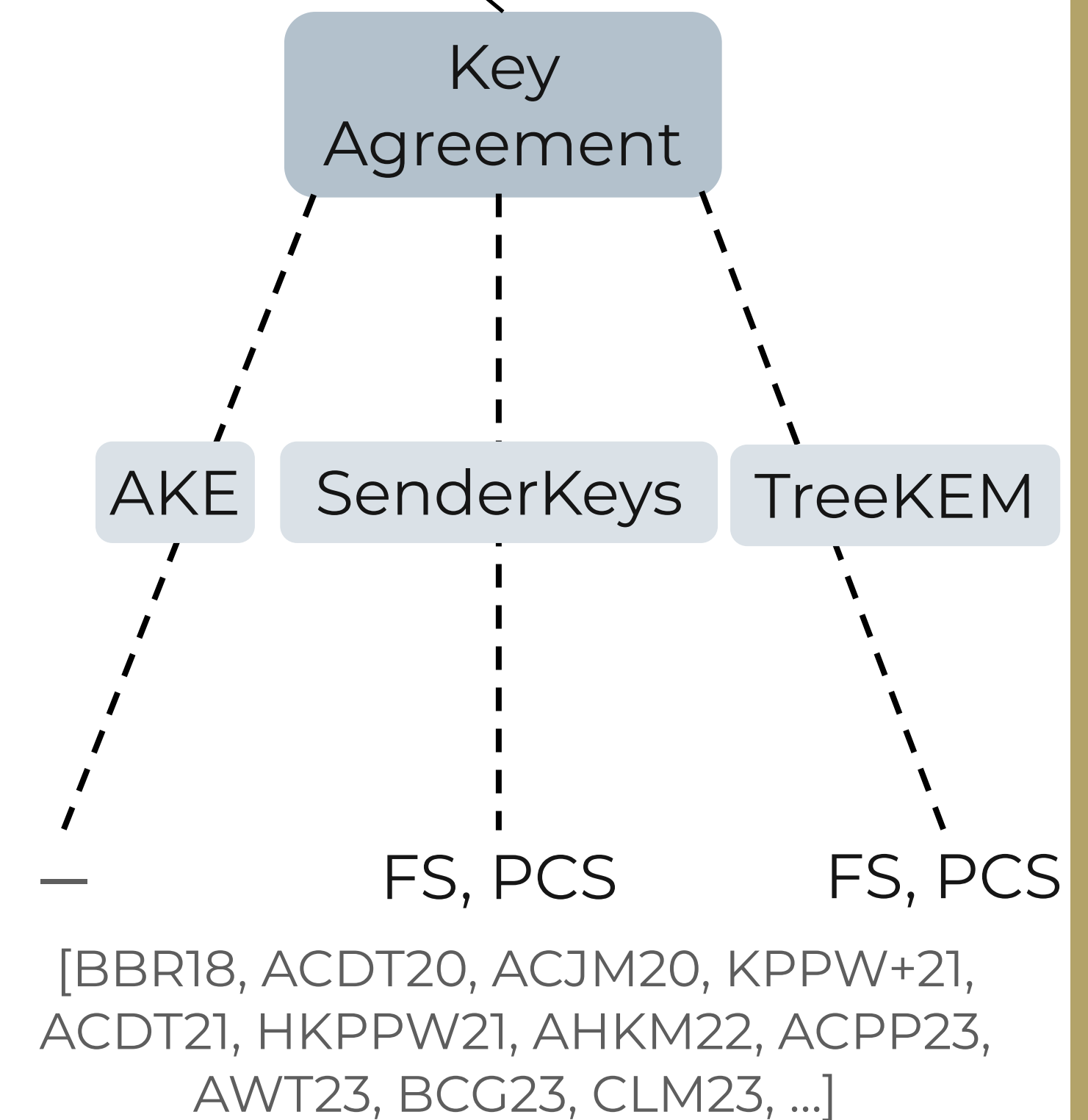
Secure Group Messaging



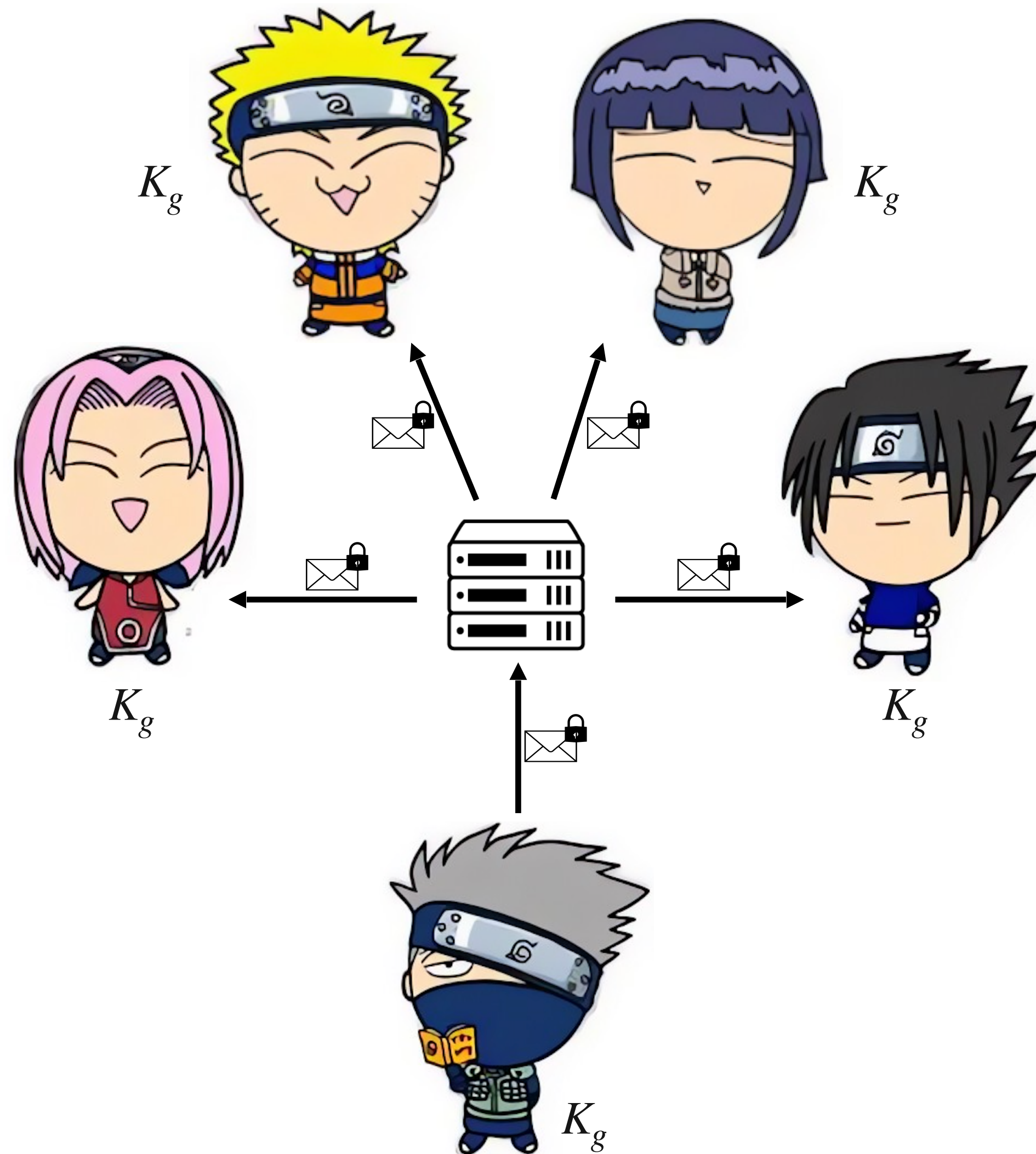
Secure Group Messaging



Secure Group Messaging



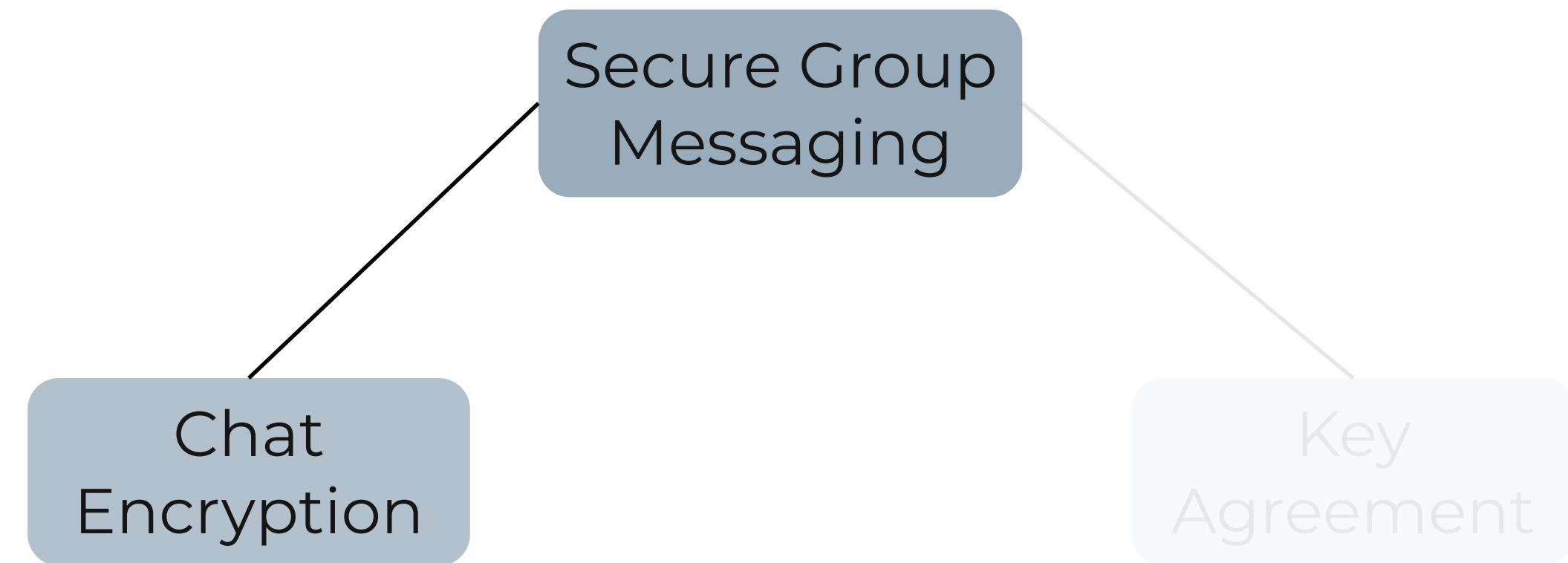
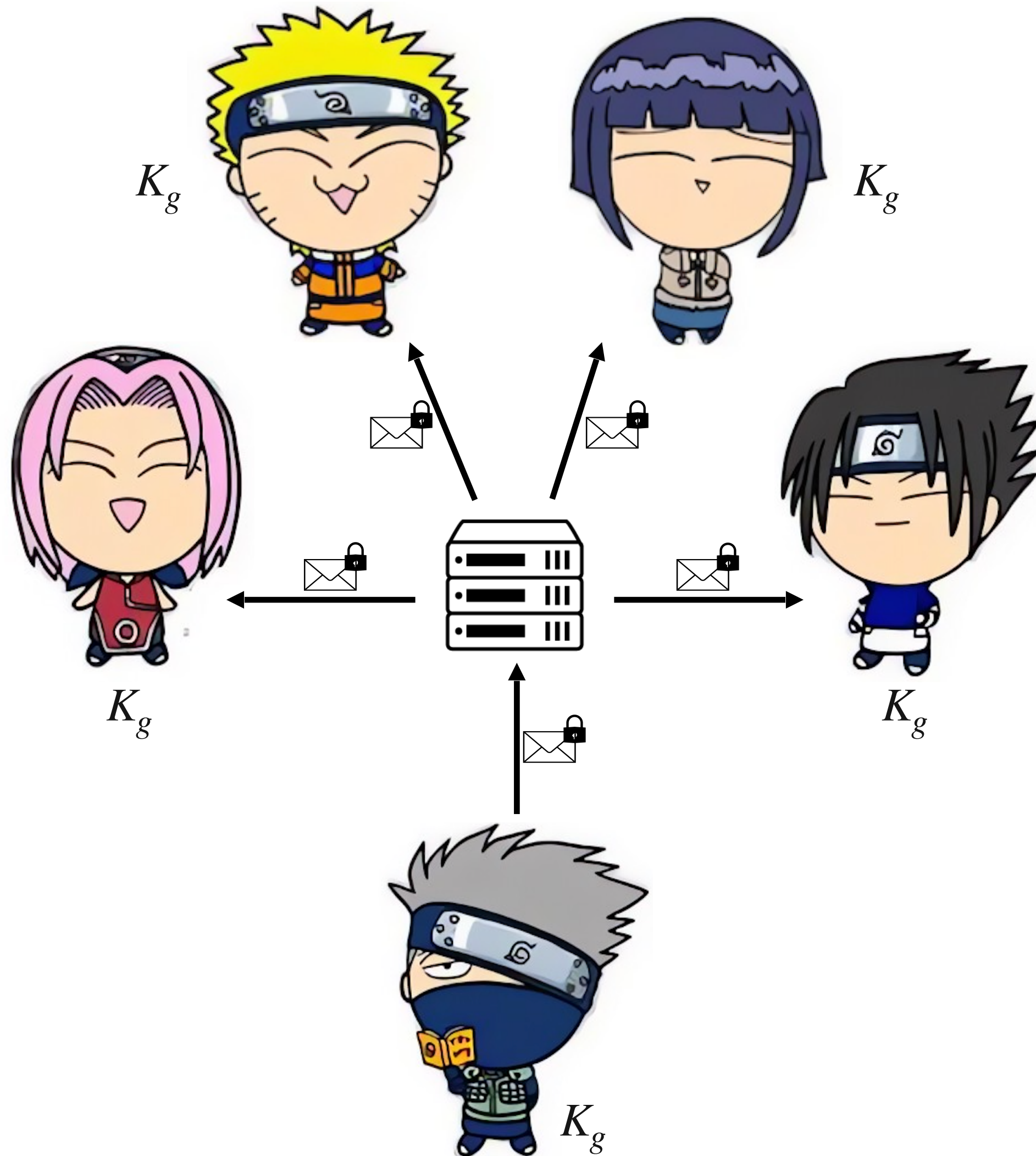
Secure Group Messaging



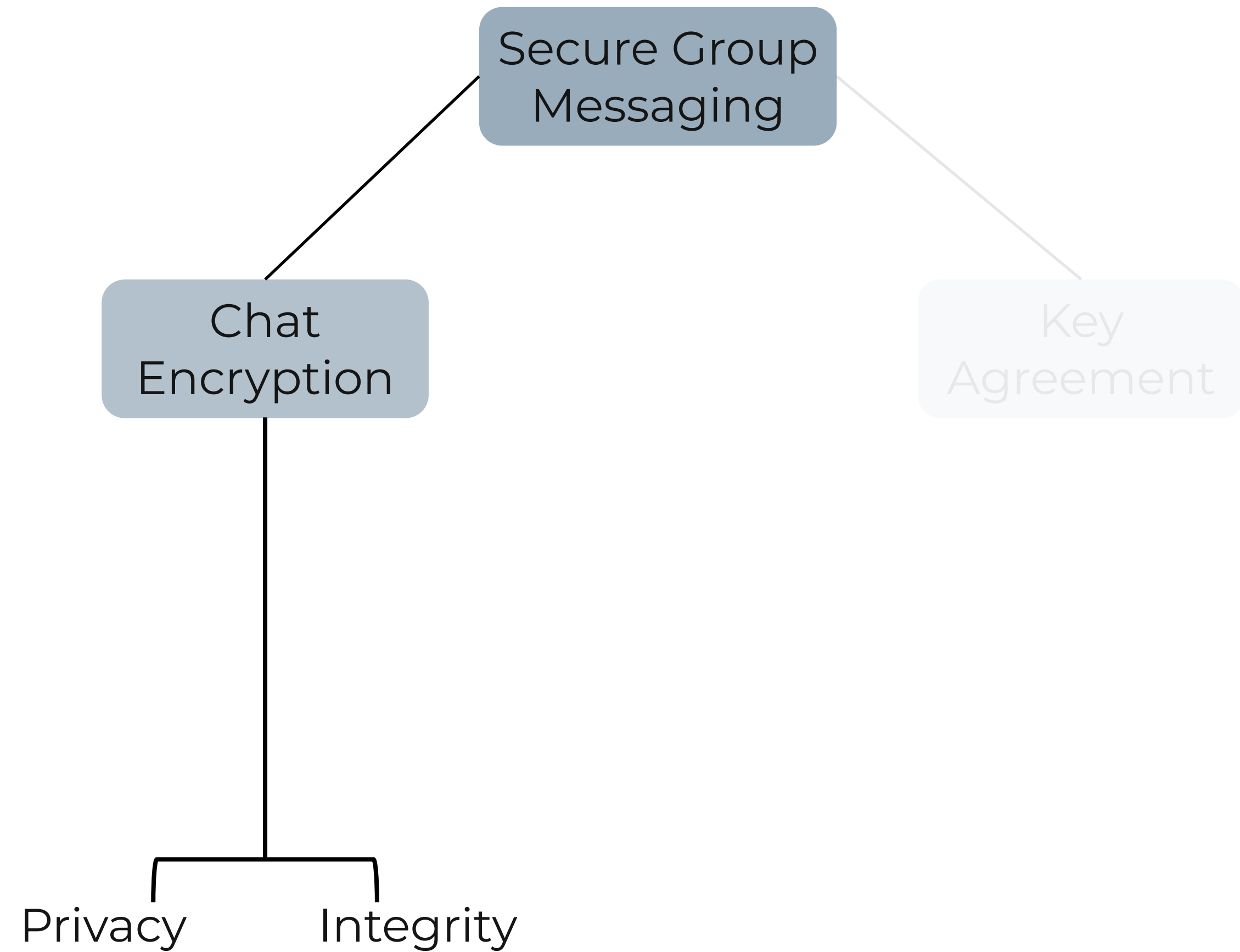
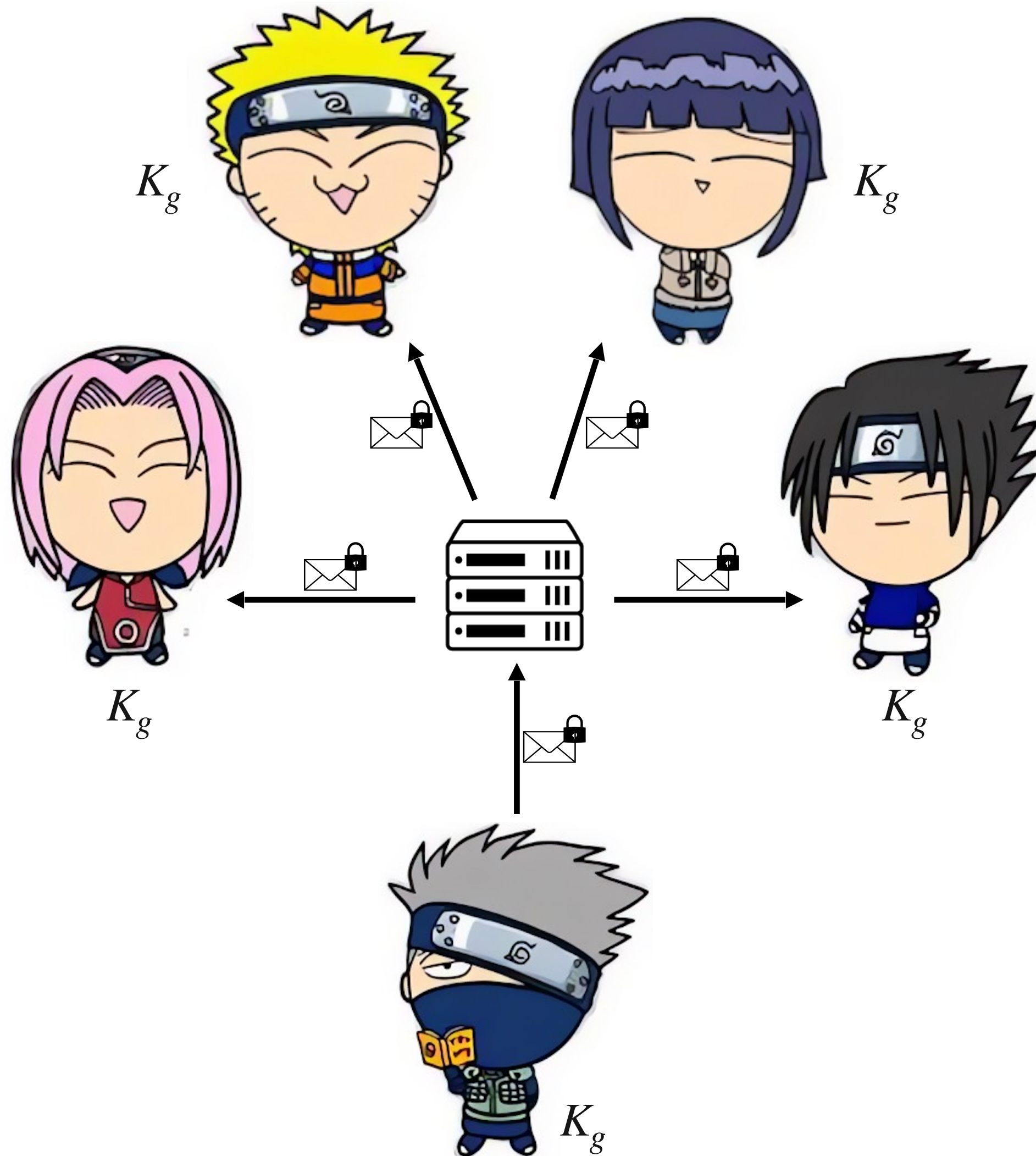
Secure Group
Messaging

Key
Agreement

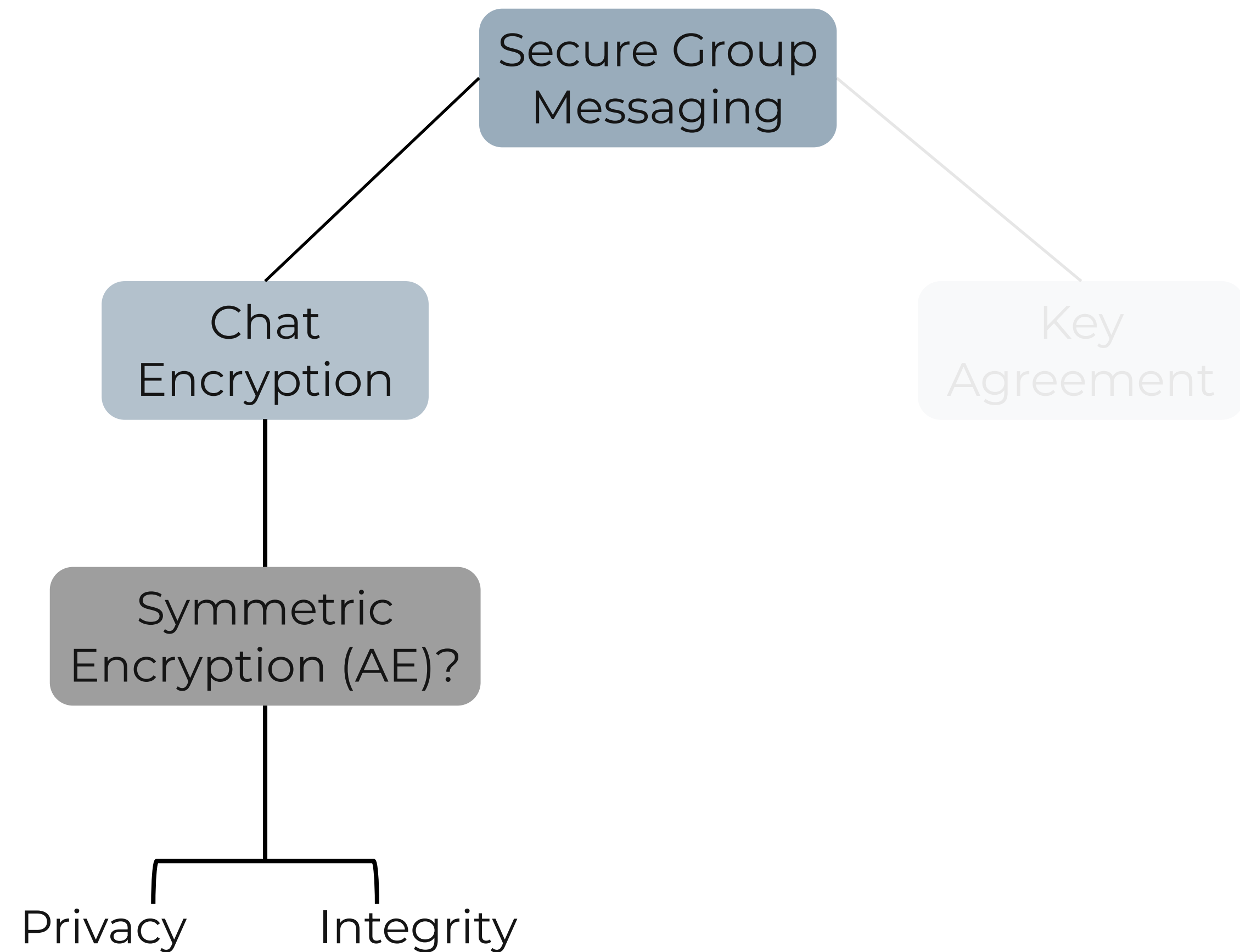
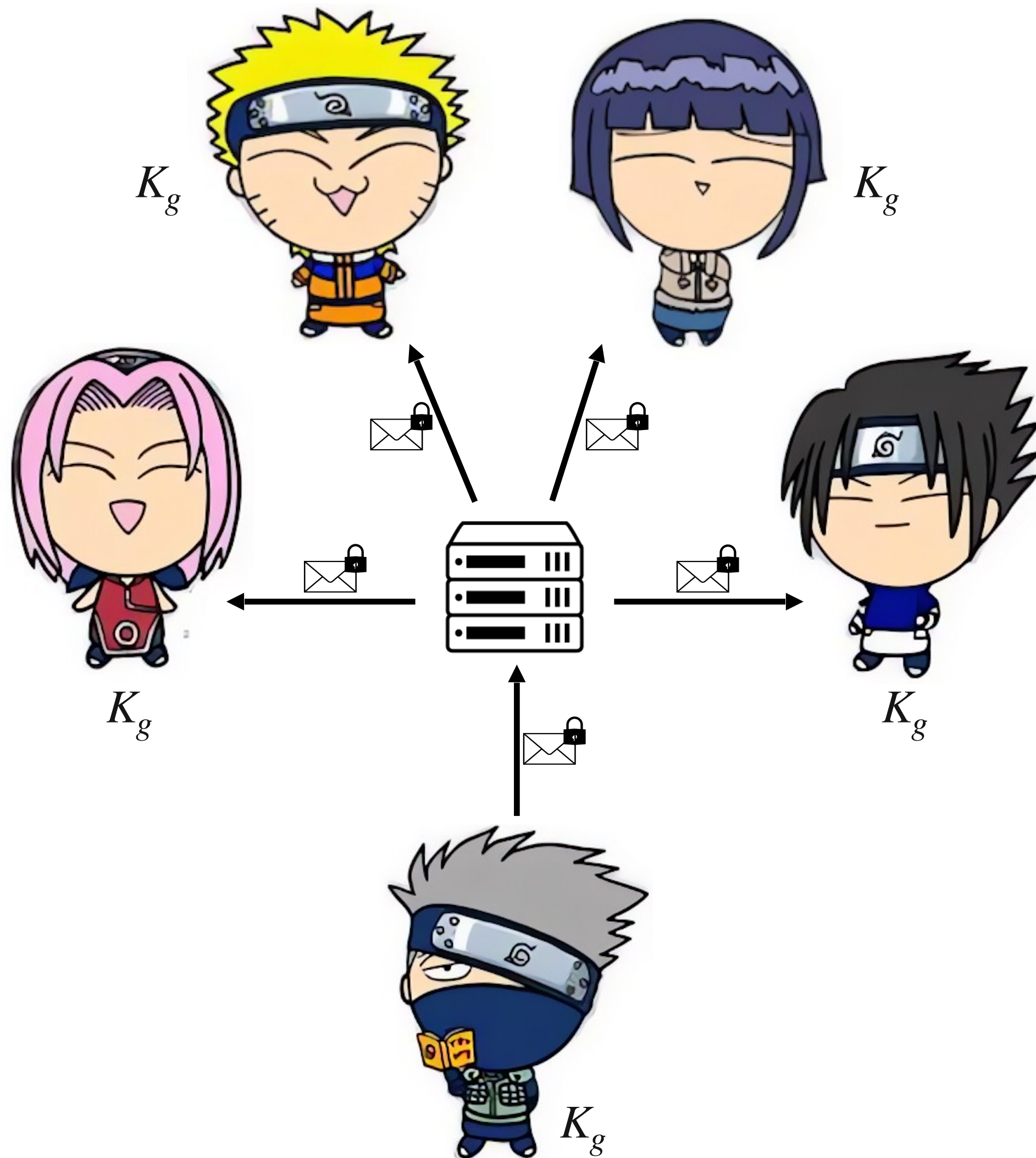
Secure Group Messaging



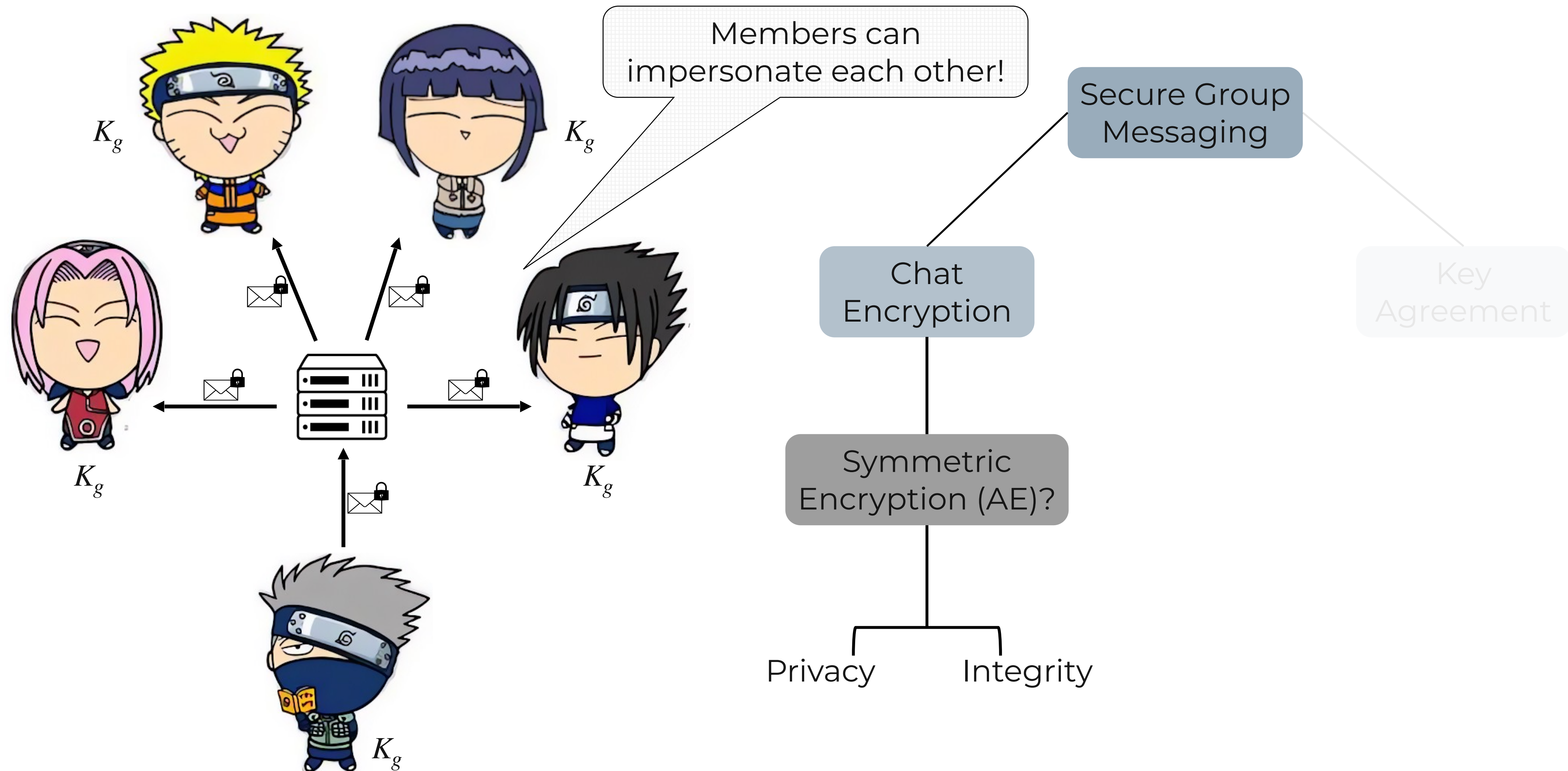
Secure Group Messaging



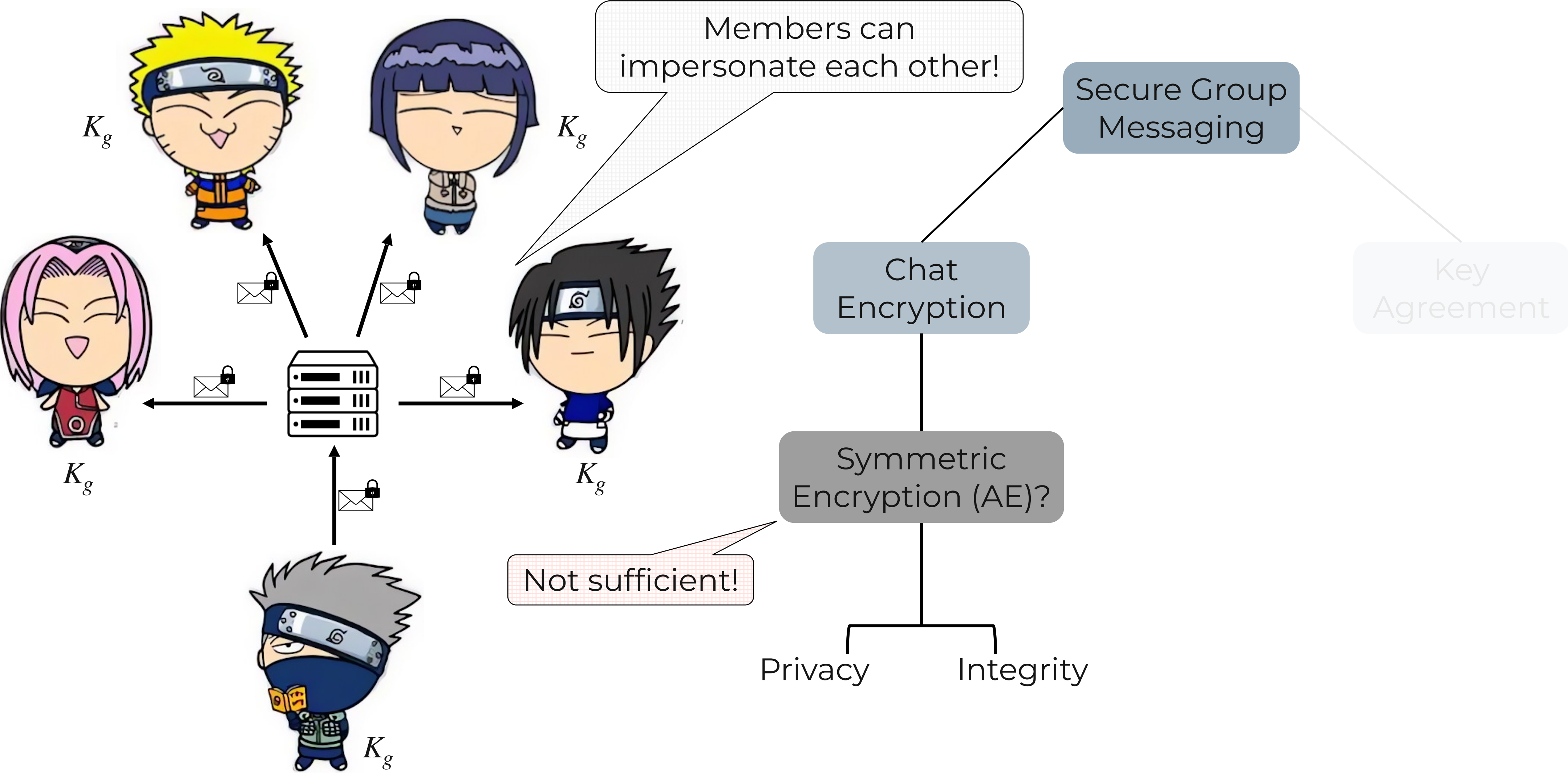
Secure Group Messaging



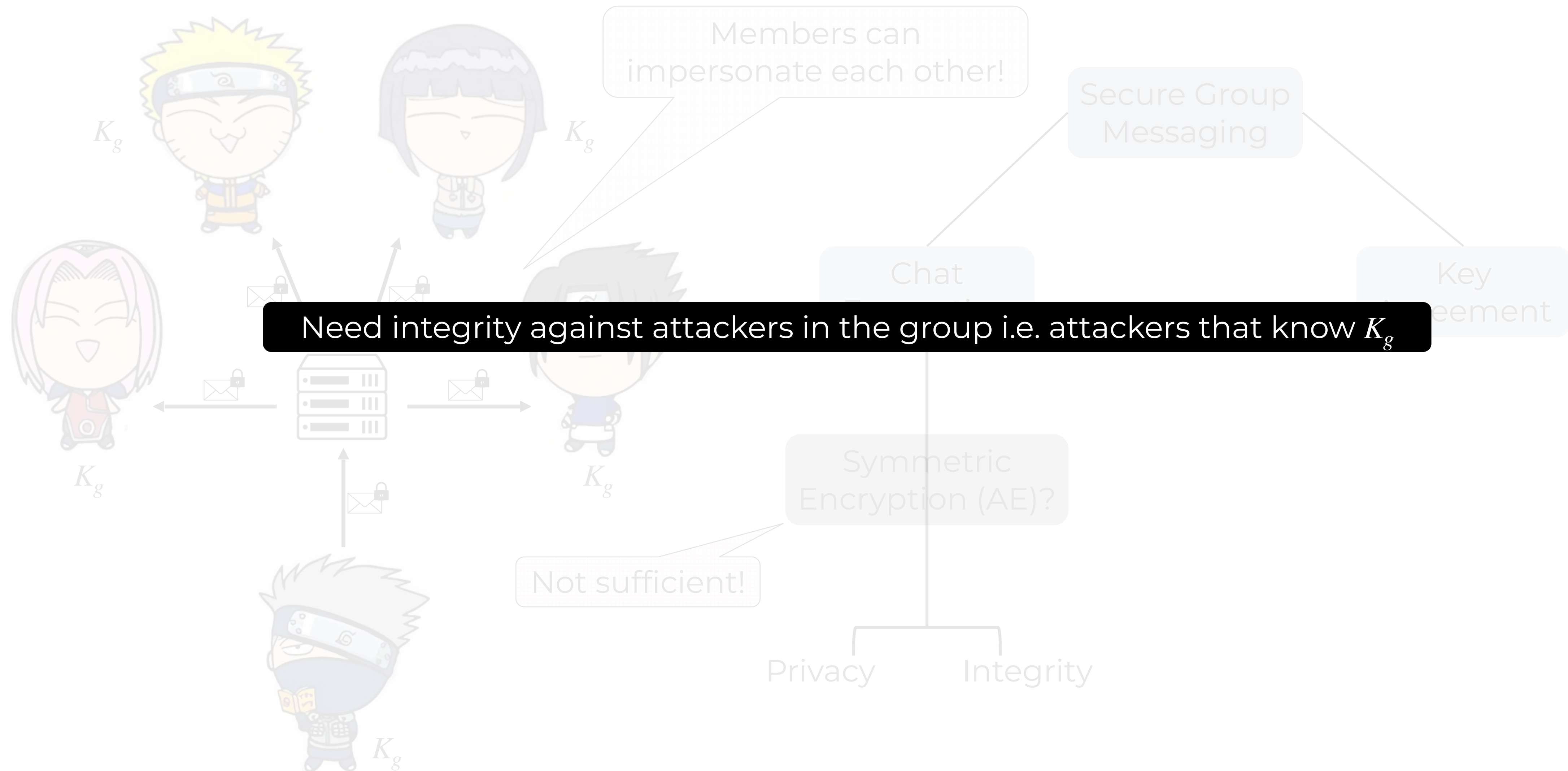
Secure Group Messaging



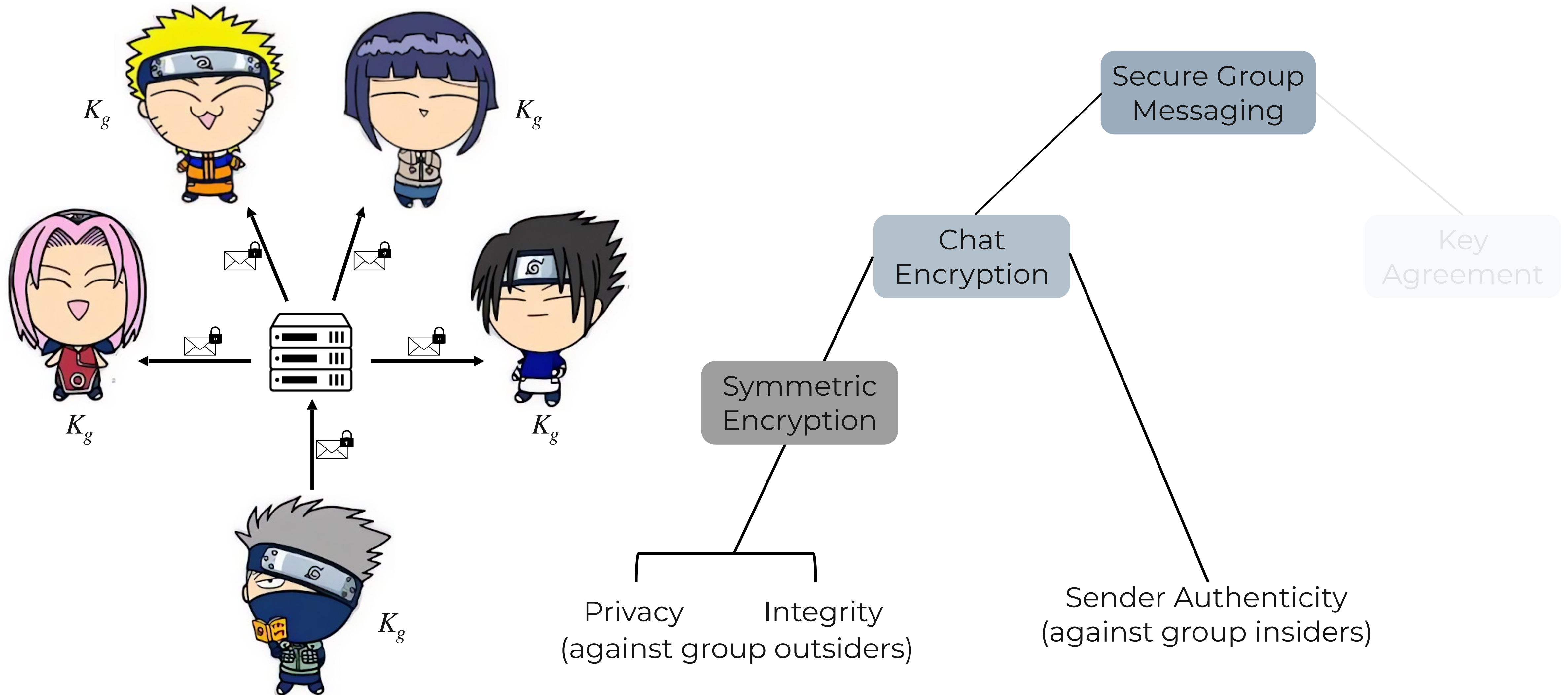
Secure Group Messaging



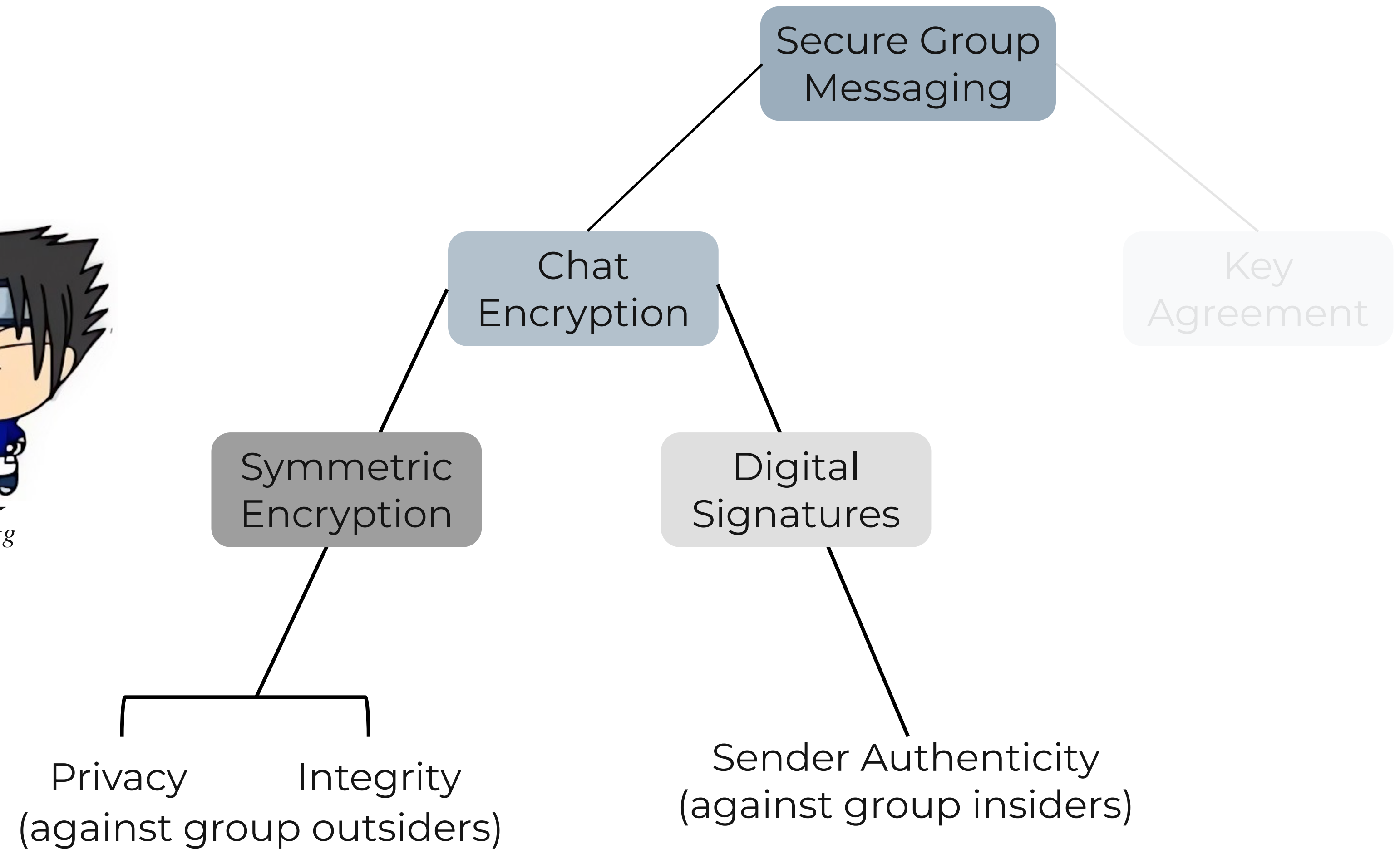
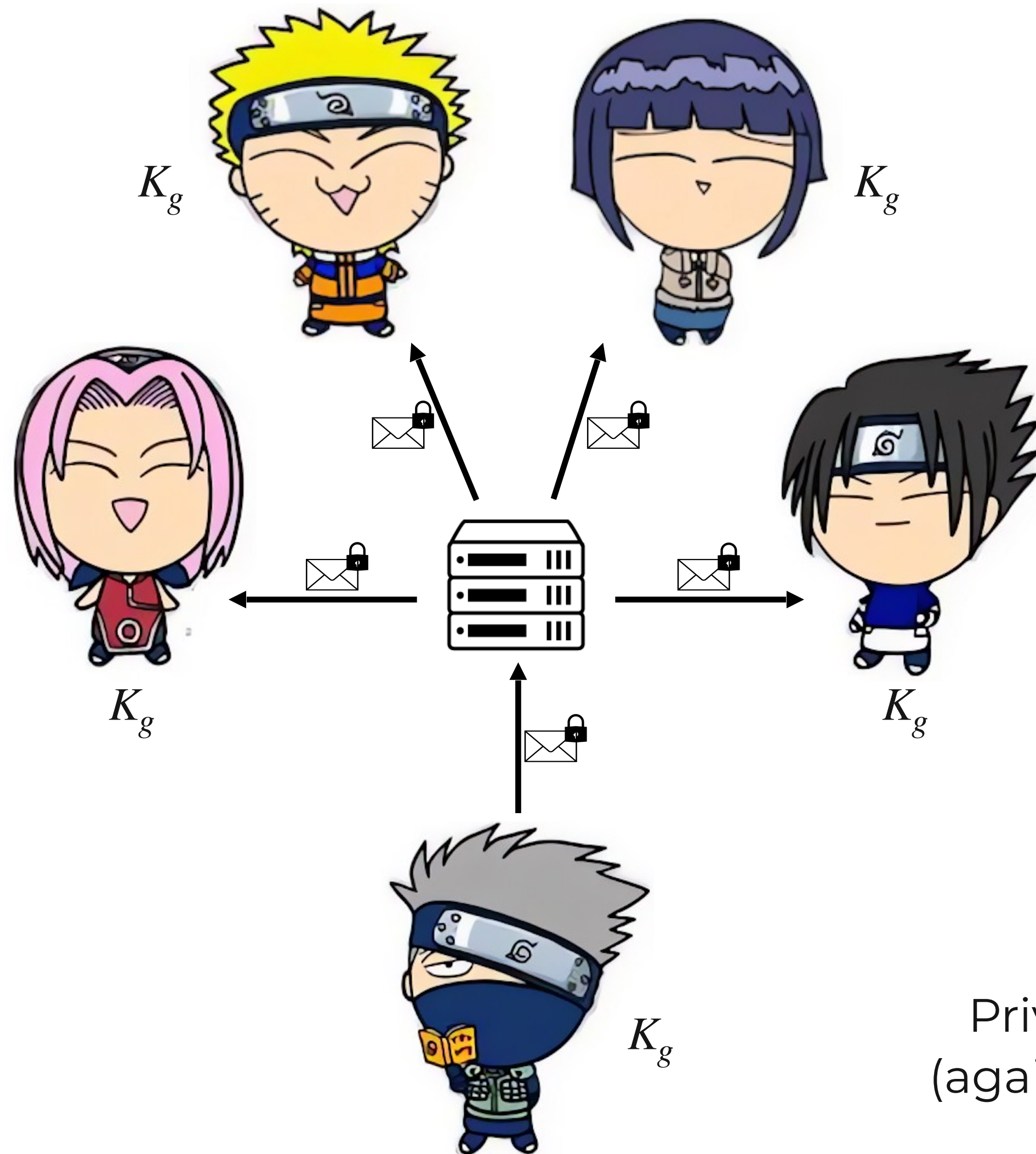
Secure Group Messaging



