Privacy-Preserving Authenticated Key Exchange in the Standard Model

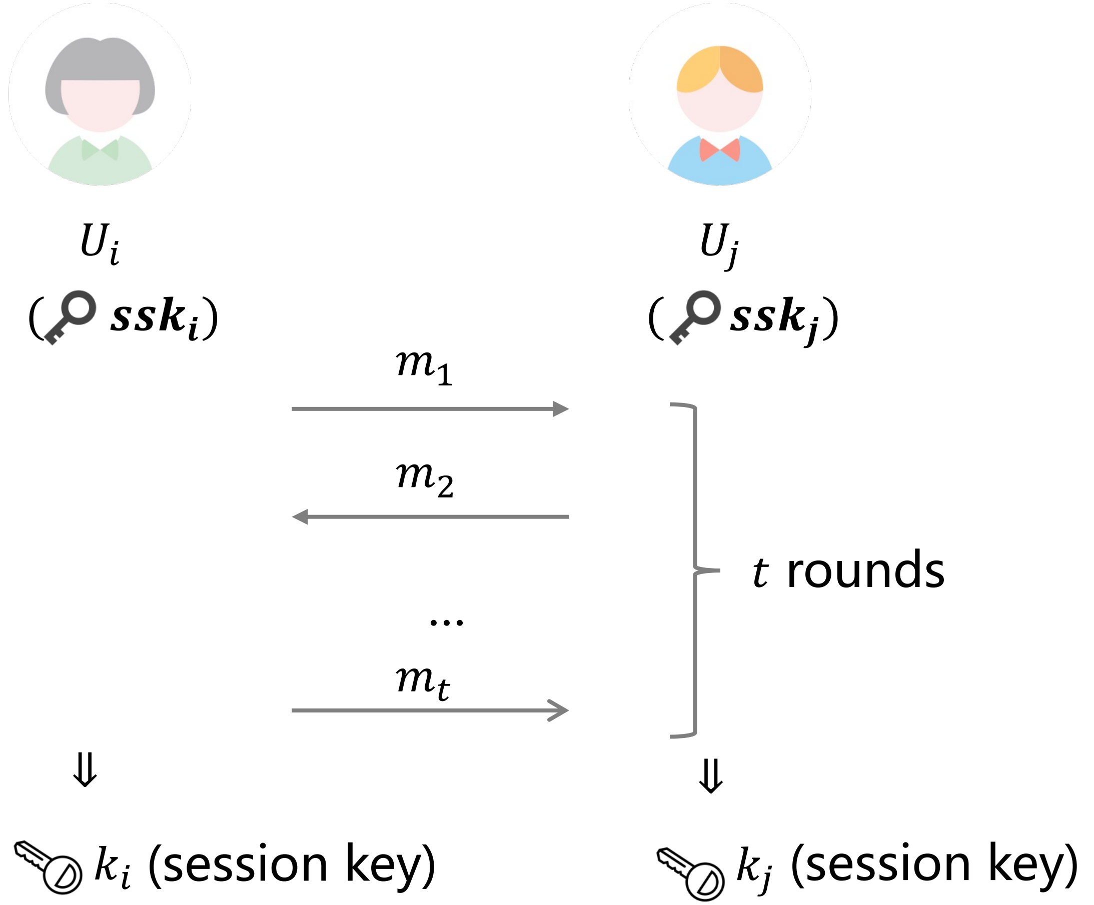
You Lyu, Shengli Liu, Shuai Han, Dawu Gu
Shanghai Jiao Tong University

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Contents

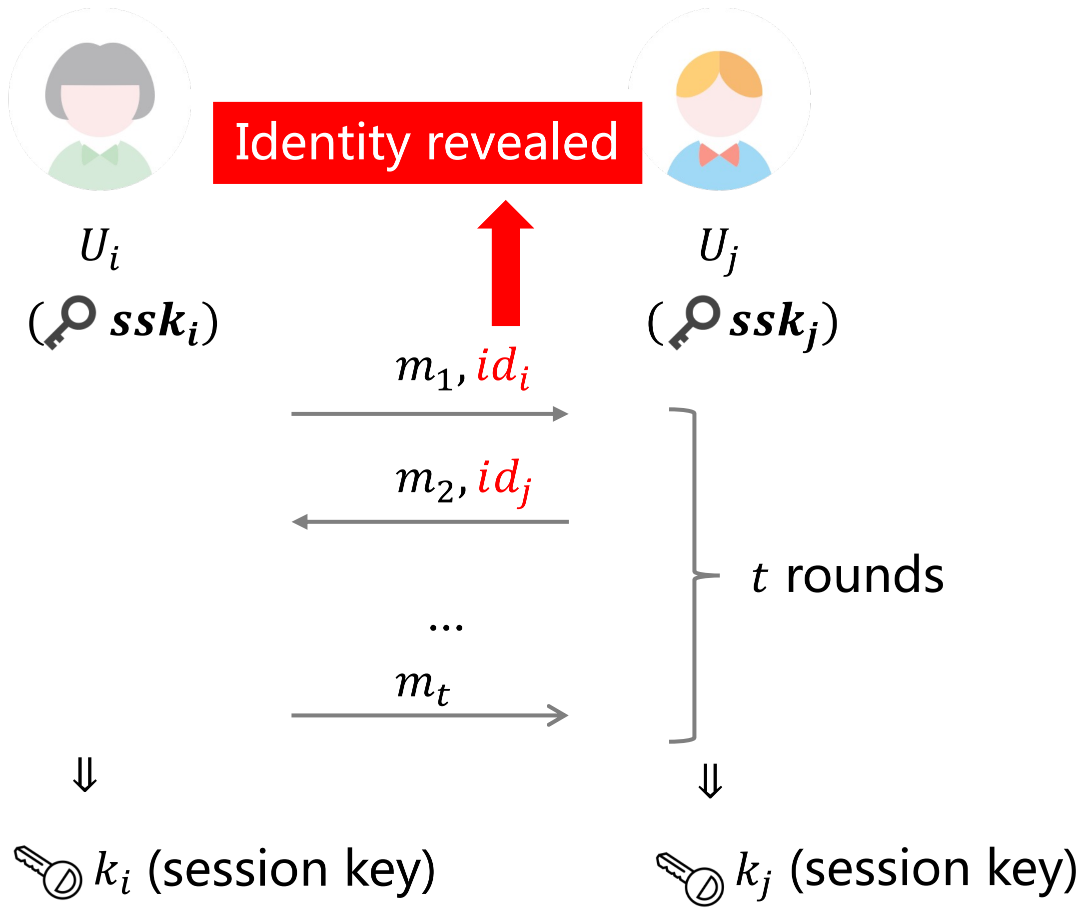
Contents

Authenticated Key Exchange (AKE)



- AKE allows two parties to authenticate each other and securely share a session key.
- It has been widely used in data sharing, electronic notebooks, etc.