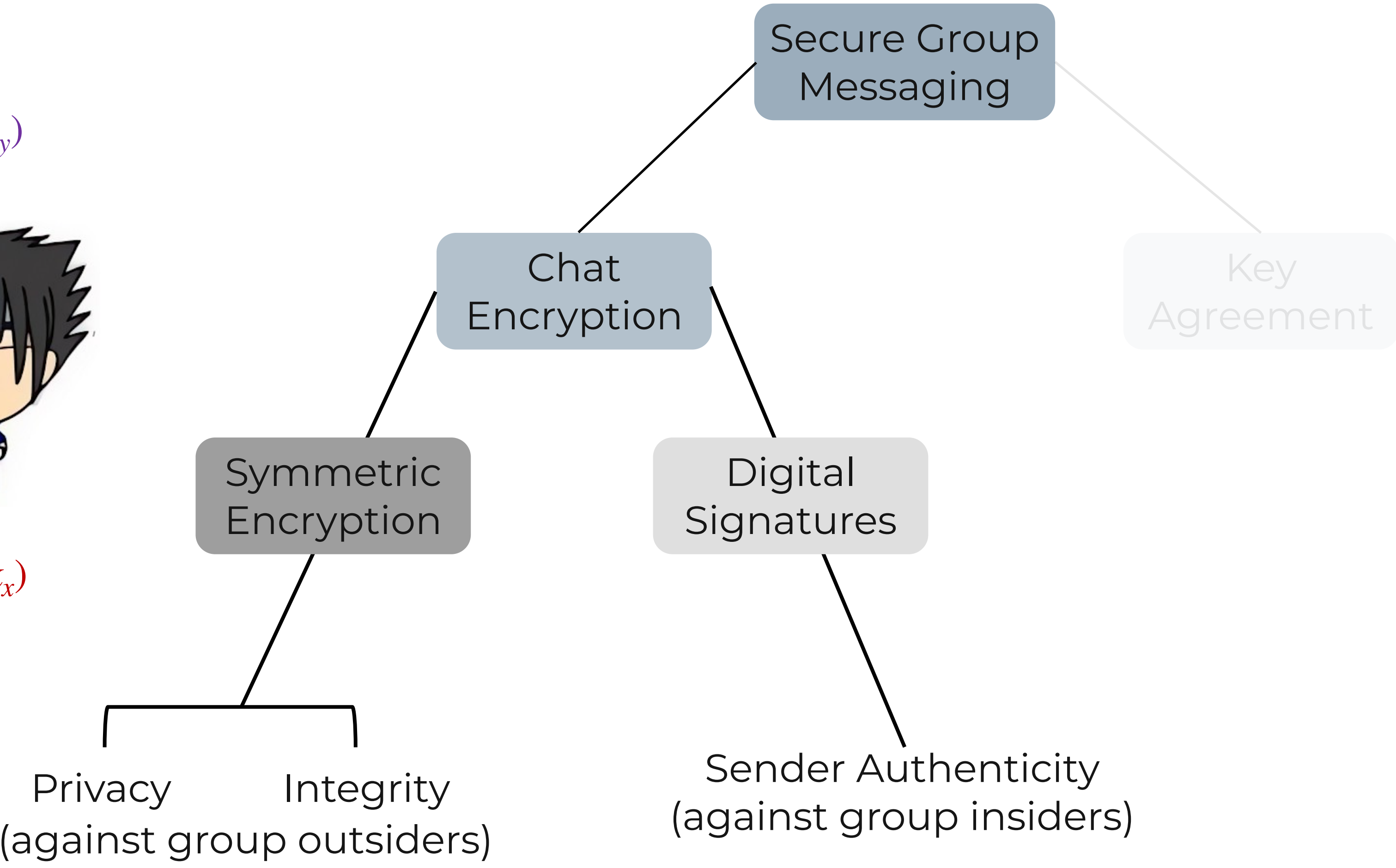
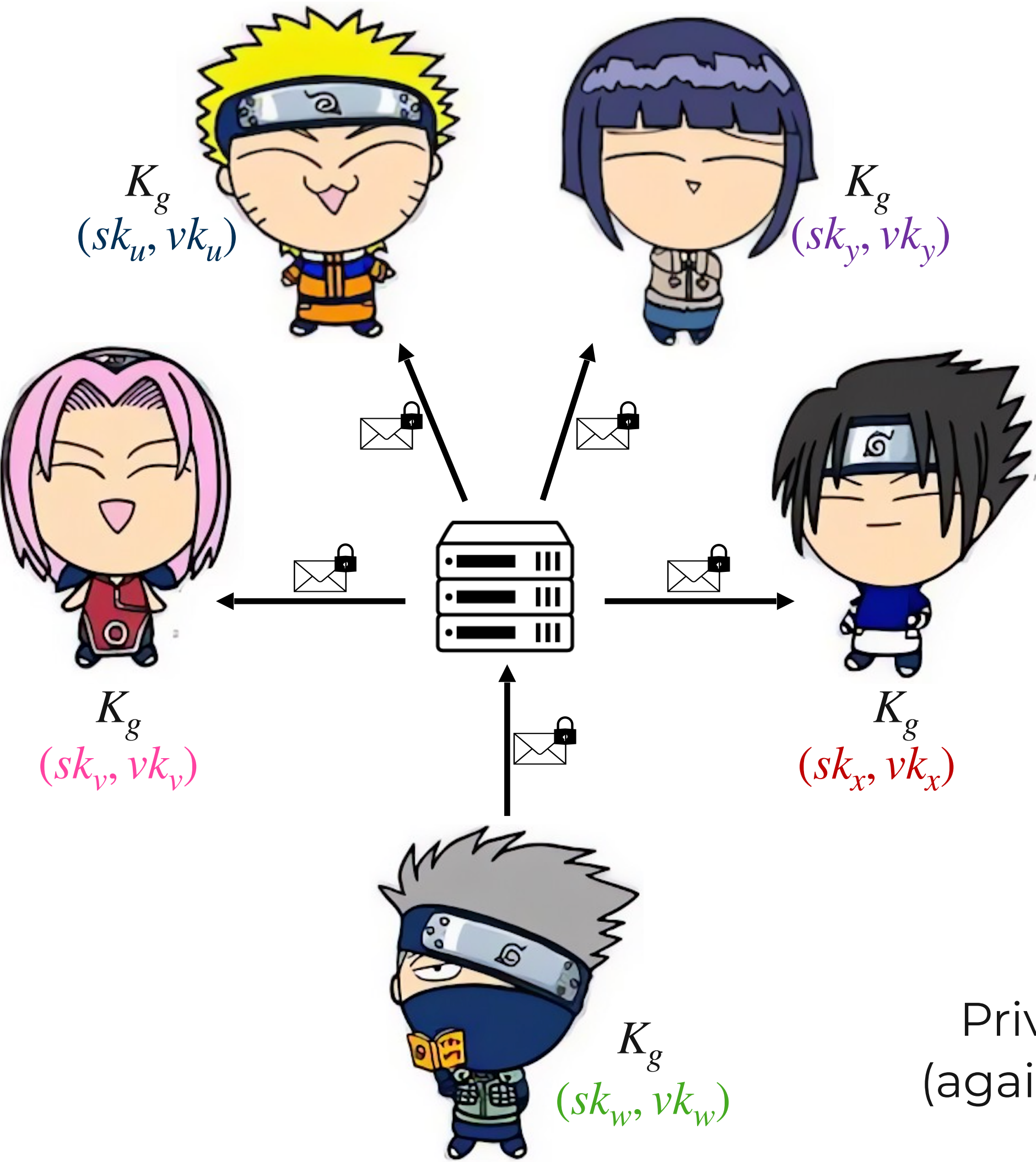
Secure Group Messaging



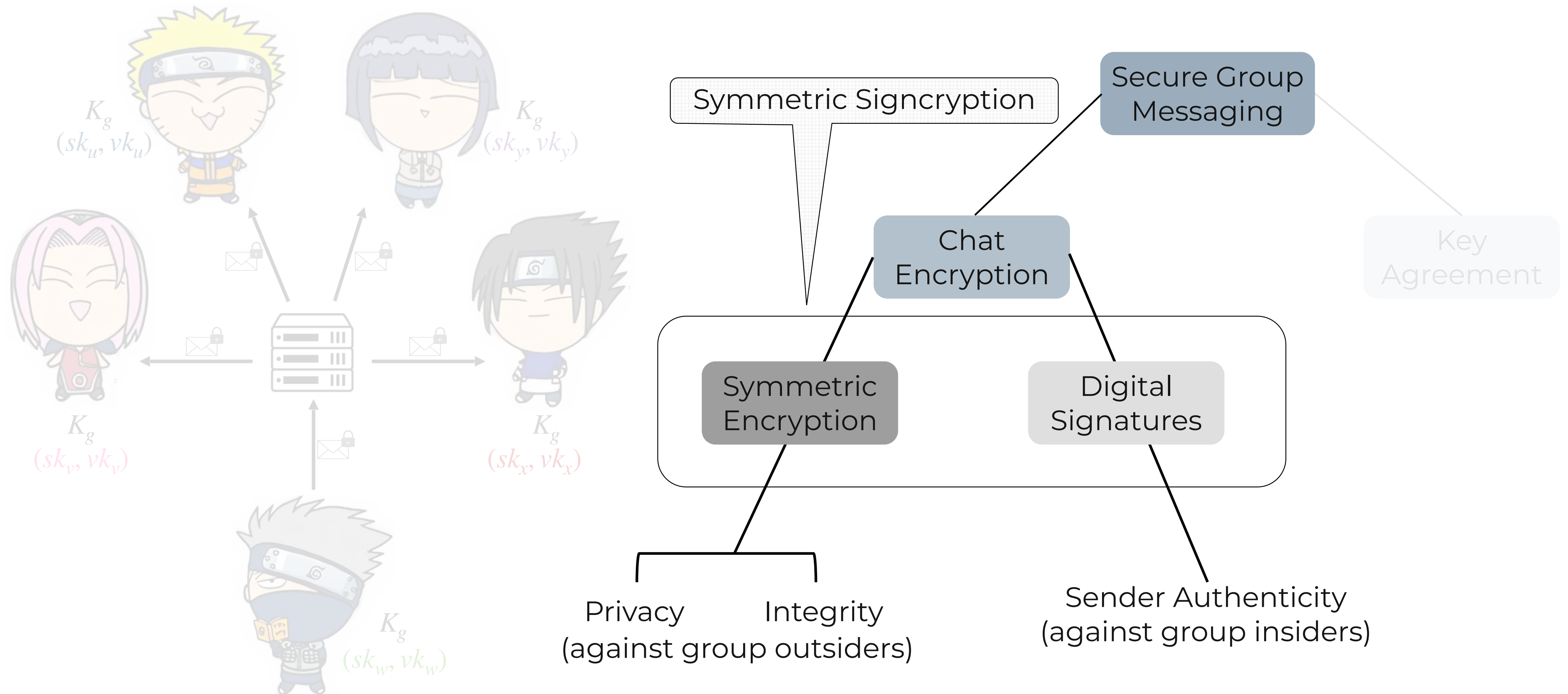
Secure Group Messaging



Secure Group Messaging






Secure Group Messaging





Methodology

Methodology

Symmetric Signcryption and E2EE Group Messaging in Keybase




Joseph Jaeger¹ , Akshaya Kumar¹ , and Igors Stepanovs² 

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



Joseph Jaeger  and Akshaya Kumar 

Methodology

Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger¹ , Akshaya Kumar¹ , and Igors Stepanovs² 




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

Joseph Jaeger  and Akshaya Kumar 



Symmetric Signcryption
Model

Methodology

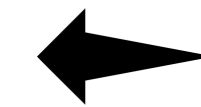
Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger¹ , Akshaya Kumar¹ , and Igors Stepanovs² 

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

Joseph Jaeger  and Akshaya Kumar 

Symmetric Signcryption
Model






Application





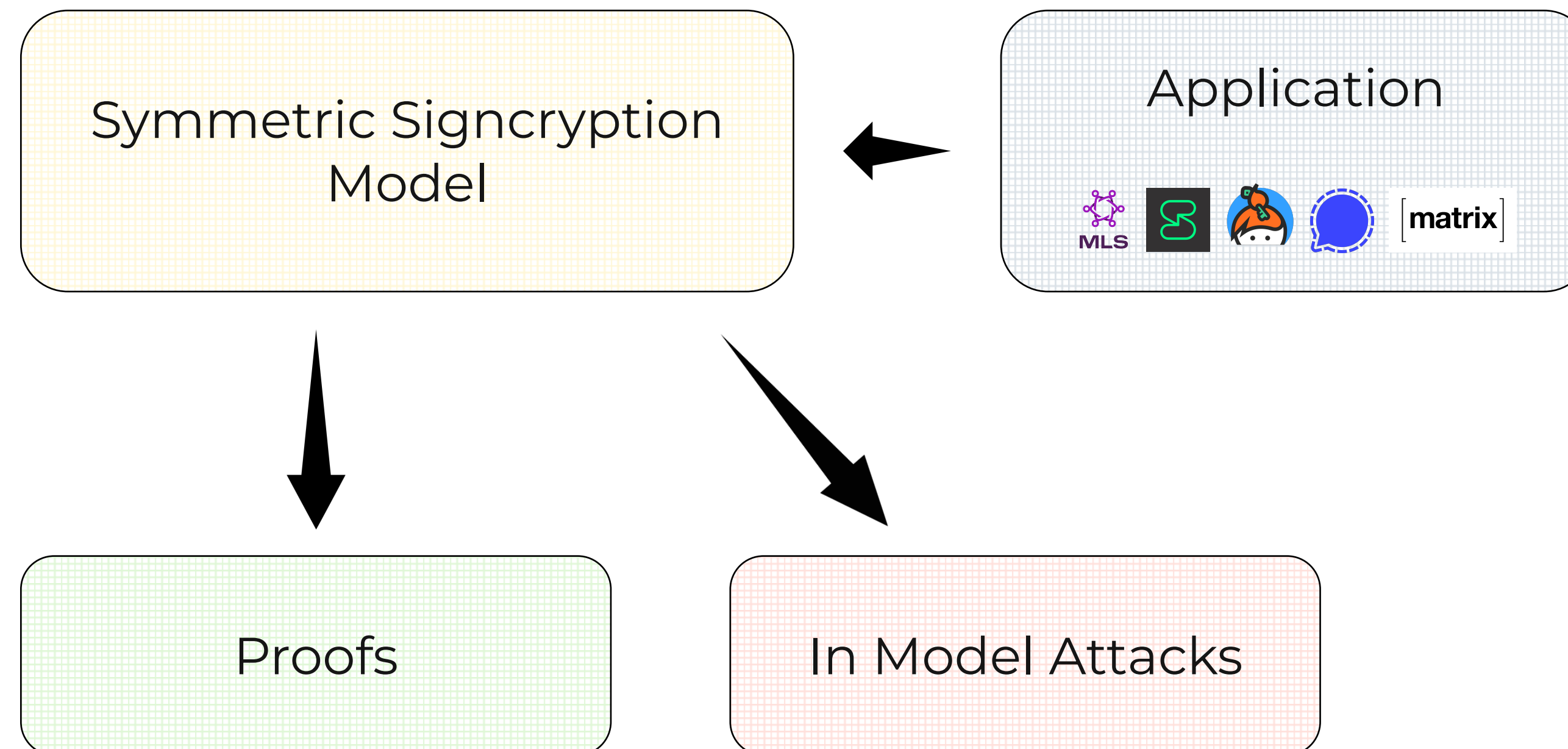
Methodology

Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger¹ , Akshaya Kumar¹ , and Igors Stepanovs² 




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

Joseph Jaeger  and Akshaya Kumar 





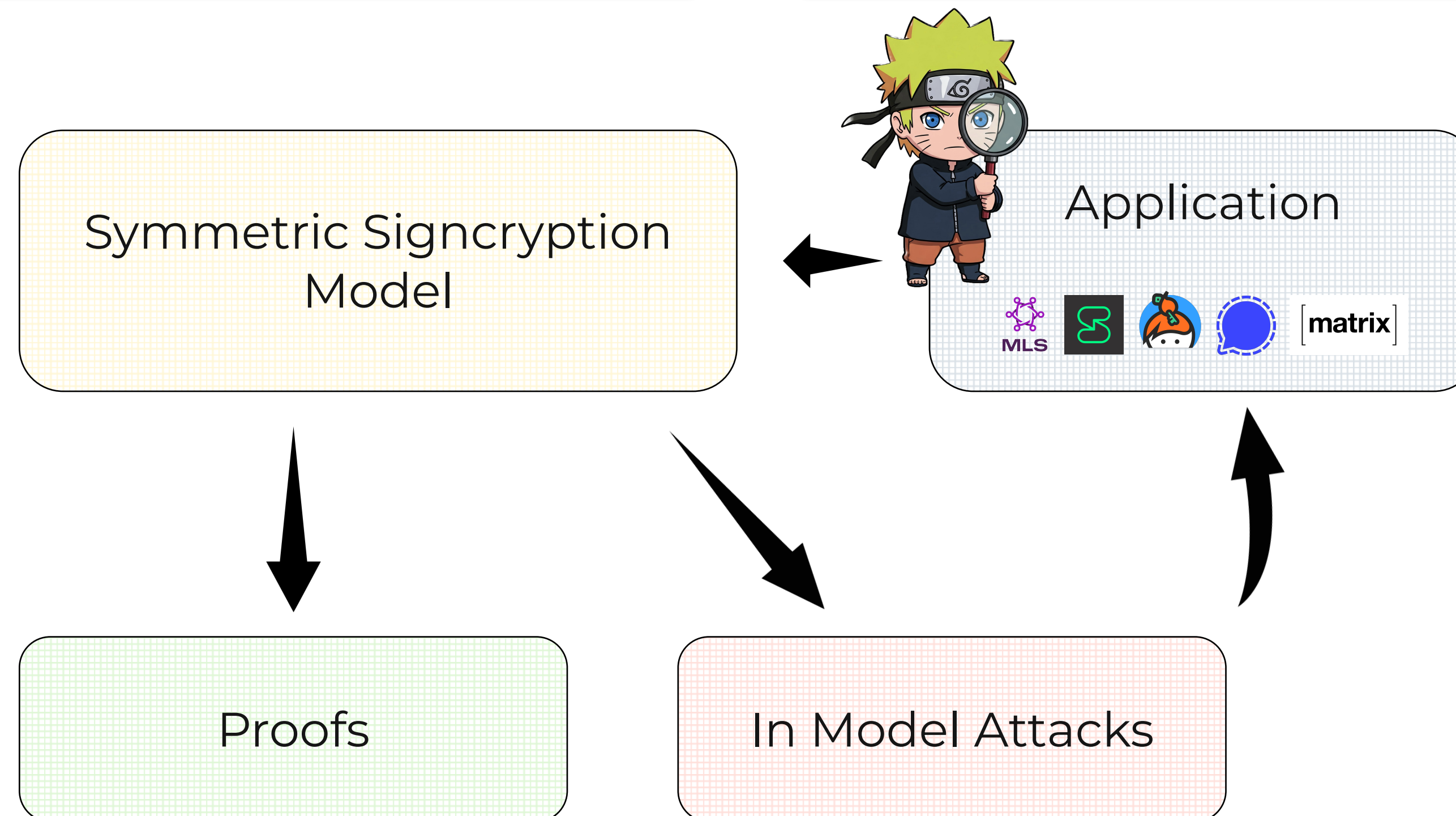
Methodology

Symmetric Signcryption and E2EE Group Messaging in Keybase

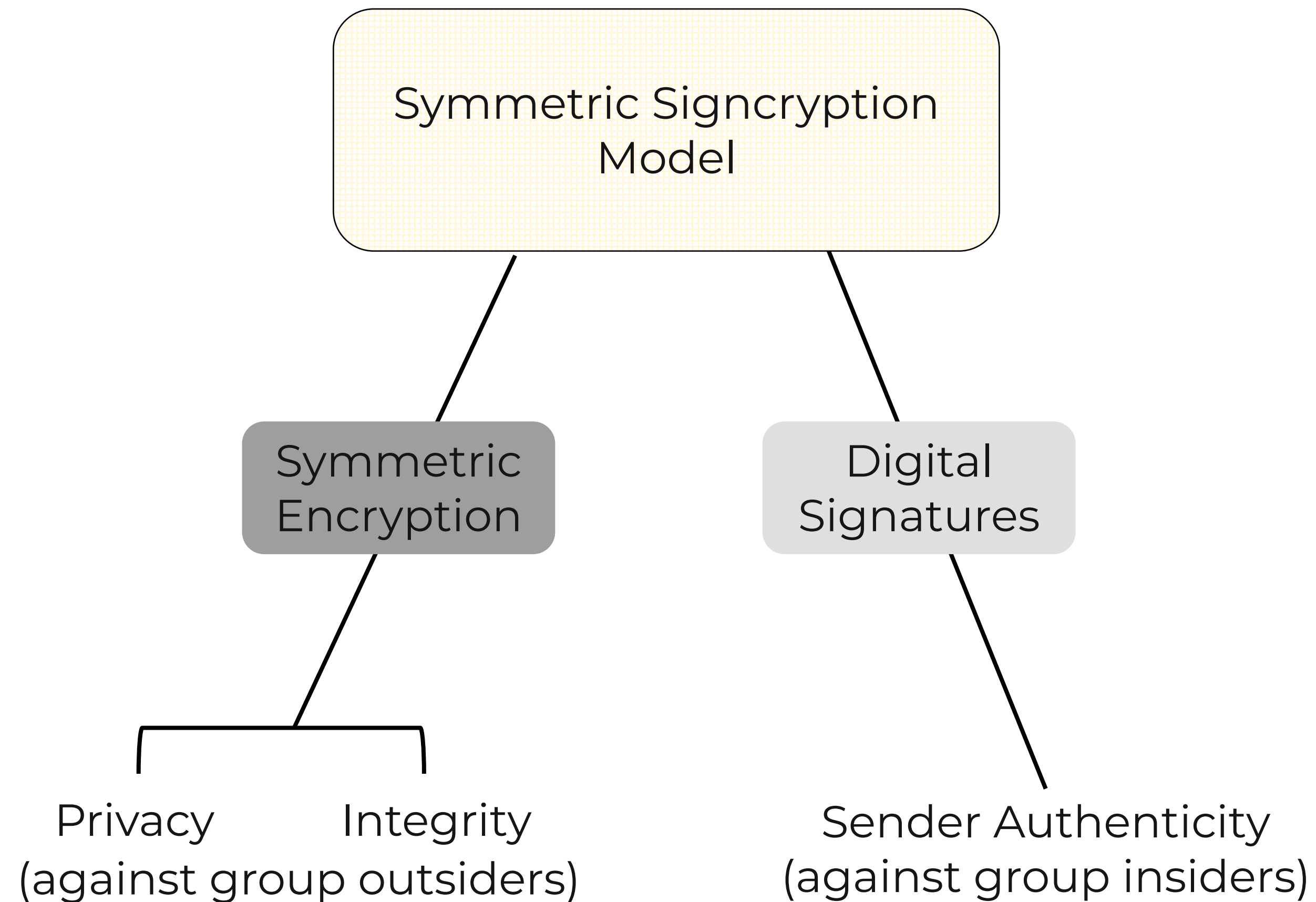
Joseph Jaeger¹ , Akshaya Kumar¹ , and Igors Stepanovs² 

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

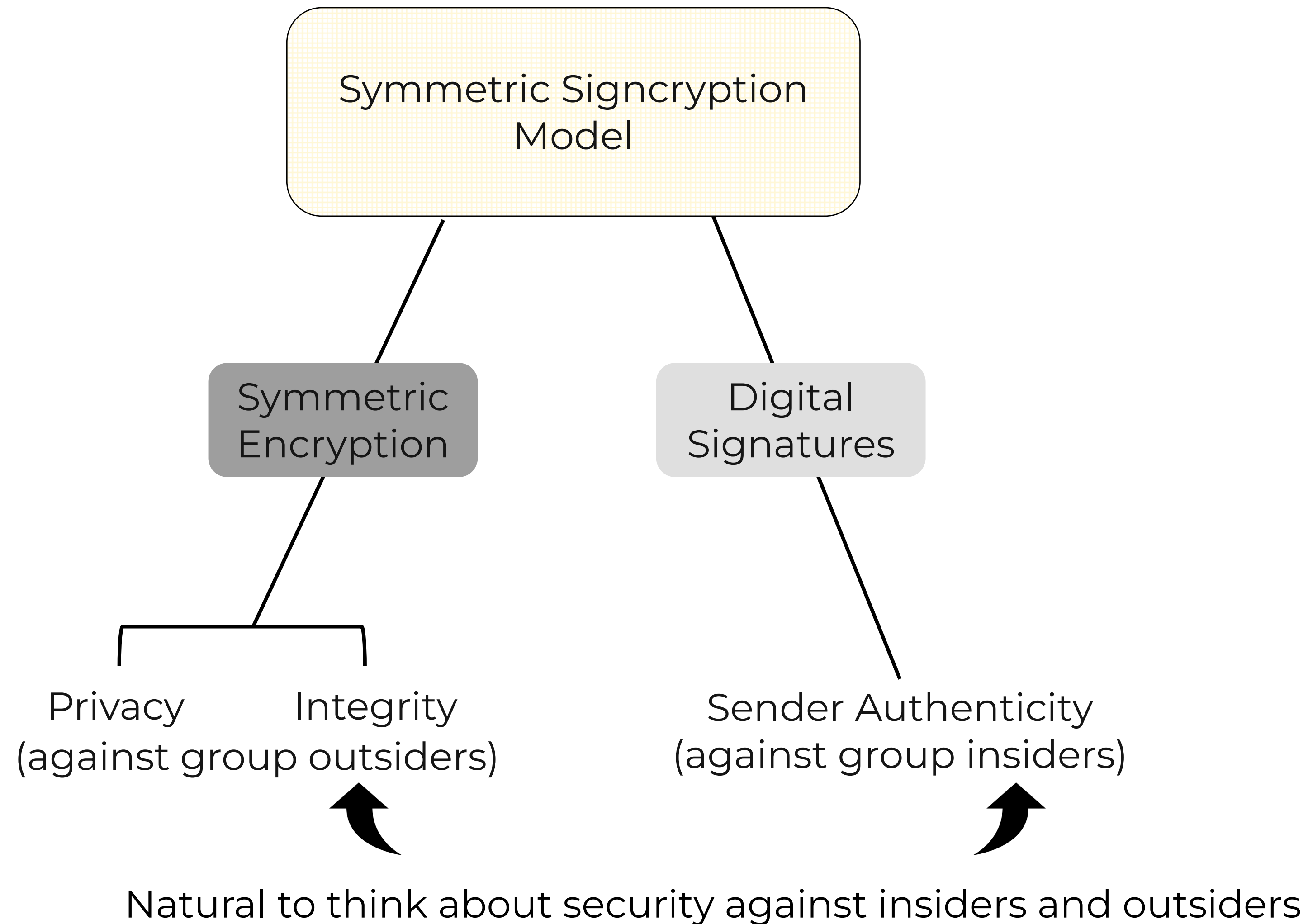
Joseph Jaeger  and Akshaya Kumar 



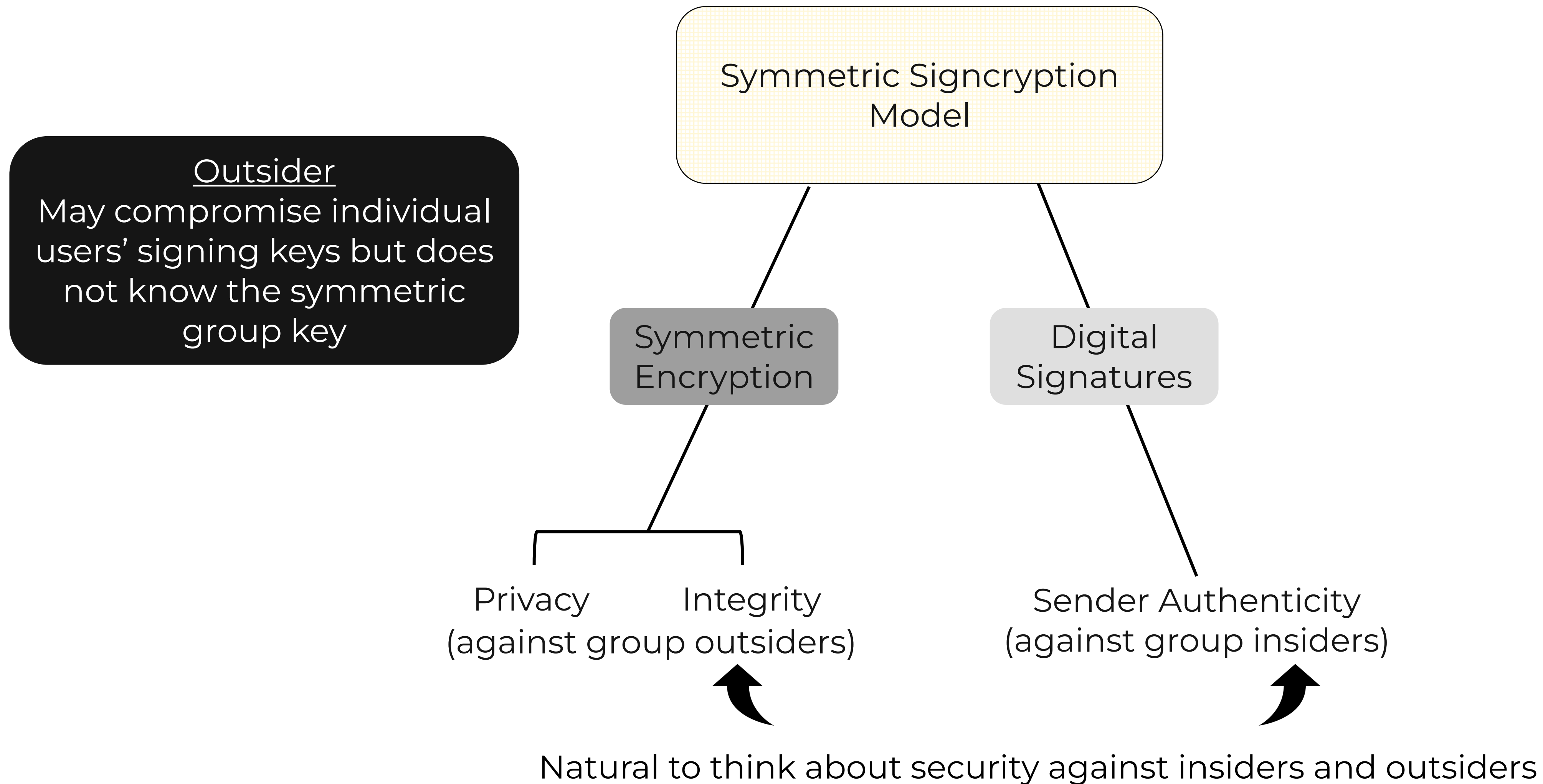
Insider and Outsider Attacks



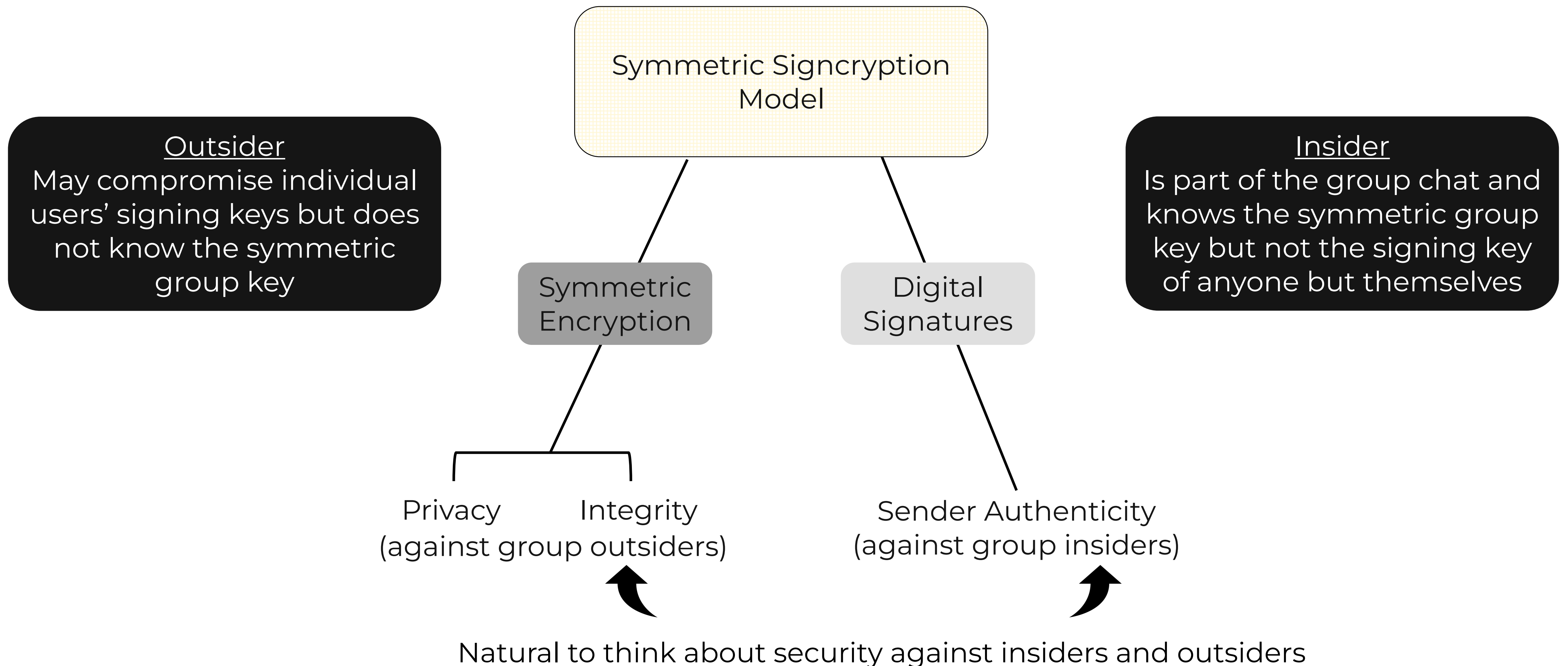
Insider and Outsider Attacks



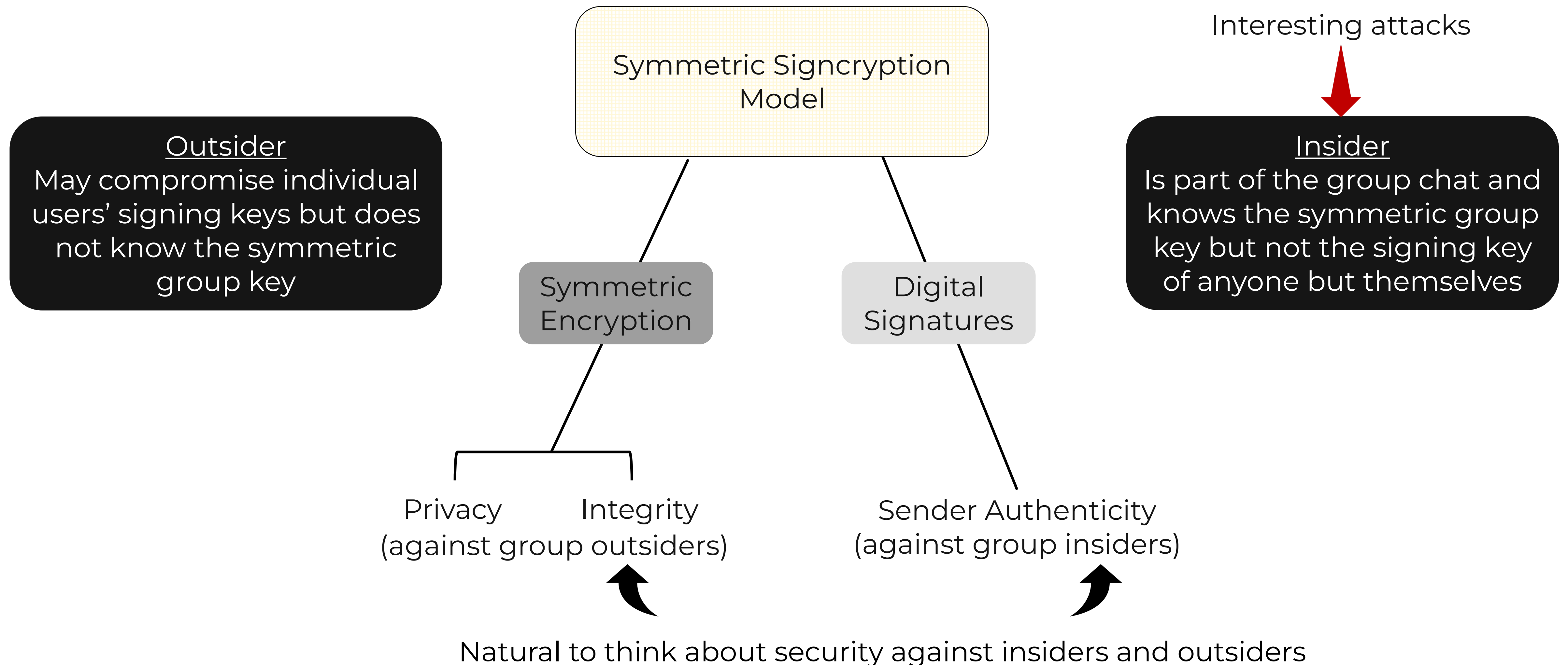
Insider and Outsider Attacks



Insider and Outsider Attacks



Insider and Outsider Attacks



Natural Constructions

Natural Constructions

Sign-then-Encrypt (StE)

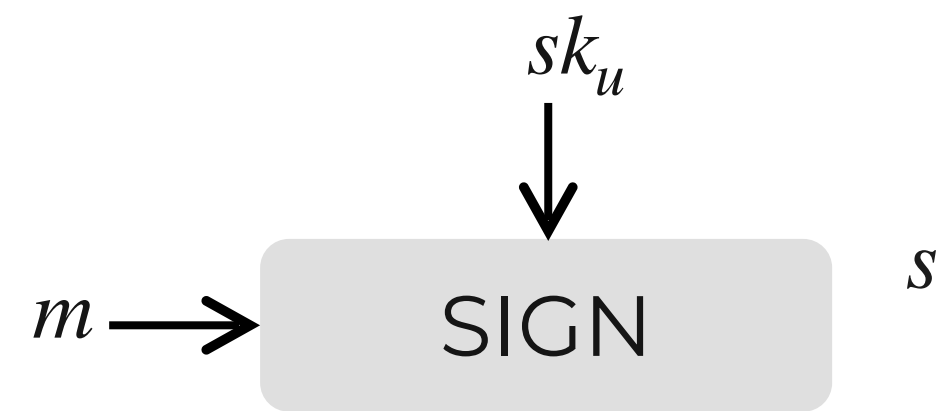
Natural Constructions

Sign-then-Encrypt (StE)

Encrypt-then-Sign (EtS)

Natural Constructions

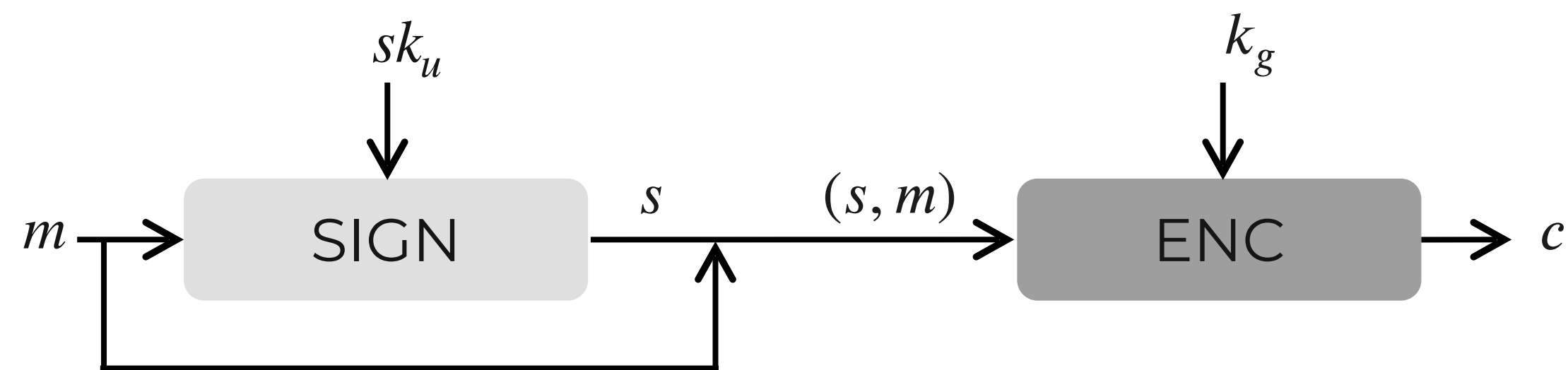
Sign-then-Encrypt (StE)



Encrypt-then-Sign (EtS)

Natural Constructions

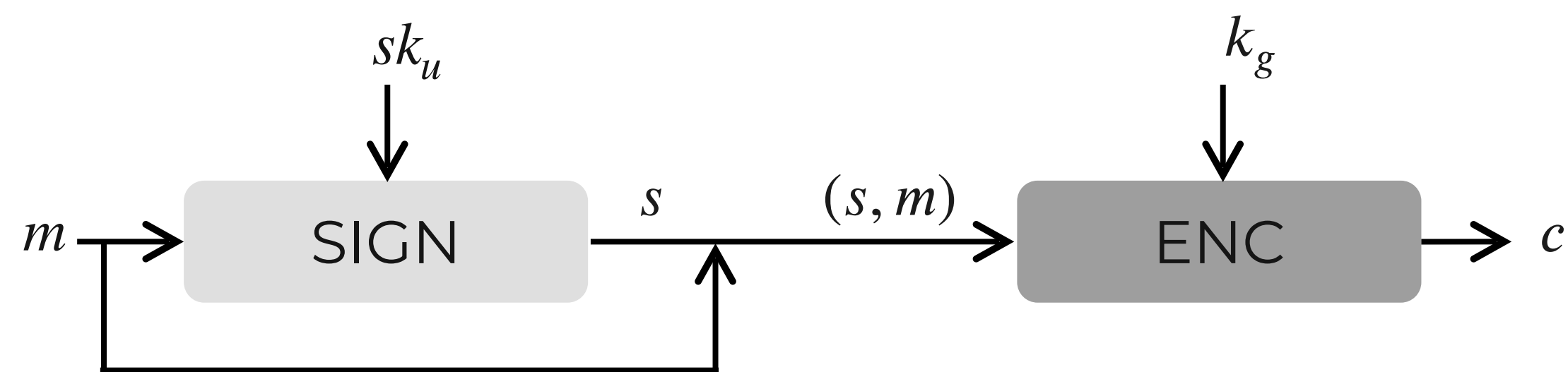
Sign-then-Encrypt (StE)



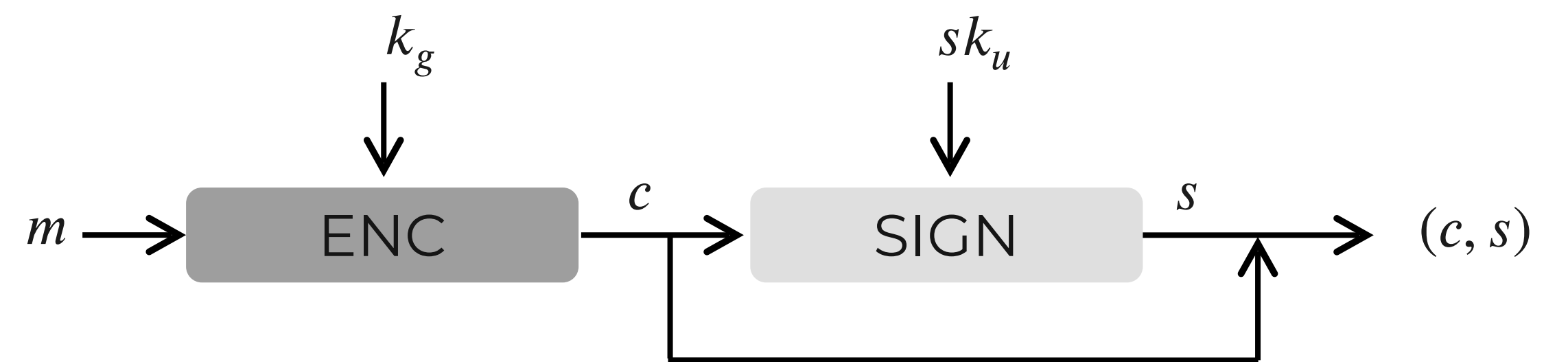
Encrypt-then-Sign (EtS)

Natural Constructions

Sign-then-Encrypt (StE)

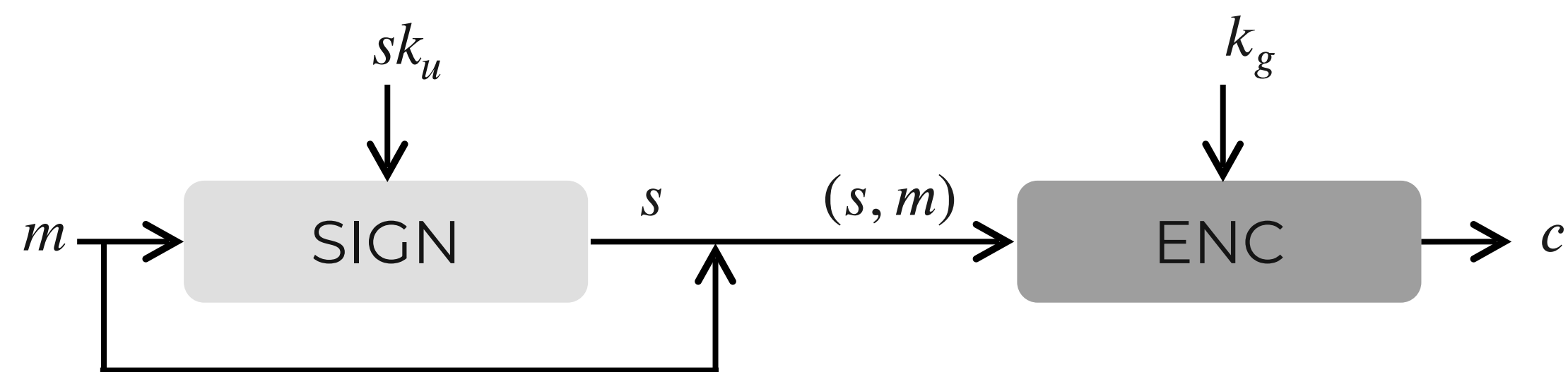


Encrypt-then-Sign (EtS)

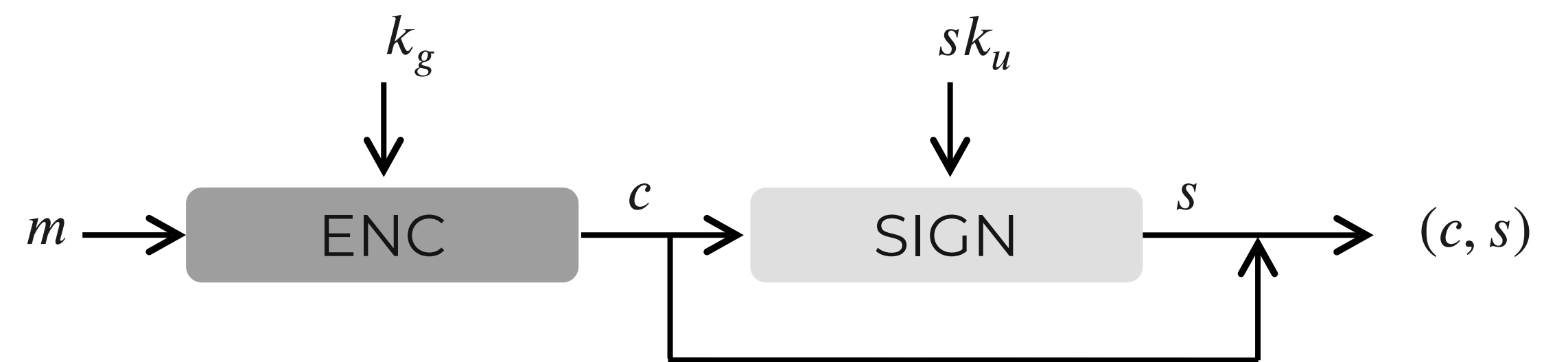


Natural Constructions

Sign-then-Encrypt (StE)

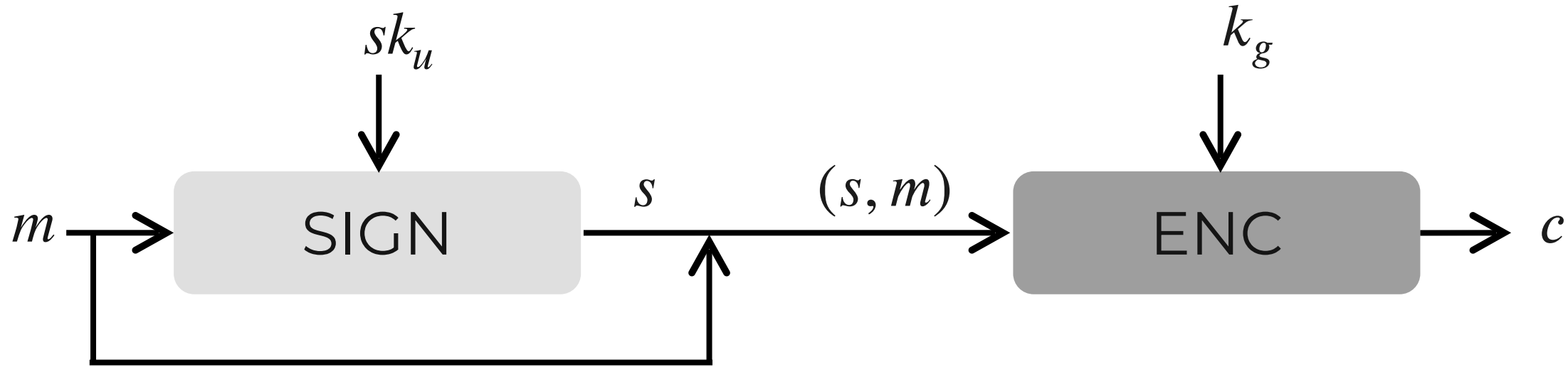


Encrypt-then-Sign (EtS)

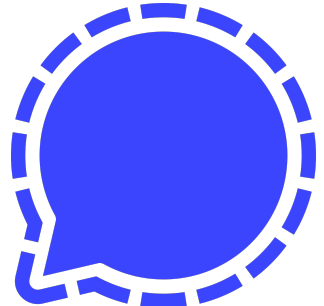
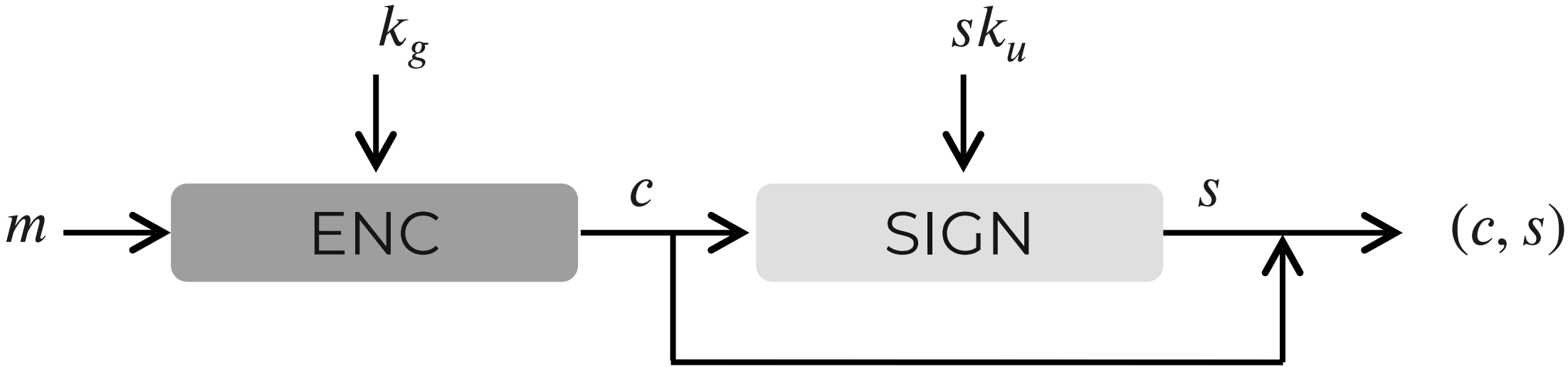


Natural Constructions

Sign-then-Encrypt (StE)

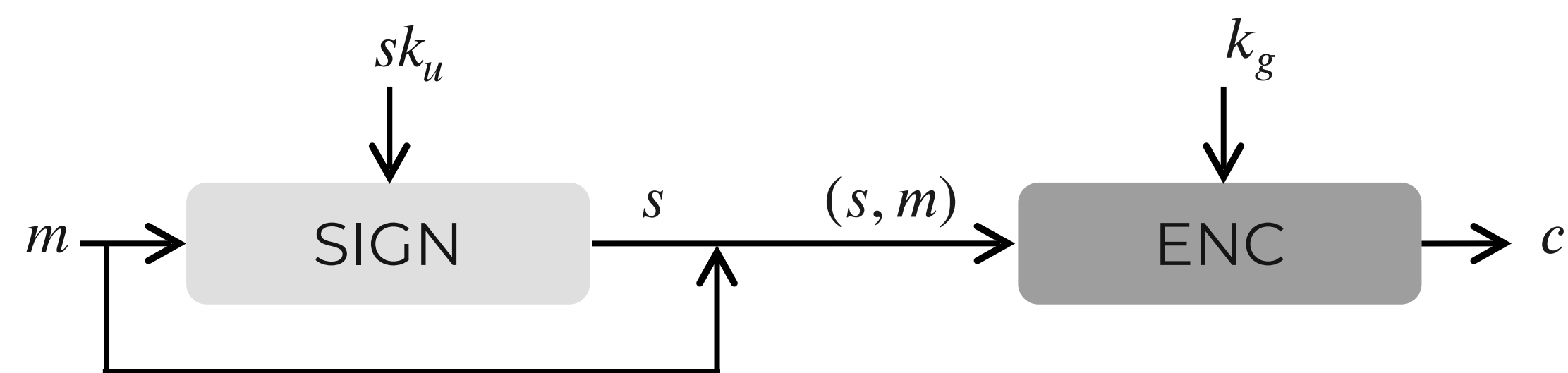


Encrypt-then-Sign (EtS)

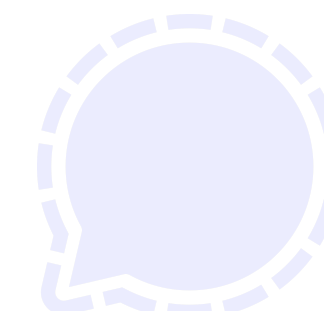
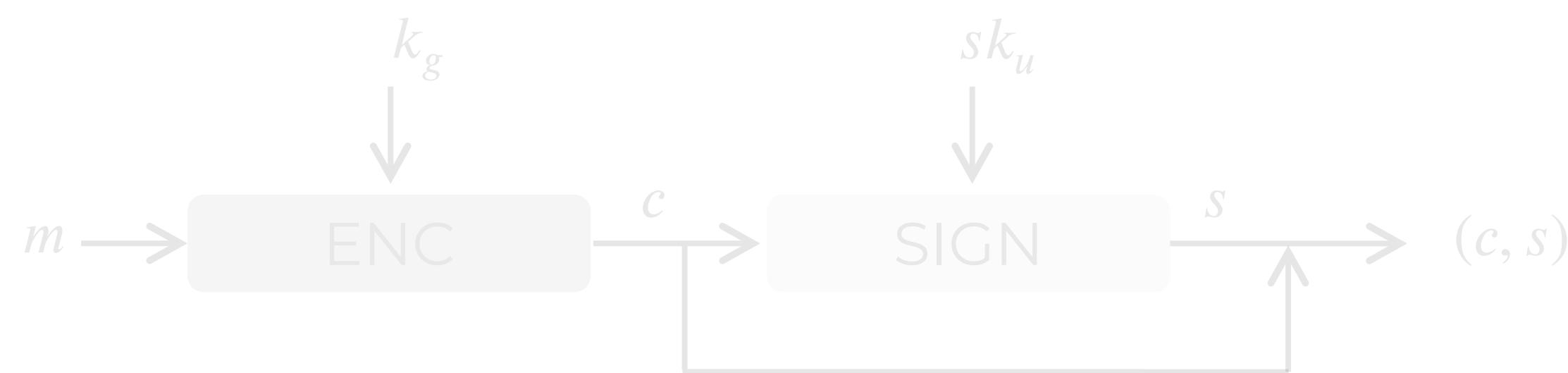


Natural Constructions

Sign-then-Encrypt (StE)



Encrypt-then-Sign (EtS)



Context-Switching Attacks

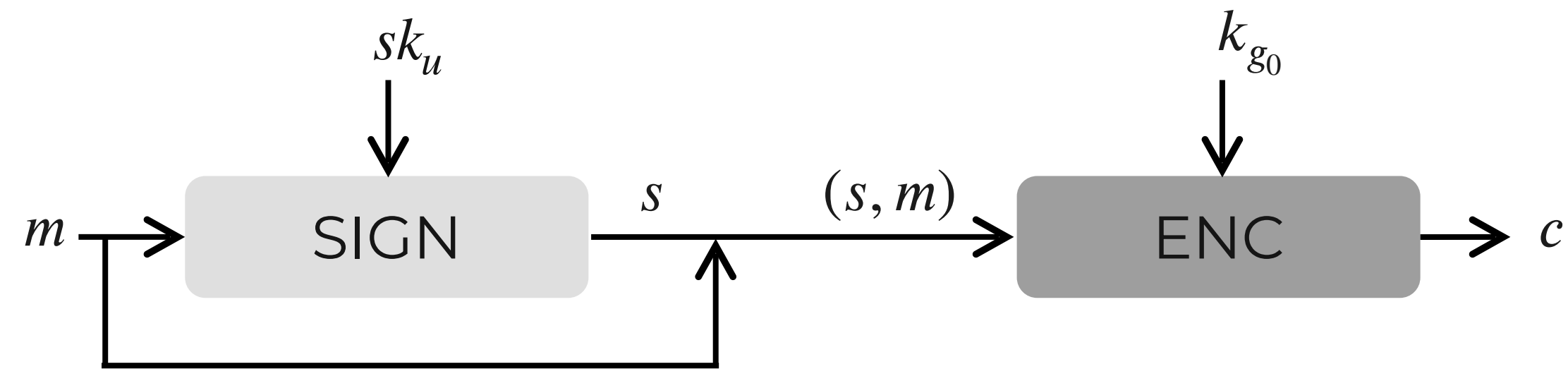
Context-Switching Attacks



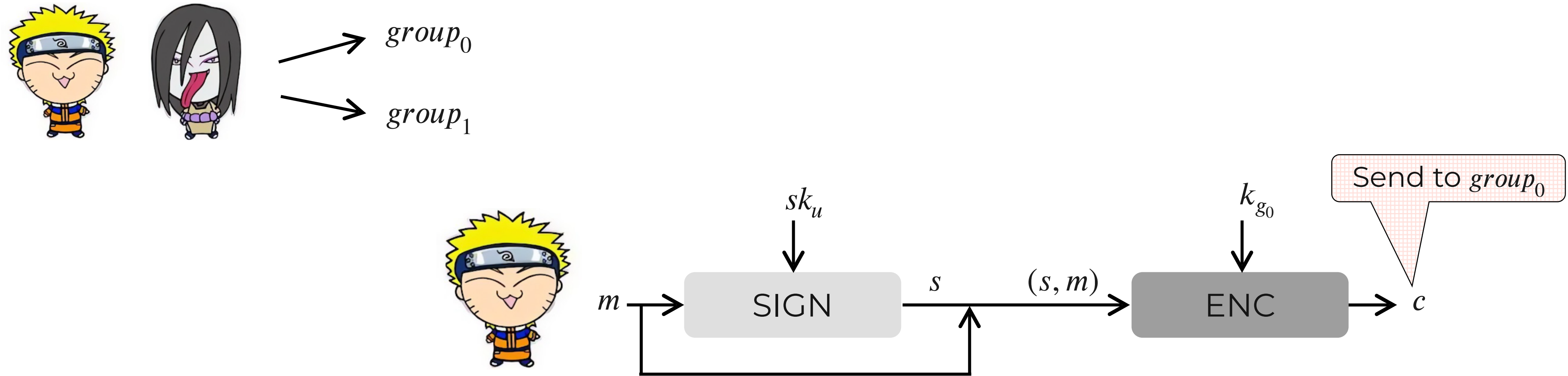
Context-Switching Attacks



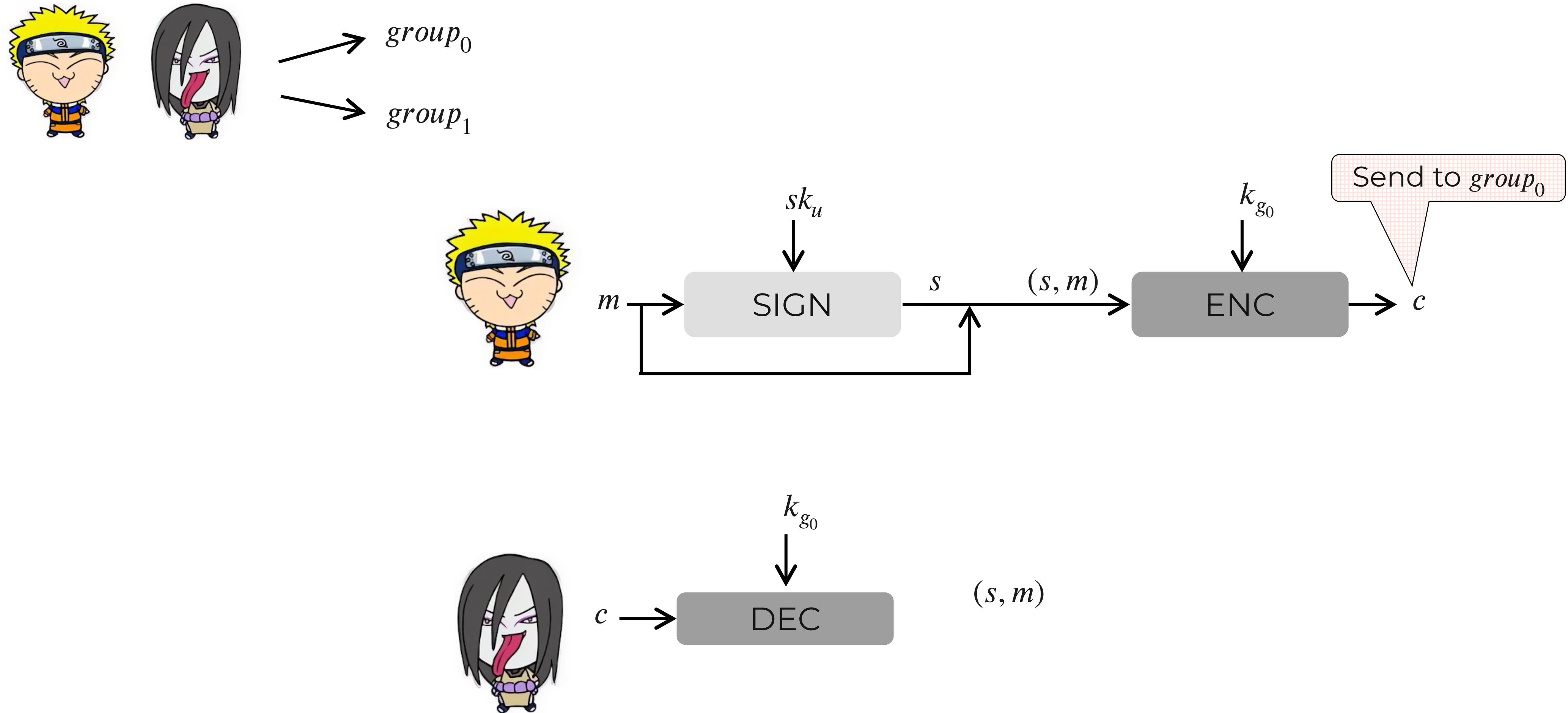
$\nearrow group_0$
 $\nearrow group_1$



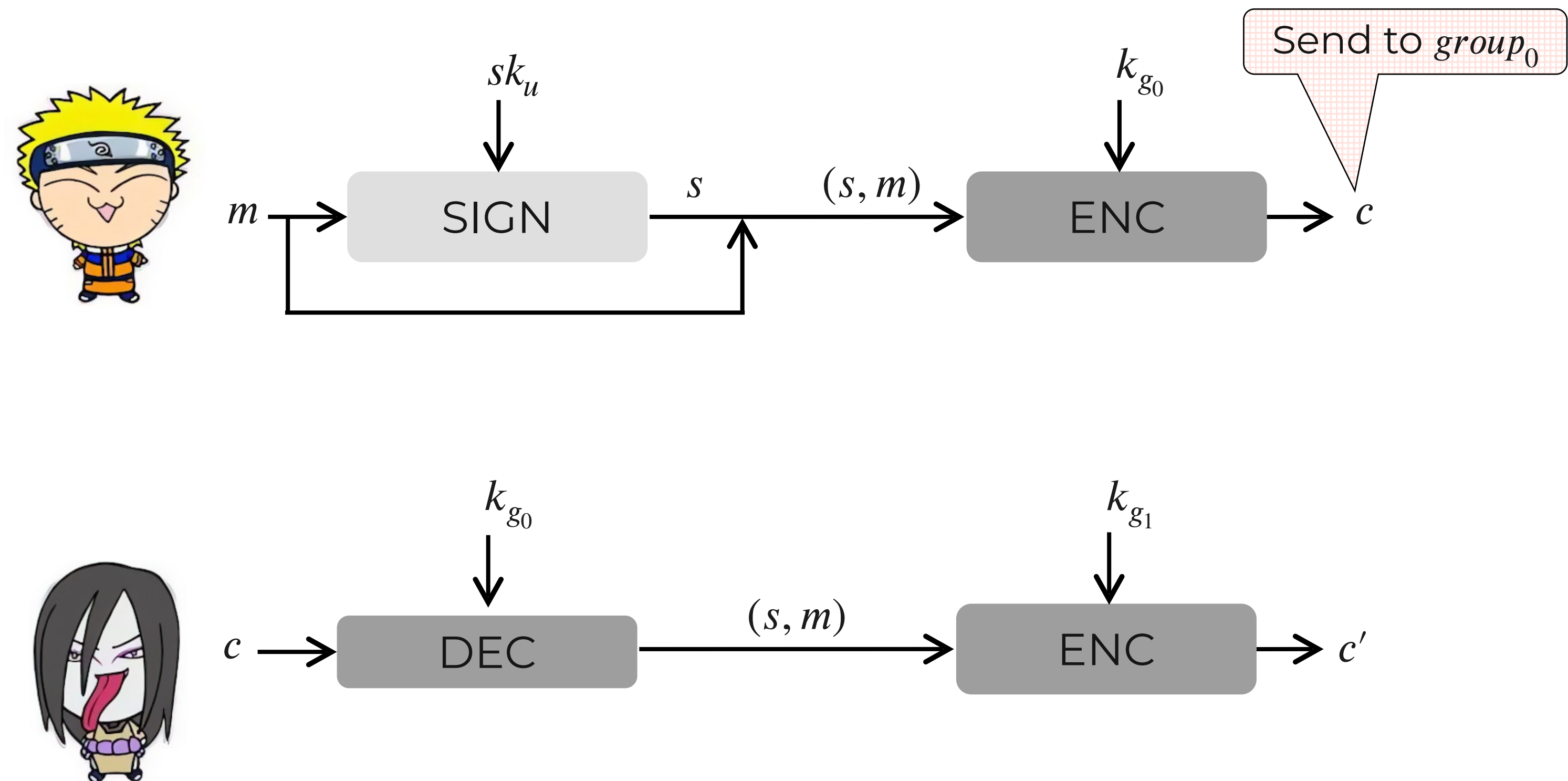
Context-Switching Attacks



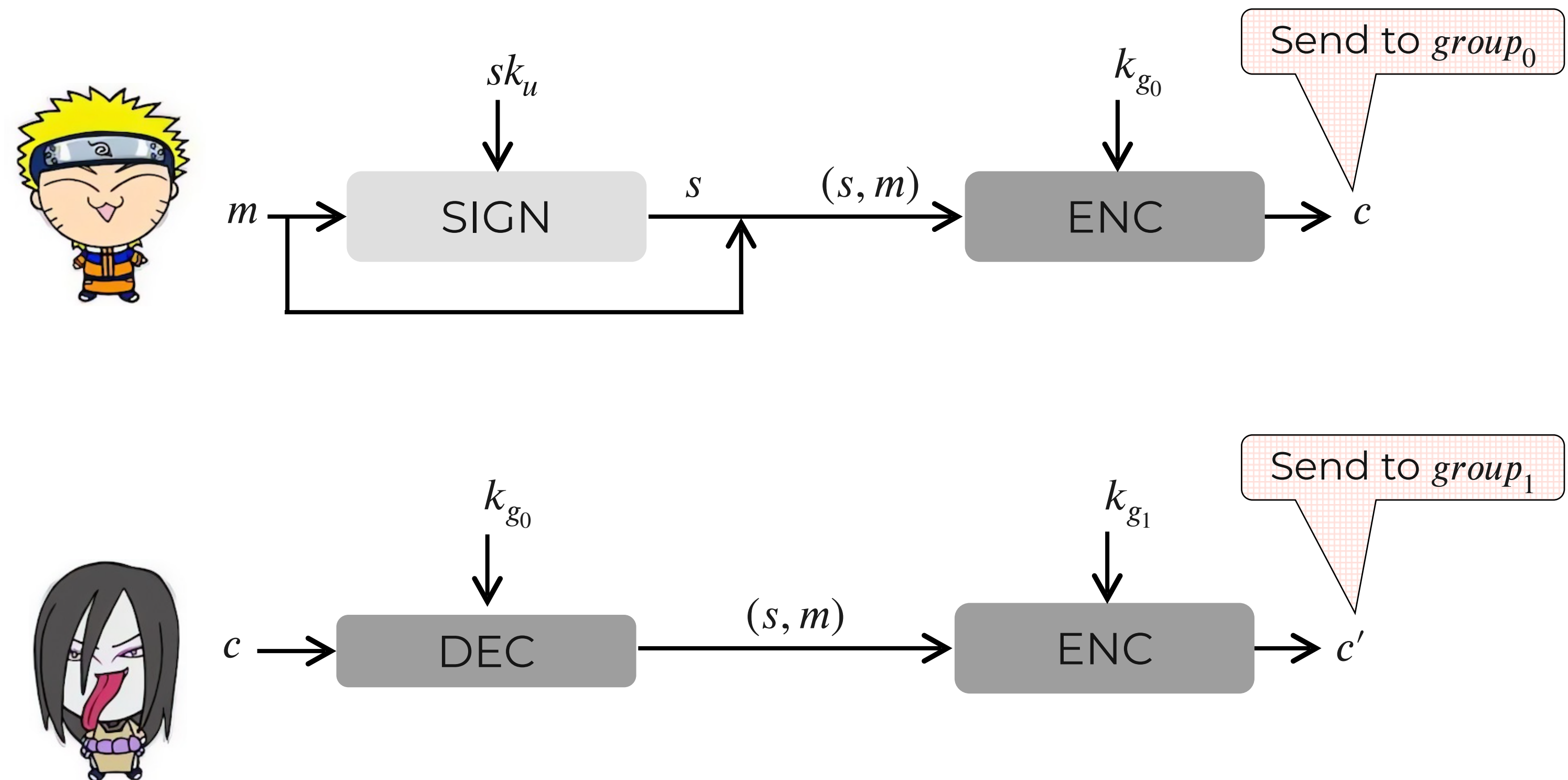
Context-Switching Attacks



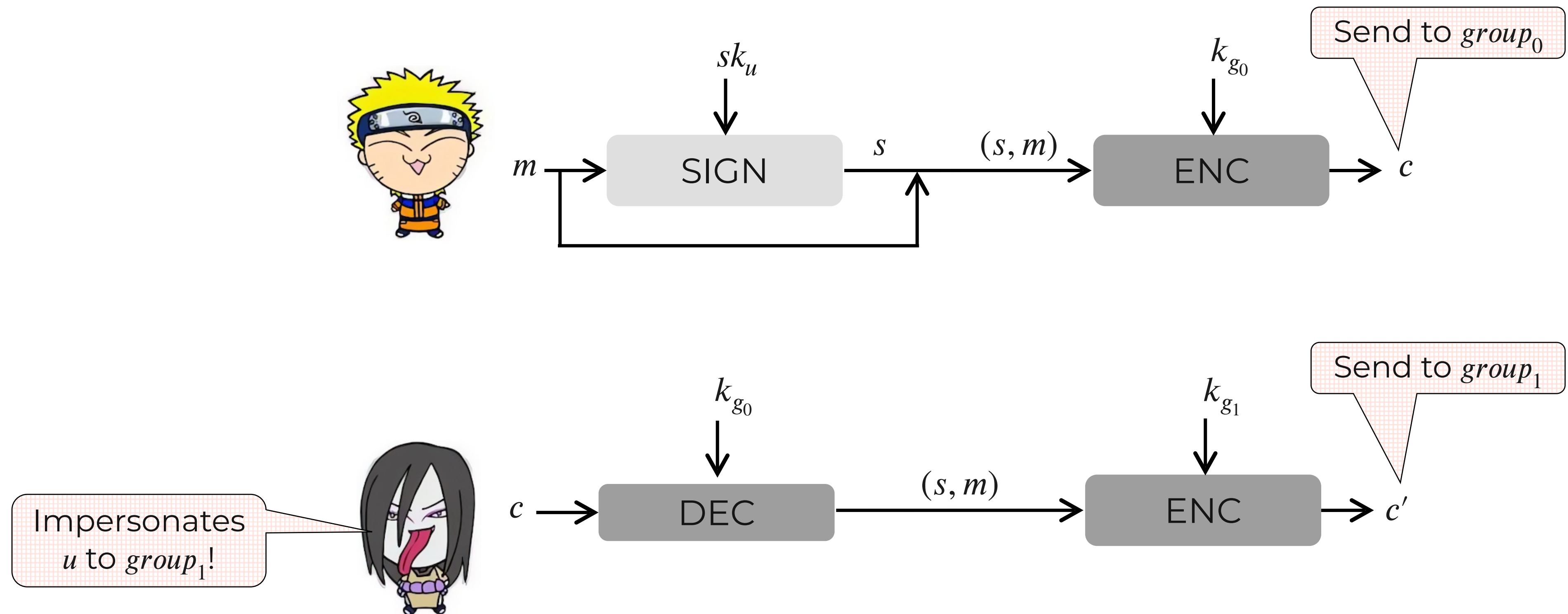
Context-Switching Attacks



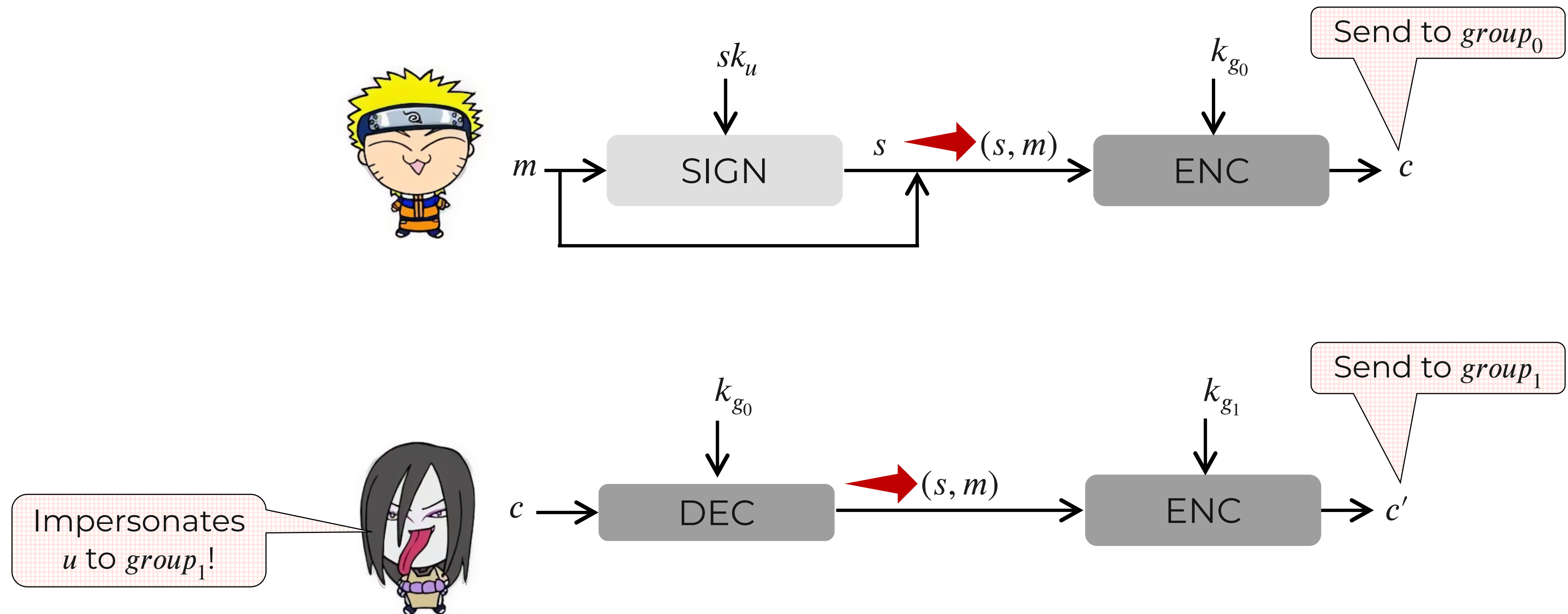
Context-Switching Attacks



Context-Switching Attacks



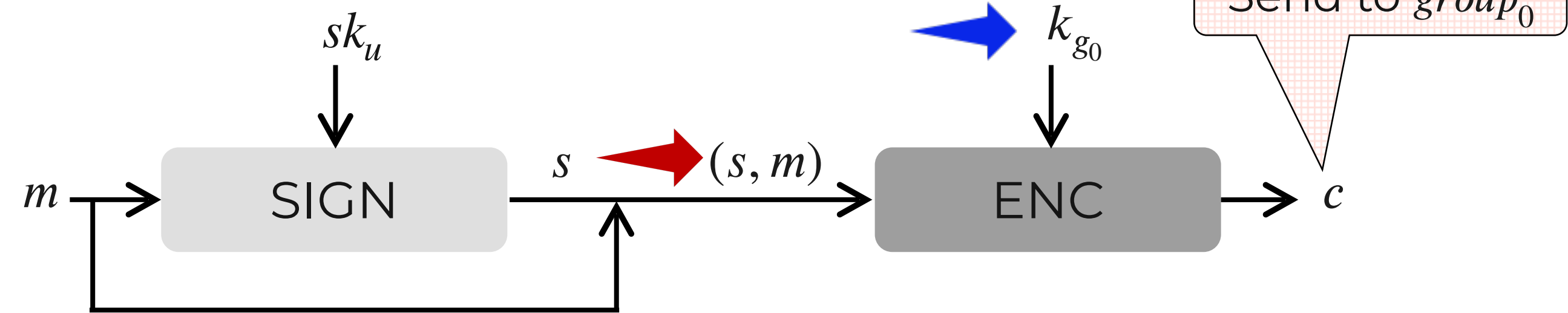
Context-Switching Attacks



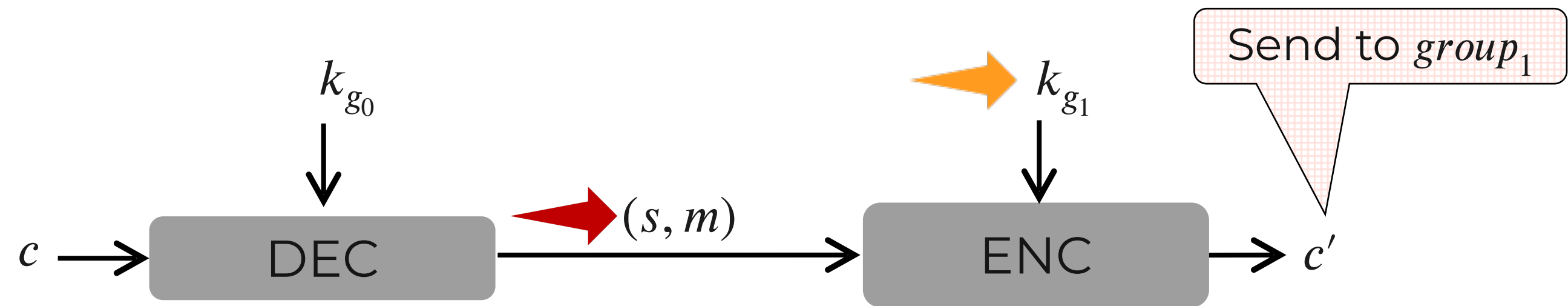
Context-Switching Attacks



$\rightarrow group_0$
 $\rightarrow group_1$

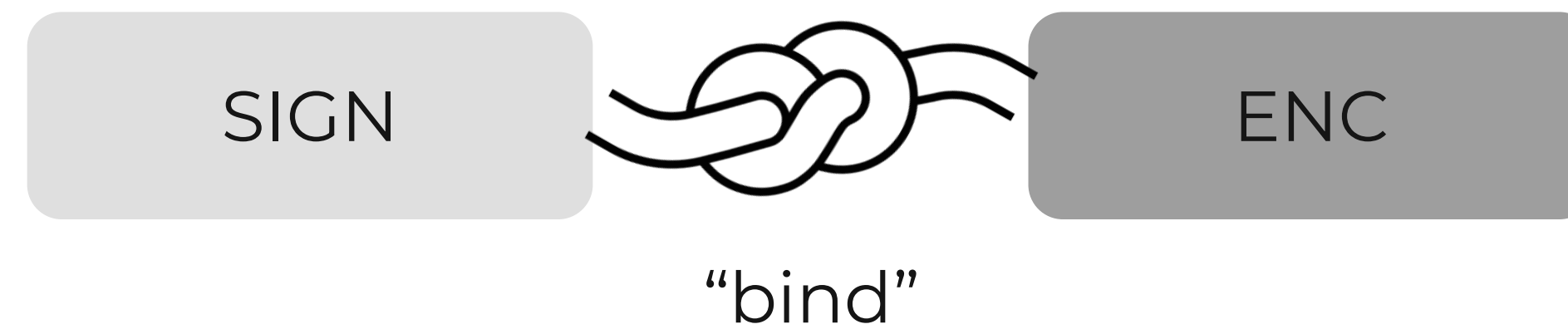


Impersonates
 u to $group_1$!

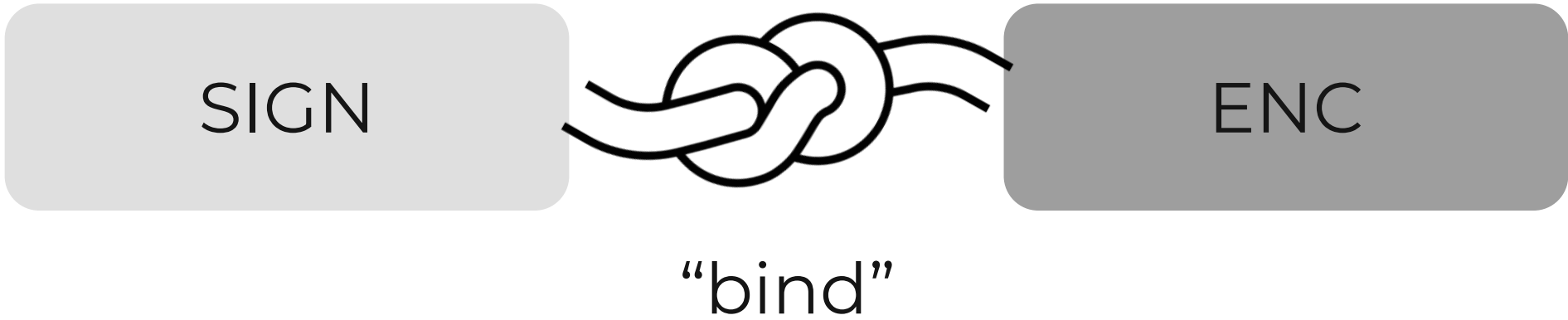


Context Binding

Context Binding

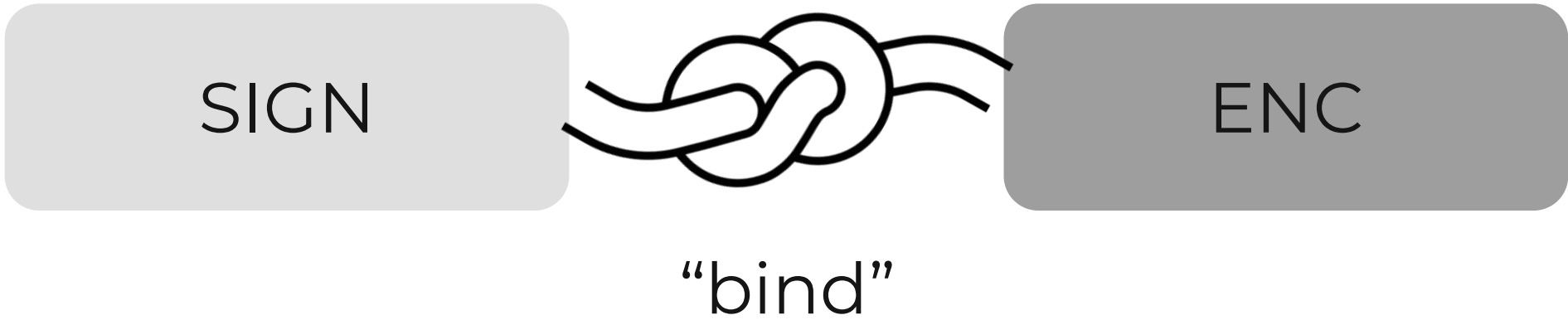


Context Binding



SIGN
group_key_id
ad
n

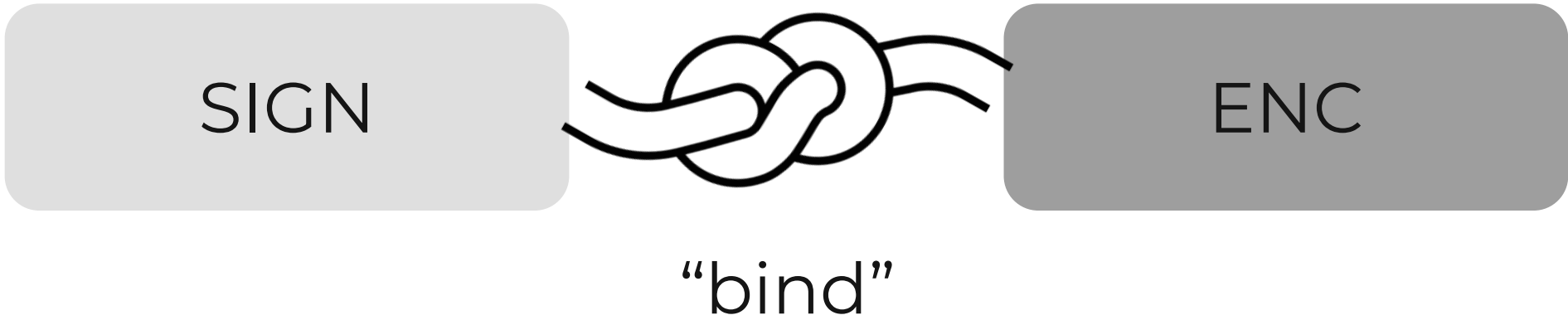
Context Binding



Value that uniquely identifies
the group's encryption key

SIGN
group_key_id
ad
n

Context Binding

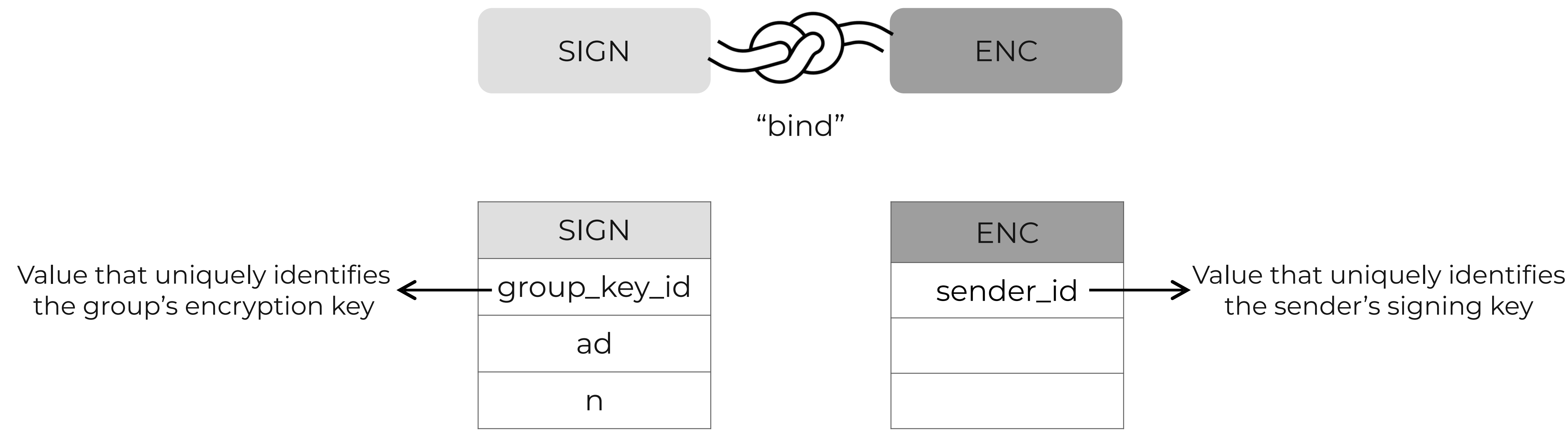


Value that uniquely identifies
the group's encryption key

SIGN
group_key_id
ad
n

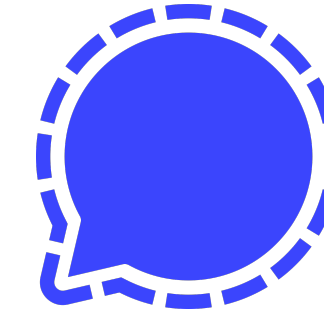
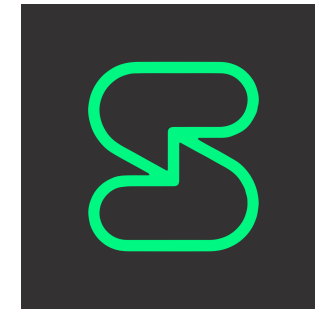
ENC
sender_id

Context Binding



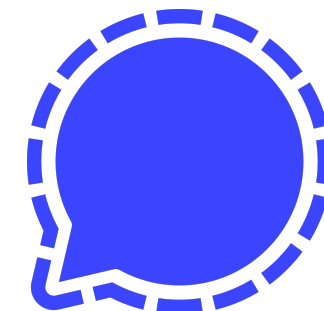
Our Analysis: An Overview

Our Analysis: An Overview



[matrix]

Our Analysis: An Overview

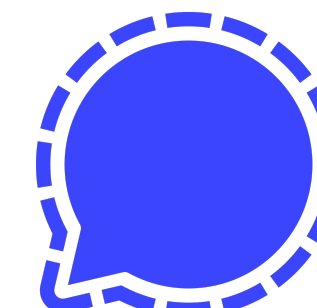


[matrix]



Insider Replay
Insider Re-ordering

Our Analysis: An Overview



[**matrix**]

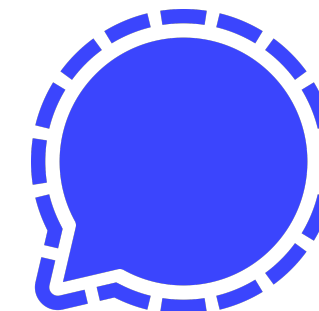


Insider Replay
Insider Re-ordering



No context binding

Our Analysis: An Overview



[**matrix**]



Insider Replay
Insider Re-ordering



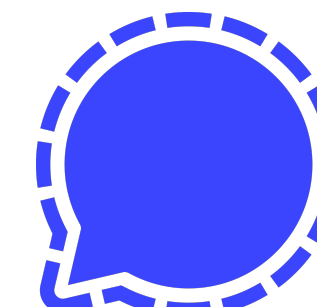
Insider Replay
Outsider Replay
Outsider Forgery*