Privacy-Preserving AKE (PPAKE)



- AKE allows two parties to authenticate each other and securely share a session key.
- It has been widely used in data sharing, electronic notebooks, etc.
- AKE protocol provides no security on users' identities.
- To solve this problem, PPAKE was proposed.
- Privacy-Preserving: It requires anonymity, which means the adversary cannot identify the users who are communicating.

Most AKE protocols are not Privacy-Preserving

Signed-DH AKE

U_i $\boldsymbol{vk_i}$

$(\boldsymbol{sk_i})$

U_j $\boldsymbol{vk_j}$

$(\boldsymbol{sk_j})$

$$g^a, Sign(sk_i, i|j|g^a) \longrightarrow$$

$$\longleftarrow g^b, Sign(sk_j, i|j|g^a|g^b)$$

$\Downarrow$

$k_i = g^{ab}$

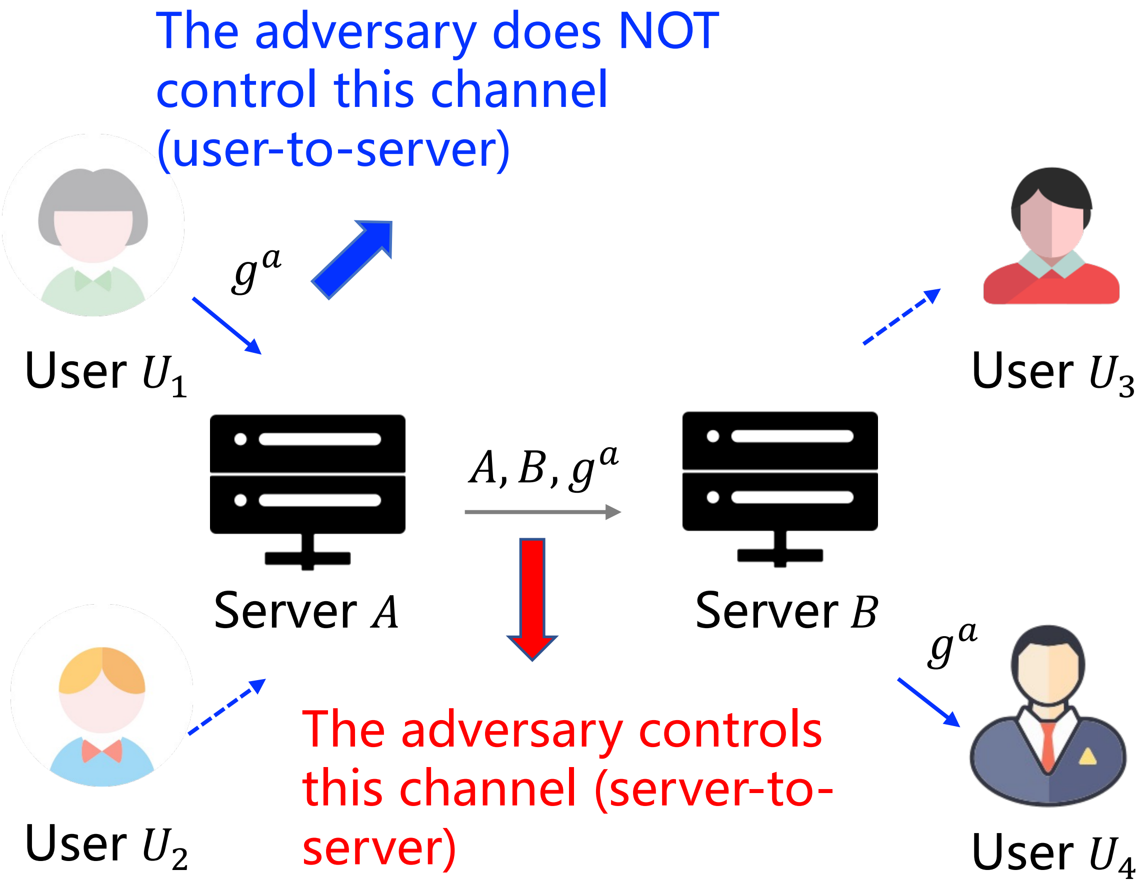
$\Downarrow$

$k_j = g^{ab}$

- Most AKE protocols are not privacy-preserving.
- For example, the well-know Signed-DH AKE is not PPAKE
- Anonymity: No other user can identify which two people are communicating

The signature leaks the identity of both initiator and responder.