No context binding

* stolen signing key

Our Analysis: An Overview



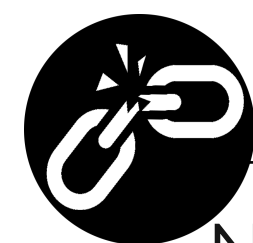
[**matrix**]



Insider Replay
Insider Re-ordering



Insider Replay
Outsider Replay
Outsider Forgery*



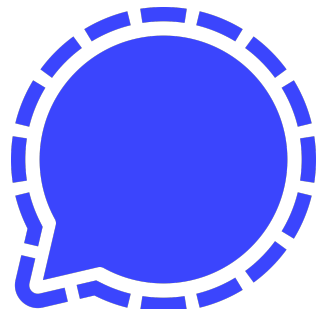
No context binding



No context binding

* stolen signing key

Our Analysis: An Overview



[**matrix**]



Insider Replay
Insider Re-ordering



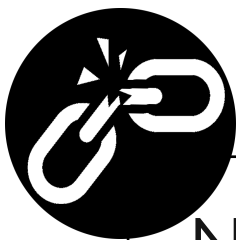
Insider Replay
Outsider Replay
Outsider Forgery*



--



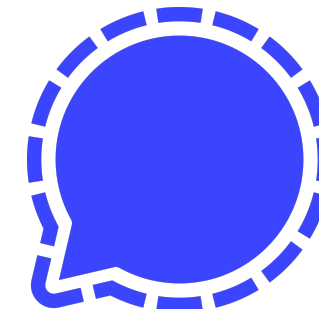
No context binding



No context binding

* stolen signing key

Our Analysis: An Overview



[**matrix**]



Insider Replay
Insider Re-ordering



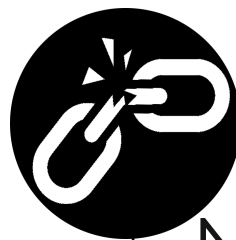
Insider Replay
Outsider Replay
Outsider Forgery*



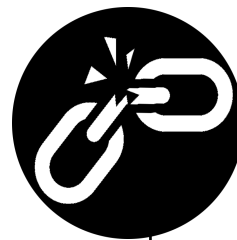
--



No context binding



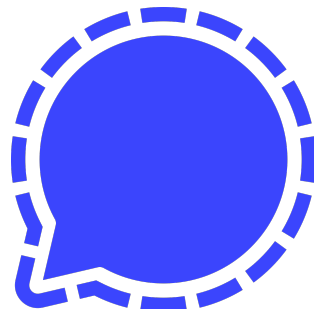
No context binding



Key Reuse
Key Cycle

* stolen signing key

Our Analysis: An Overview



[matrix]



Insider Replay
Insider Re-ordering



Insider Replay
Outsider Replay
Outsider Forgery*



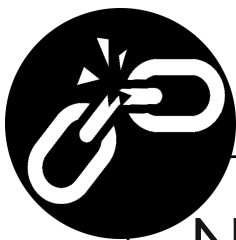
--



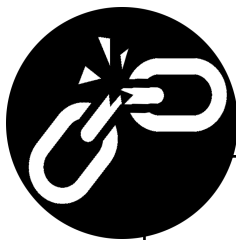
Outsider Forgery*



No context binding



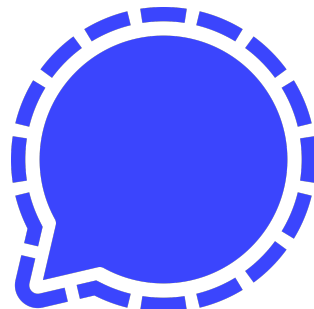
No context binding



Key Reuse
Key Cycle

* stolen signing key * discovered by [BCG23]

Our Analysis: An Overview



[matrix]



Insider Replay
Insider Re-ordering



Insider Replay
Outsider Replay
Outsider Forgery*



--



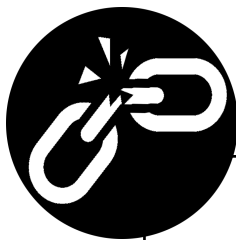
Outsider Forgery*



No context binding



No context binding



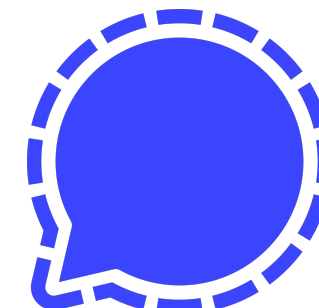
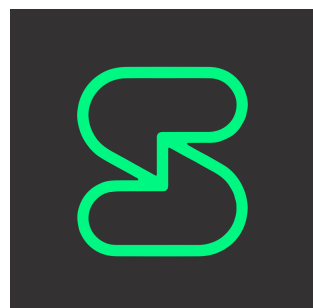
Key Reuse
Key Cycle



Unauthenticated
Symmetric
Encryption

* stolen signing key * discovered by [BCG23]

Our Analysis: An Overview



Insider Replay
Insider Re-ordering



Insider Replay
Outsider Replay
Outsider Forgery*



--



Outsider Forgery*



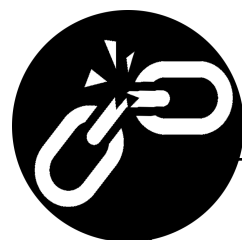
In-model Insider Replay



No context binding



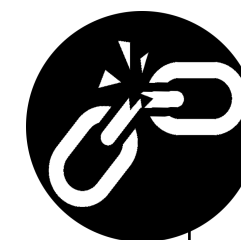
No context binding



Key Reuse
Key Cycle



Unauthenticated
Symmetric
Encryption



No context binding

* stolen signing key * discovered by [BCG23]

Our Analysis: An Overview



Insider Replay
Insider Re-ordering



Insider Replay
Outsider Replay
Outsider Forgery*



--



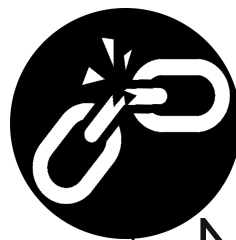
Outsider Forgery*



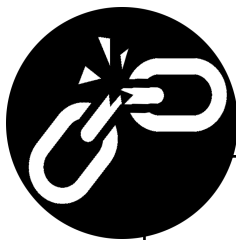
In-model Insider Replay



No context binding



No context binding



Key Reuse
Key Cycle



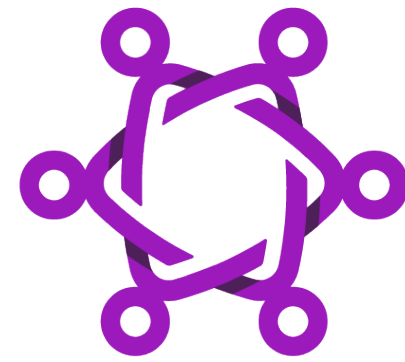
Unauthenticated
Symmetric
Encryption



No context binding

* stolen signing key * discovered by [BCG23]