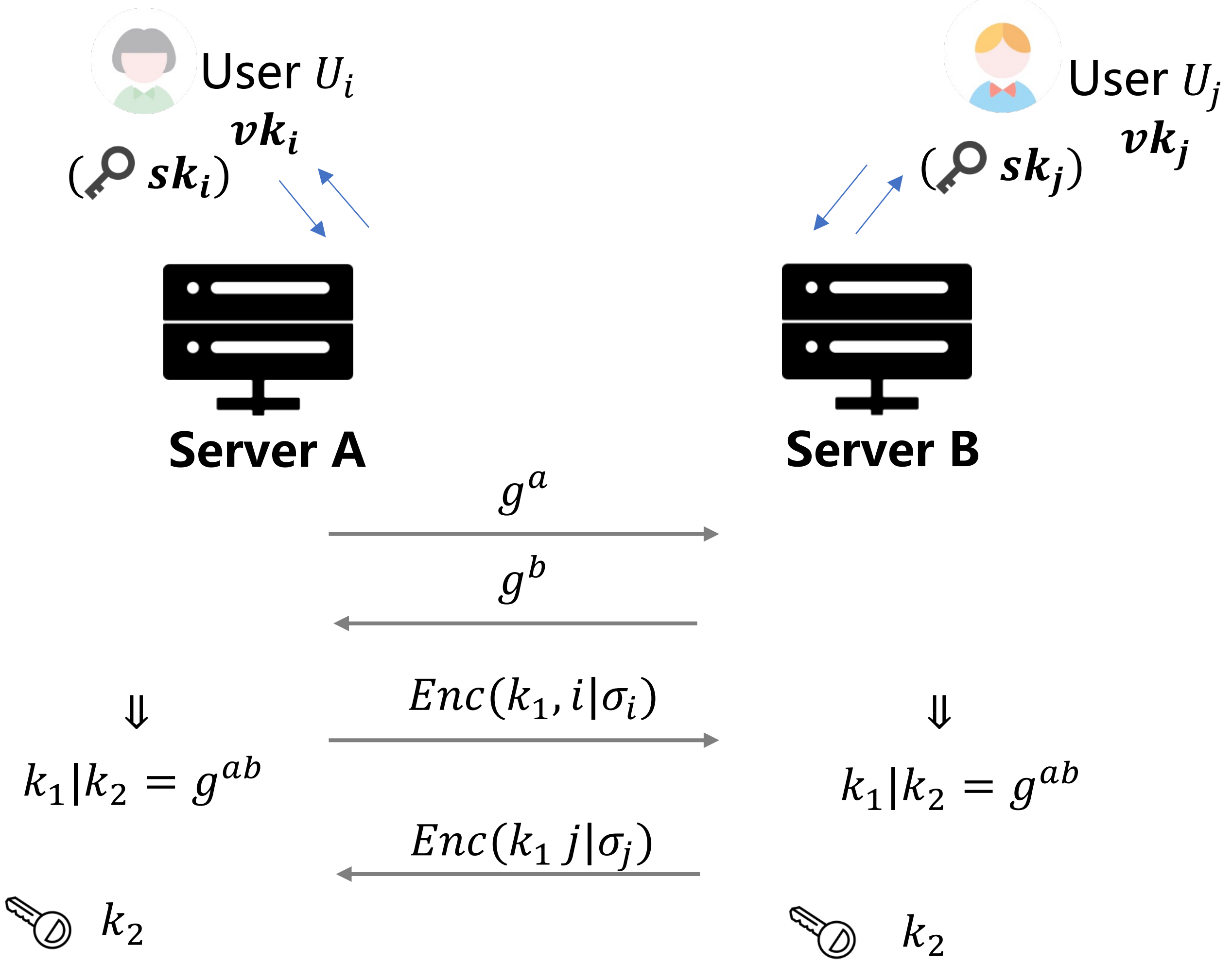
Previous work on PPAKE: SSL-PPAKE



- [SSL20] proposed a way to protect the identity of users with PPAKE.
- It considers the Server-to-Server scenario.
- Many users sit behind some servers. But The adversary does NOT control channel of user-to-server
- Anonymity requires the adversary cannot distinguish which user sits behind the server.

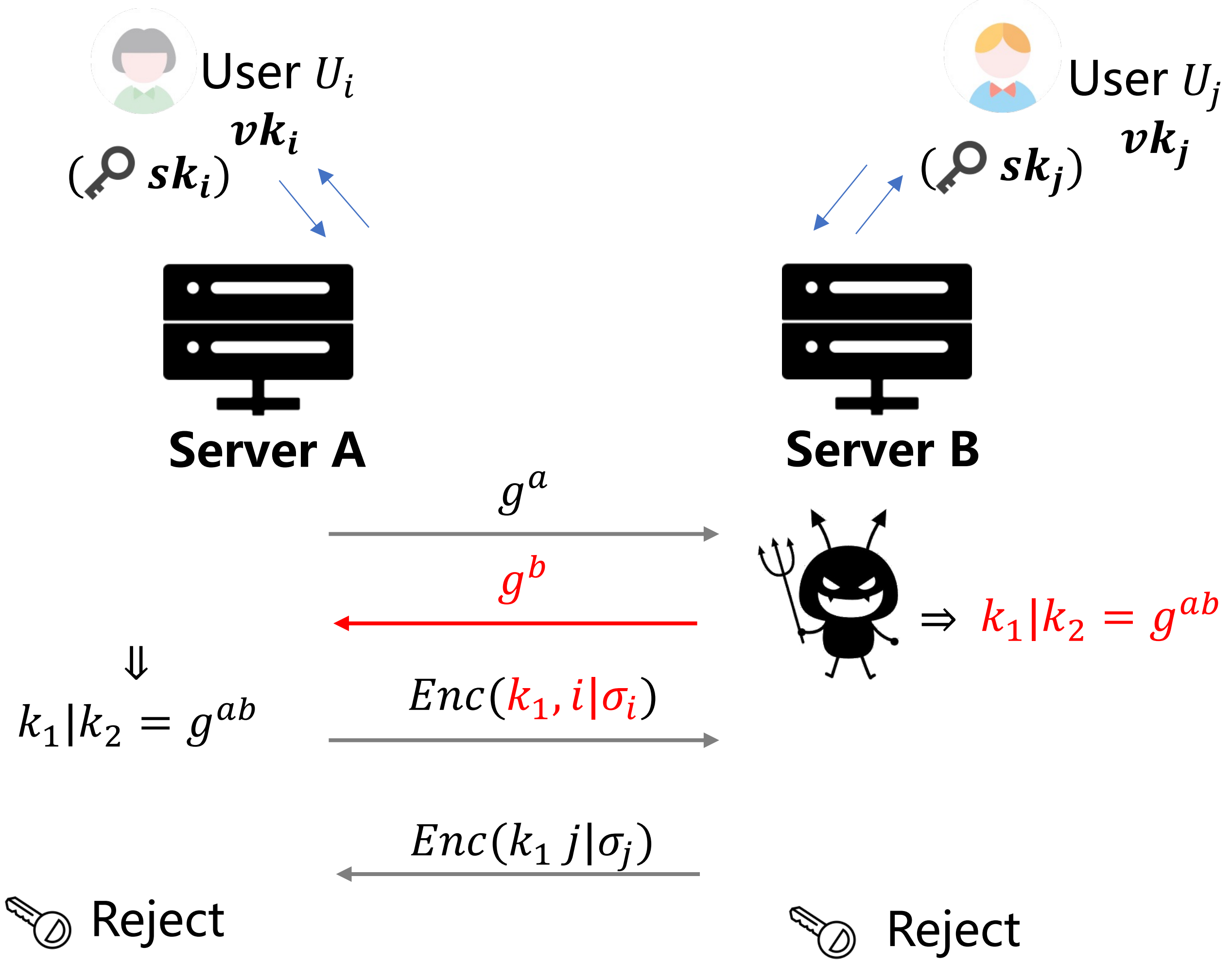
Privacy-Preserving Authenticated Key Exchange and the Case of IKEv2.
Sven Schäge, Jörg Schwenk, Sebastian Lauer PKC 2020

Previous work: SSL-PPAKE



- Both users first do an anonymous DH key exchange to get an ephemeral key.
- Then they use the ephemeral key to encrypt the signature to hide their identity.

Previous work: SSL-PPAKE

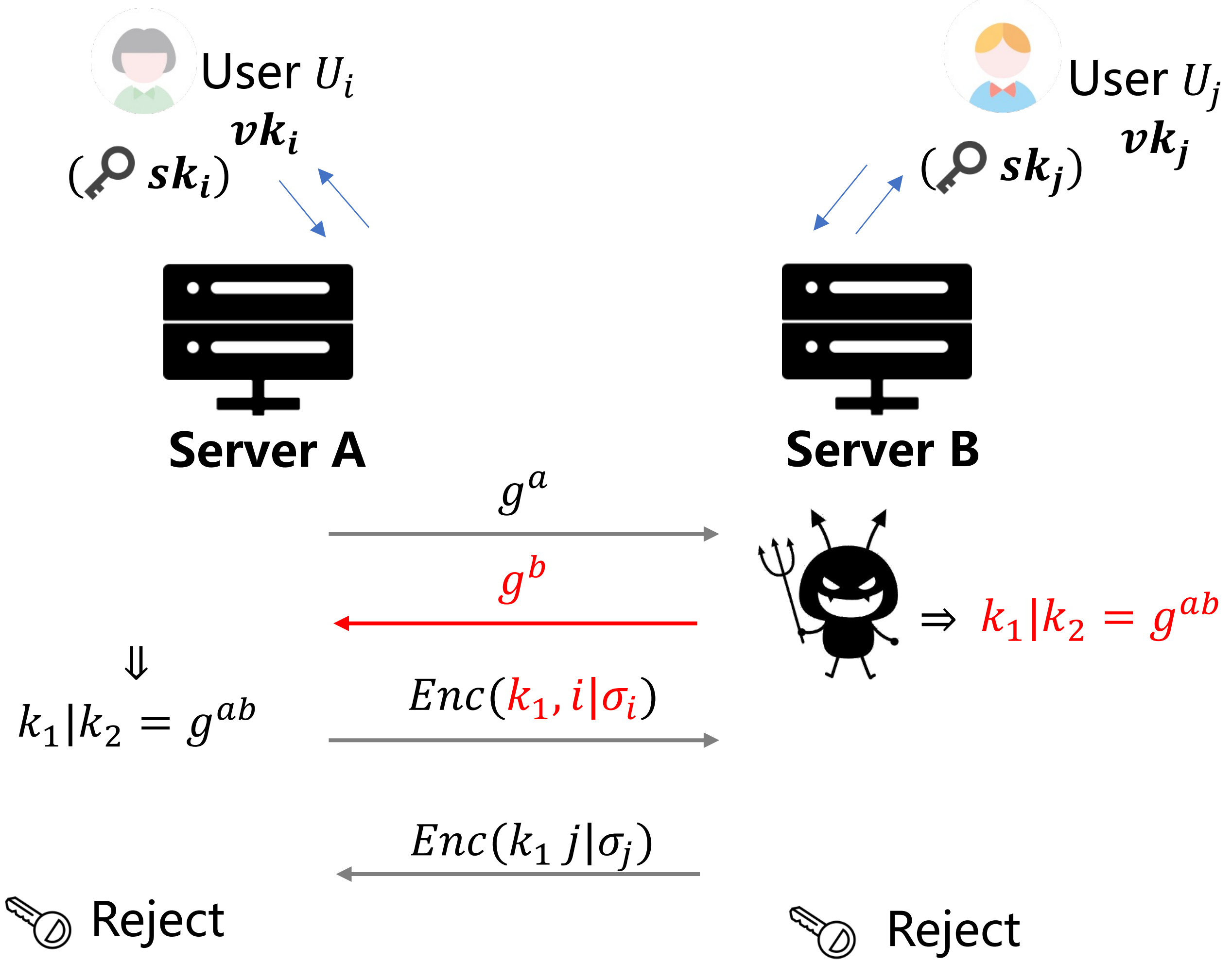


- Due to the lack of authenticity in the first two rounds, it suffers an active attack.

  An adversary can send the second message to get the identity of the initiator.

- It considers the server-to-server scenario (e.g. network protocol)
- It does not apply to the user-to-user scenario (e.g. WLAN, Bluetooth, Apple Airdrop)， which we will discuss later.

Previous work: SSL-PPAKE



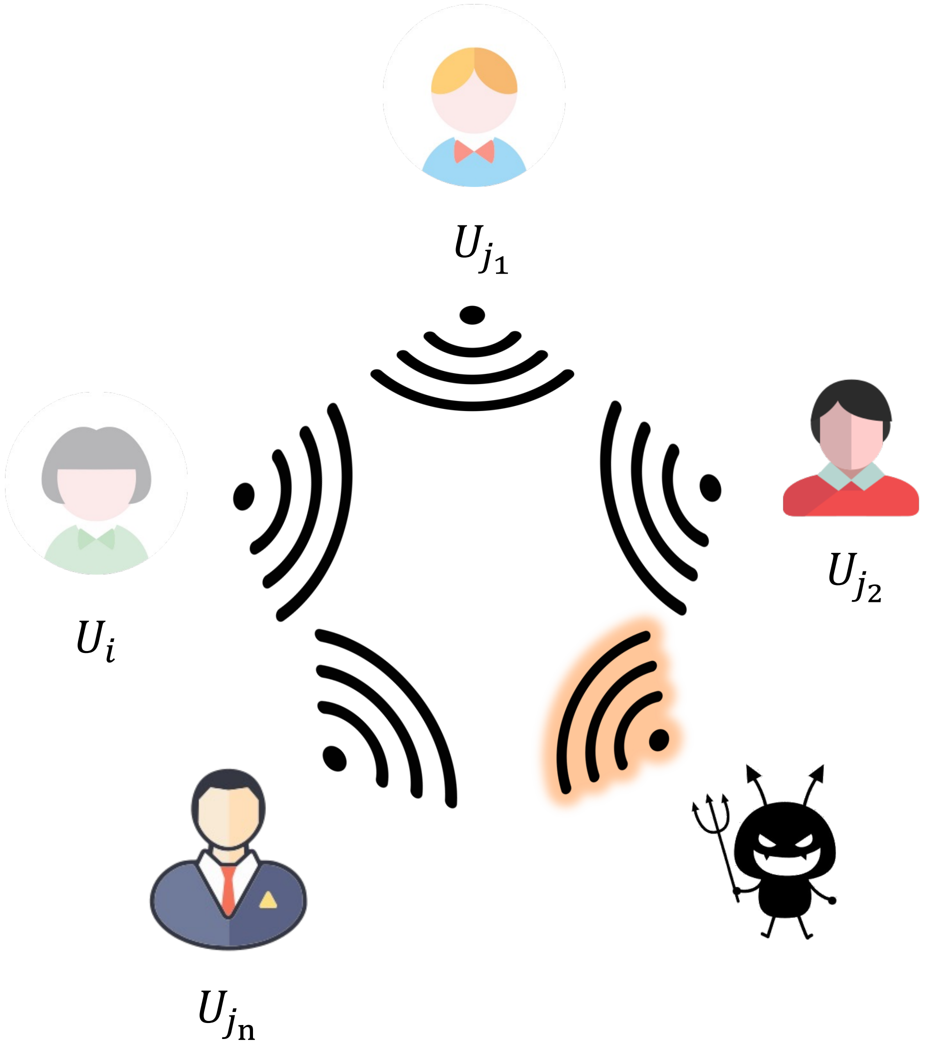
- Due to the lack of authenticity in the first two rounds, it suffers an active attack.