Case Study I: MLS



MLS

Encryption Key Derivation in MLS

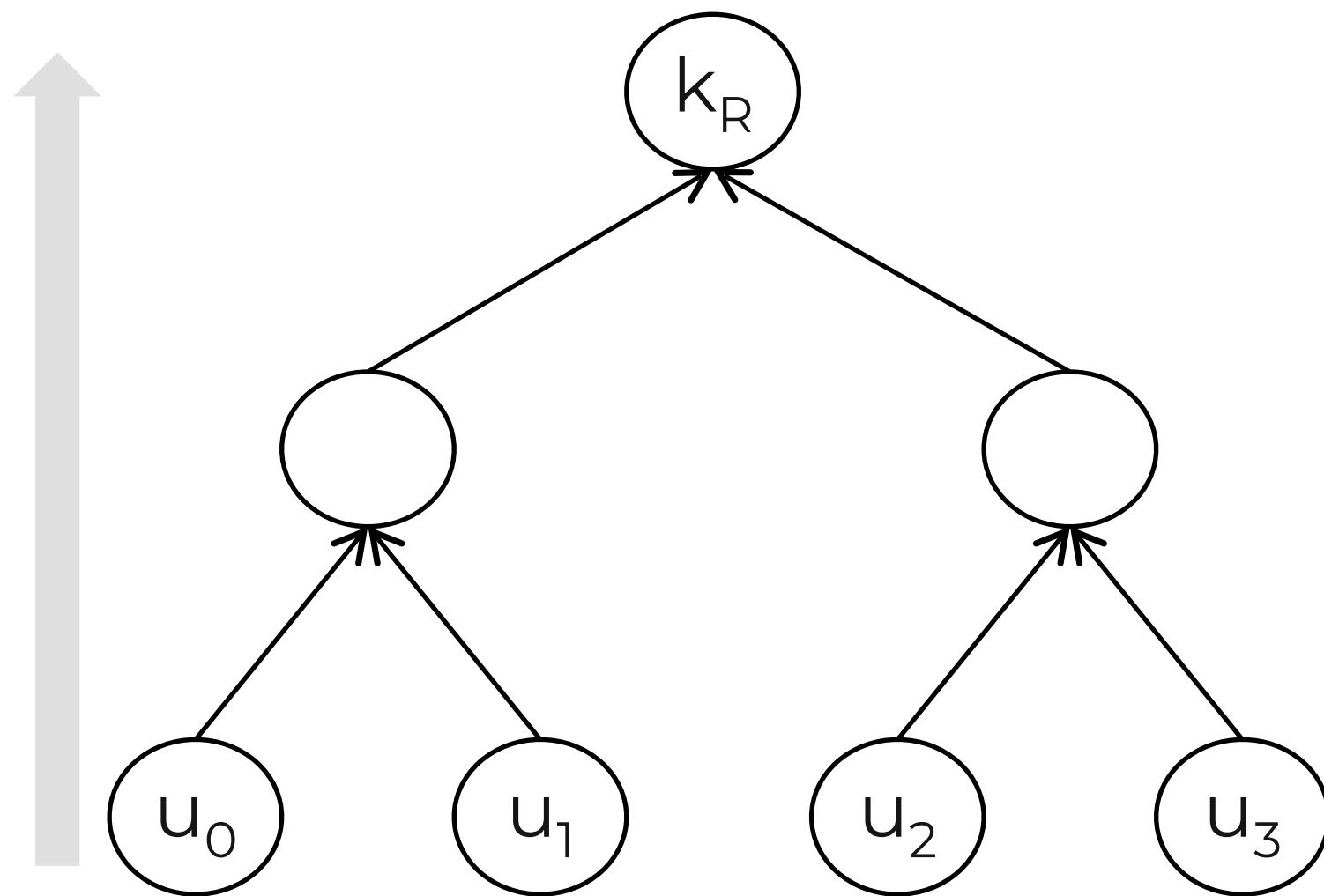
Encryption Key Derivation in MLS

Ratchet tree



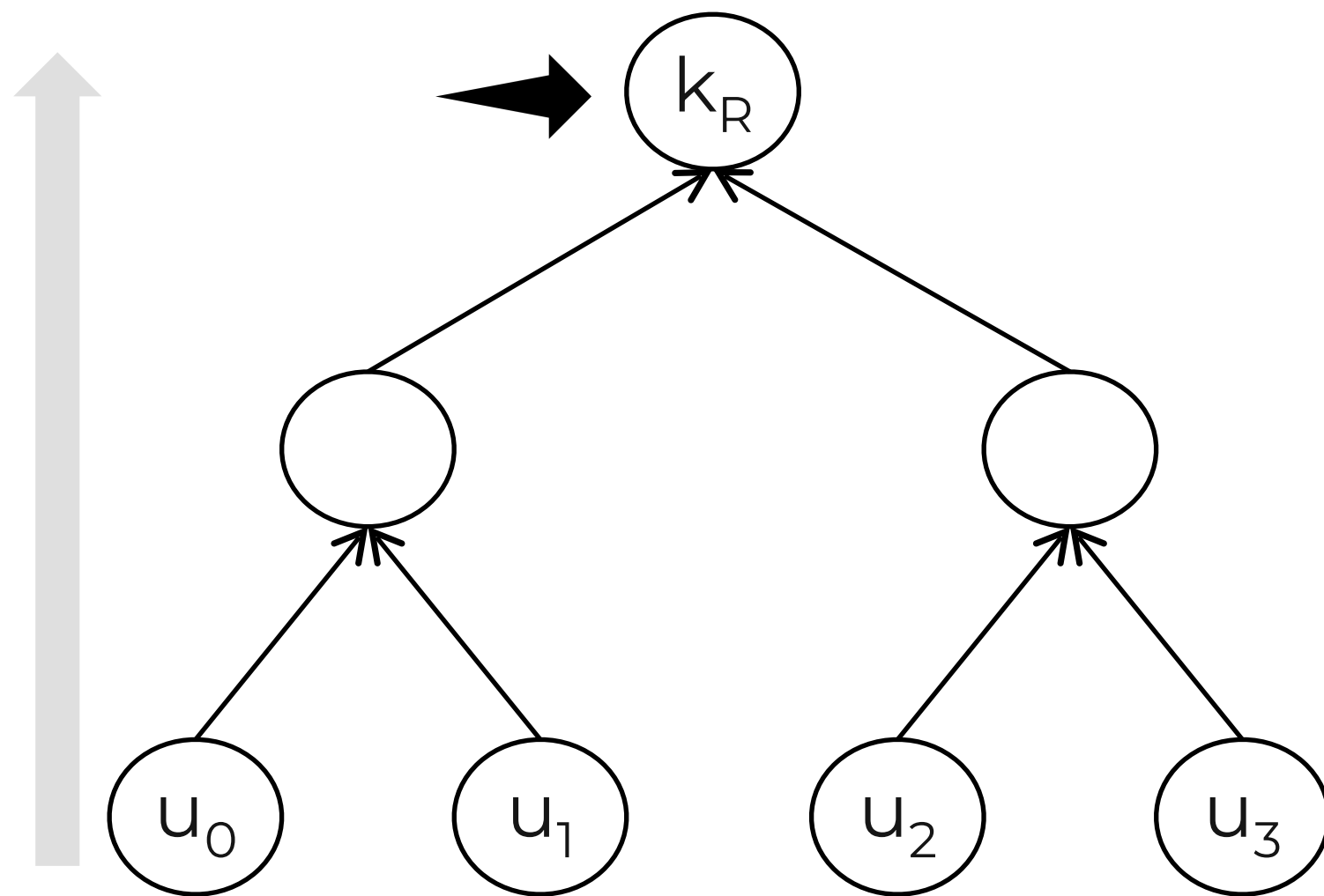
Encryption Key Derivation in MLS

Ratchet tree



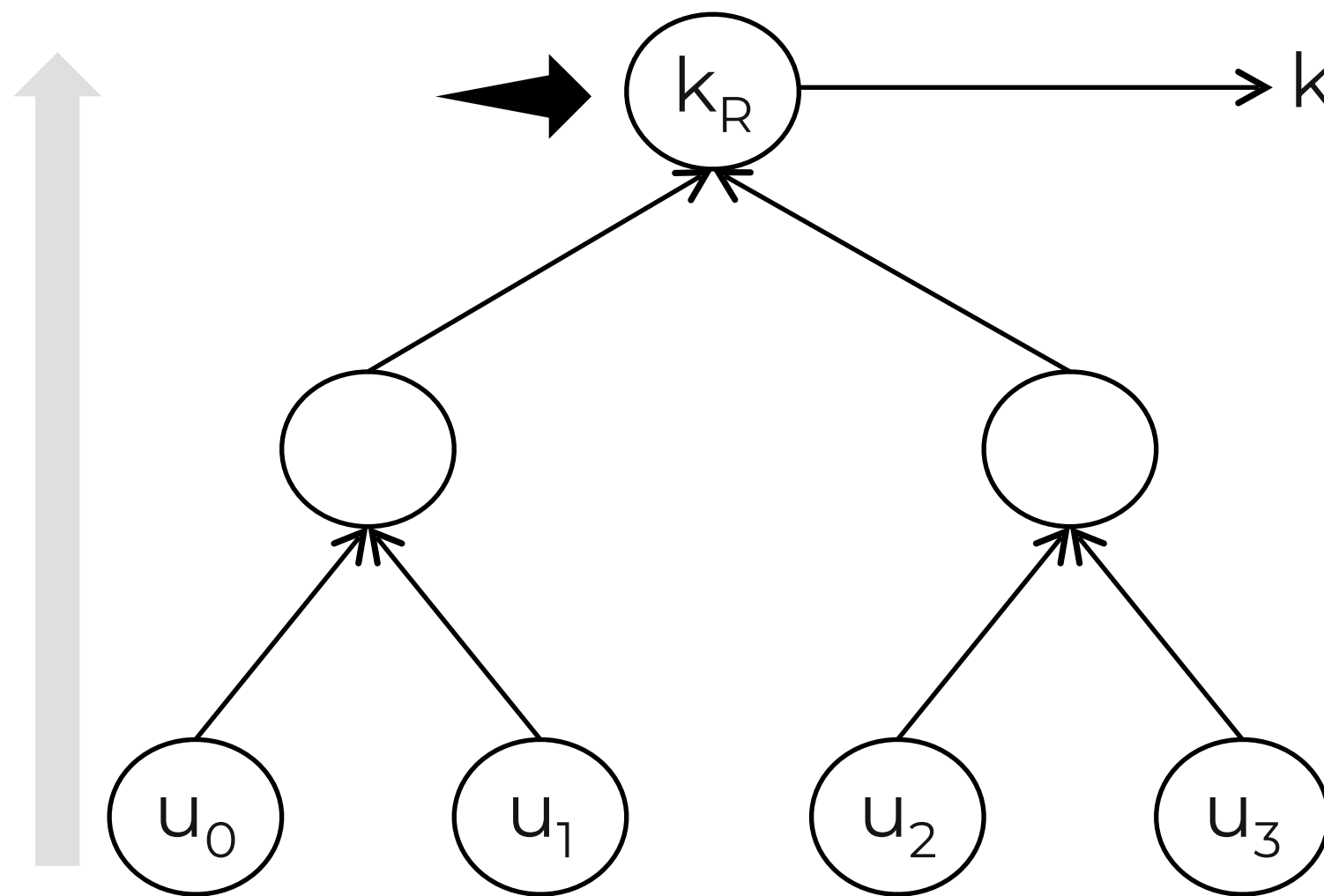
Encryption Key Derivation in MLS

Ratchet tree

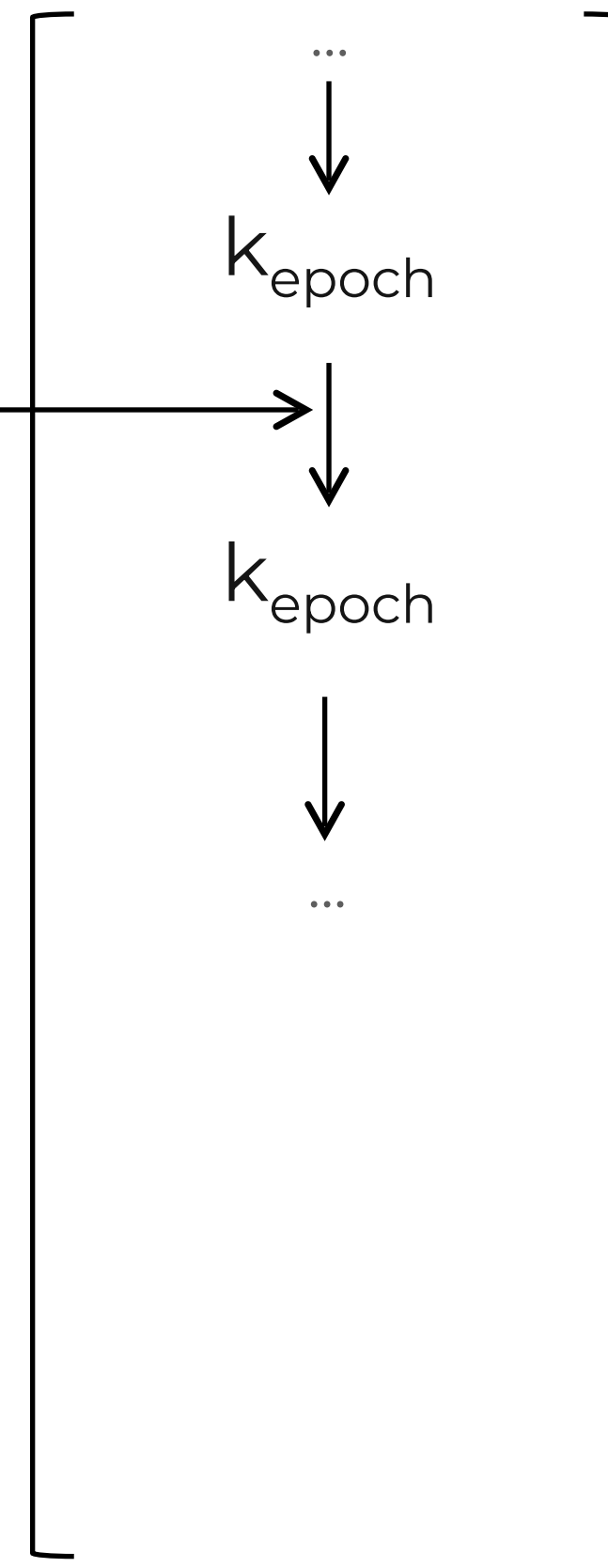


Encryption Key Derivation in MLS

Ratchet tree

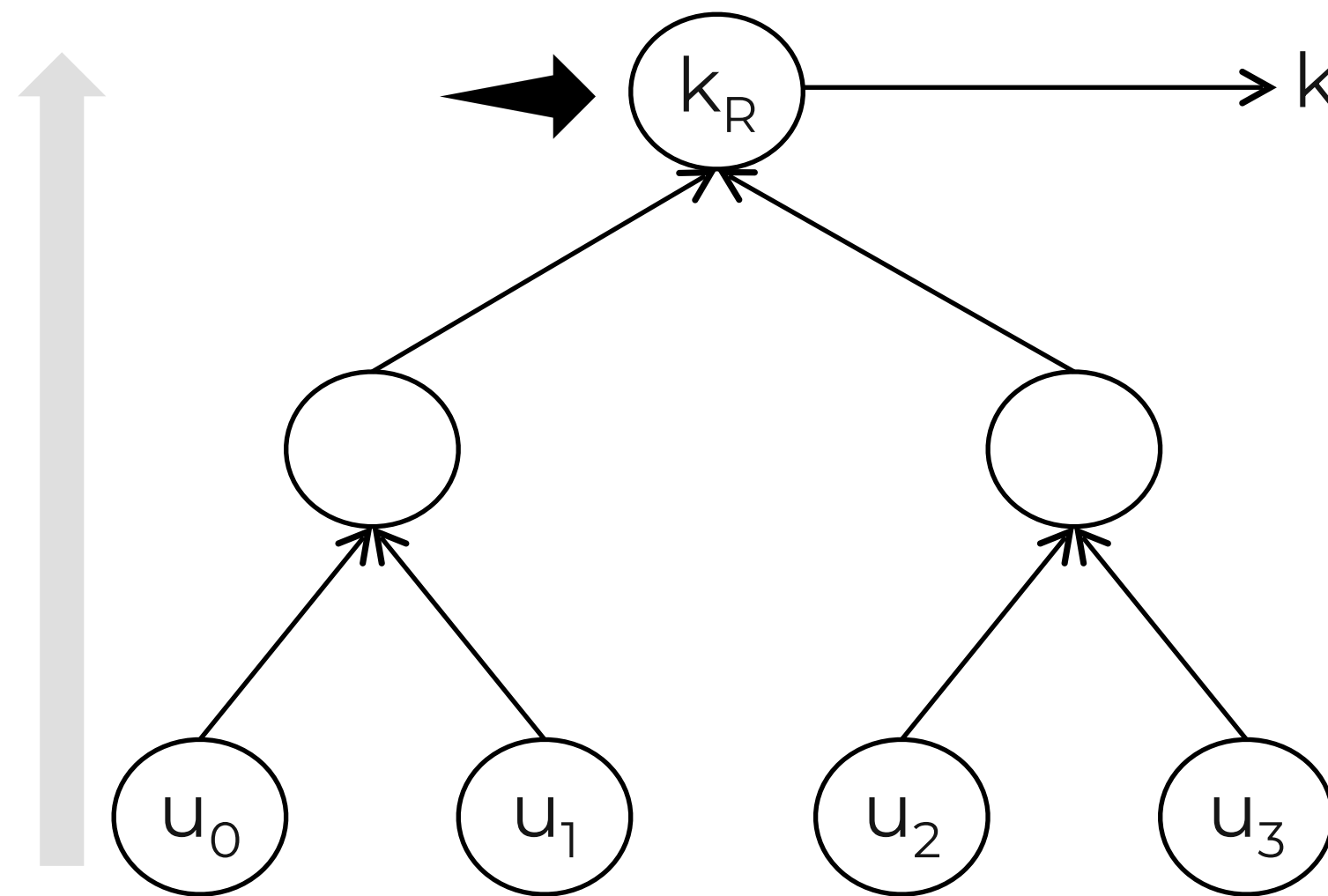


Key schedule

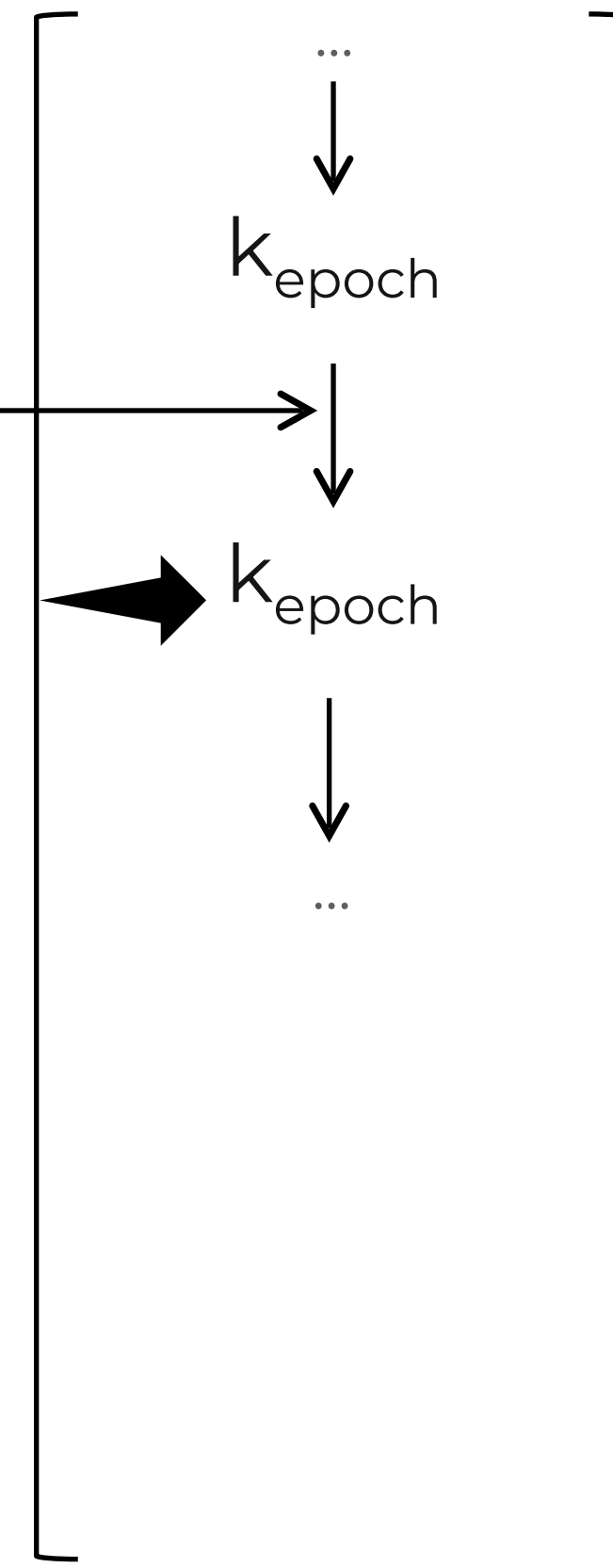


Encryption Key Derivation in MLS

Ratchet tree

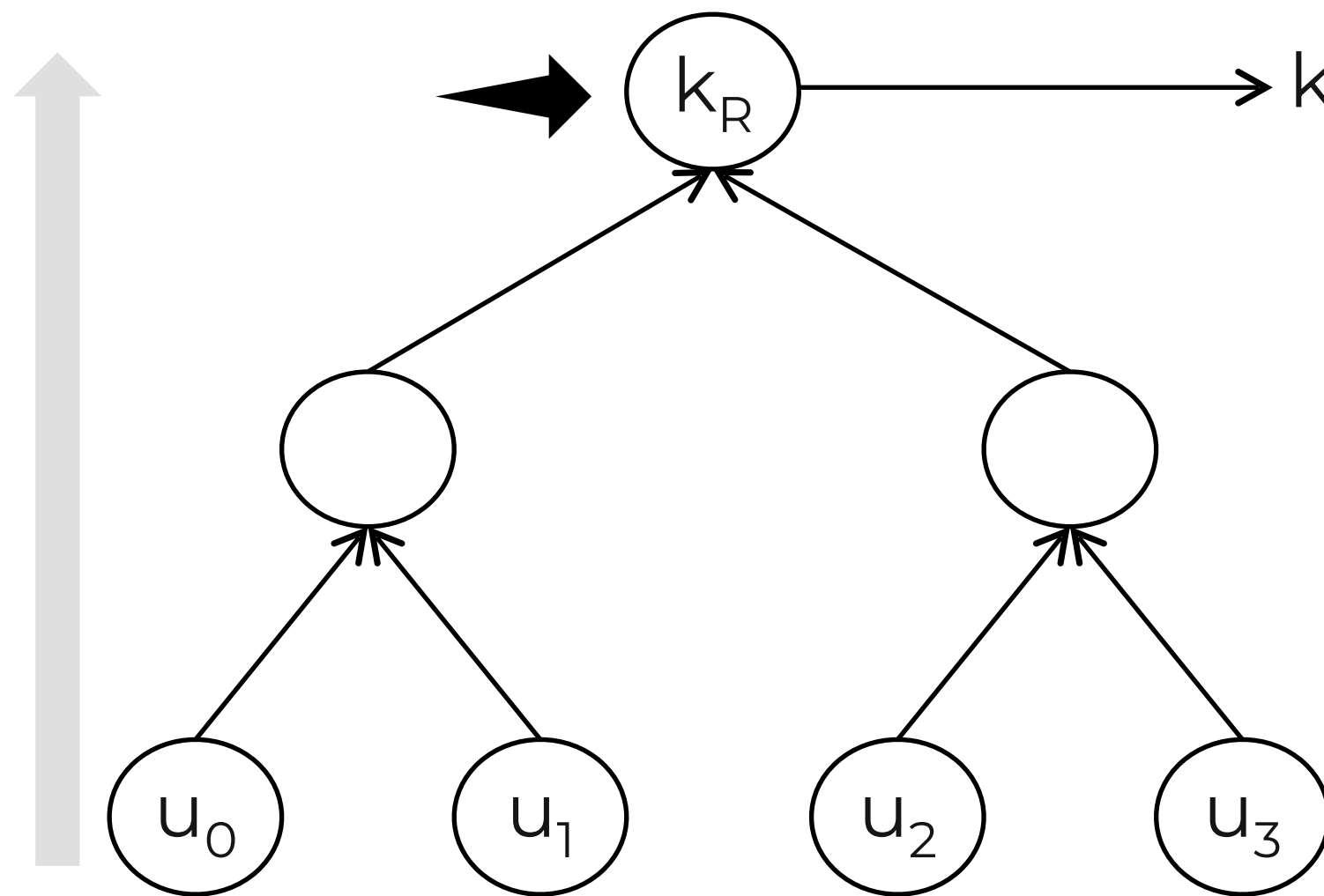


Key schedule

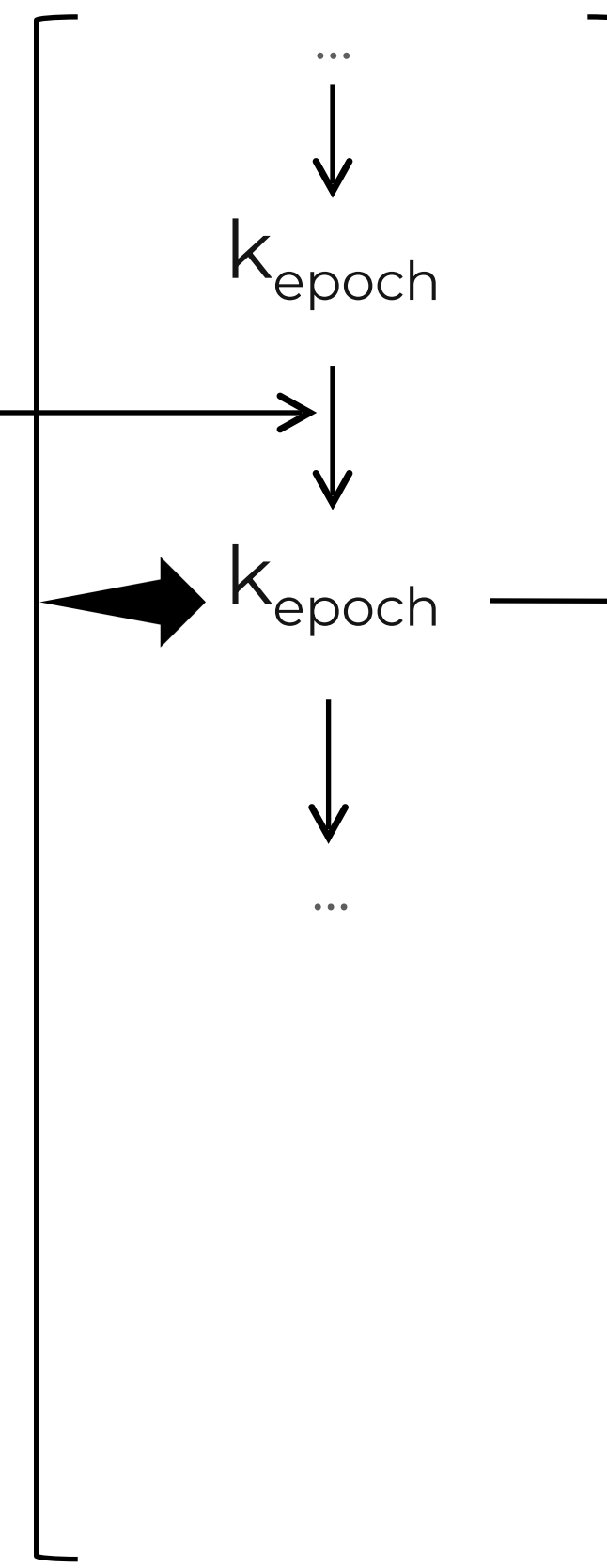


Encryption Key Derivation in MLS

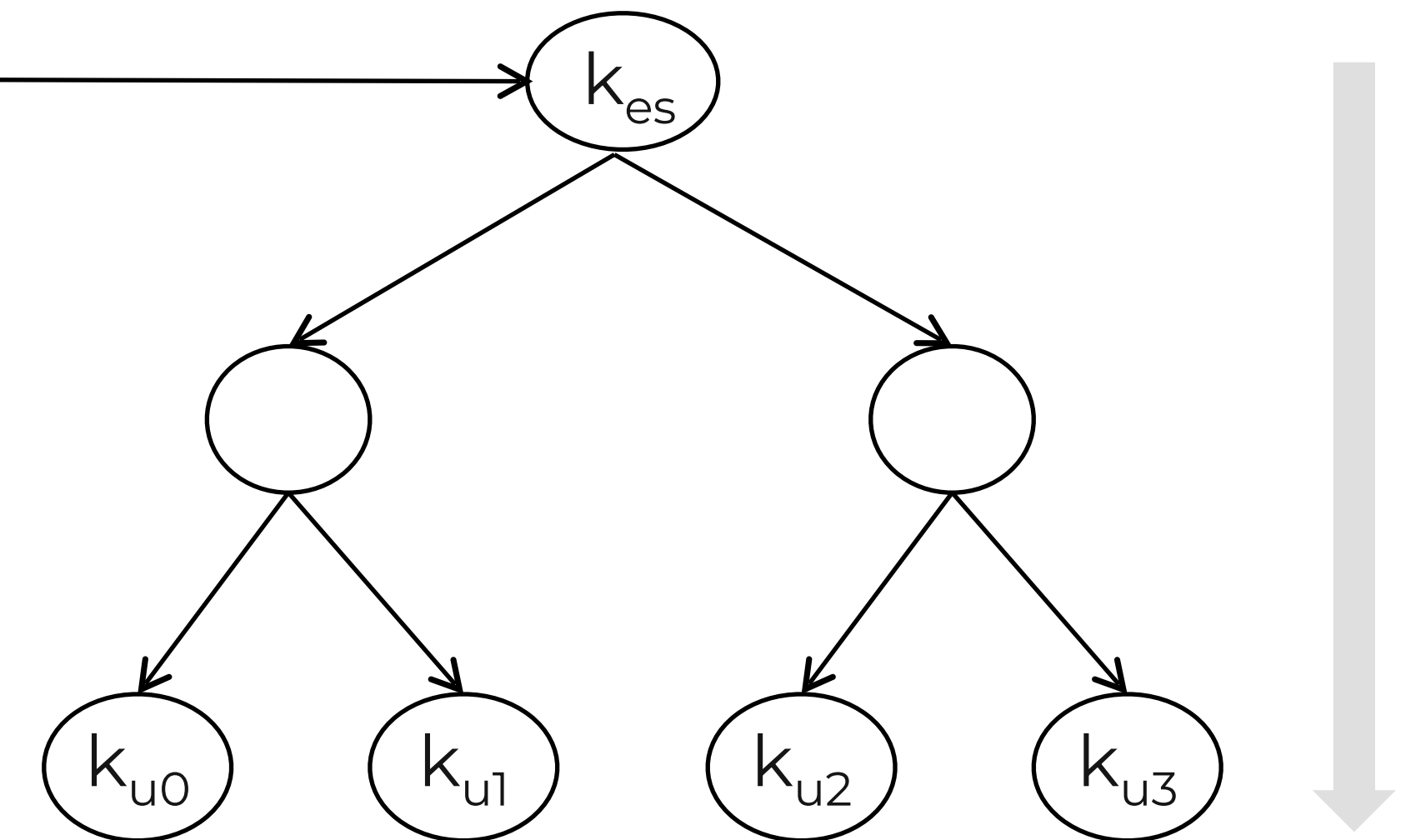
Ratchet tree



Key schedule

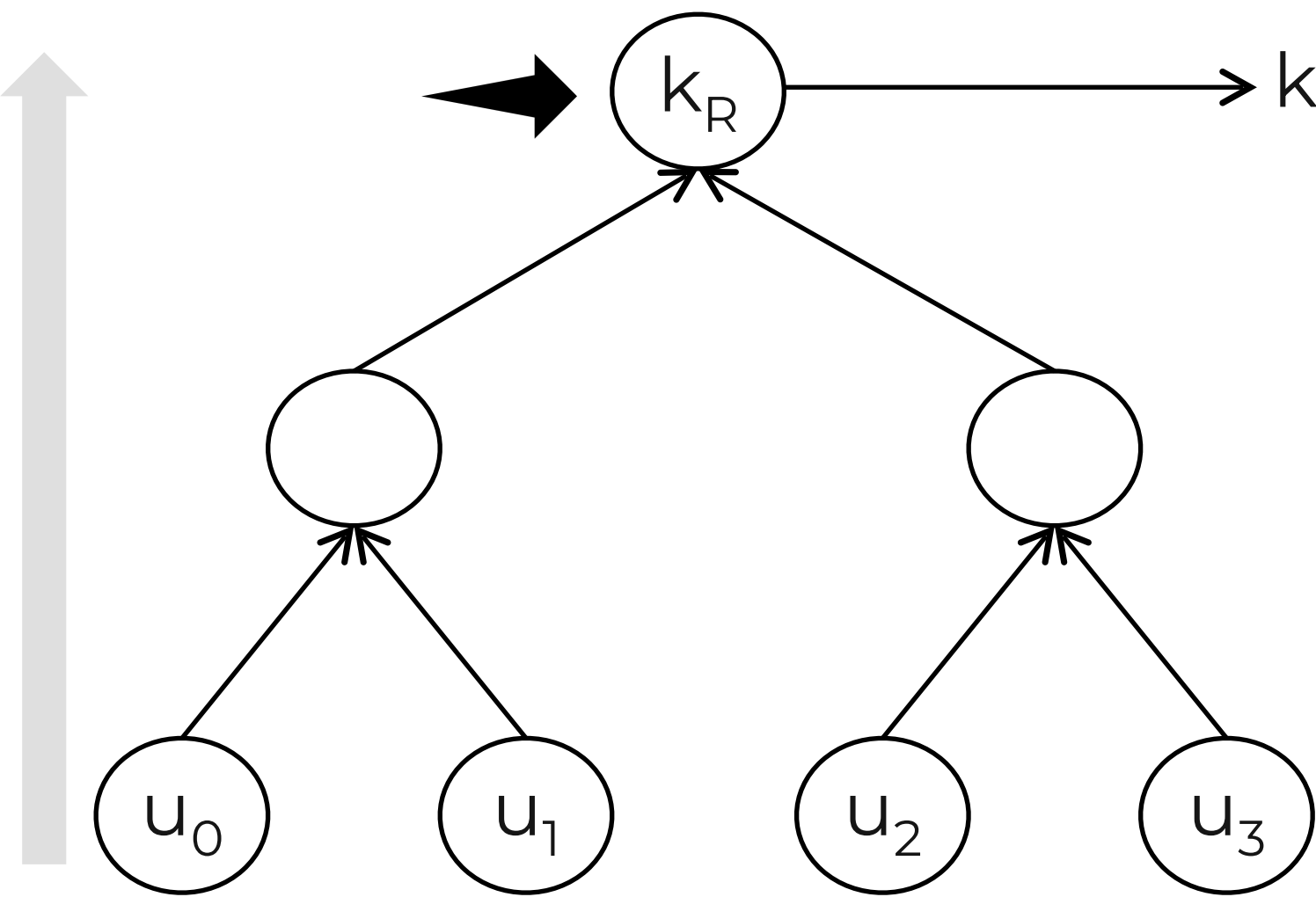


Secret tree

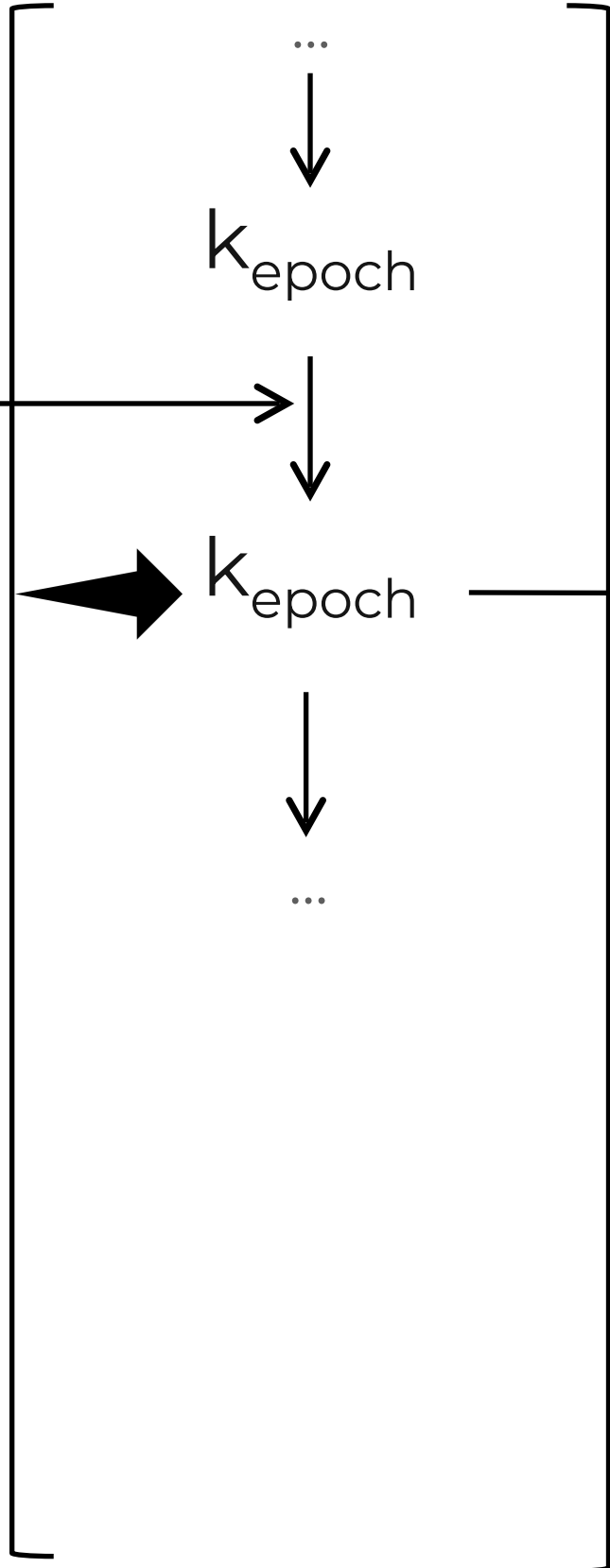


Encryption Key Derivation in MLS

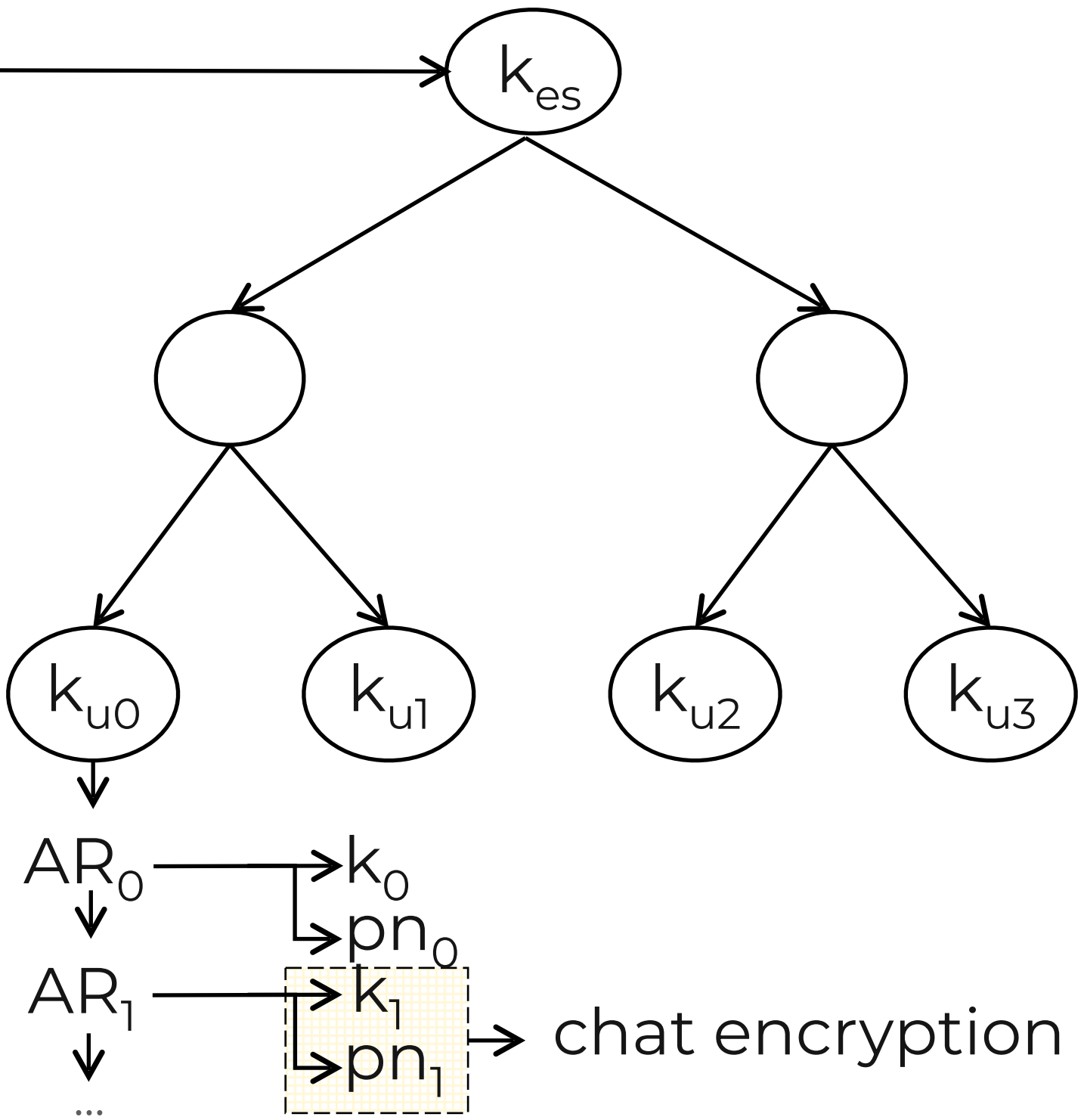
Ratchet tree



Key schedule

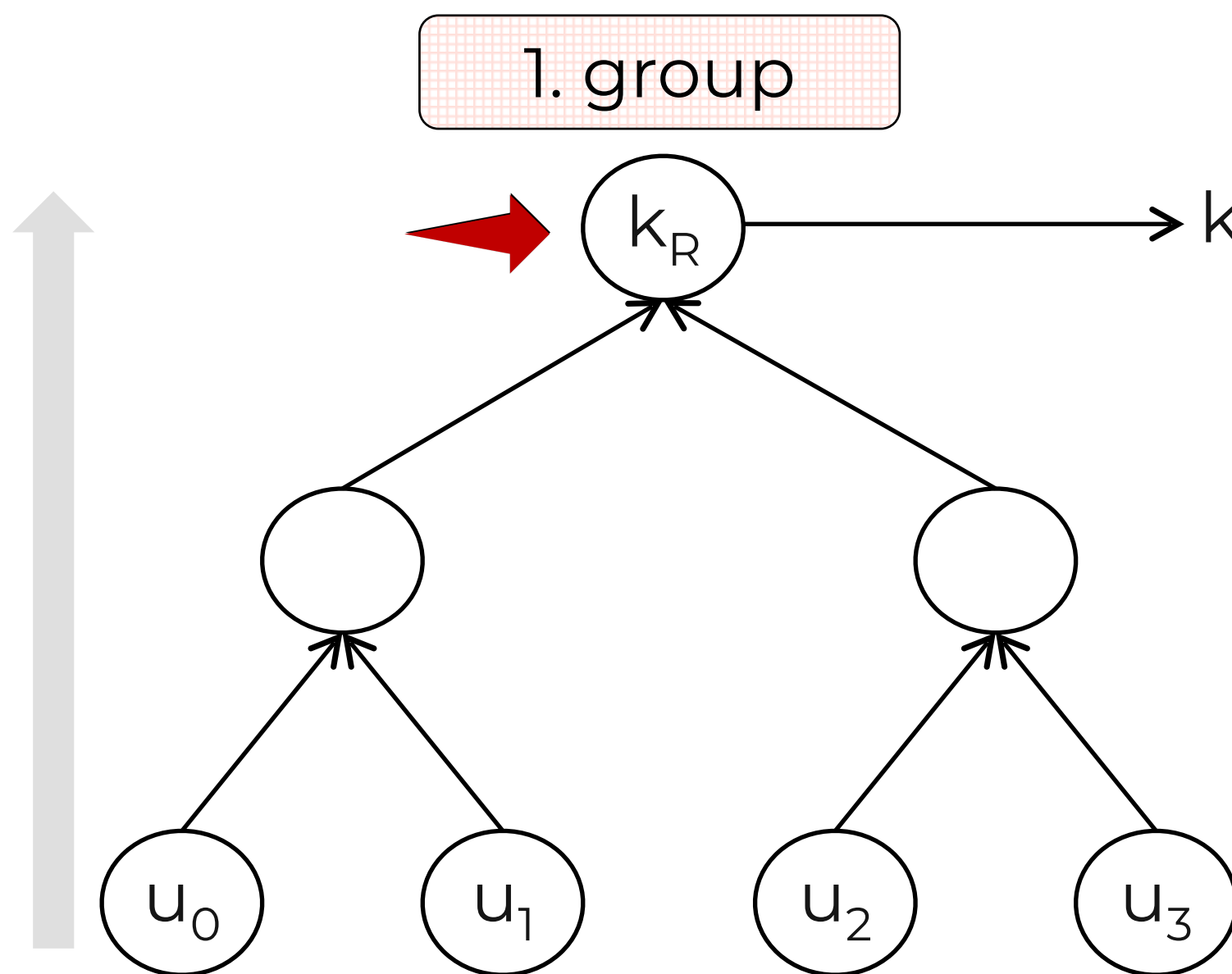


Secret tree

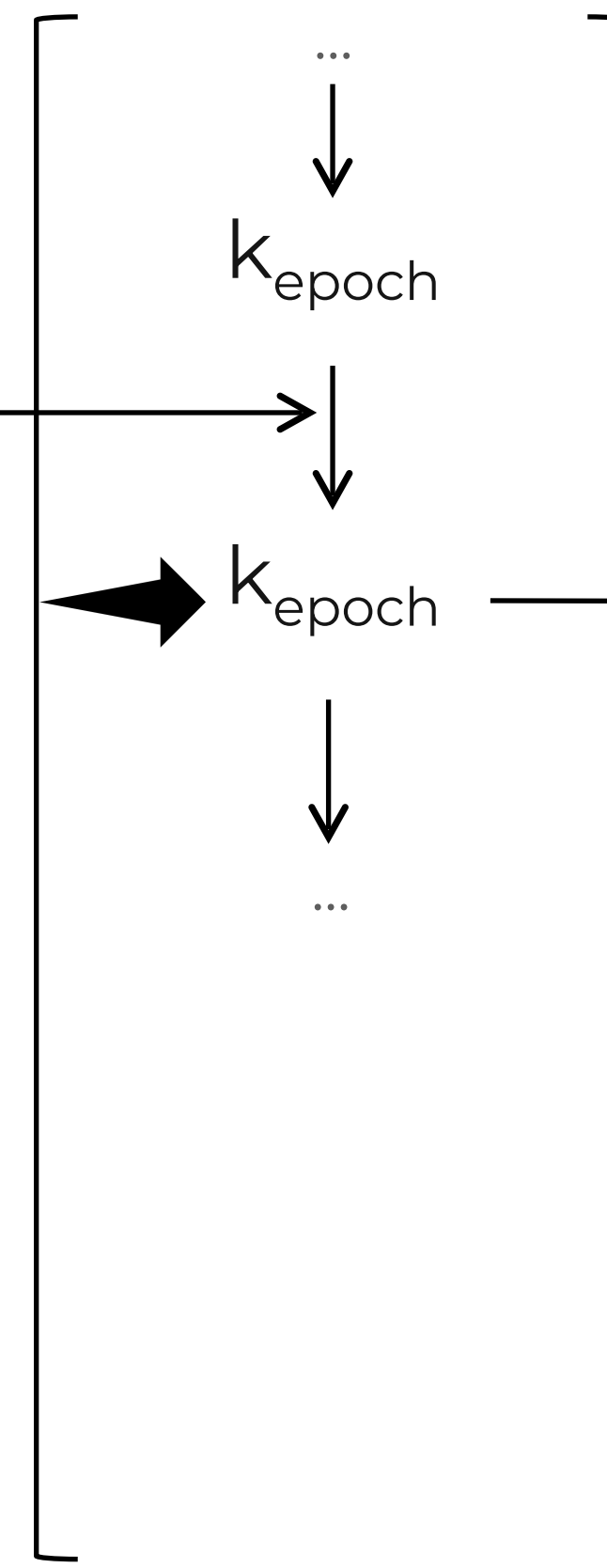


Encryption Key Derivation in MLS

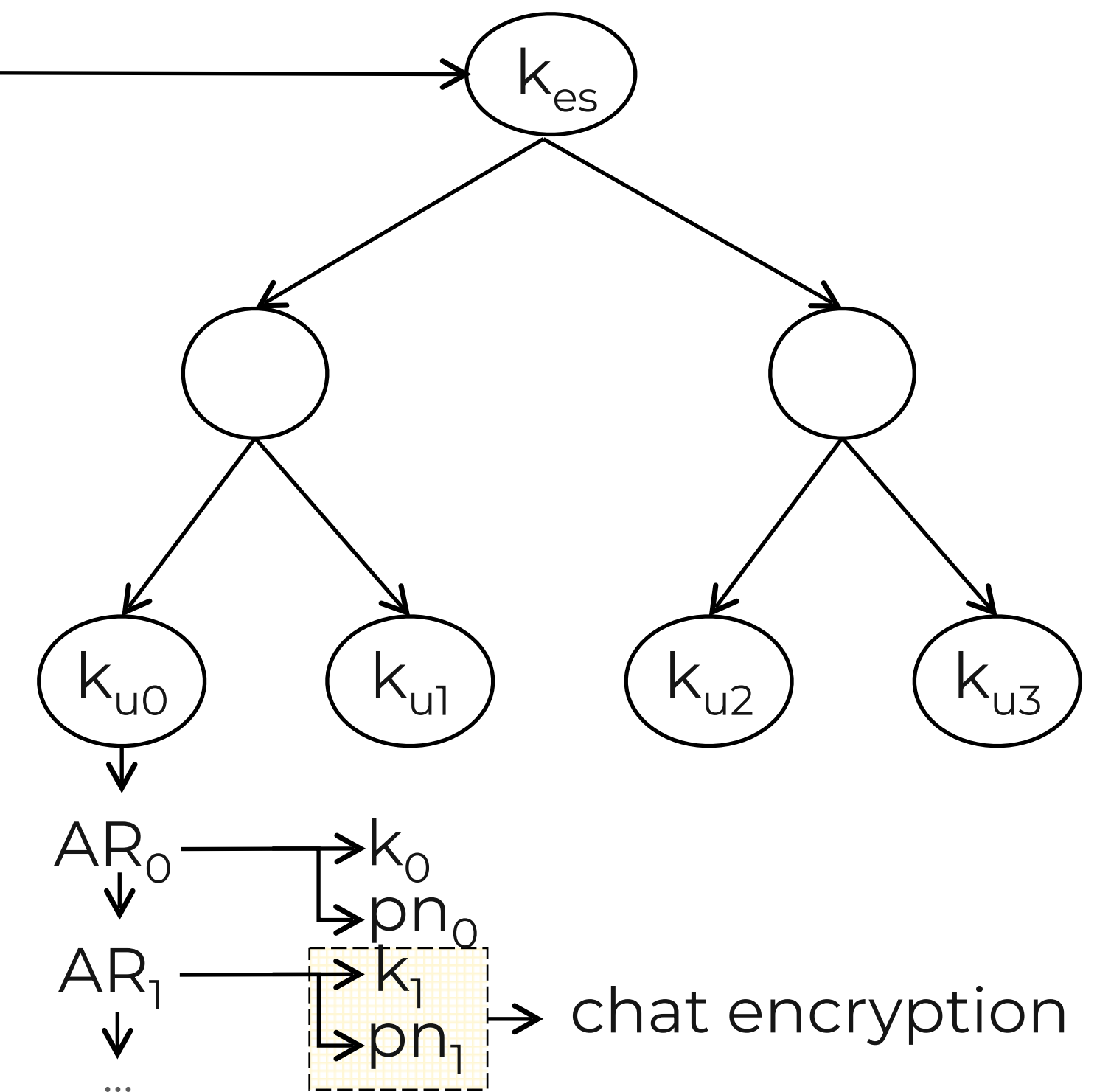
Ratchet tree



Key schedule

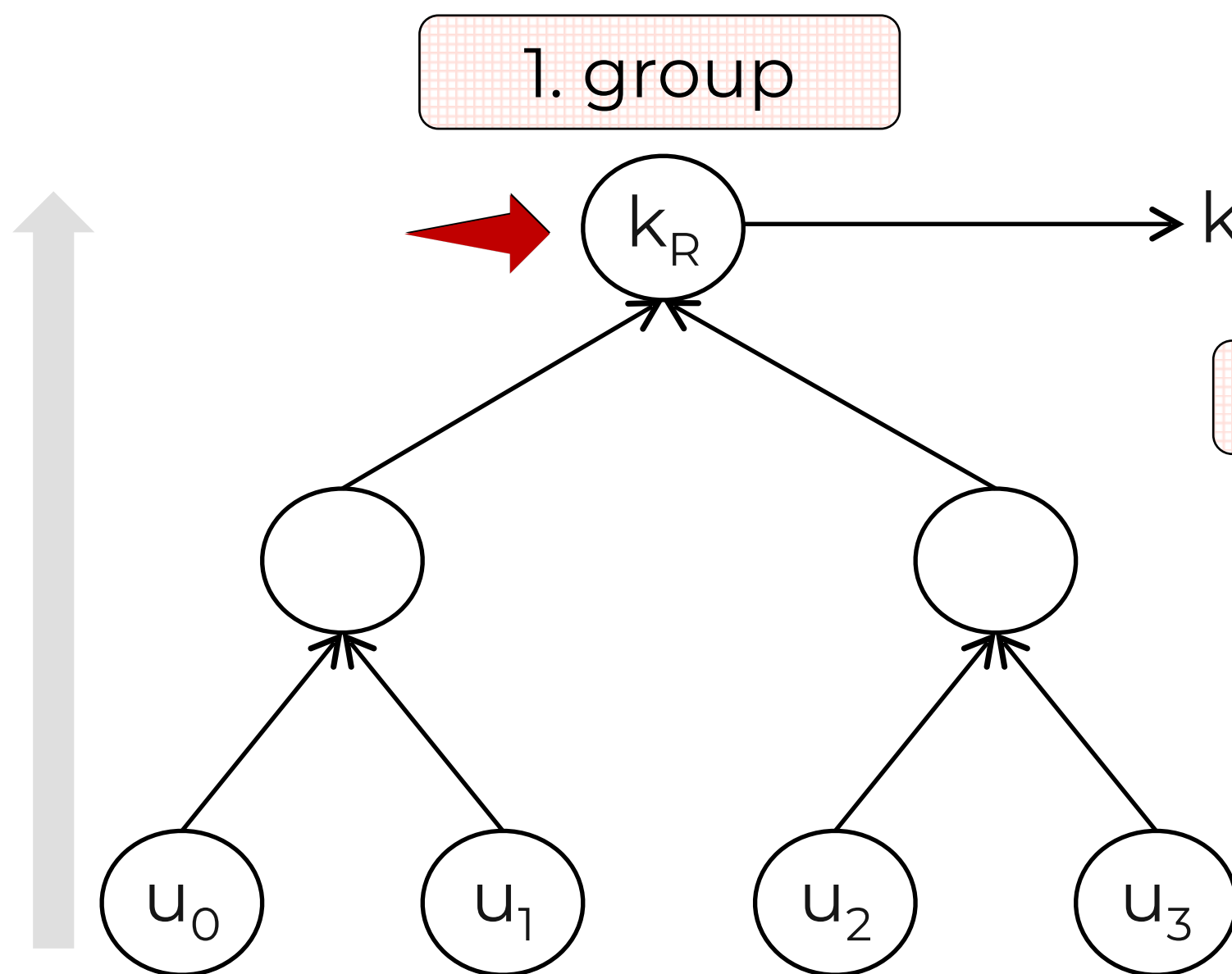


Secret tree

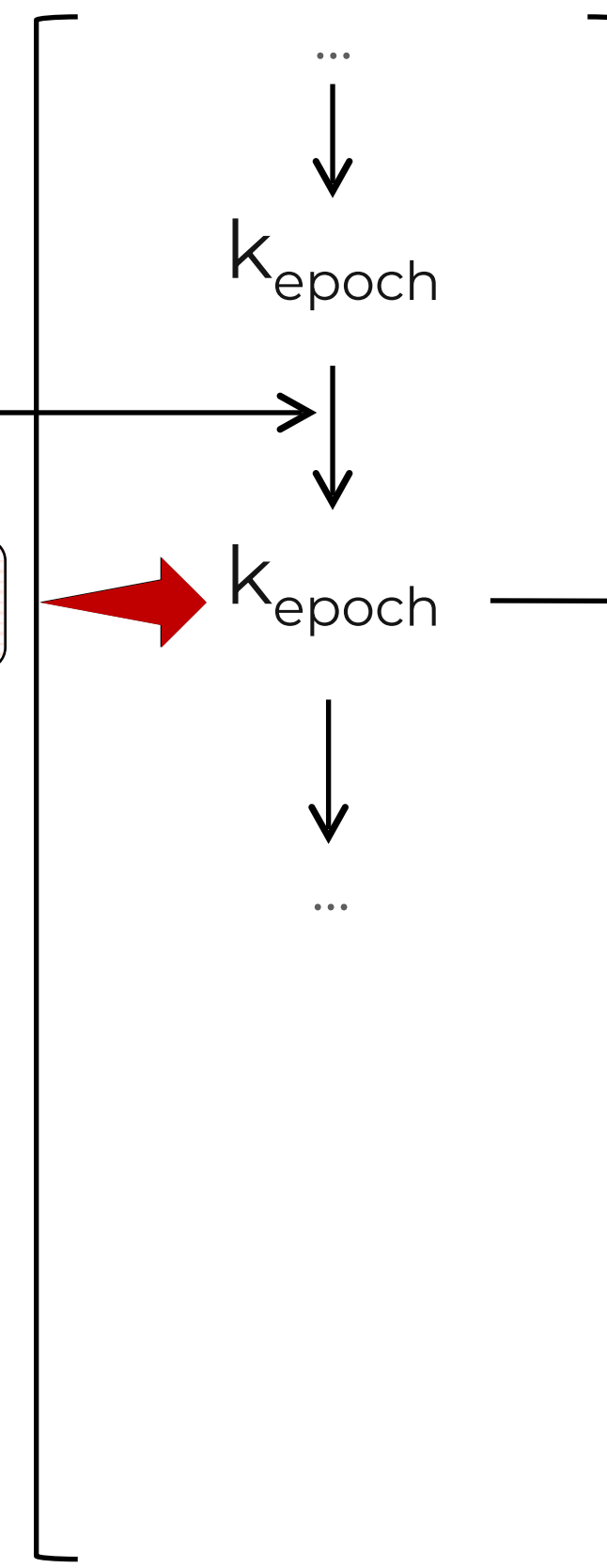


Encryption Key Derivation in MLS

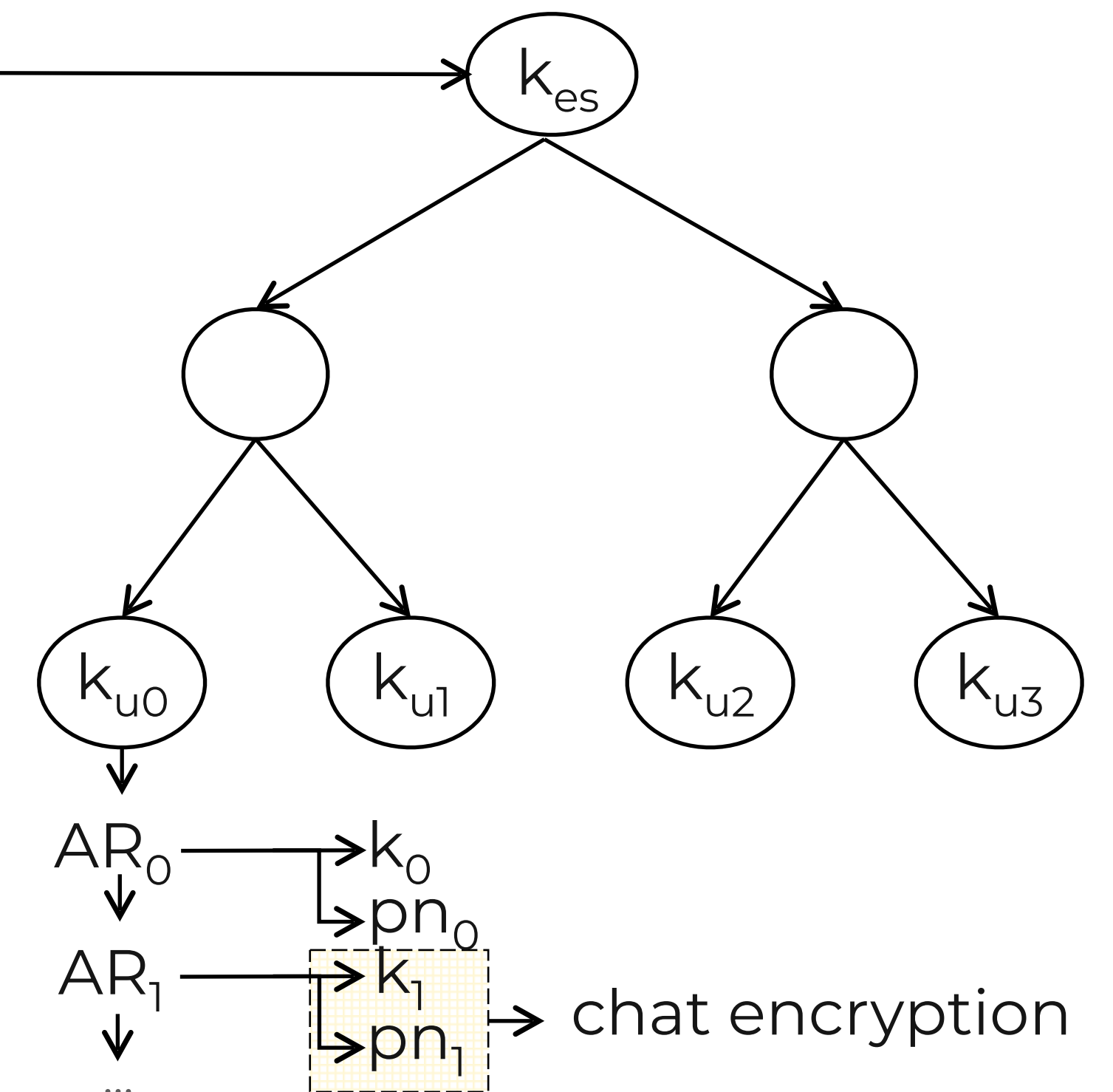
Ratchet tree



Key schedule

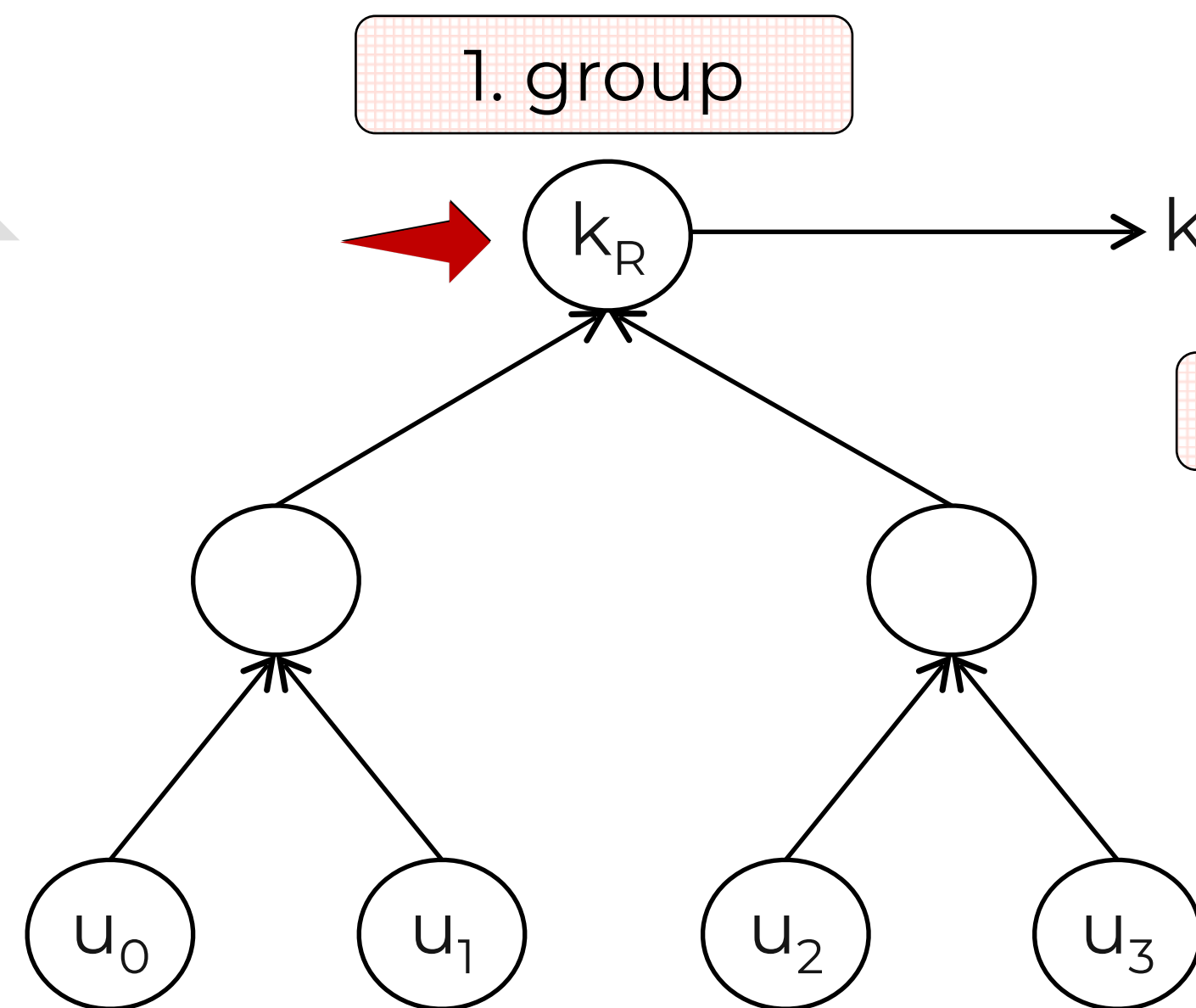


Secret tree

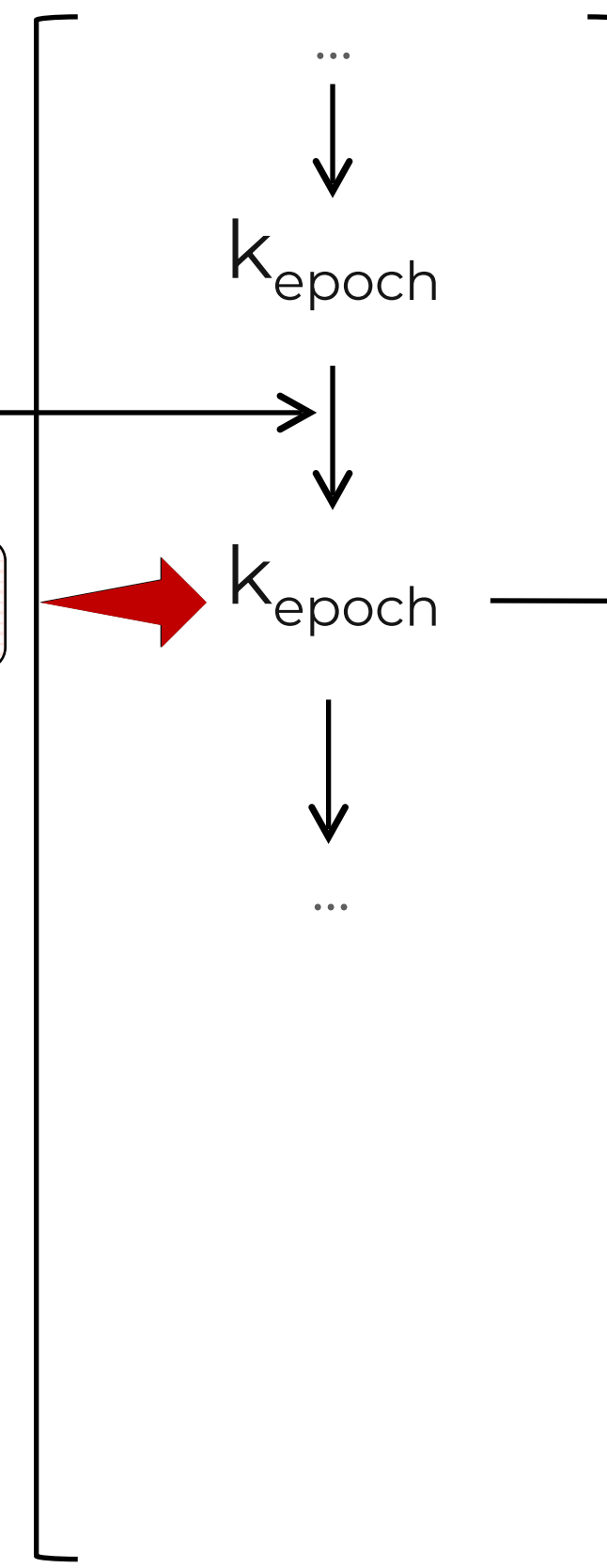


Encryption Key Derivation in MLS

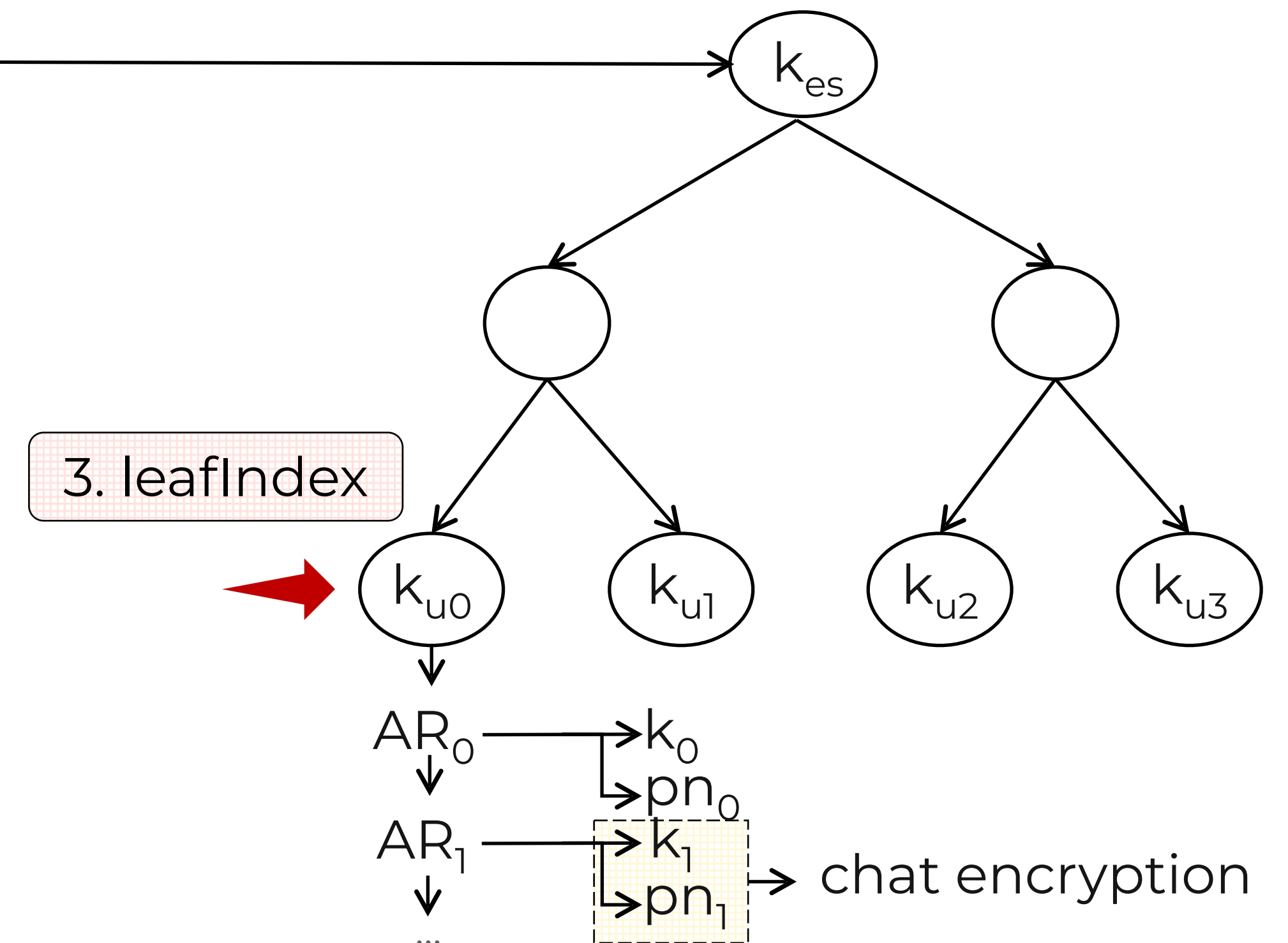
Ratchet tree



Key schedule

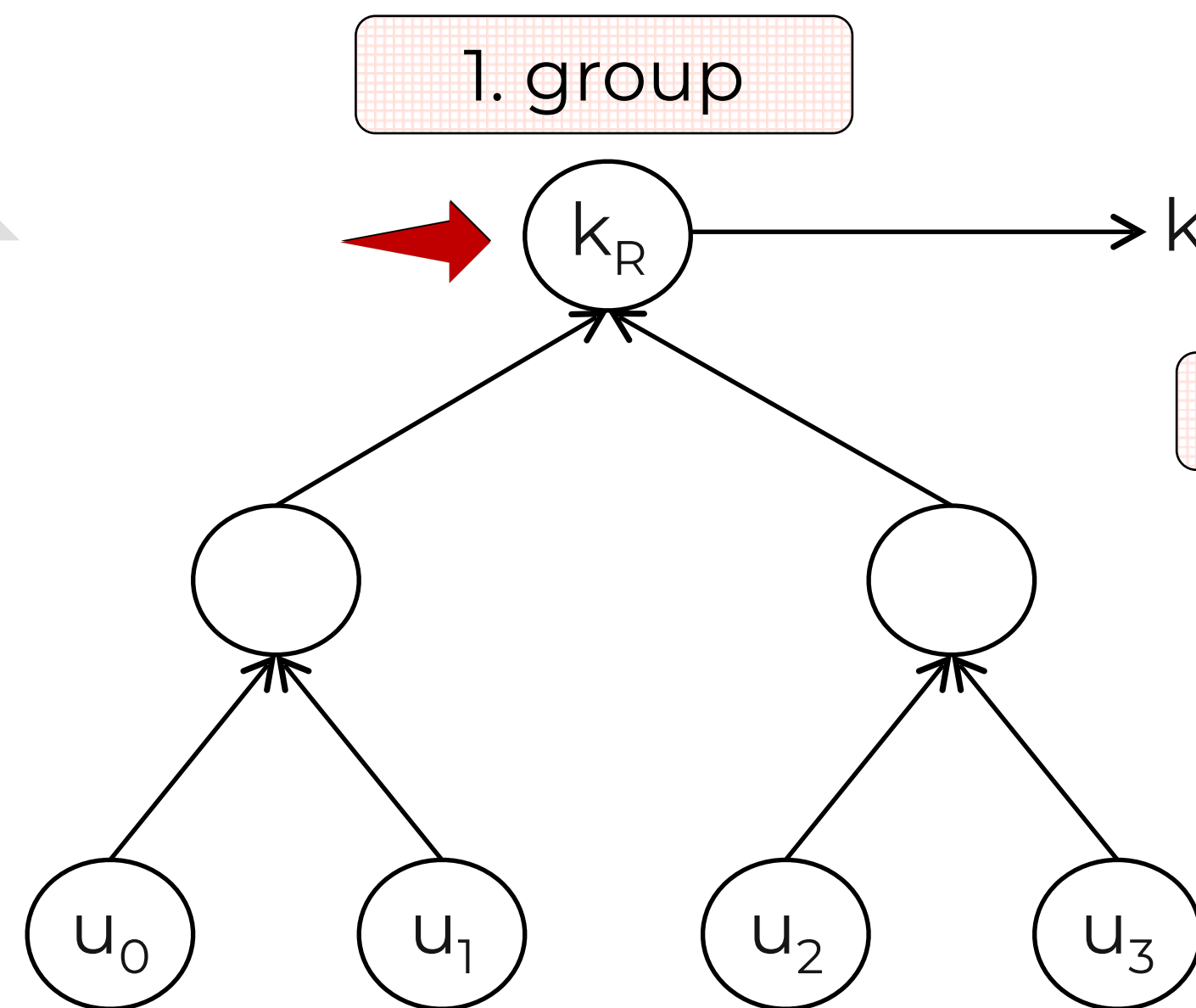


Secret tree

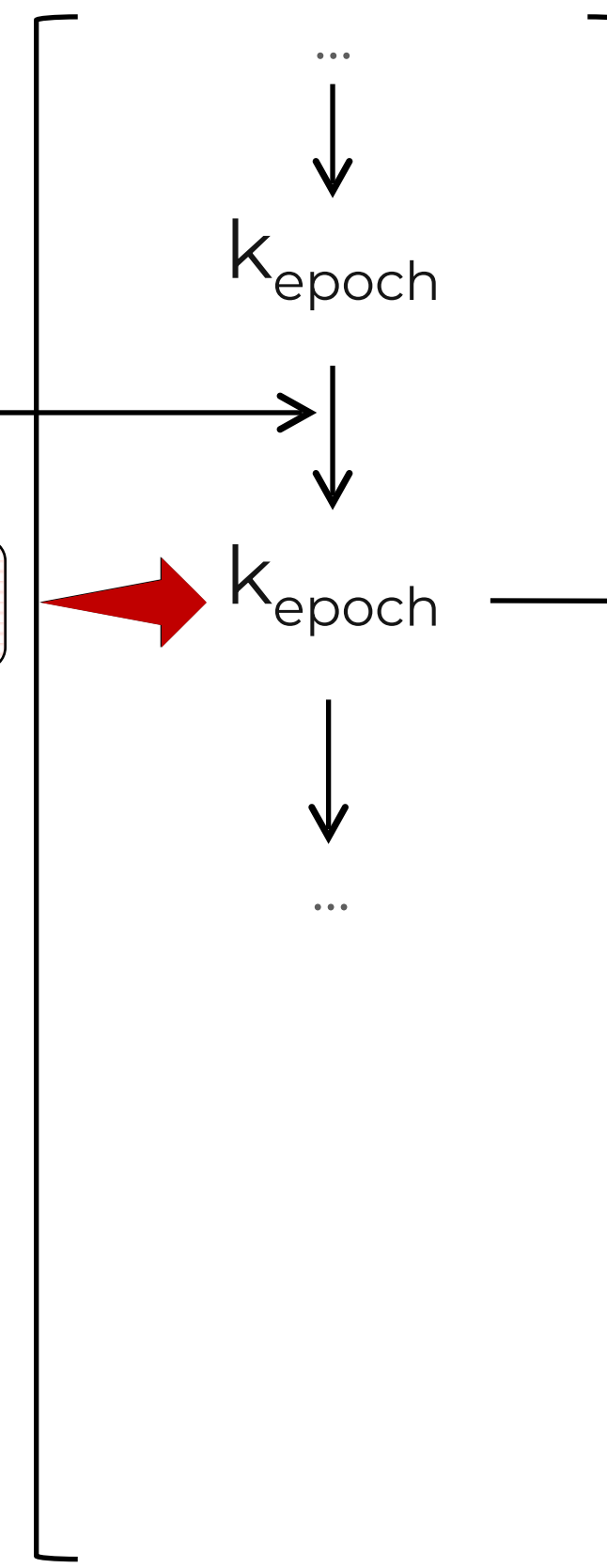


Encryption Key Derivation in MLS

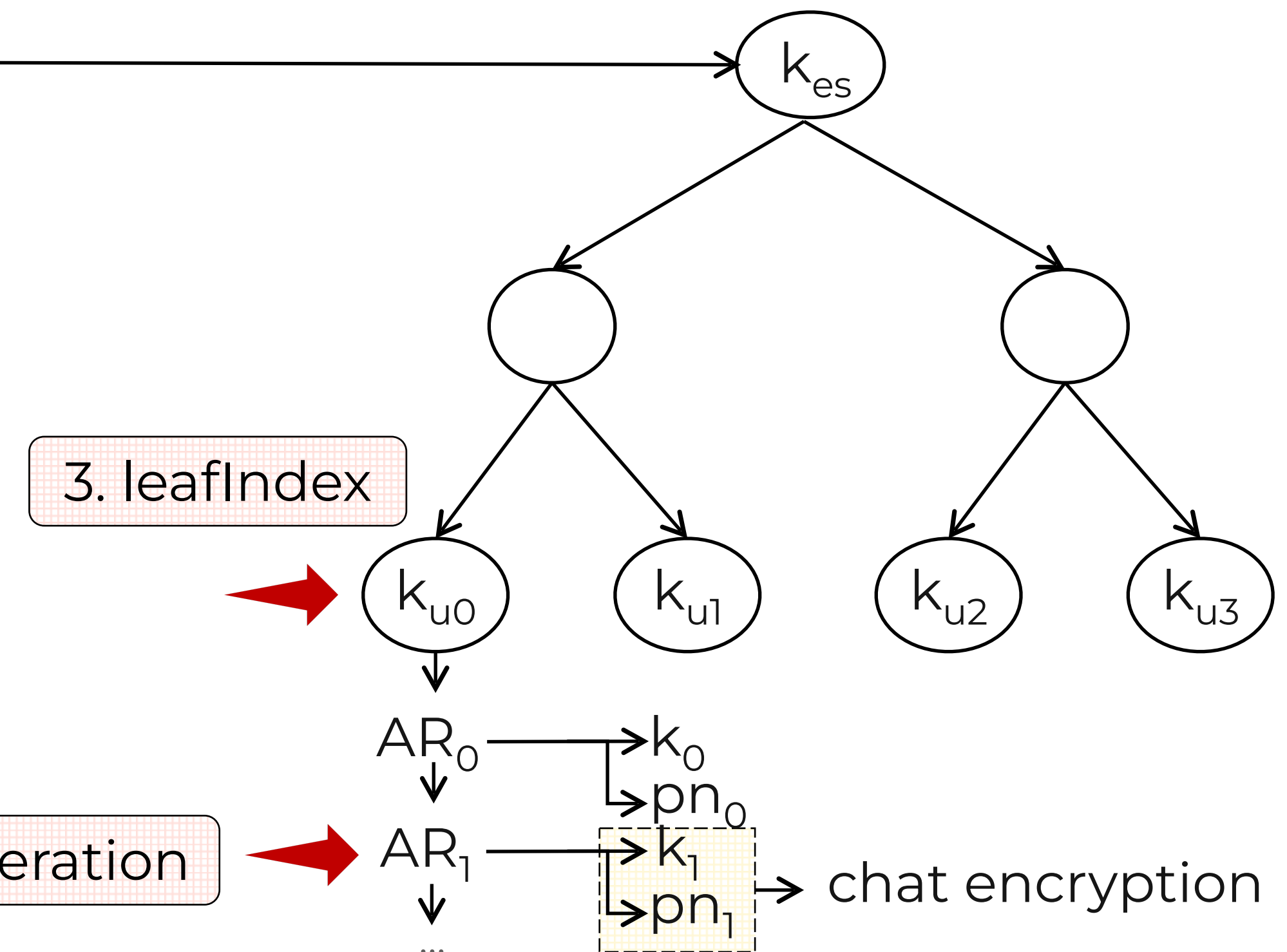
Ratchet tree



Key schedule



Secret tree



Encryption Key Derivation in MLS

Ratchet tree

Key schedule

Secret tree

1. group

k_R

k_{commit}

k_{epoch}

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

u_0

u_1

u_2

u_3

3. leafIndex

k_{u0}

k_{u1}

k_{u2}

k_{u3}

AR_0

k_0

pn_0

AR_1

k_1

pn_1

4. generation

chat encryption

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

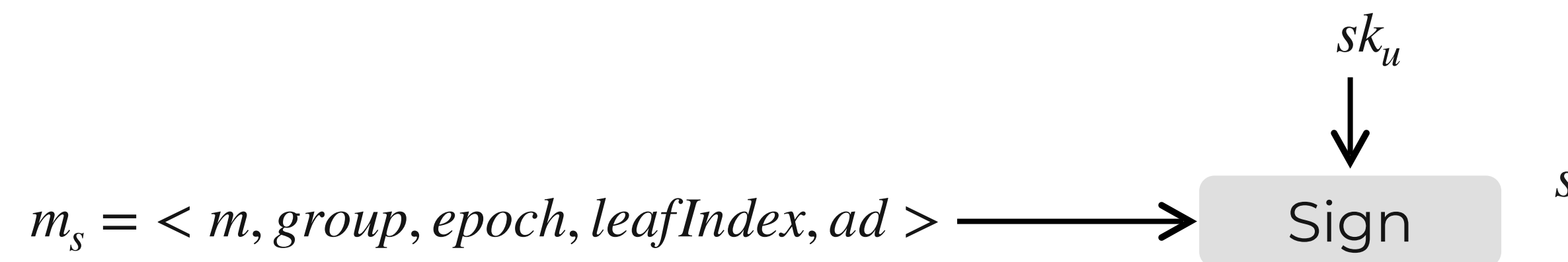
MLS-Sign-then-Encrypt

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

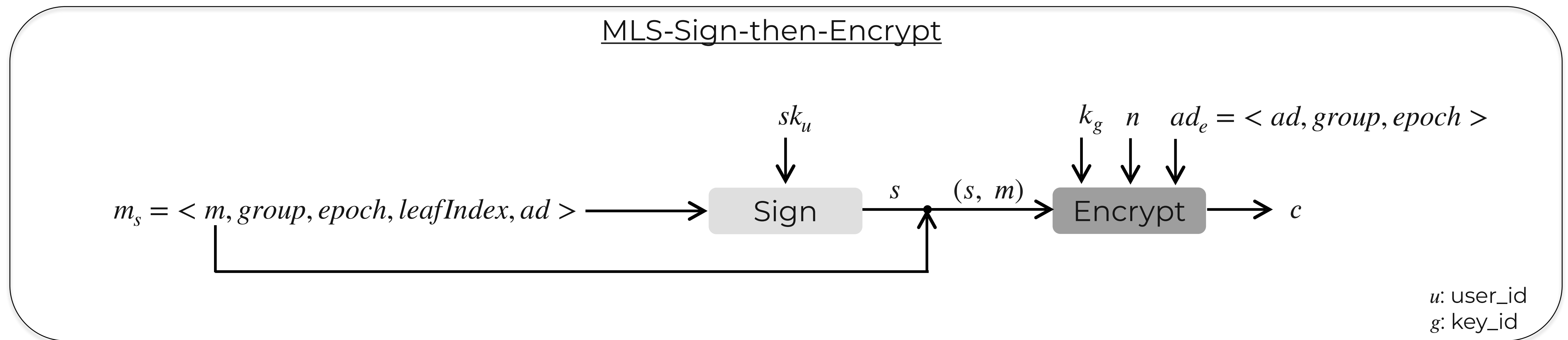
MLS-Sign-then-Encrypt



u : user_id
 g : key_id

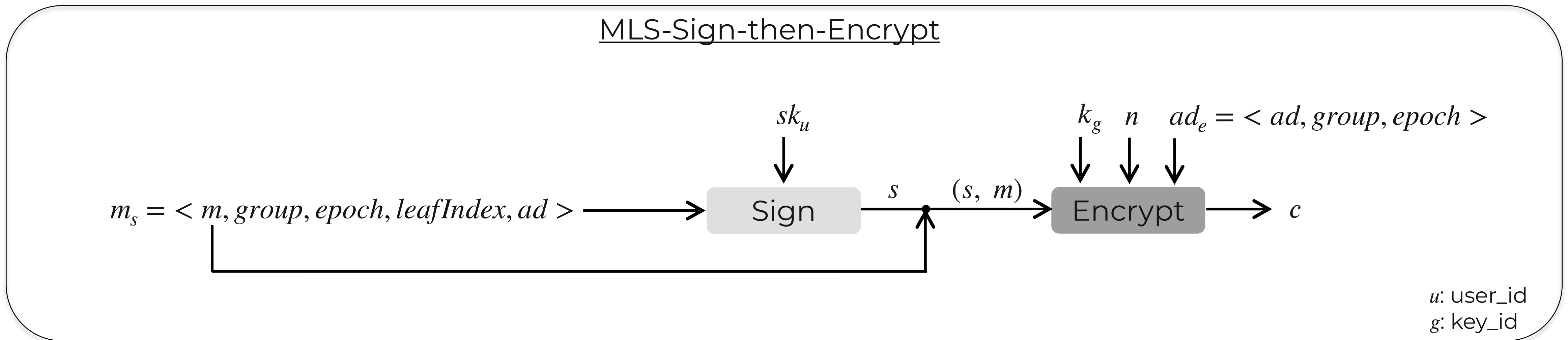
Chat Encryption in MLS

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

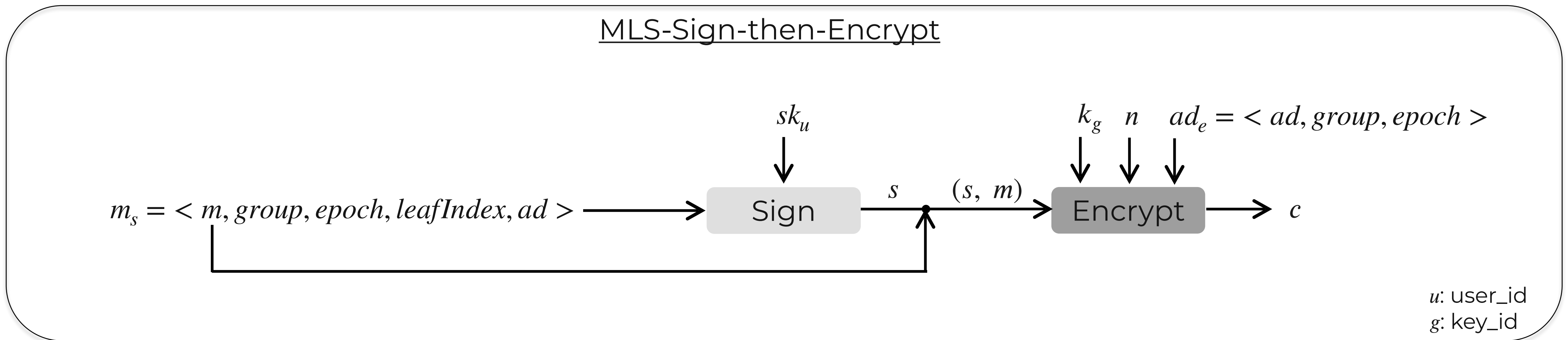
Chat encryption in MLS composes a digital signature scheme and a nonce-based encryption scheme in a Sign-then-Encrypt fashion



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

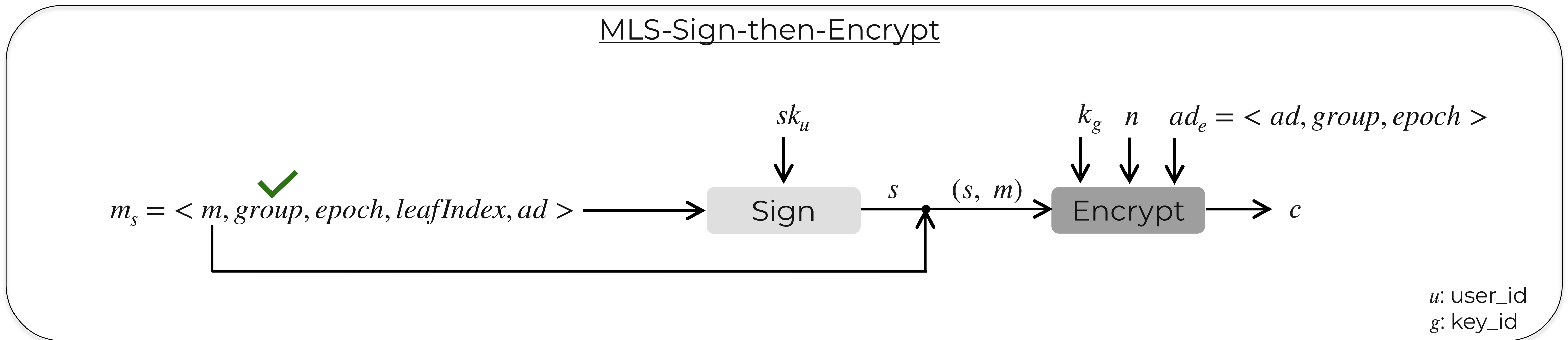


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt (s, m) using a different k and replay message to group

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

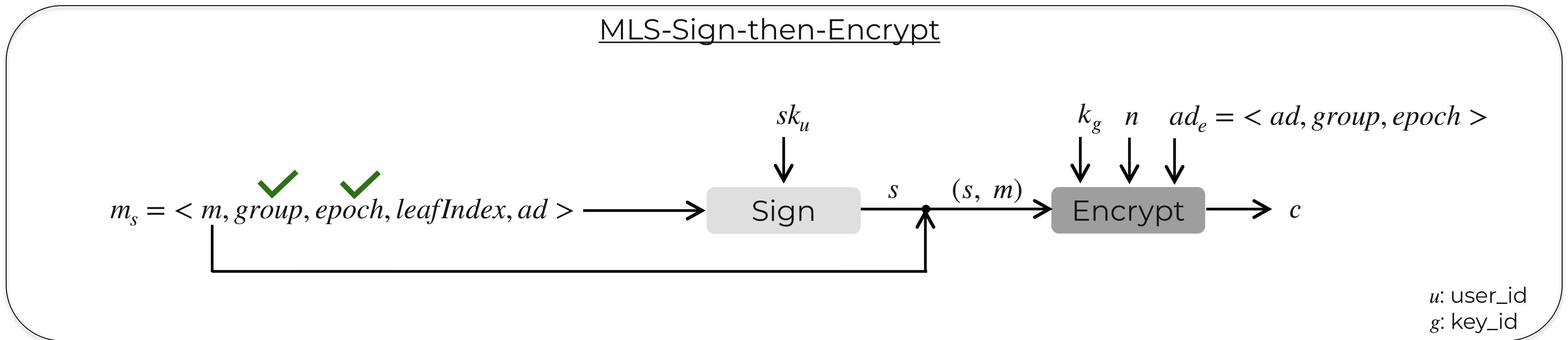


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt (s, m) using a different k and replay message to group

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

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

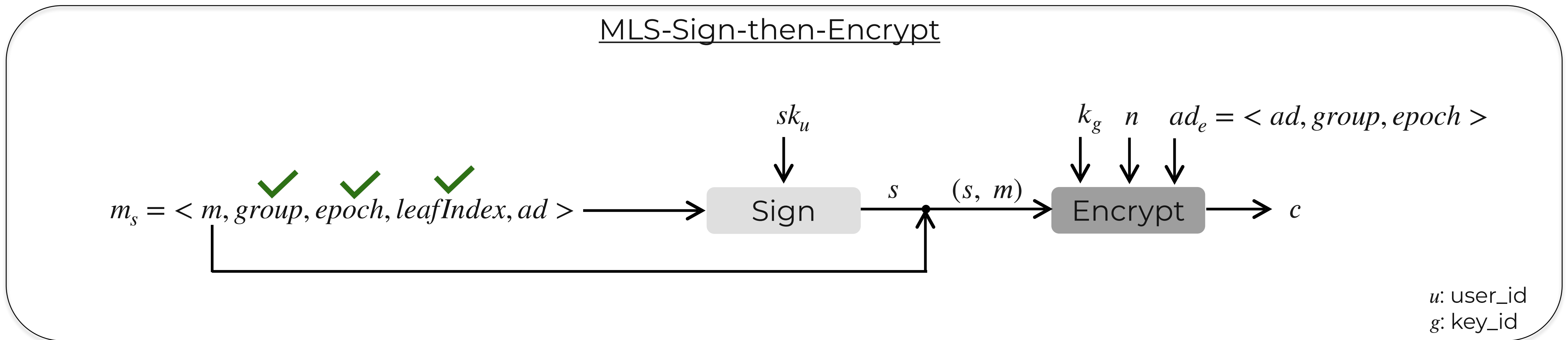


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt (s, m) using a different k and replay message to group

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

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

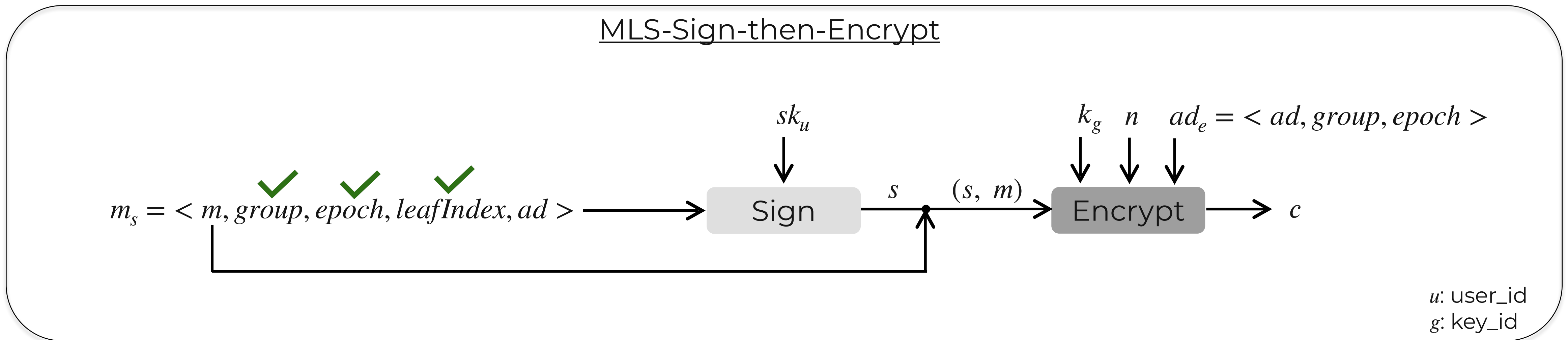


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt (s, m) using a different k and replay message to group

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

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

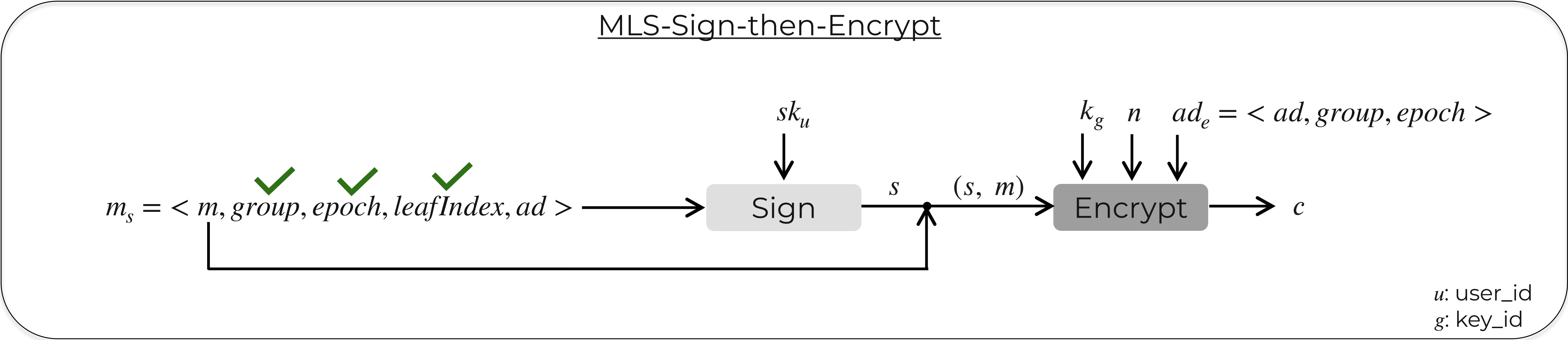


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt (s, m) using a different k and replay message to group

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

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

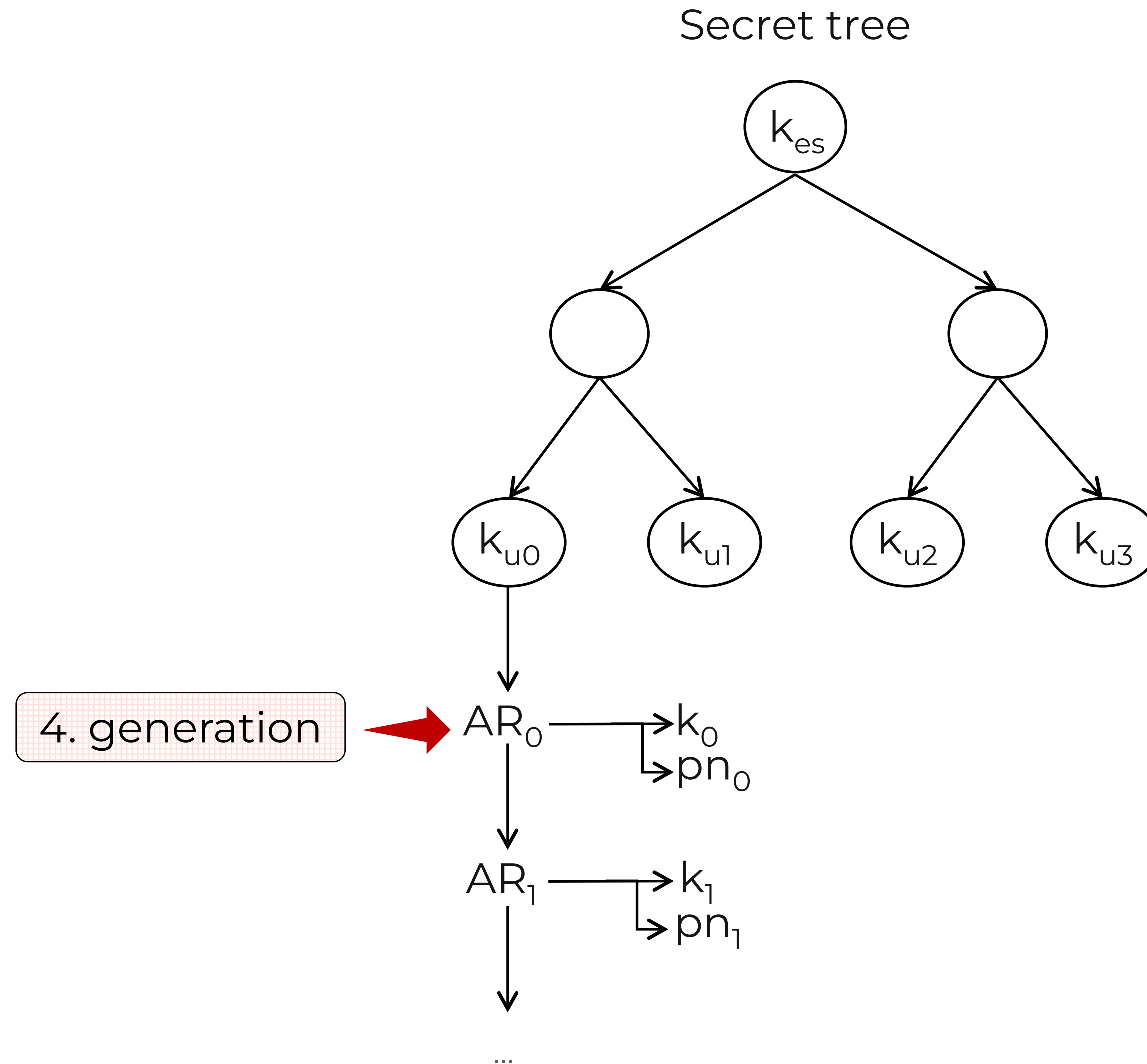


Intuition: s should authenticate the key identifier so that group insider cannot re-encrypt (s, m) using a different k and replay message to group

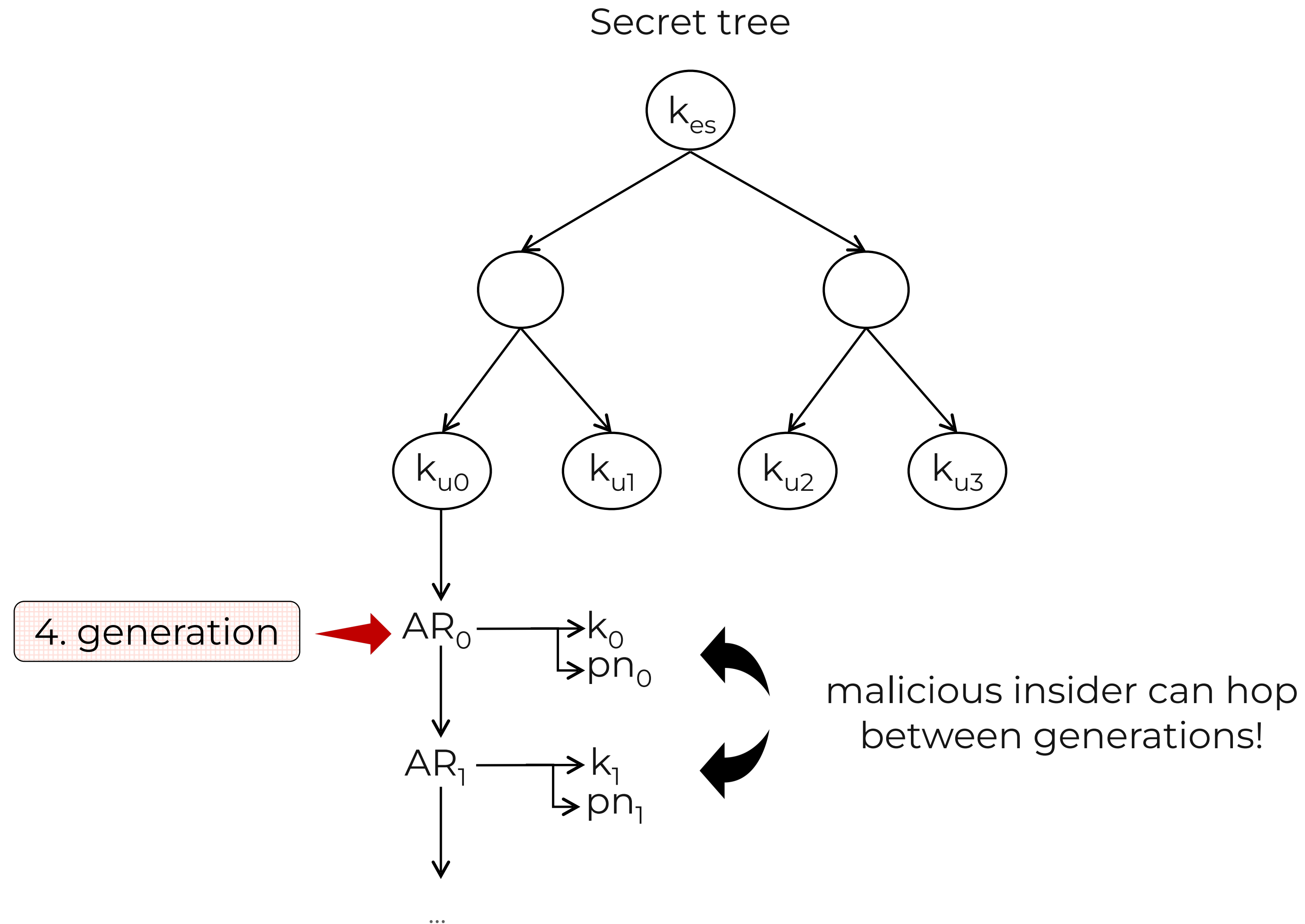
Recall: key identifier $g = (\text{group}, \text{epoch}, \text{leafIndex}, \text{generation})$

SIGN
group_key_id ✗

Encryption Key Derivation in MLS



Encryption Key Derivation in MLS



1. Insider Replay Attack

SIGN
group_key_id X

Insider Replay Attack



Insider Replay Attack

Insider Replay Attack



Insider Replay Attack



m ← Yes!



Insider Replay Attack



$m \leftarrow \text{Yes!}$

MLS-Sign-then-Encrypt

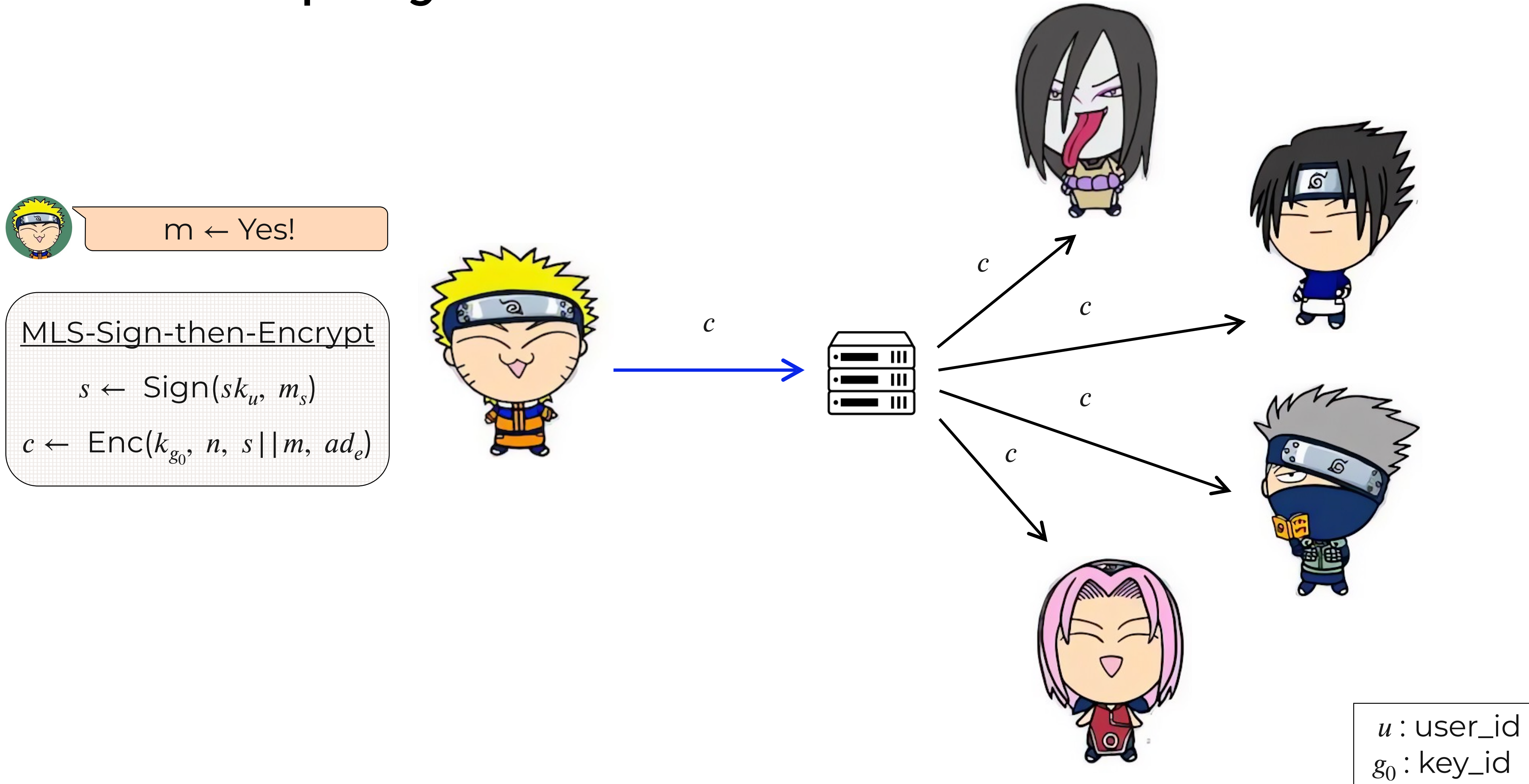
$s \leftarrow \text{Sign}(sk_u, m_s)$

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



$u : \text{user_id}$
 $g_0 : \text{key_id}$

Insider Replay Attack



Insider Replay Attack



$m \leftarrow \text{Yes!}$

MLS-Sign-then-Encrypt

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

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



c



c



c



c



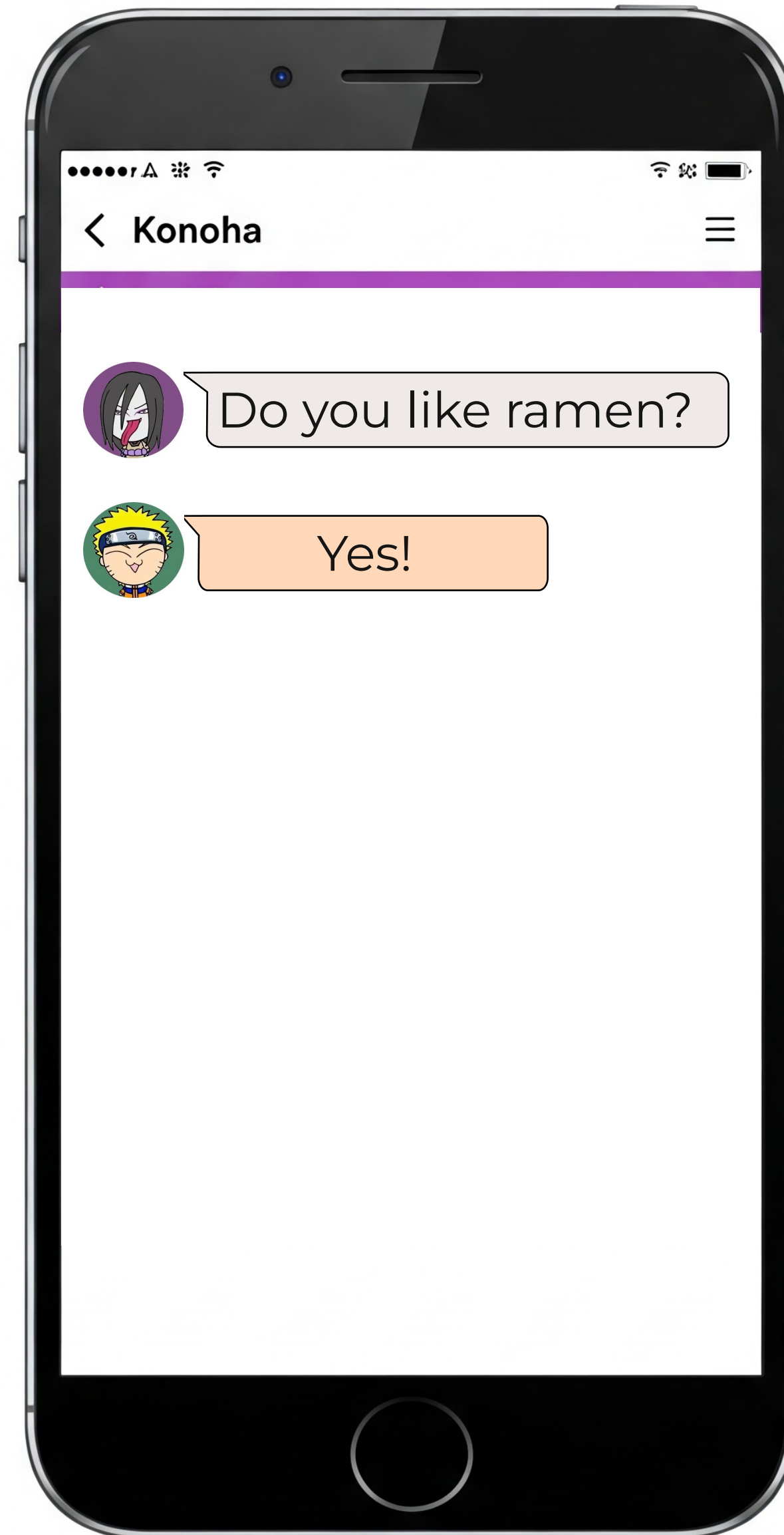
c



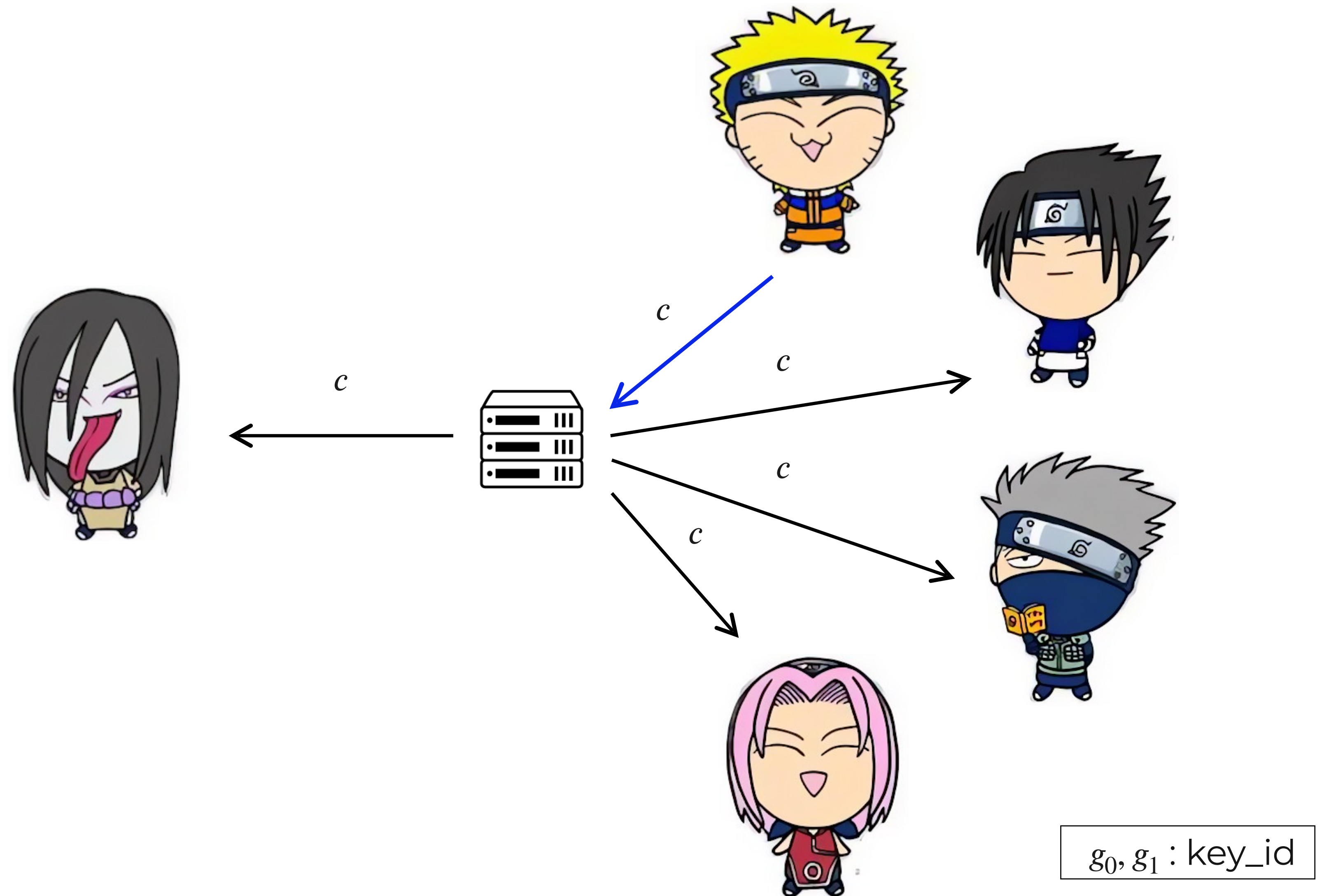
Let key identifier $g_0 = (group, epoch, leafIndex, generation_0)$

$u : \text{user_id}$
 $g_0 : \text{key_id}$

Insider Replay Attack

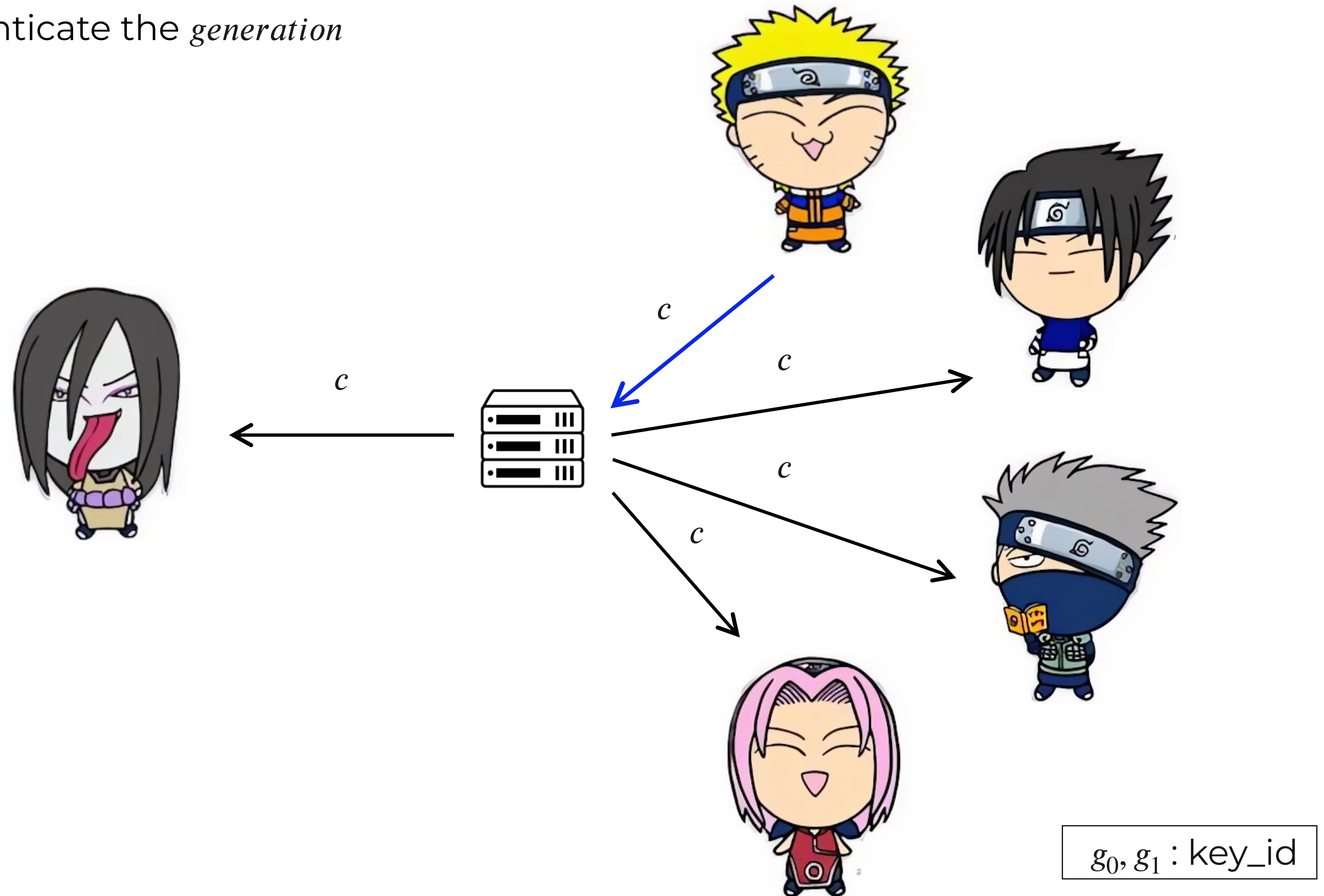


Insider Replay Attack



Insider Replay Attack

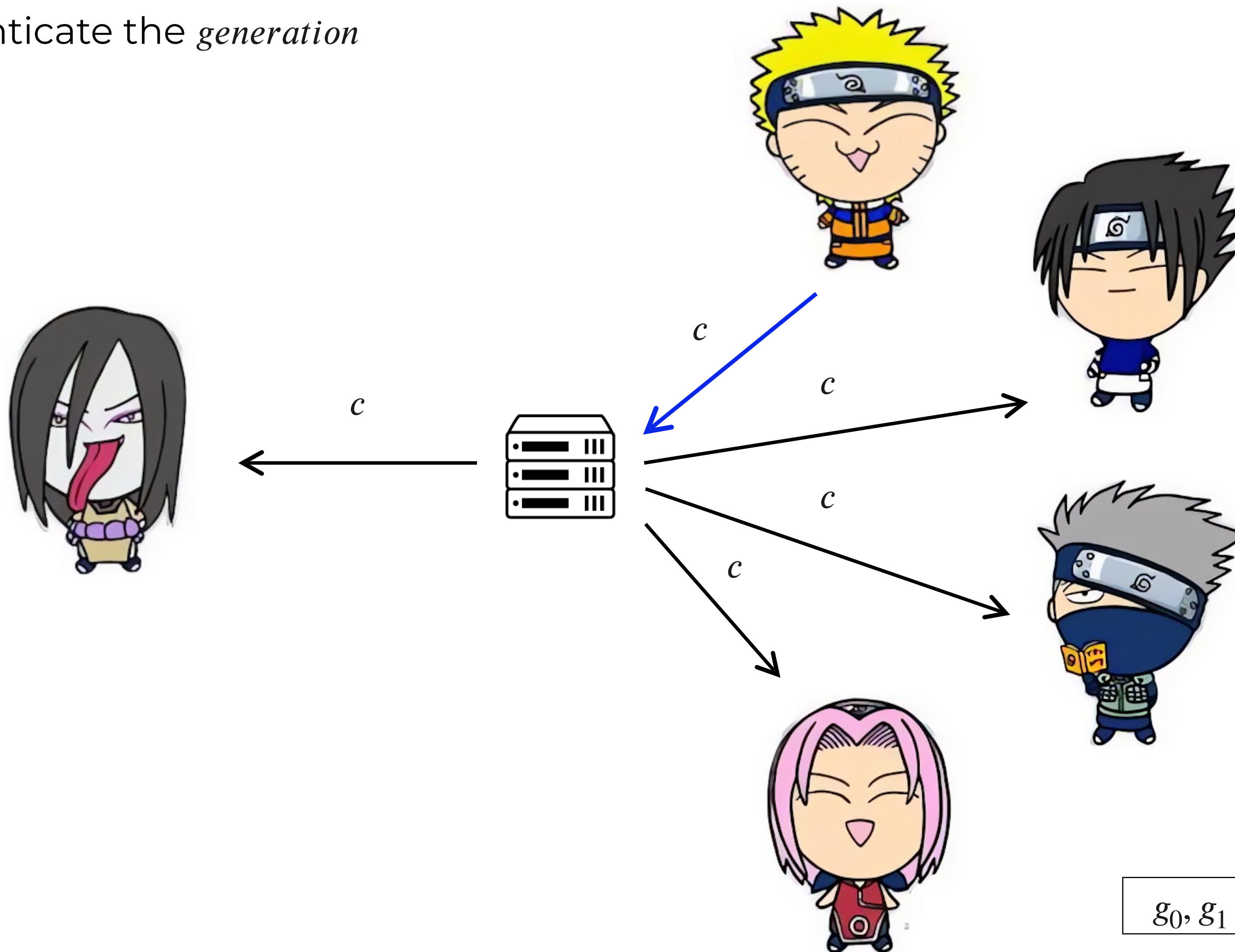
Recall: signature s does not authenticate the *generation*



Insider Replay Attack

Recall: signature s does not authenticate the *generation*

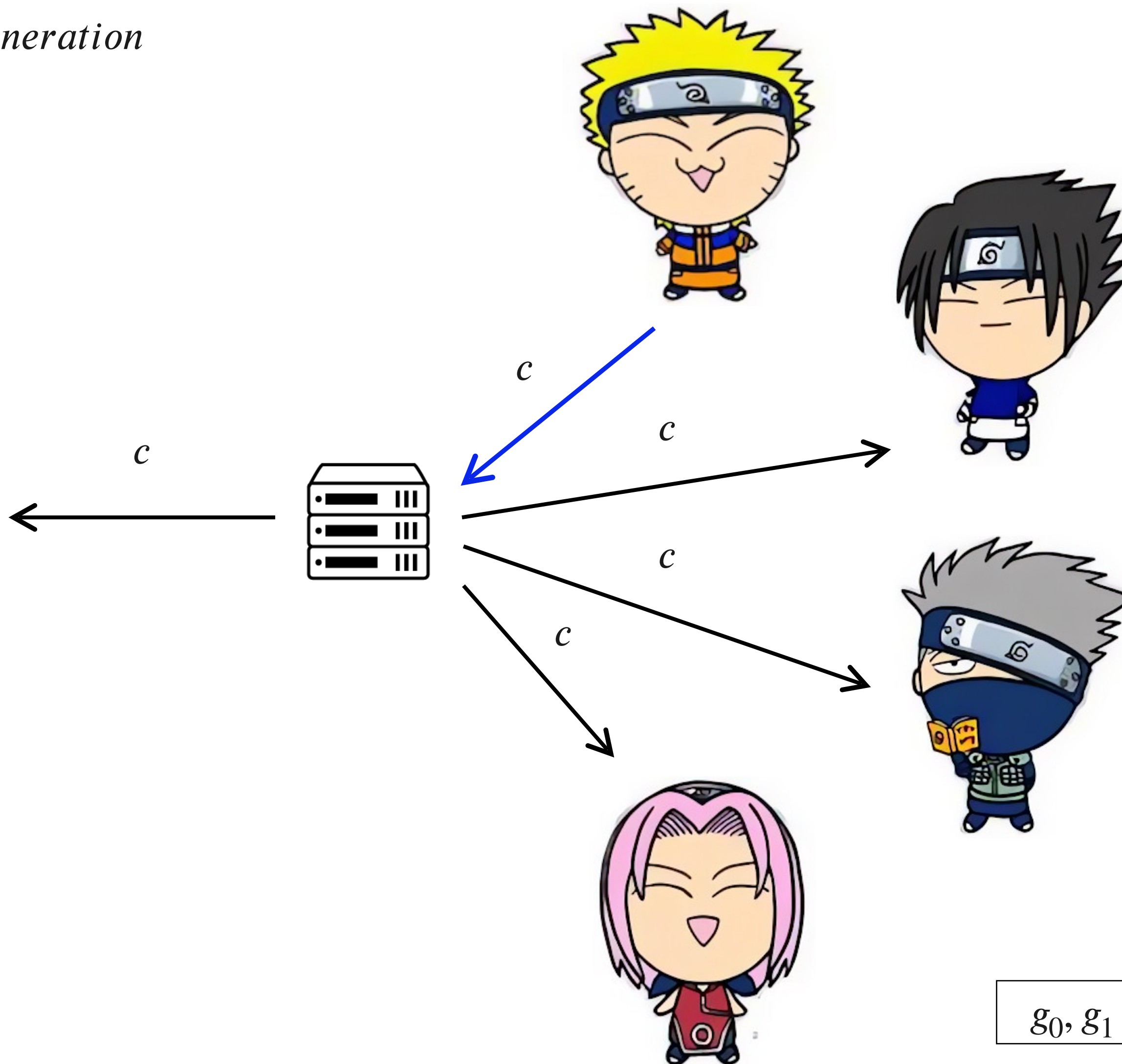
Decrypt-then-Re-Encrypt
 $s || m \leftarrow \text{Enc}(k_{g_0}, n, c, ad_e)$
 $c' \leftarrow \text{Dec}(k_{g_1}, n, s || m, ad_e)$



Insider Replay Attack

Recall: signature s does not authenticate the *generation*

Decrypt-then-Re-Encrypt

$$s || m \leftarrow \text{Enc}(k_{g_0}, n, c, ad_e)$$
$$c' \leftarrow \text{Dec}(k_{g_1}, n, s || m, ad_e)$$
$$g_0 = (\text{group}, \text{epoch}, \text{leafIndex}, \text{generation}_0)$$
$$g_1 = (\text{group}, \text{epoch}, \text{leafIndex}, \text{generation}_1)$$
$$(\text{generation}_1 > \text{generation}_0)$$


Insider Replay Attack

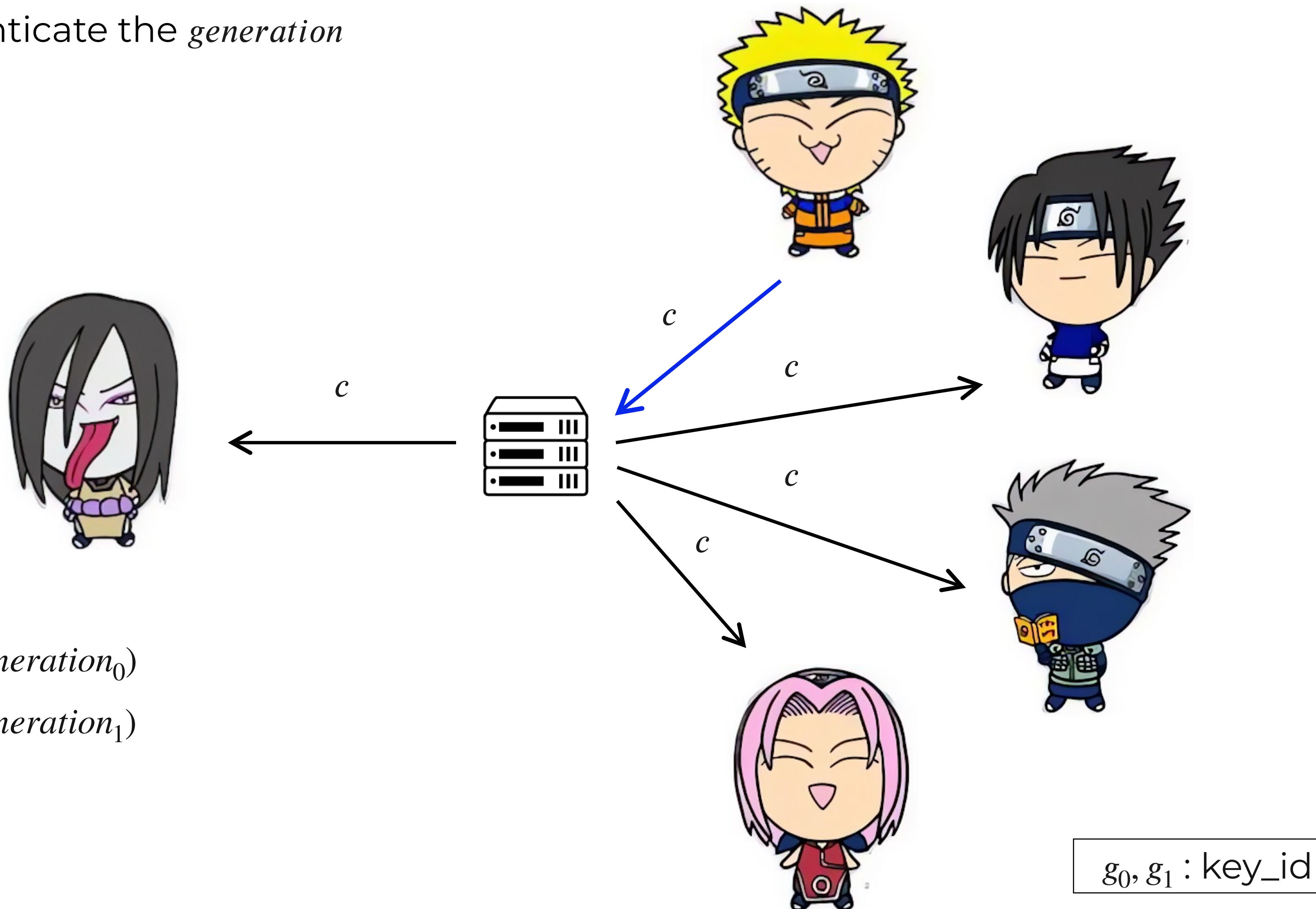
Recall: signature s does not authenticate the *generation*

Decrypt-then-Re-Encrypt
 $s || m \leftarrow \text{Enc}(k_{g_0}, n, c, ad_e)$
 $c' \leftarrow \text{Dec}(k_{g_1}, n, s || m, ad_e)$
Save for later