  An adversary can send the second message to get the identity of the initiator.

- It considers the server-to-server scenario (e.g. network protocol)
- It does not apply to the user-to-user scenario (e.g. WLAN, Bluetooth, Apple Airdrop).

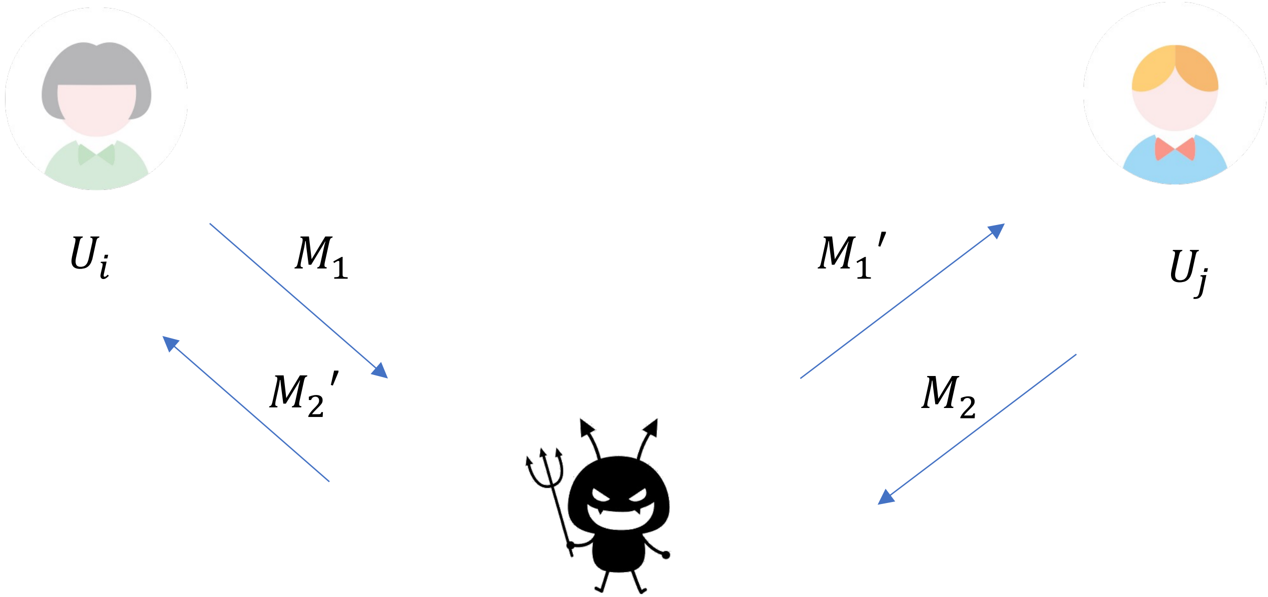
Question: How to design a PPAKE protocol for the user-to-user scenario?

The user-to-user scenario



- In the user-to-user scenario, there are no agent servers.
- We consider the broadcast channel (just like the scenario of Bluetooth, WLAN, and Airdrop).
- The adversary can see the message in the broadcast channel and involve in the communication of users.