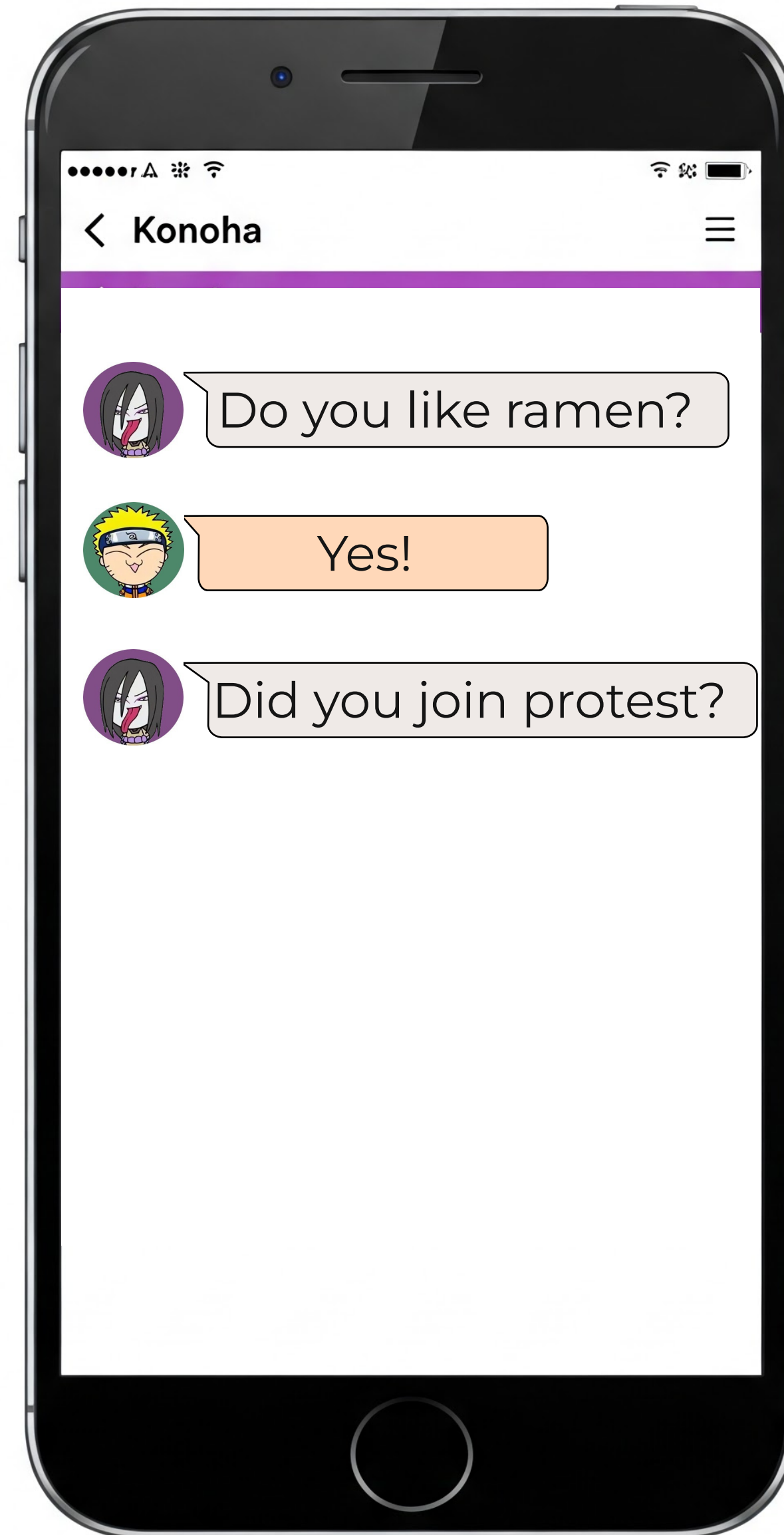
$g_0 = (\text{group}, \text{epoch}, \text{leafIndex}, \text{generation}_0)$

$g_1 = (\text{group}, \text{epoch}, \text{leafIndex}, \text{generation}_1)$

$(\text{generation}_1 > \text{generation}_0)$



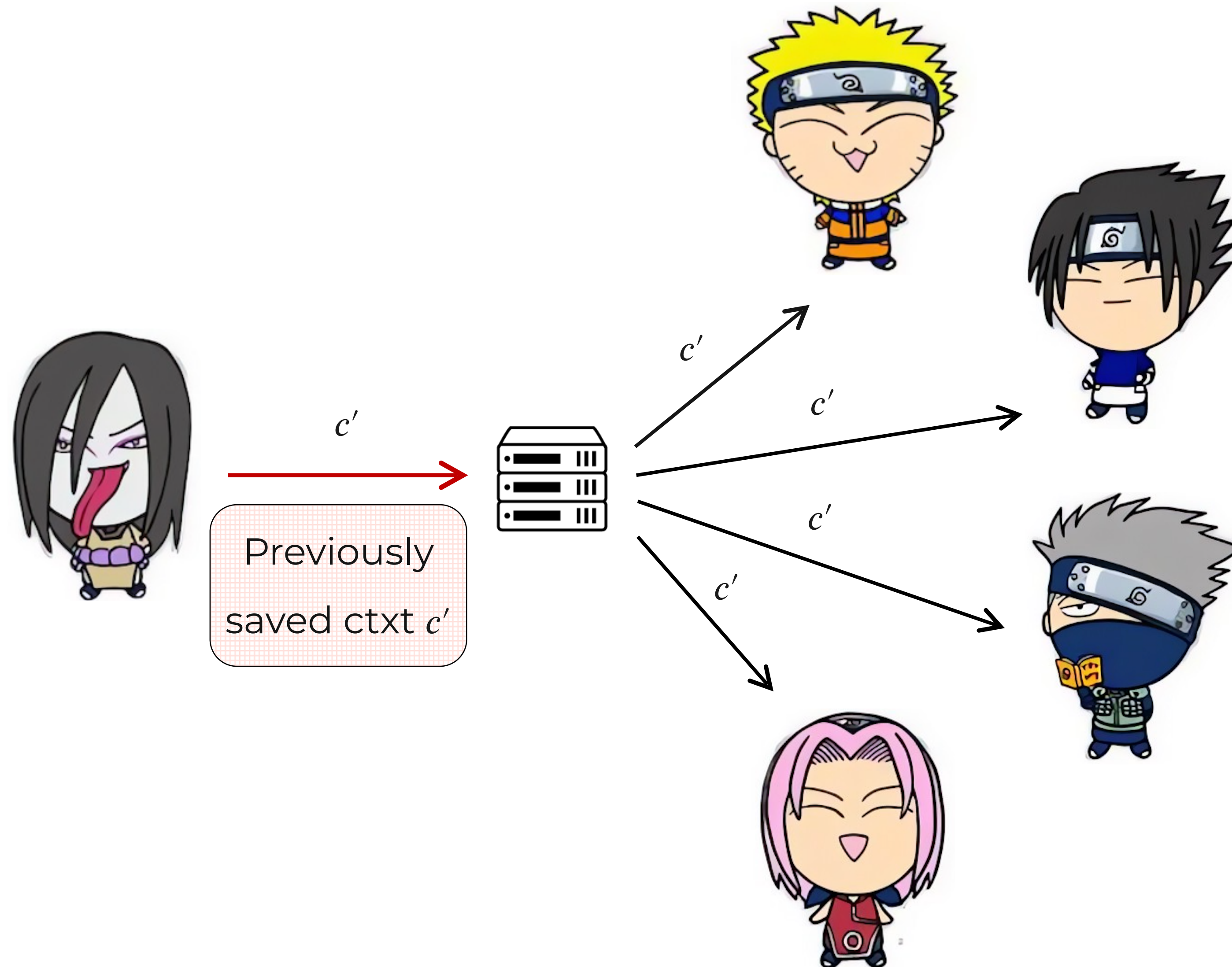
Insider Replay Attack



Insider Replay Attack



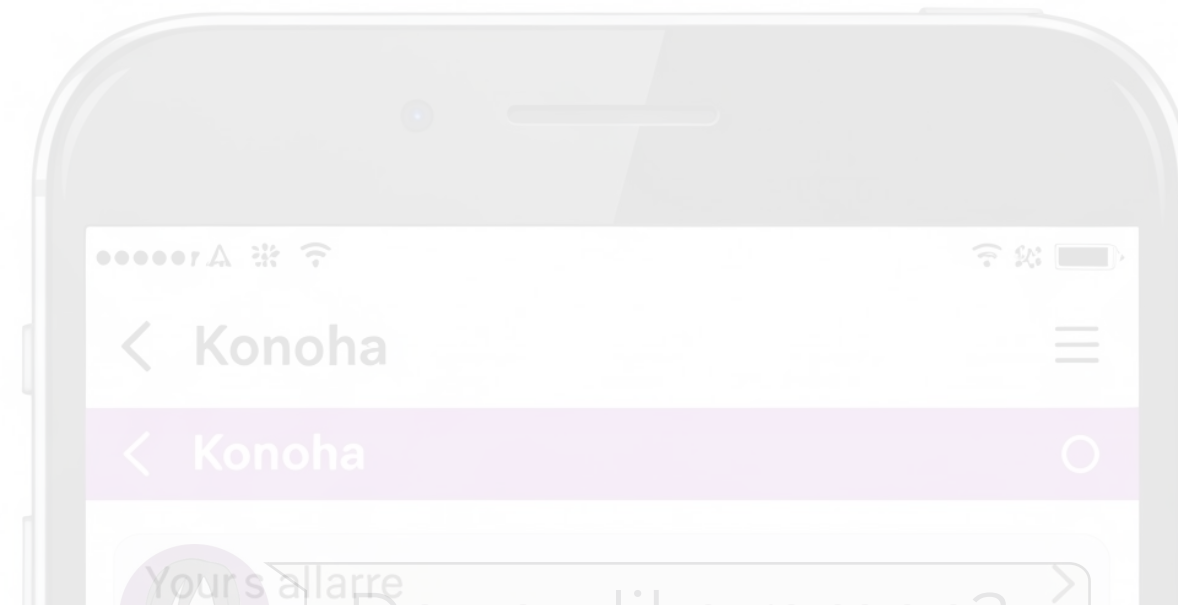
Insider Replay Attack



Insider Replay Attack

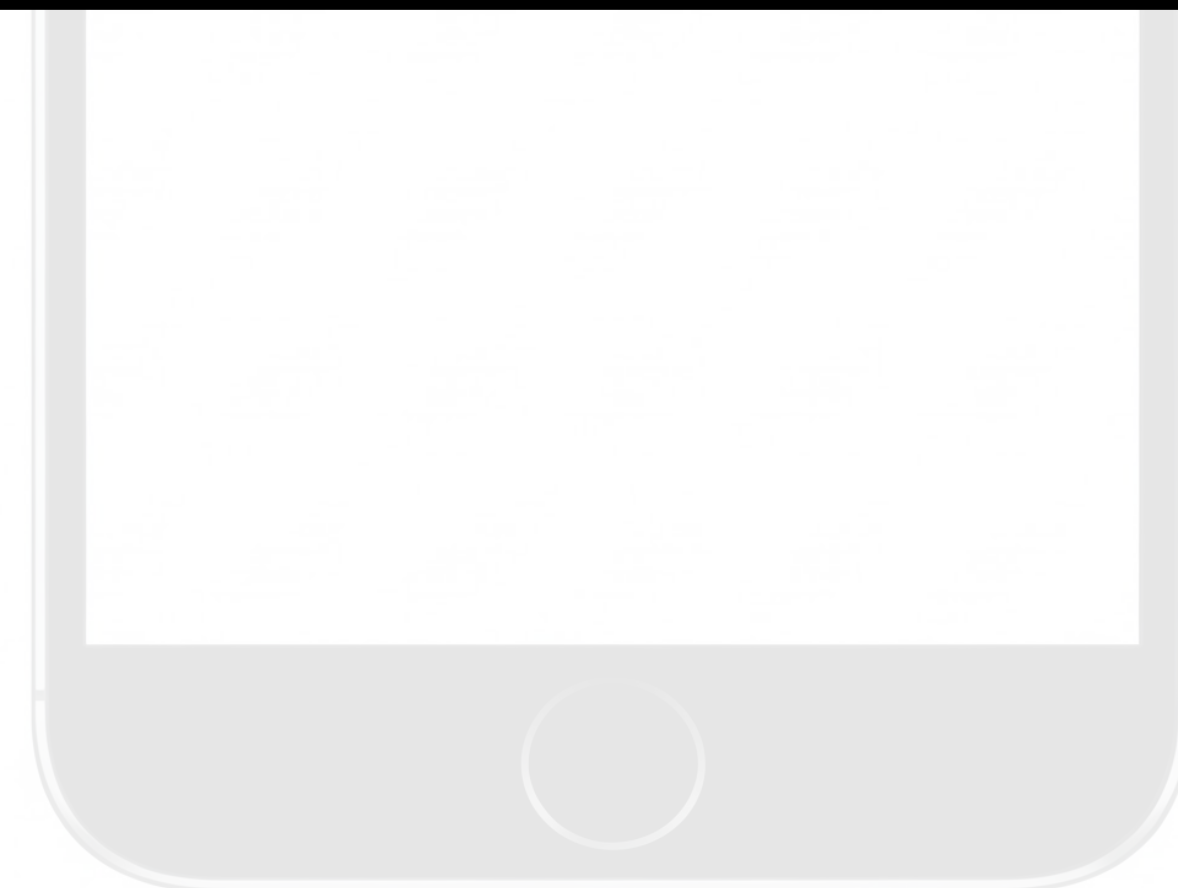


Insider Replay Attack



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

$$m_s = \langle m, group, epoch, leafIndex, ad \rangle$$

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

$$m_s = \langle m, group, epoch, leafIndex, ad \rangle$$

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

$$m_s = \langle m, group, epoch, leafIndex, ad \rangle$$

👉 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

$$m_s = \langle m, group, epoch, leafIndex, ad \rangle$$

☞ 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

$$m_s = \langle m, group, epoch, leafIndex, ad \rangle$$

- ☞ 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

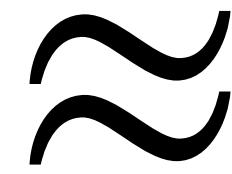
Case Study II: Session



Insider Replay Attack in Session

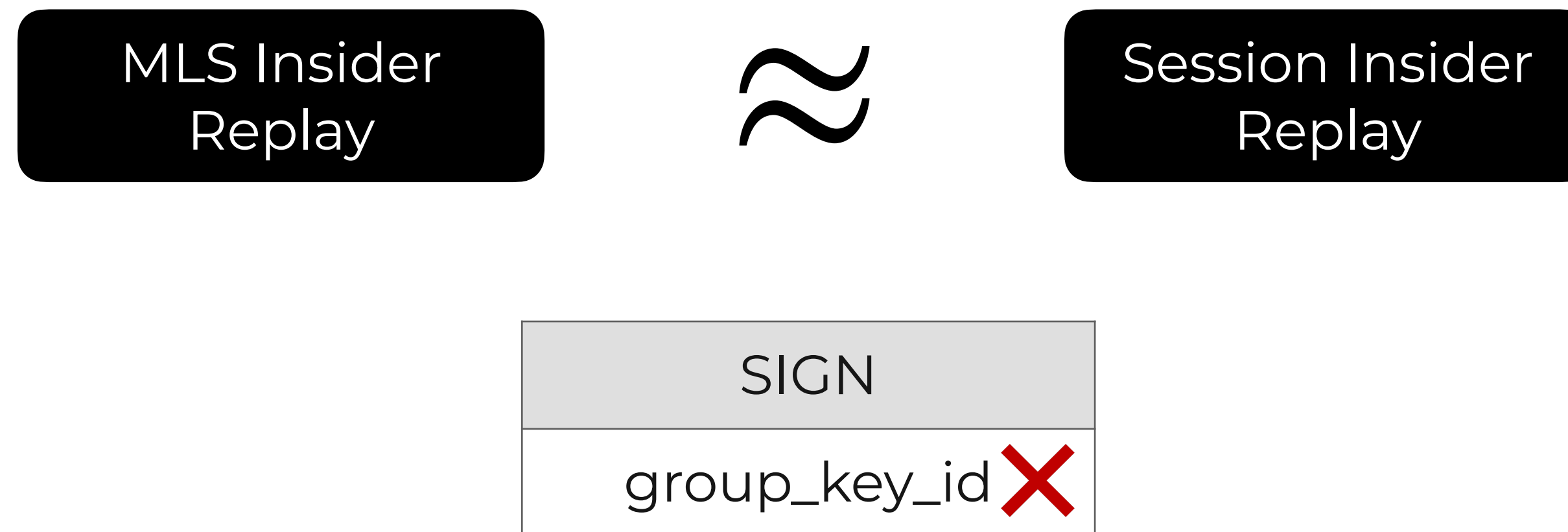
Insider Replay Attack in Session

MLS Insider
Replay



Session Insider
Replay

Insider Replay Attack in Session



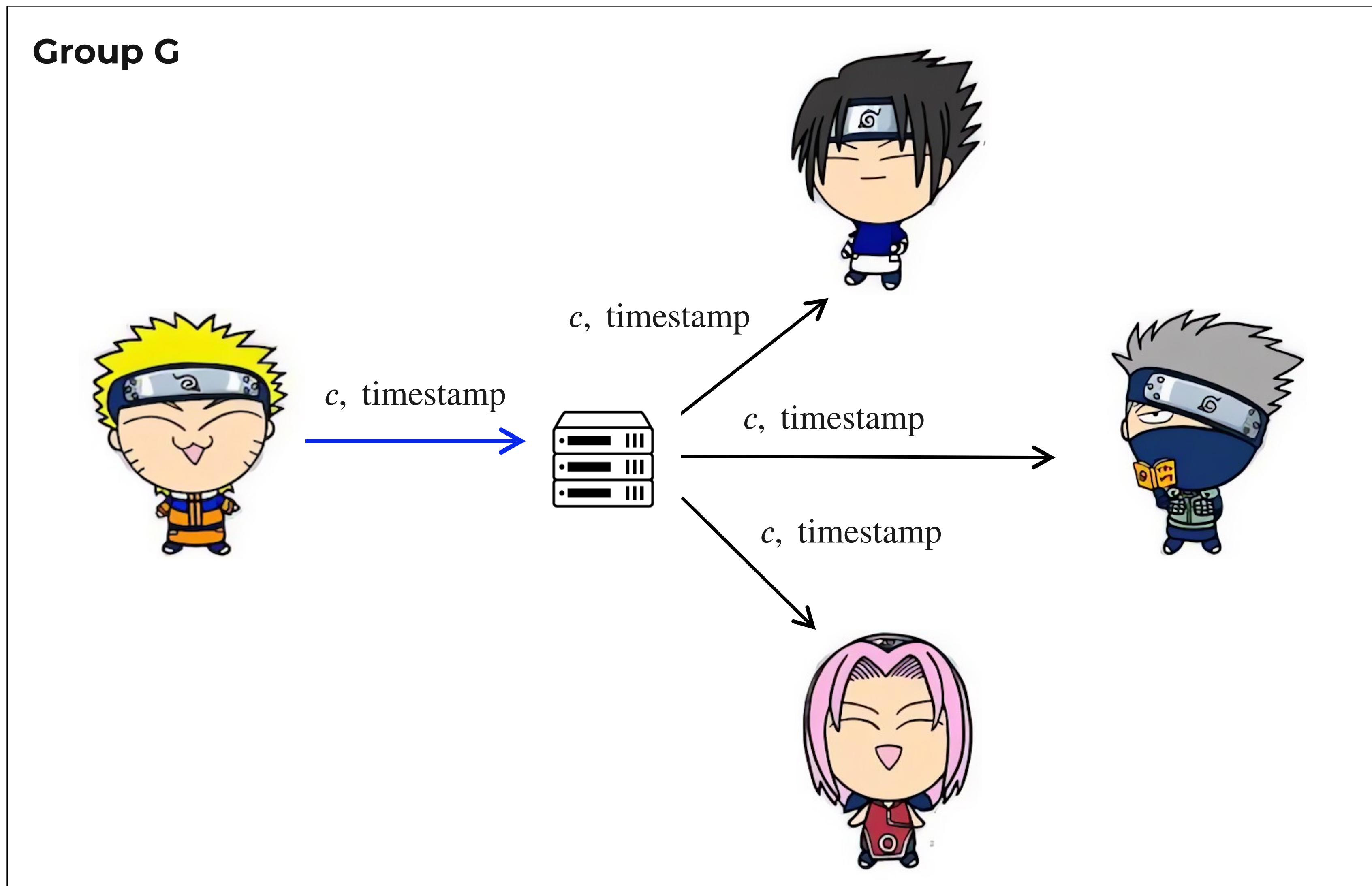
Outsider Replay Attack in Session

Outsider Replay Attack in Session

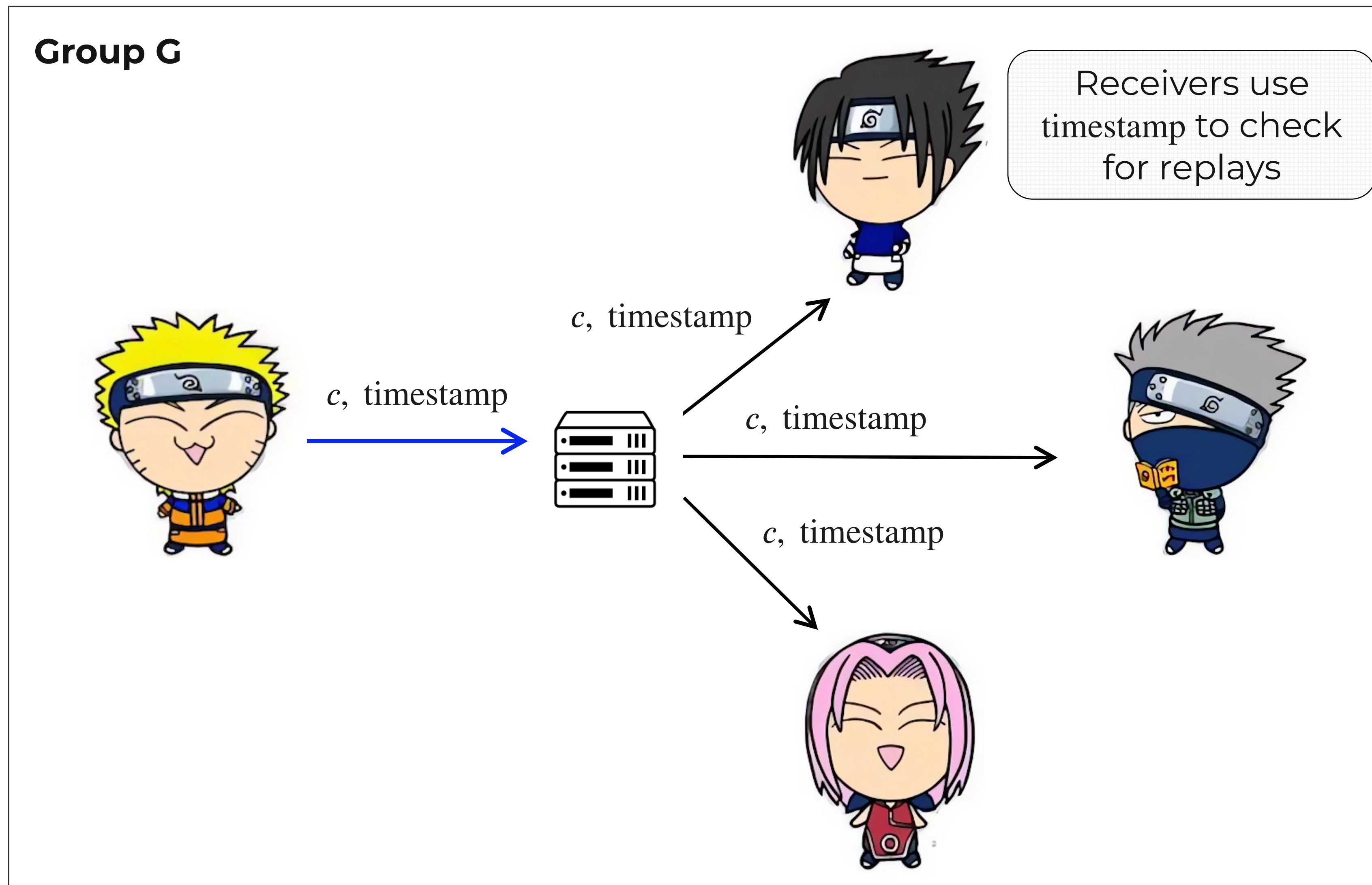
Group G



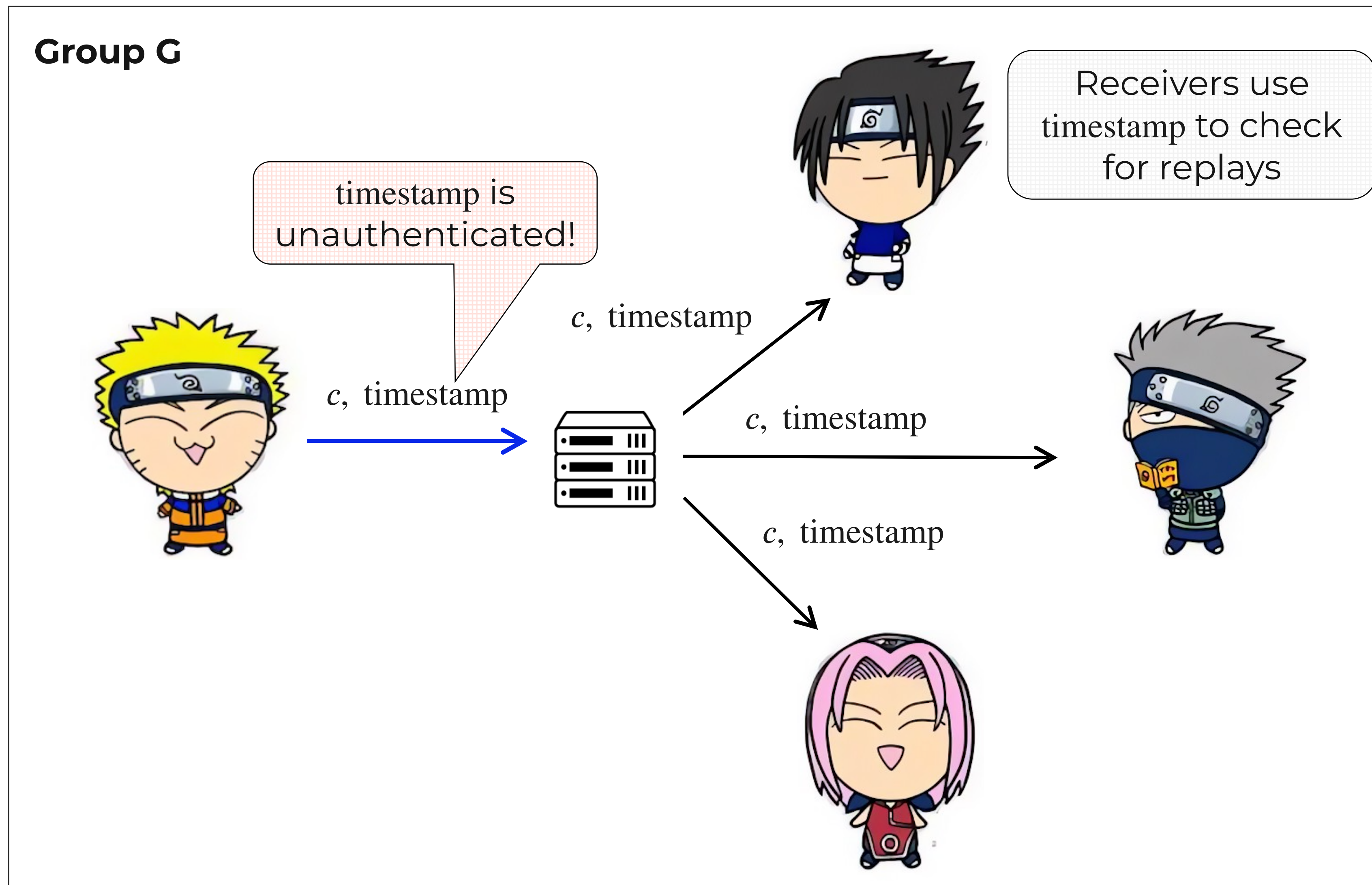
Outsider Replay Attack in Session



Outsider Replay Attack in Session



Outsider Replay Attack in Session



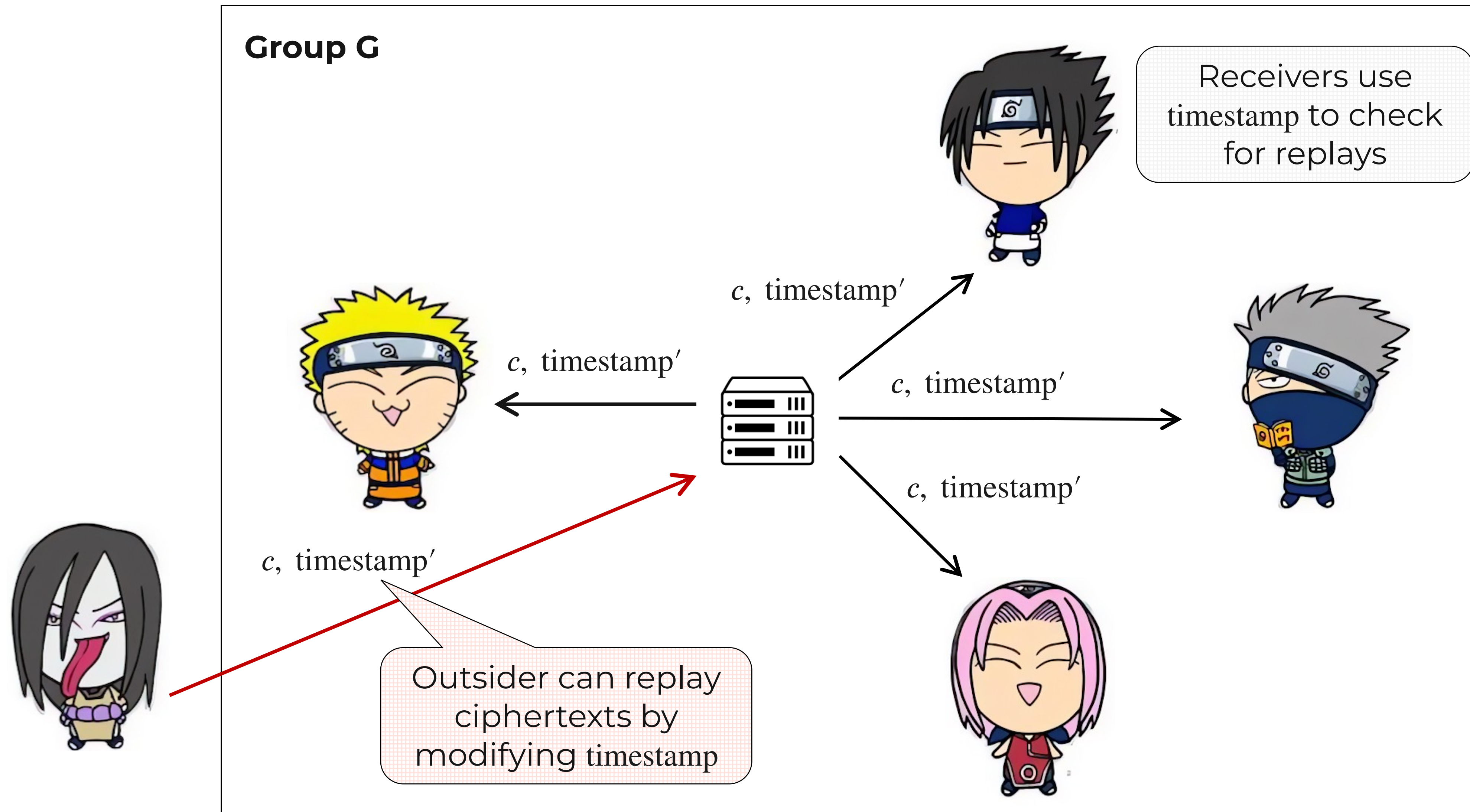
Outsider Replay Attack in Session

Group G

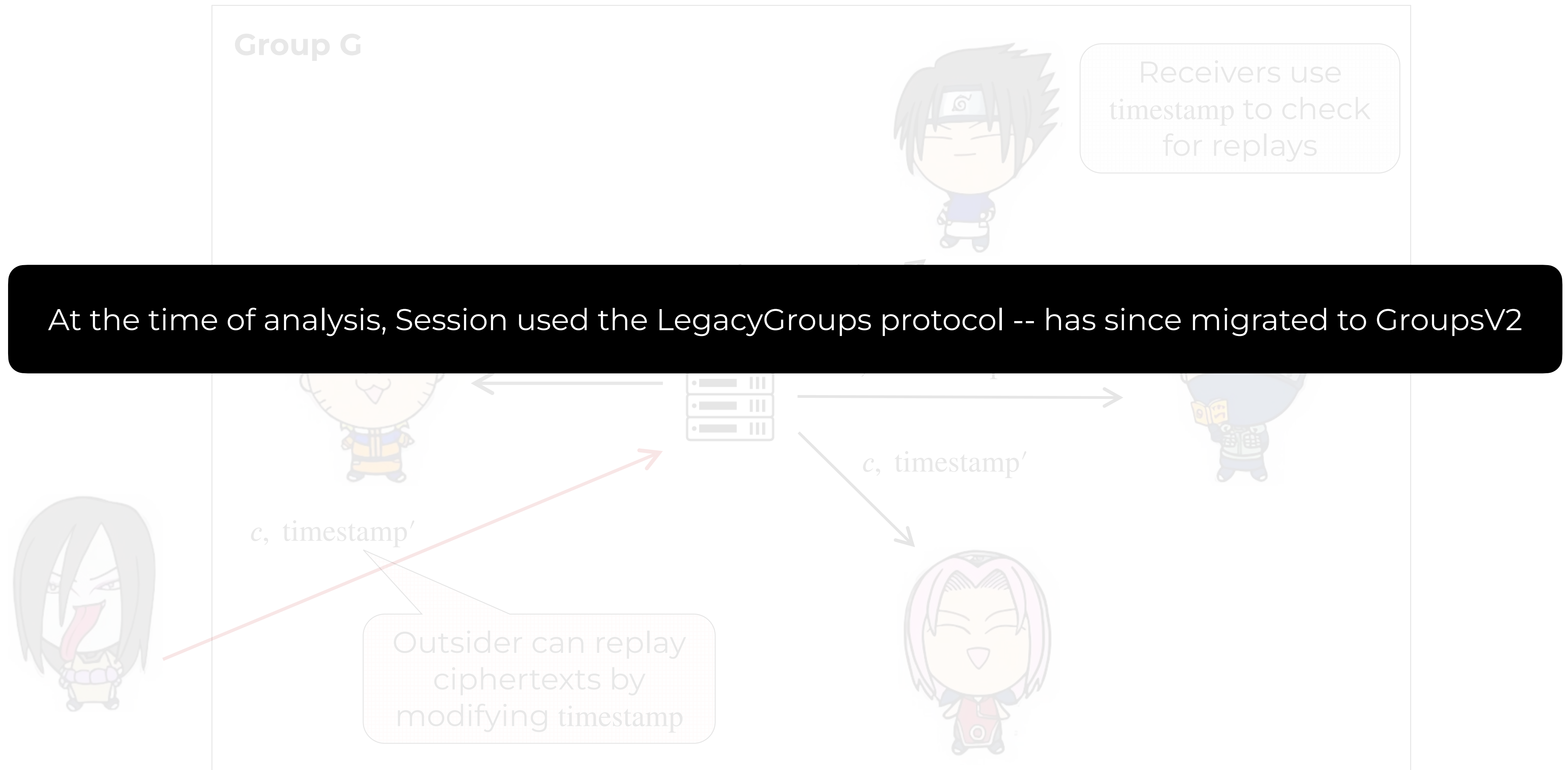
Receivers use
timestamp to check
for replays



Outsider Replay Attack in Session



Outsider Replay Attack in Session



Case Study III: Keybase



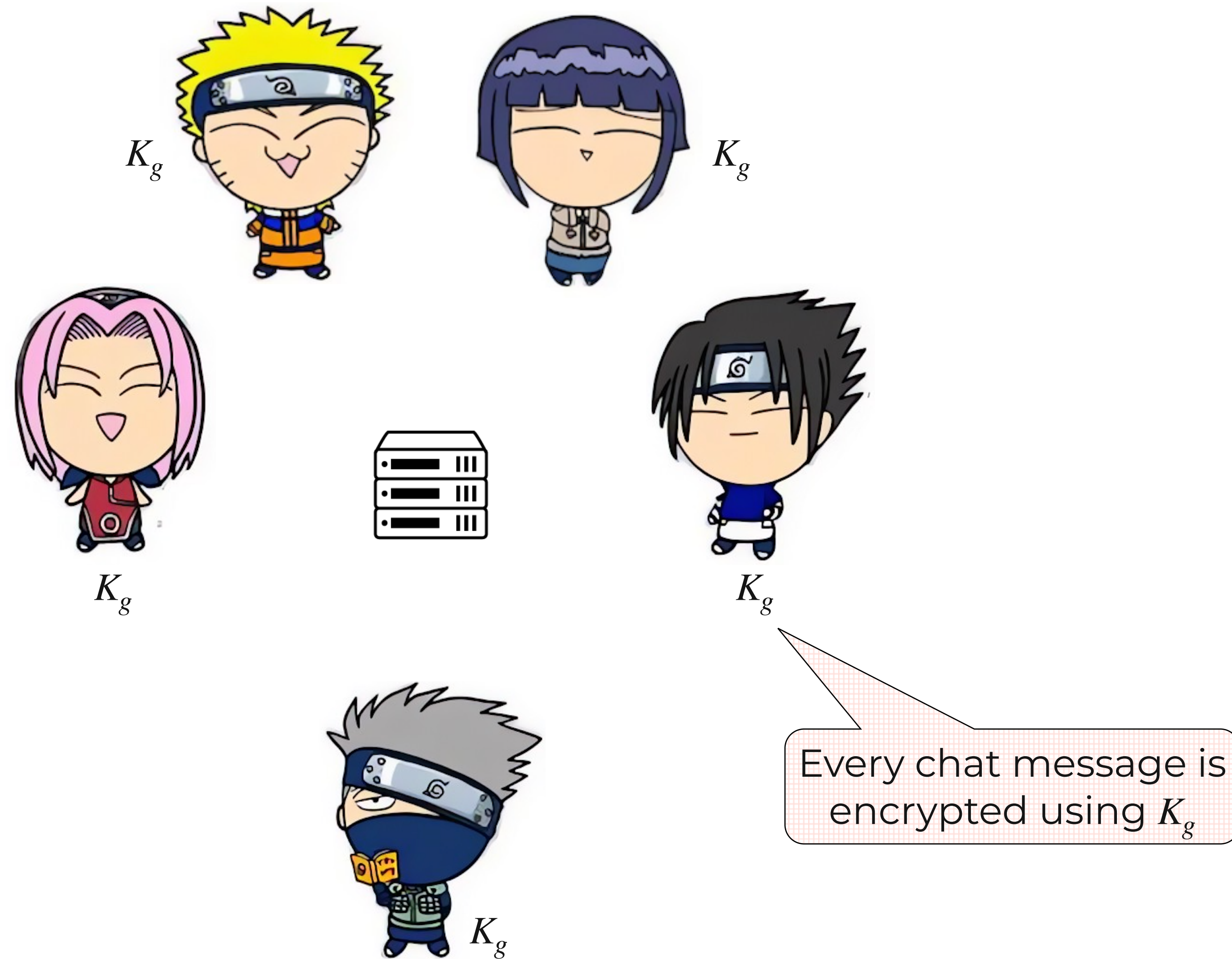
Keybase Encryption Key Derivation



Keybase Encryption Key Derivation



Keybase Encryption Key Derivation

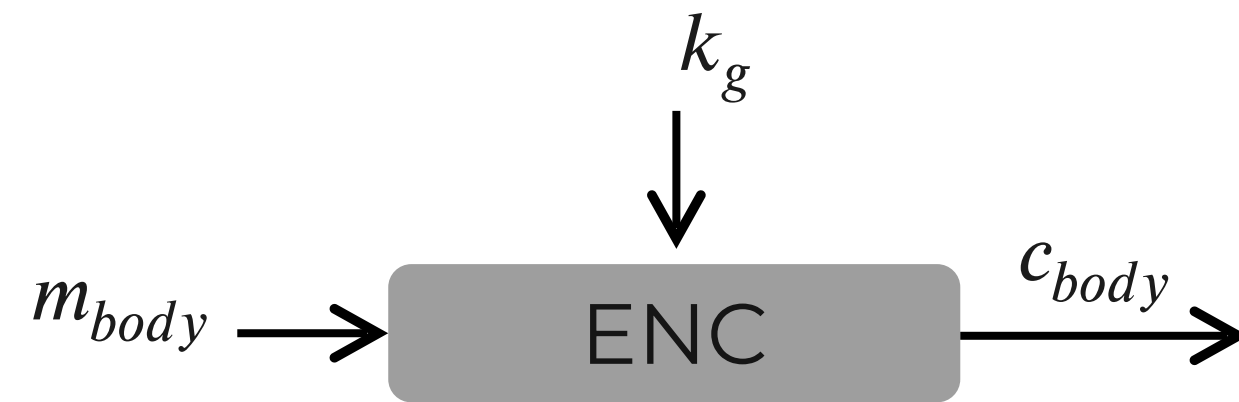


Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt

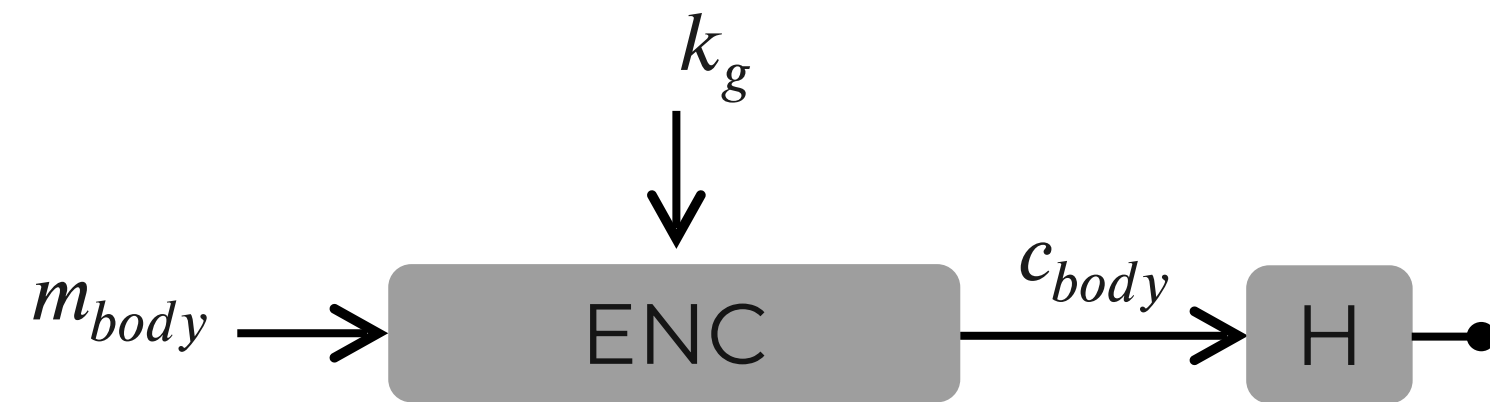
Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt



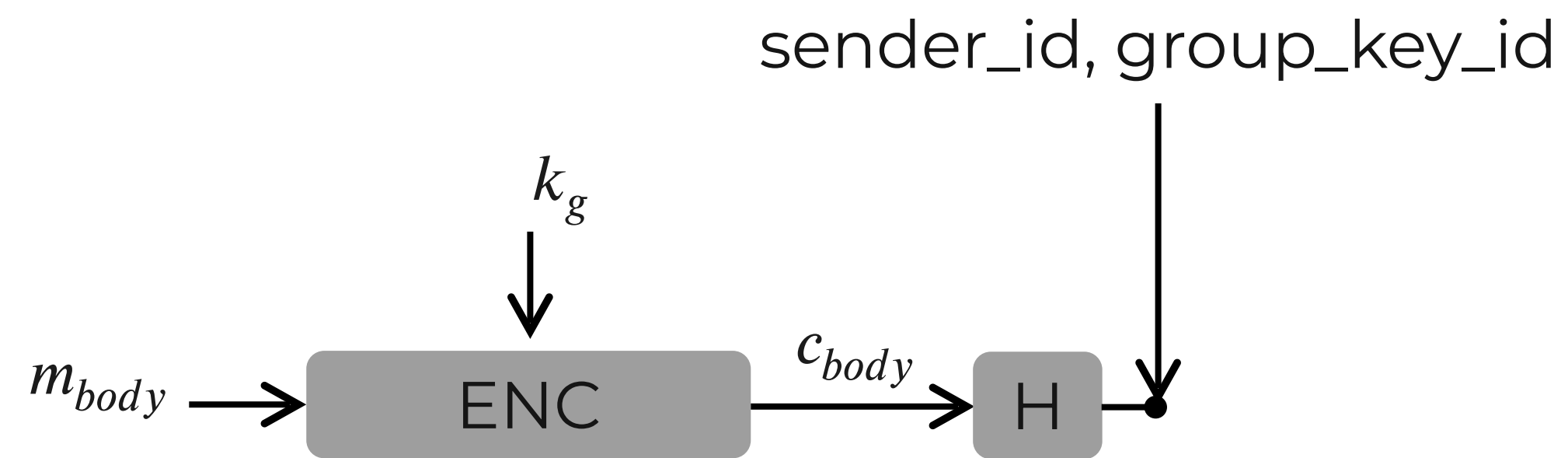
Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt



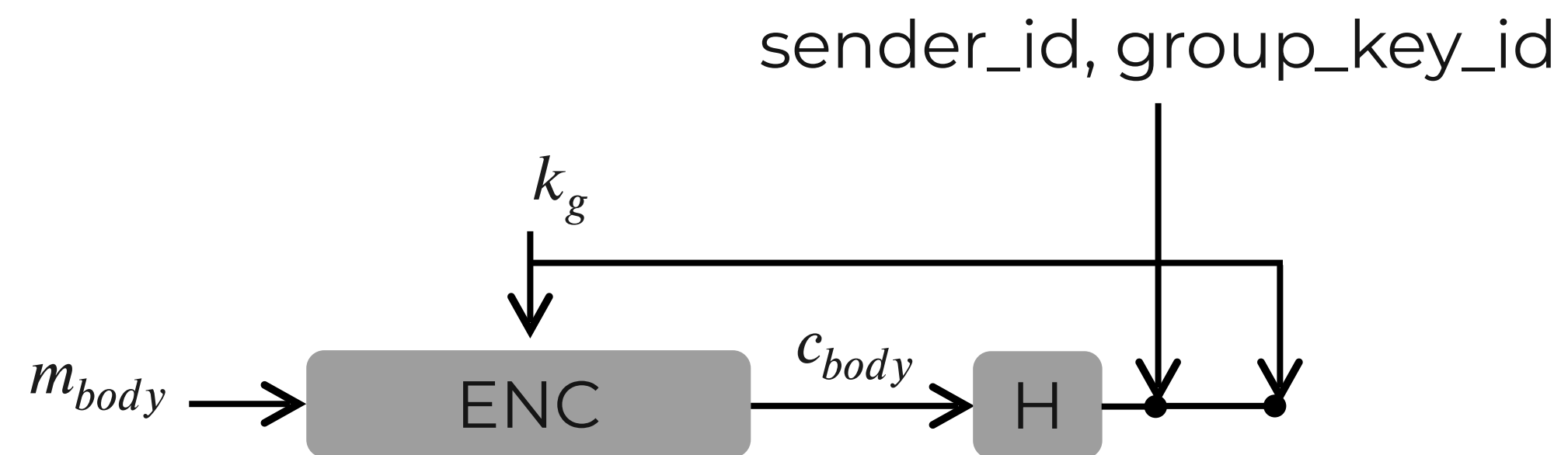
Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt