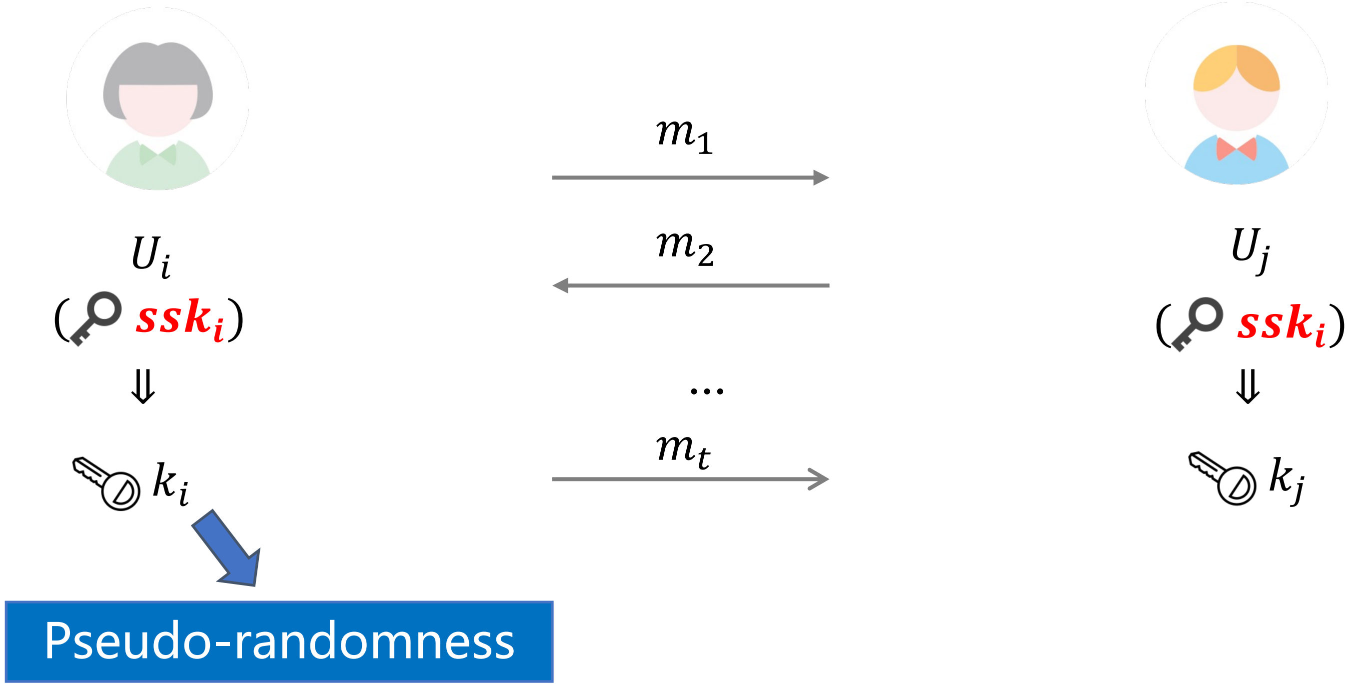
(Explicit) Authentication of PPAKE



Explicit Authentication: Active attack can be identified. For each accepted user U_i, there is a unique partner U_j such that the output of U_j is the input of U_i, and The output of U_i is the input of U_j

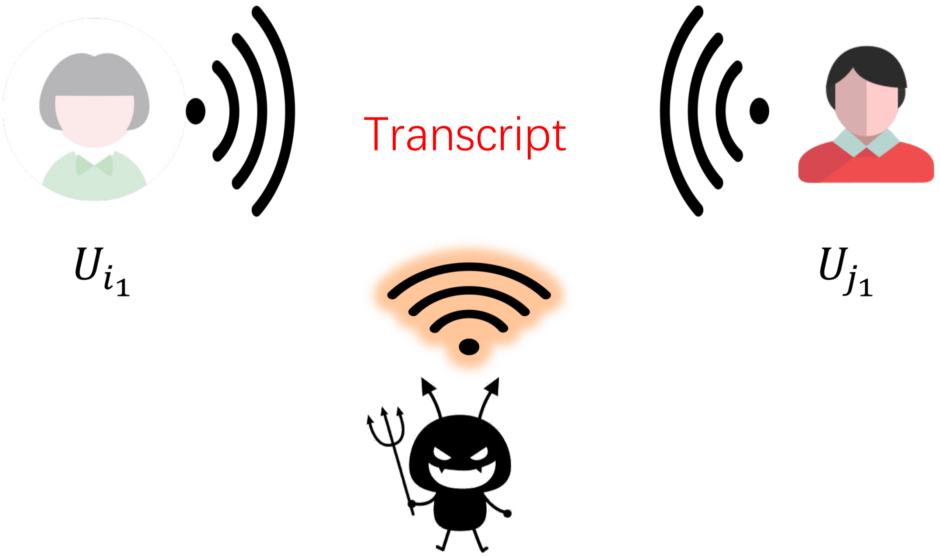
$$M_1 = M_1' \text{ and } M_2 = M_2'$$

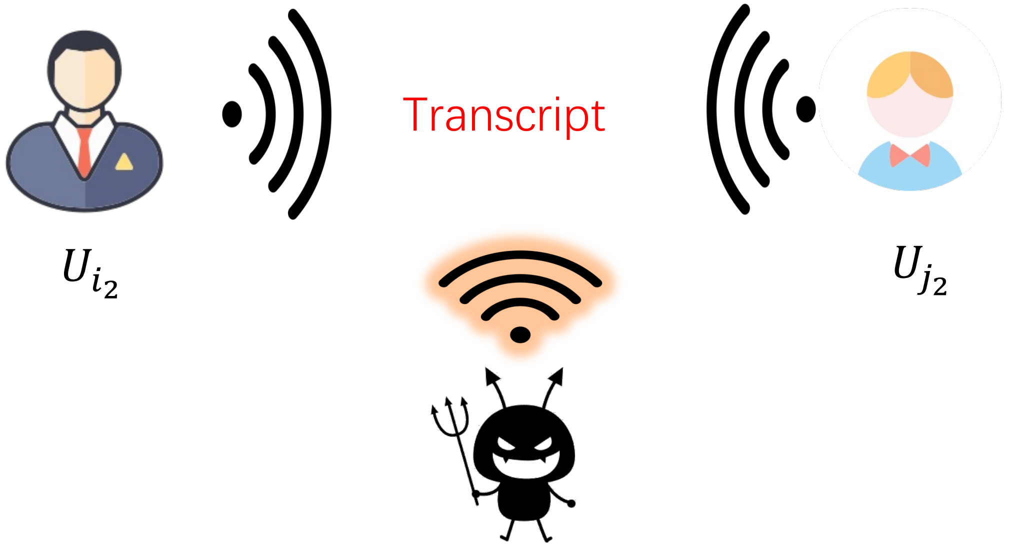
(Forward) Security for Session Key



Forward Security for Session Key: The session key is pseudo-random if there are no active attacks, even if the long-term key of users are leaked to the adversary.

Anonymity for User Identity

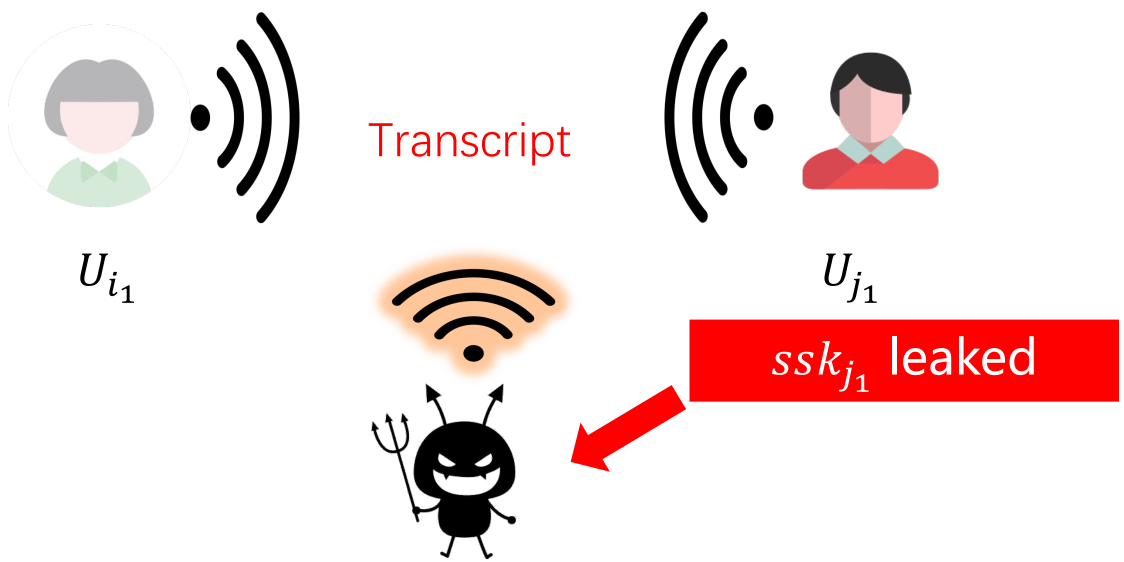




Anonymity: given the transcript, the adversary can not distinguish which two users are communicating

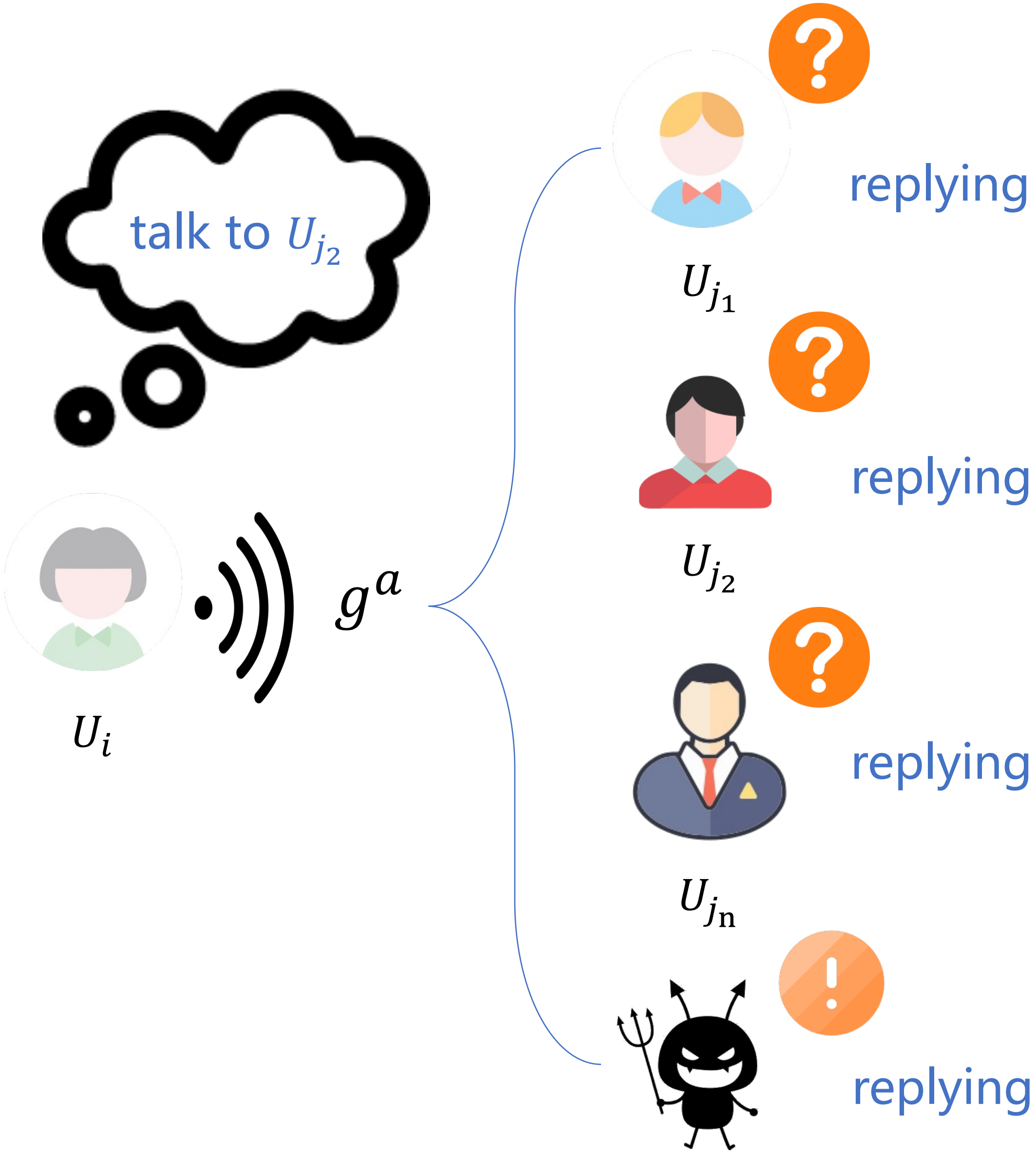
Forward Anonymity: anonymity holds even if the adversary can also corrupt these users (get their long-term keys) after it gets the transcript.