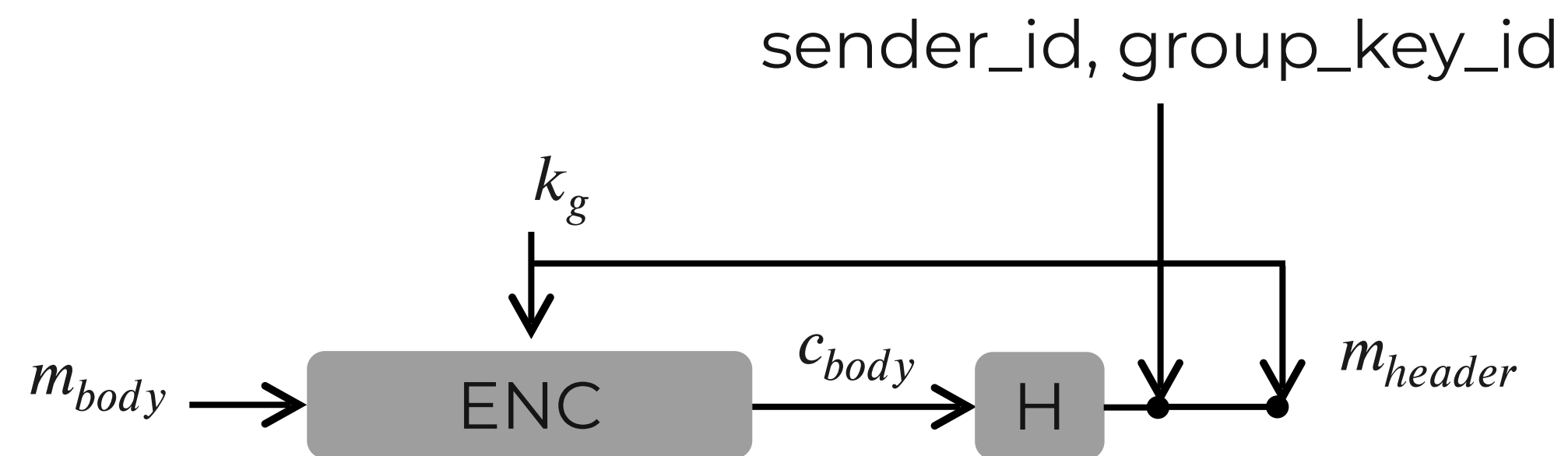
Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt



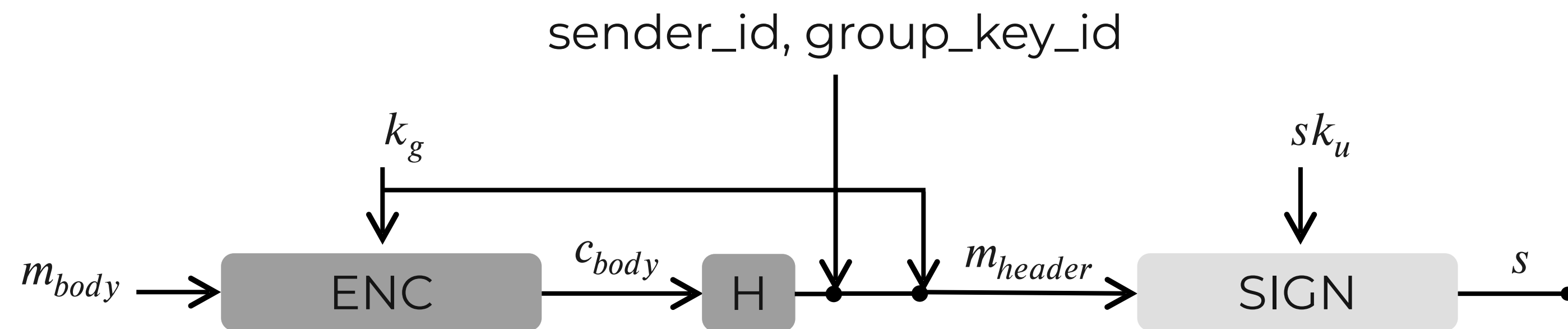
Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt



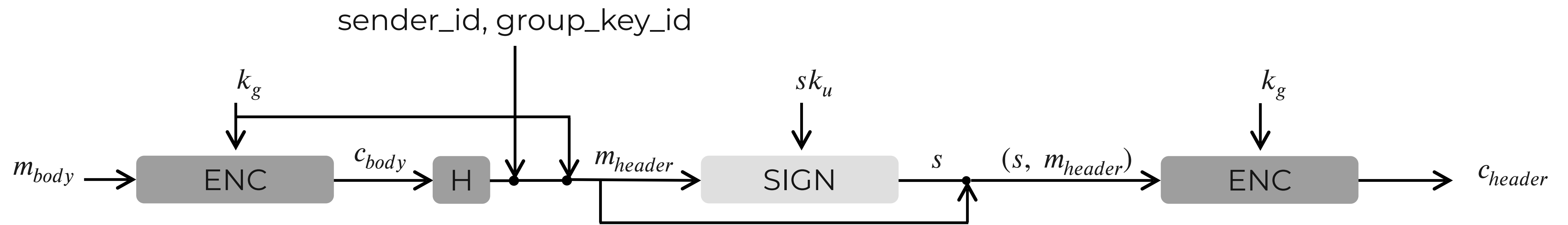
Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt



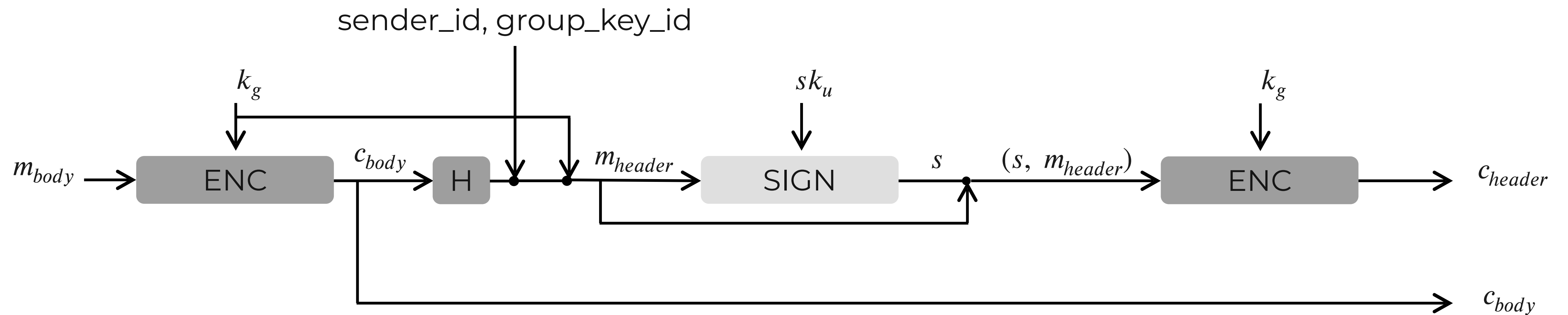
Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt



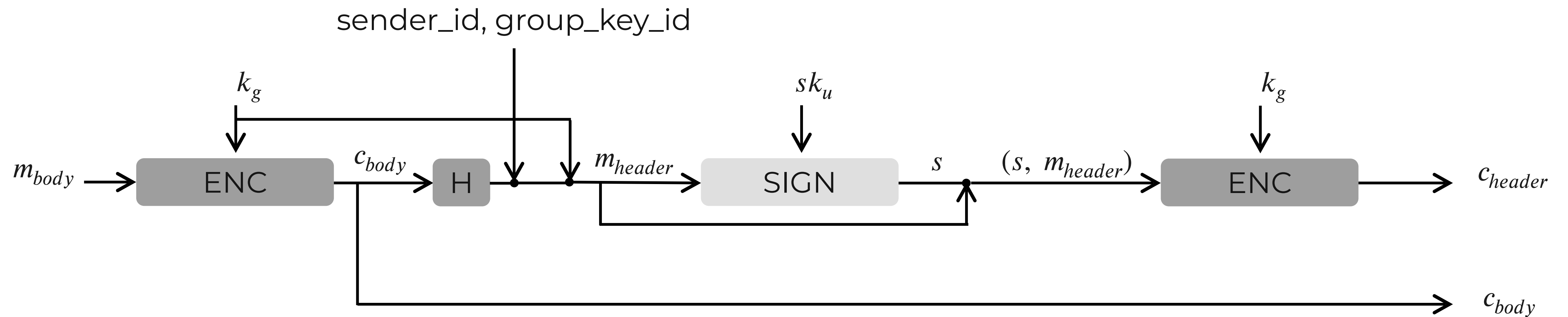
Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt



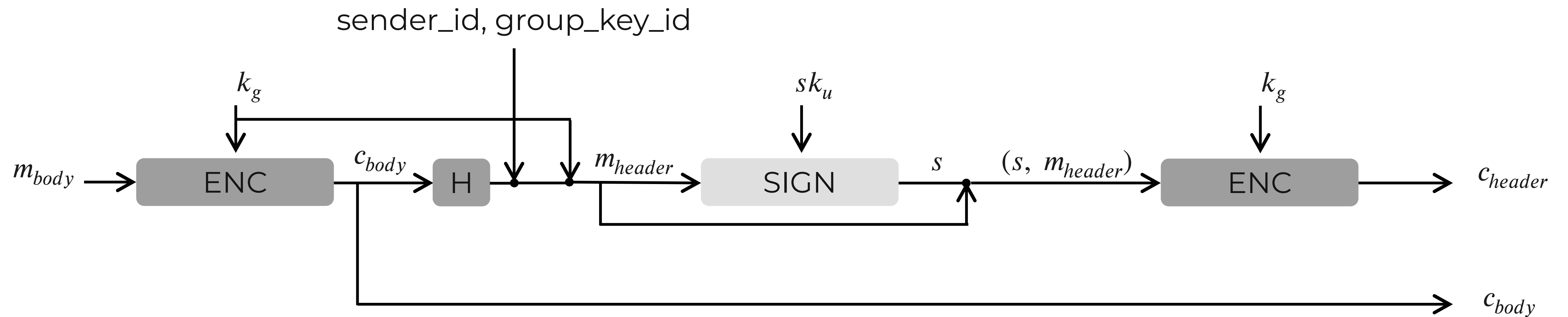
Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt



Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt

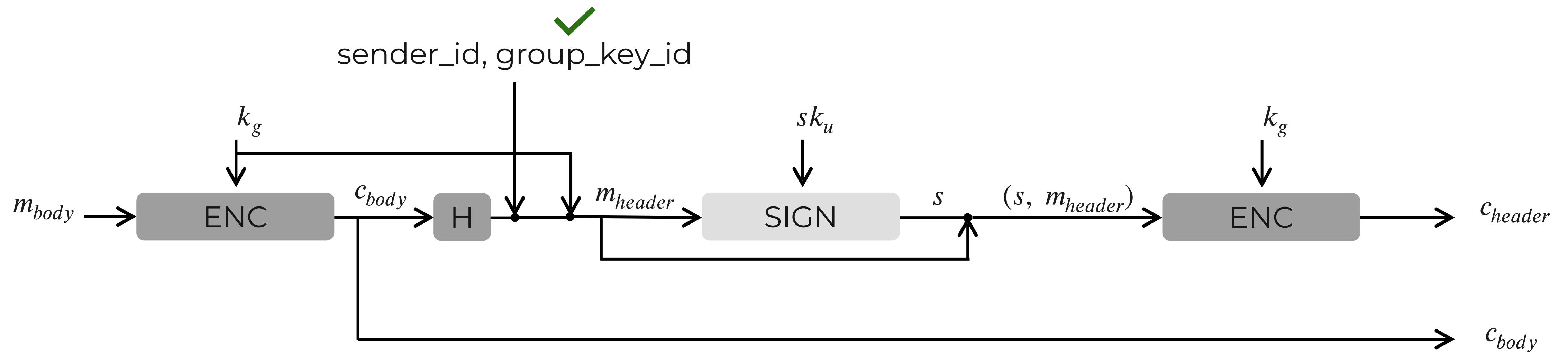


Recall:

SIGN
group_key_id

Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt

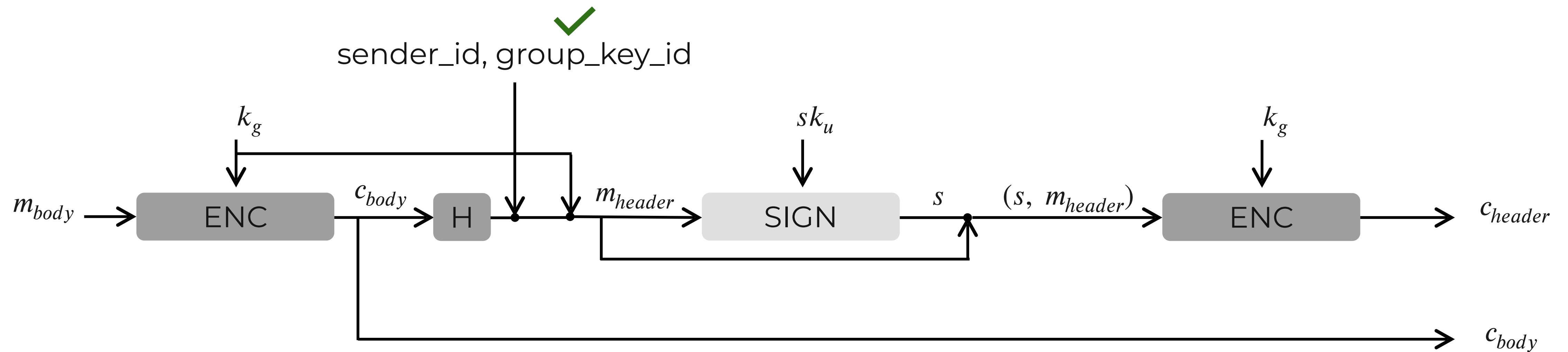


Recall:

SIGN
group_key_id ✓

Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt

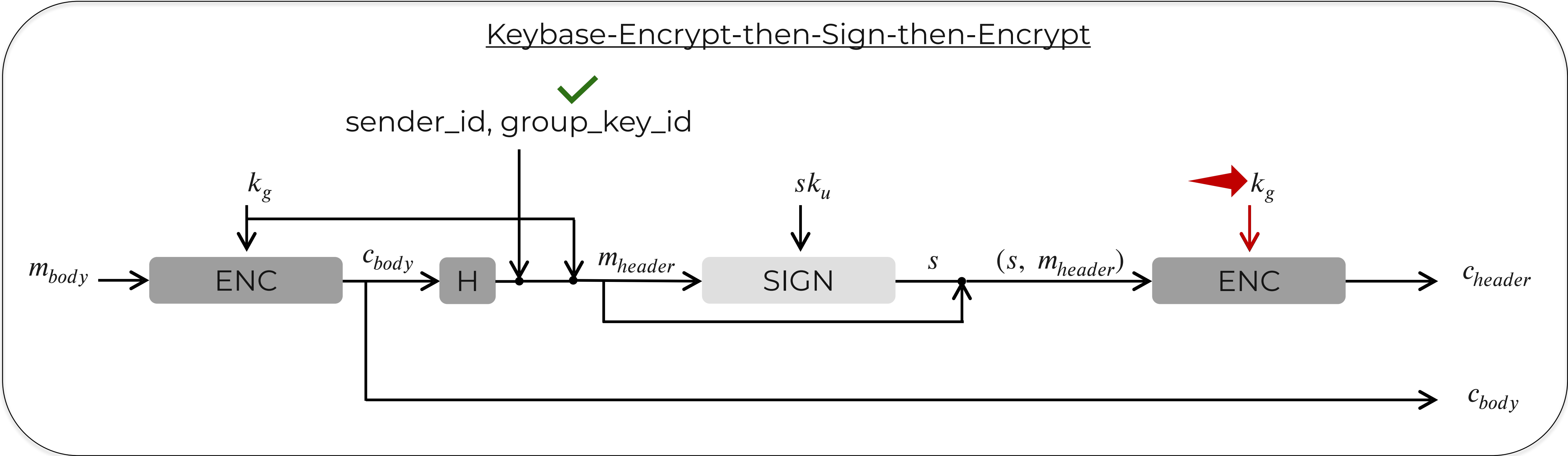


Recall:

SIGN
group_key_id ✓



Chat Encryption in Keybase



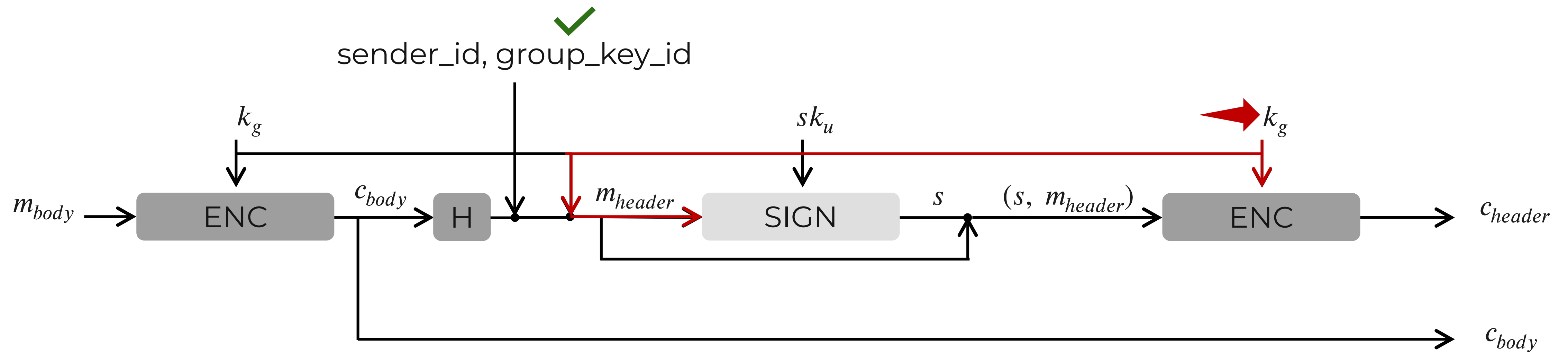
Recall:

SIGN
group_key_id ✓



Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt

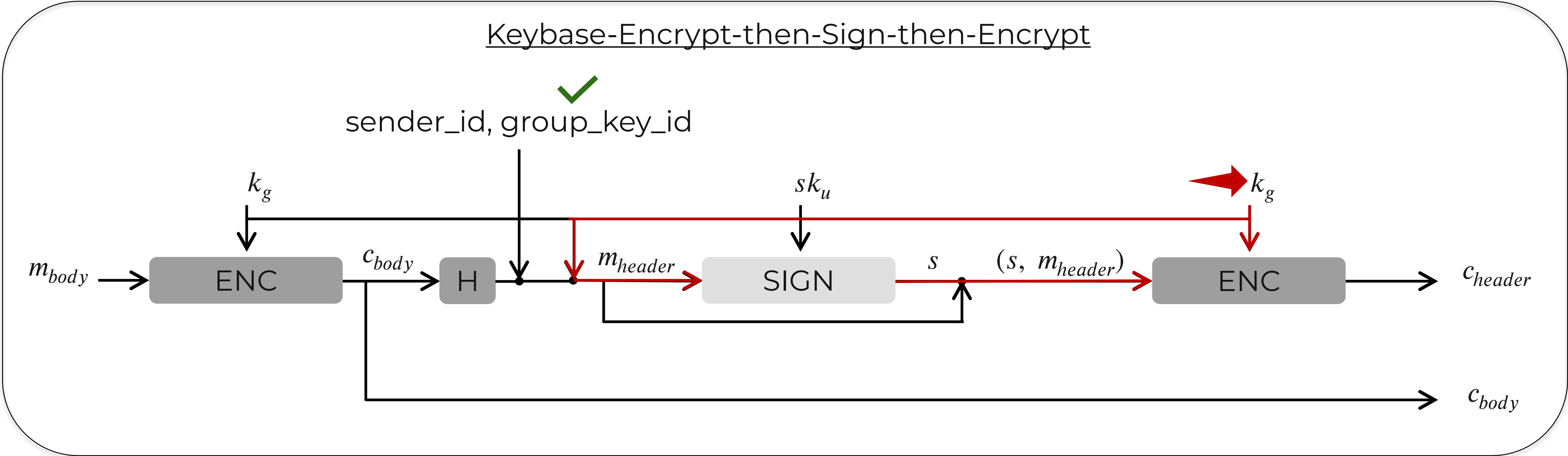


Recall:

SIGN
group_key_id ✓



Chat Encryption in Keybase



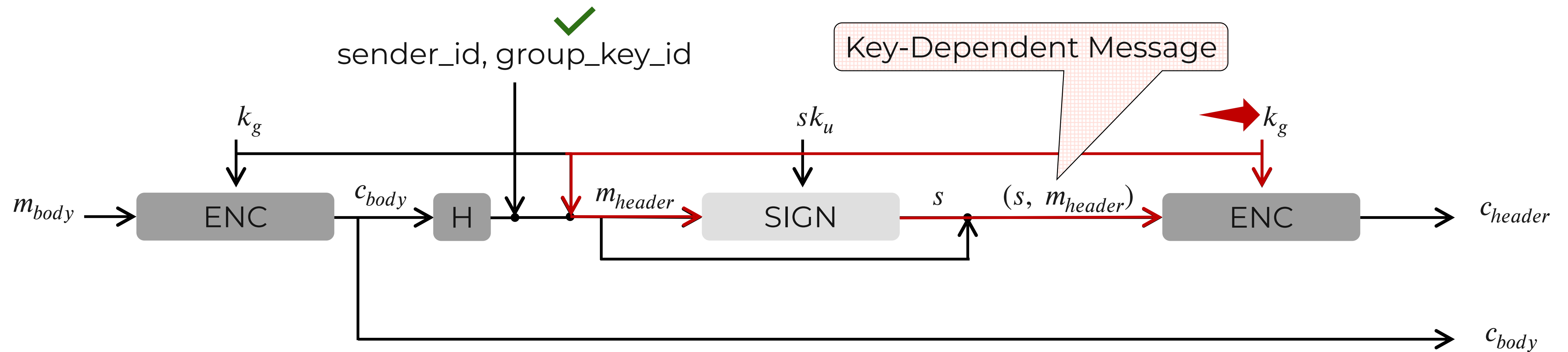
Recall:

SIGN
group_key_id ✓



Chat Encryption in Keybase

Keybase-Encrypt-then-Sign-then-Encrypt

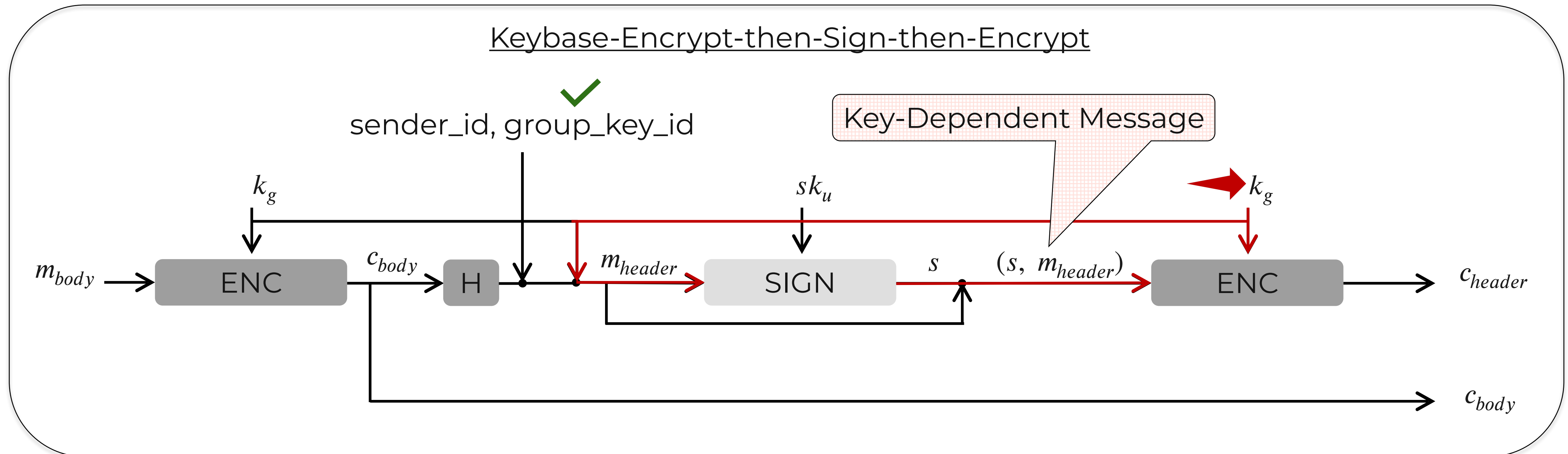


Recall:

SIGN
group_key_id ✓



Chat Encryption in Keybase

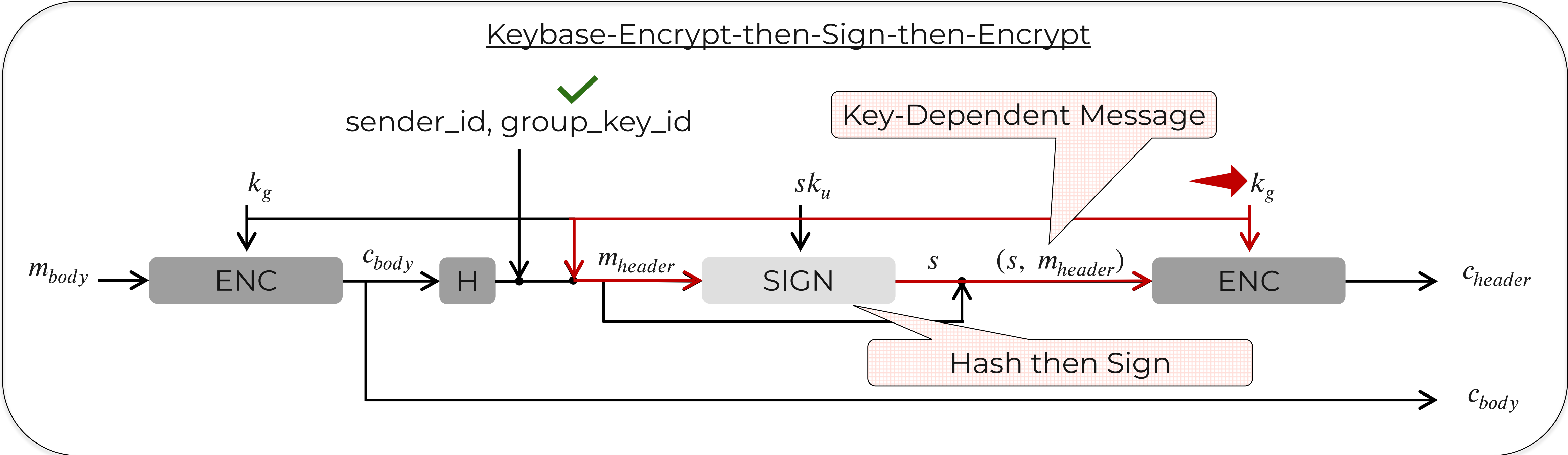


Recall:

SIGN
group_key_id ✓



Chat Encryption in Keybase



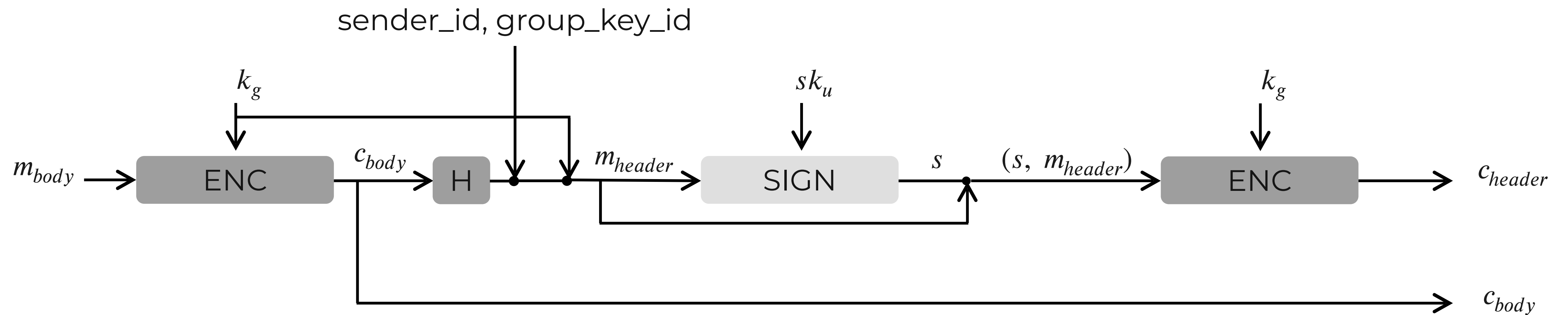
Recall:

SIGN
group_key_id ✓

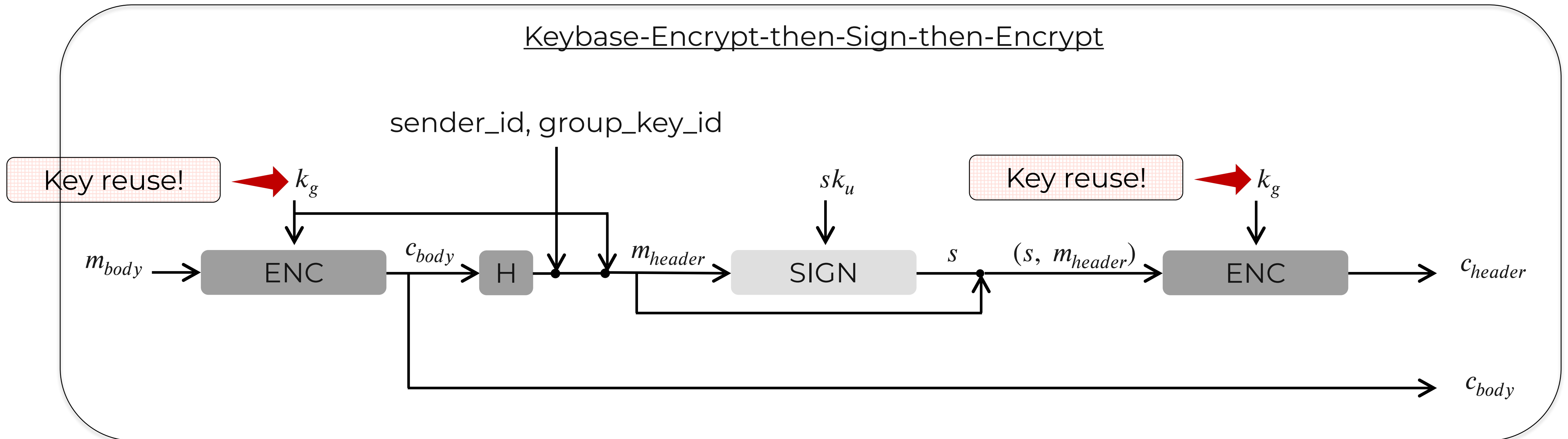


No Attacks, But ...

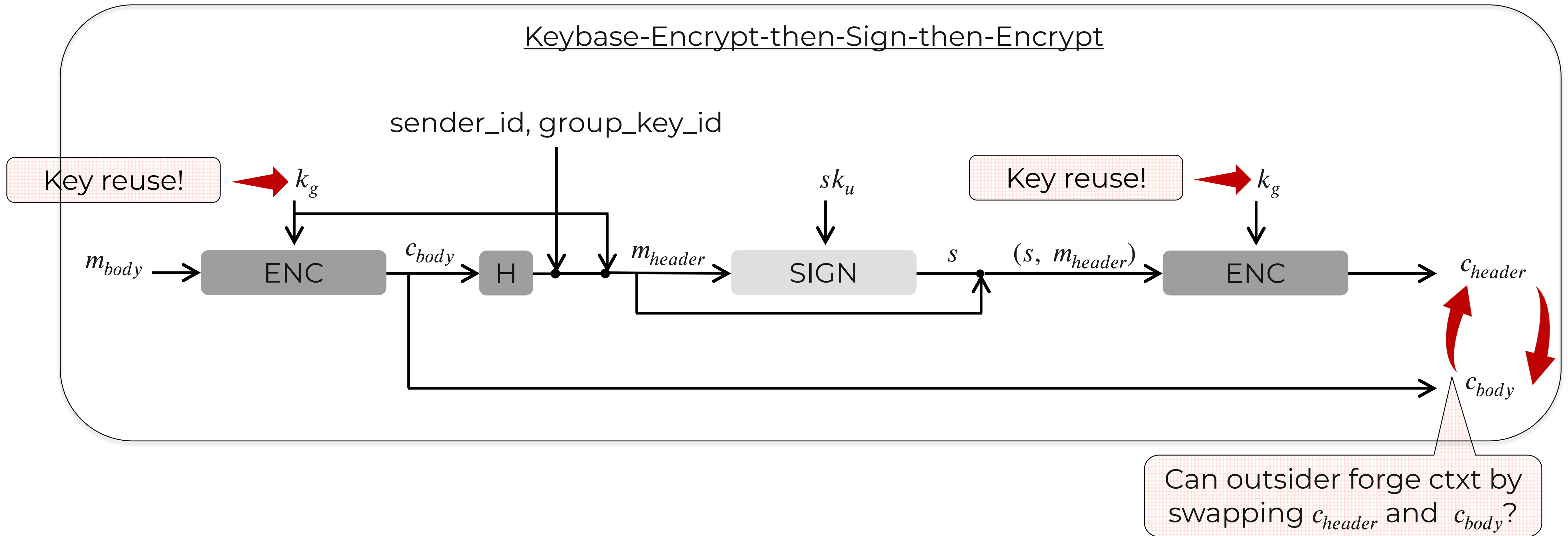
Keybase-Encrypt-then-Sign-then-Encrypt



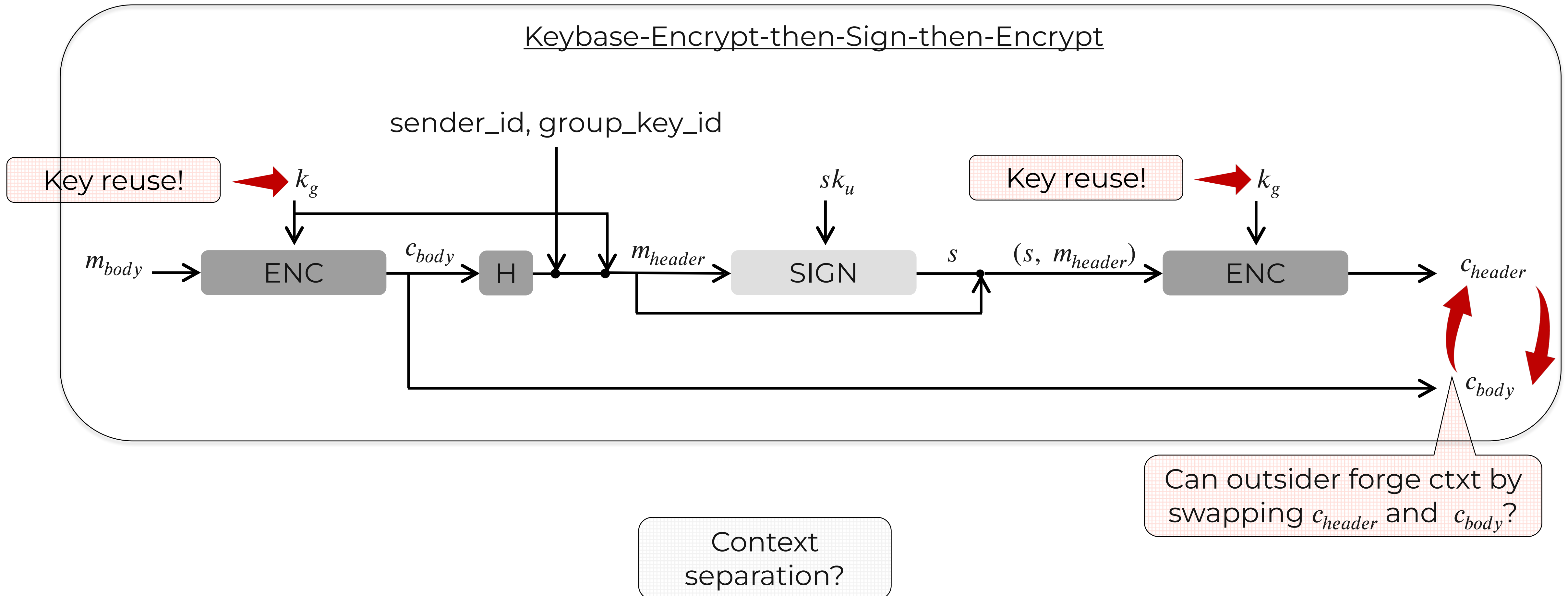
No Attacks, But ...



No Attacks, But ...

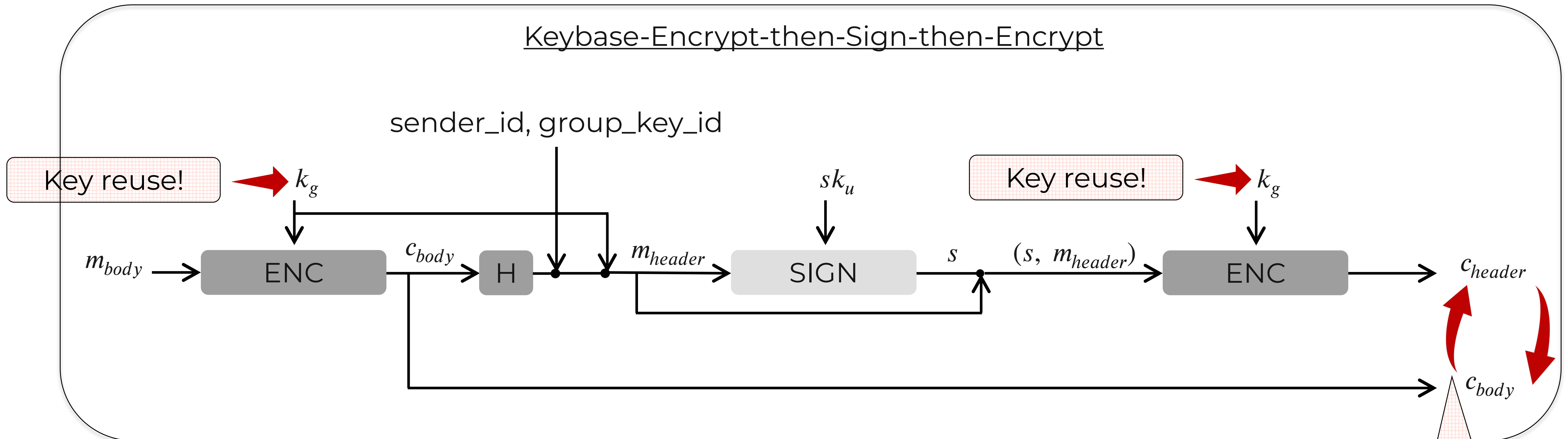


No Attacks, But ...



No Attacks, But ...

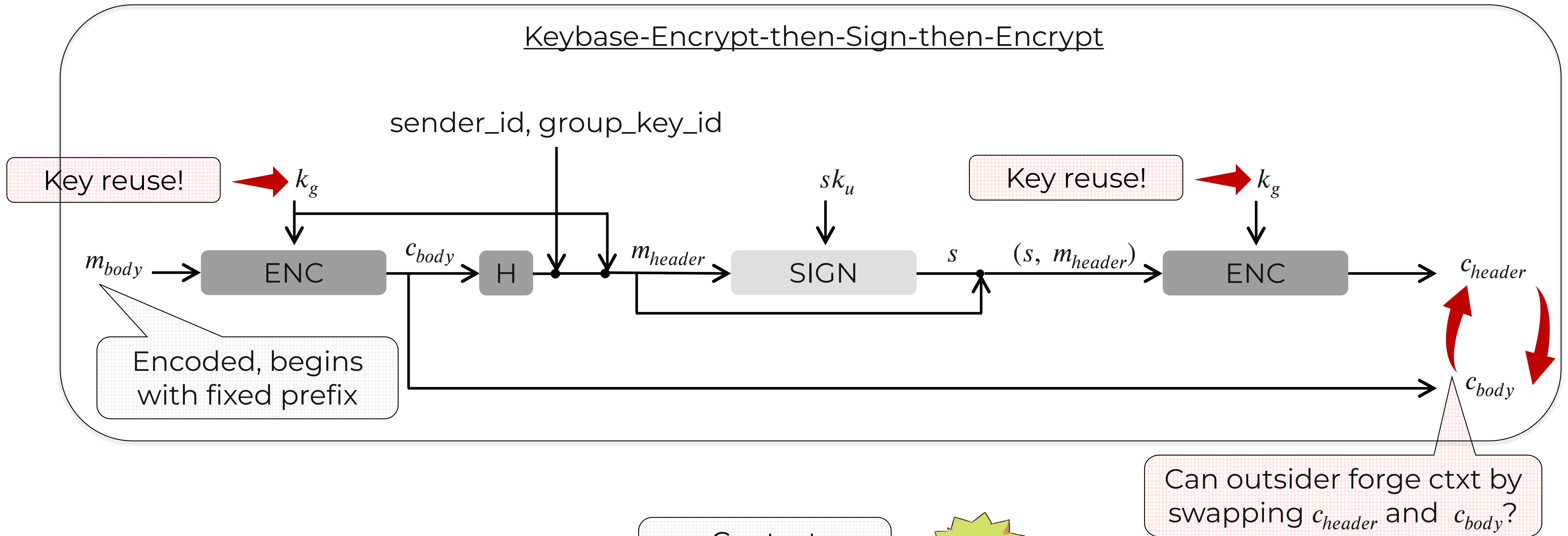
Keybase-Encrypt-then-Sign-then-Encrypt



Context
separation?



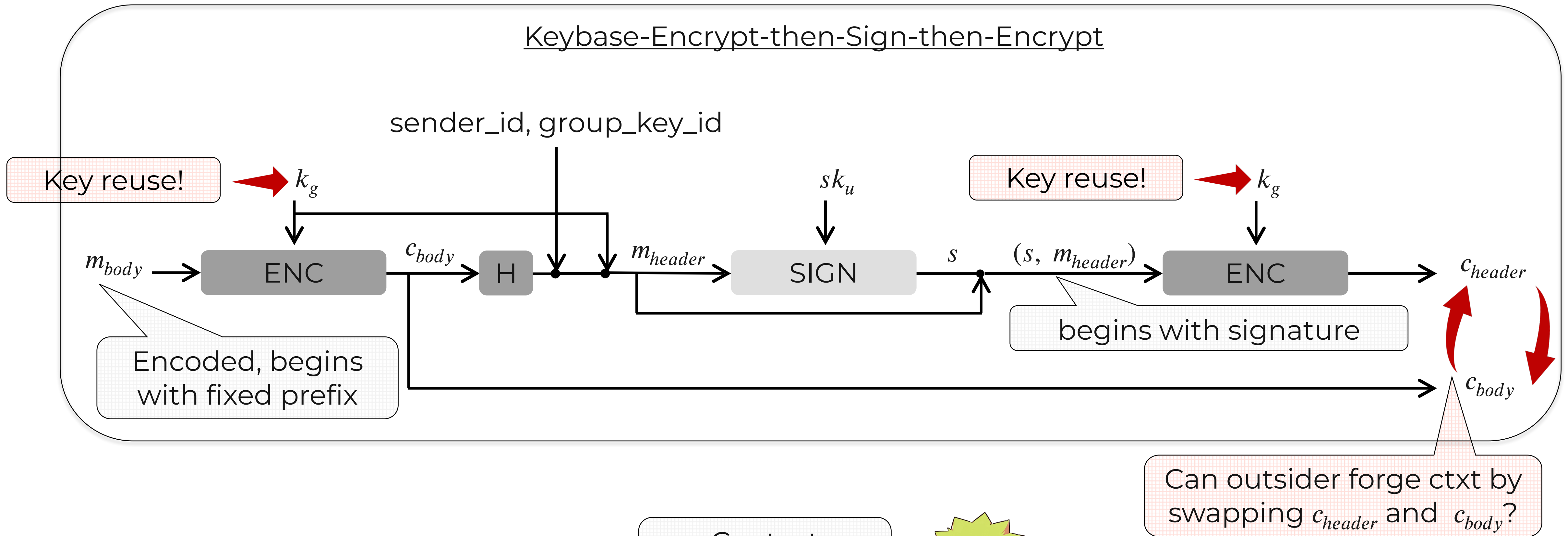
No Attacks, But ...



Context
separation?



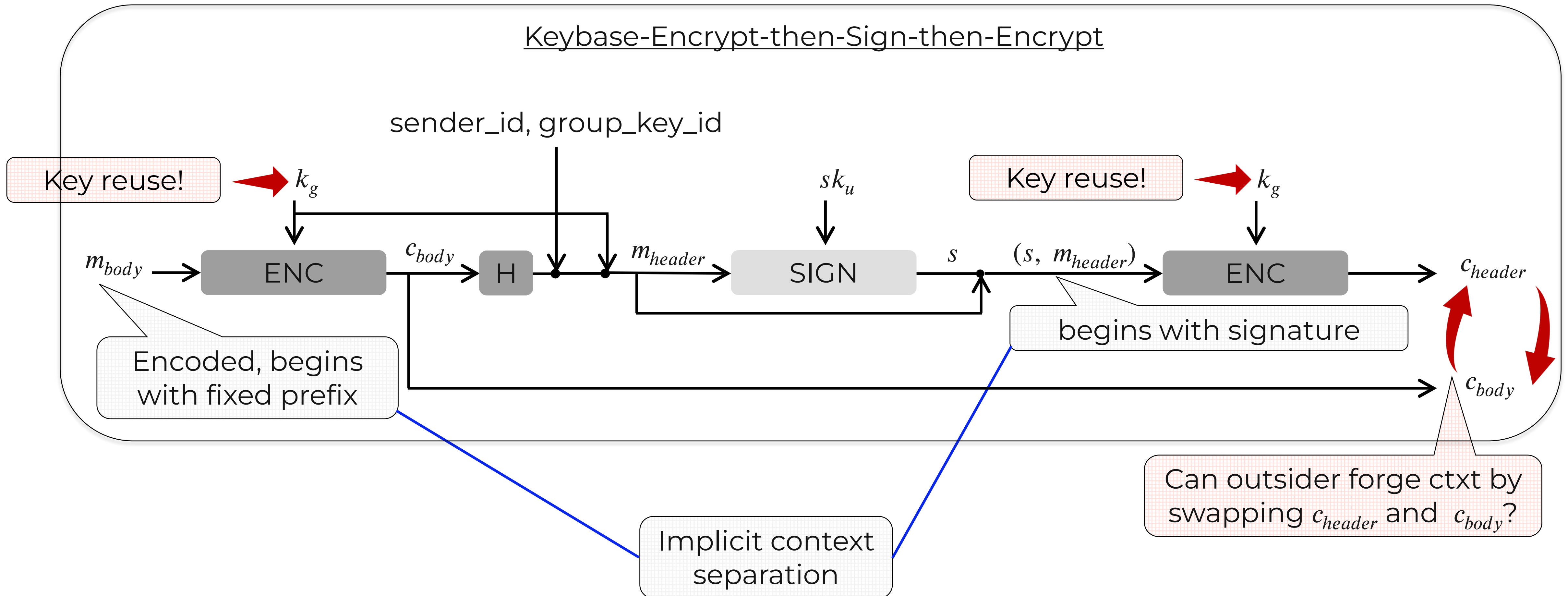
No Attacks, But ...



Context separation?



No Attacks, But ...




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)
- 🔑 Non-cryptographic encoding could be crucial for a scheme to be secure 










Takeaways

- 🔑 Message and sender authenticity are important in group messaging settings (sometimes even more than privacy)
- 🔑 Non-cryptographic encoding could be crucial for a scheme to be secure 🧑🔑
- 🔑 Reusing keys across different contexts is bad 🧑🔑







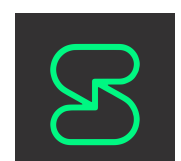


Takeaways

- 🔑 Message and sender authenticity are important in group messaging settings (sometimes even more than privacy)
- 🔑 Non-cryptographic encoding could be crucial for a scheme to be secure 
- 🔑 Reusing keys across different contexts is bad 
- 🔑 Combining encryption with MAC/signatures is non-trivial   

Takeaways

- 🔑 Message and sender authenticity are important in group messaging settings (sometimes even more than privacy)
- 🔑 Non-cryptographic encoding could be crucial for a scheme to be secure 
- 🔑 Reusing keys across different contexts is bad 
- 🔑 Combining encryption with MAC/signatures is non-trivial   
- 🔑 Formal definitions are useful to analyze real-world security     **[matrix]**

Takeaways

- 🔑 Message and sender authenticity are important in group messaging settings (sometimes even more than privacy)
- 🔑 Non-cryptographic encoding could be crucial for a scheme to be secure 
- 🔑 Reusing keys across different contexts is bad 
- 🔑 Combining encryption with MAC/signatures is non-trivial   
- 🔑 Formal definitions are useful to analyze real-world security     [matrix]



Details in the
papers!

Coming Soon

<https://eprint.iacr.org/2024/799.pdf>