Forward Anonymity



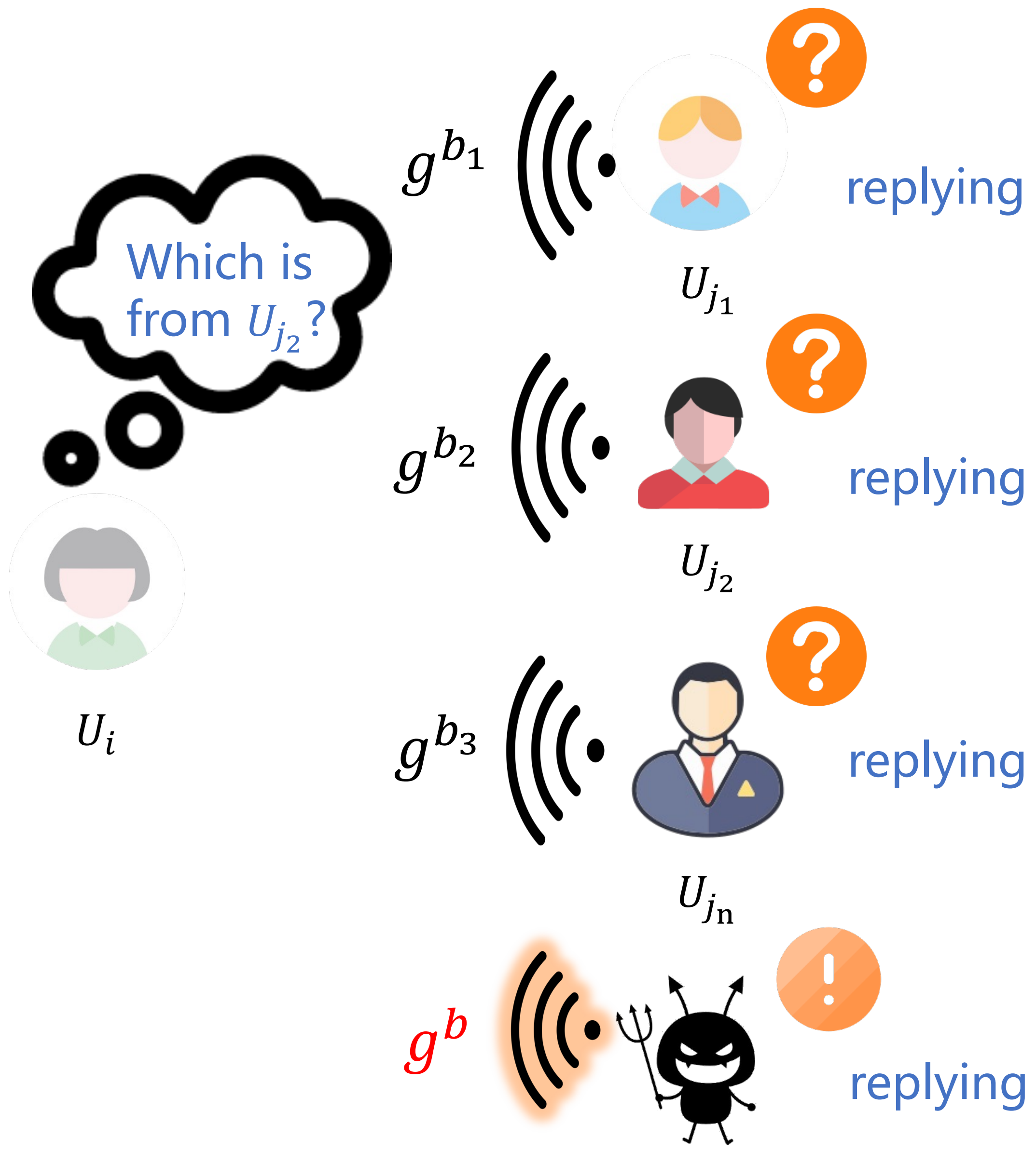
- Suppose the adversary gets the long-term key of user U_{j_2}
- Based on the previous transcript, adversary can not determine whether user U_{j_2} was involved in the previous communication.

SSL-PPAKE for user-to-user setting



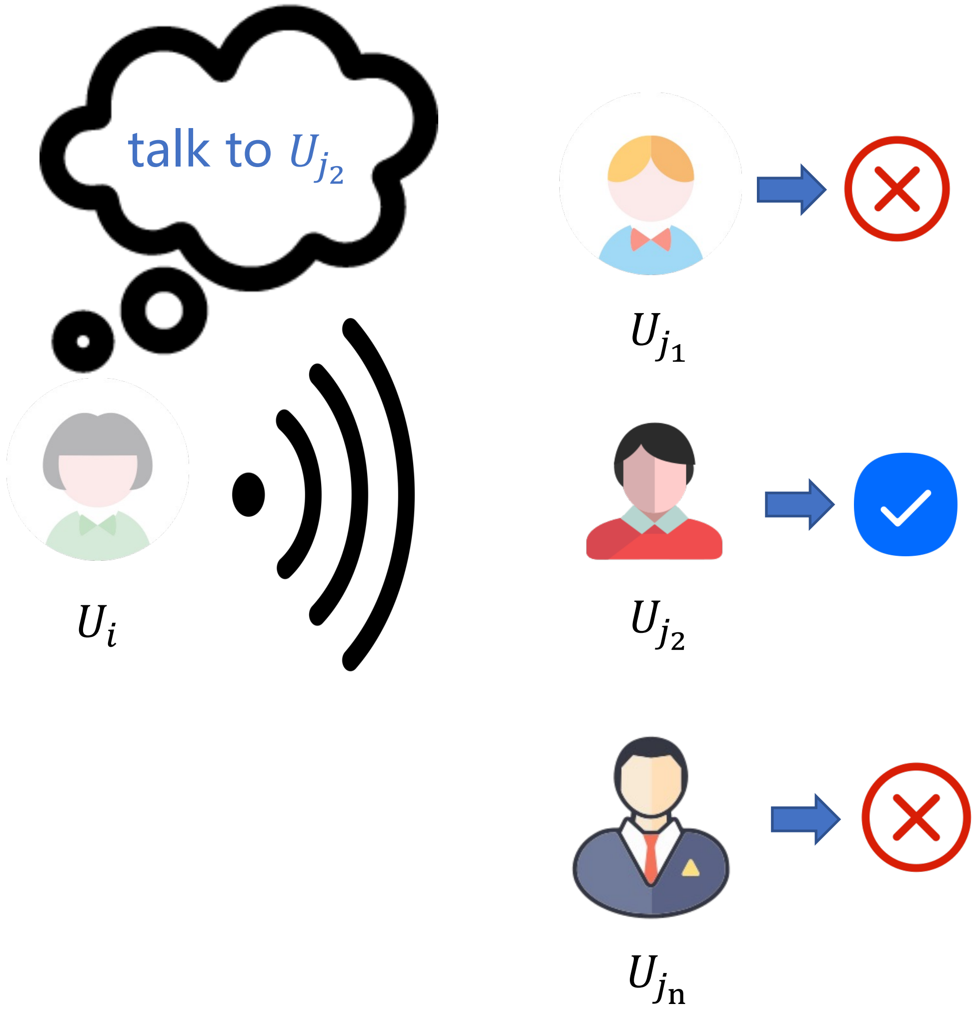
- To protect the identity of U_{j_2}, the first message does not contain any information of its target recipient.
- each user is not sure whether itself is the target recipient.
- Hence, each user must reply with a g^b

SSL-PPAKE for user-to-user setting



- To protect the identity of U_{j_2}, the first message does not contain any information of its target recipient.
- each user is not sure whether itself is the target recipient.
- Hence, each user must reply with a g^b
- This will cause large communication and computation complexity.
- Moreover, the adversary can always determine the initiator's identity.

Our Approach: Making PPAKE Robust



- Robust PPAKE: any user except the target recipient will output $\perp$ when they receive the first round message of PPAKE.
- In other words, each user is able to ascertain that the message in the first round is for him/her.
- Due to the robustness, the communication and computation complexity can be reduced.

Contents

Building Blocks of Our Construction

We propose a generic construction of PPAKE from the following building blocks:

- Signature Scheme (Sign, Verify)
- Key Encapsulation Mechanism (Encap, Decap)
- Message Authentication Code (MAC, Verify)
- Symmetric Encryption (Enc, Dec)

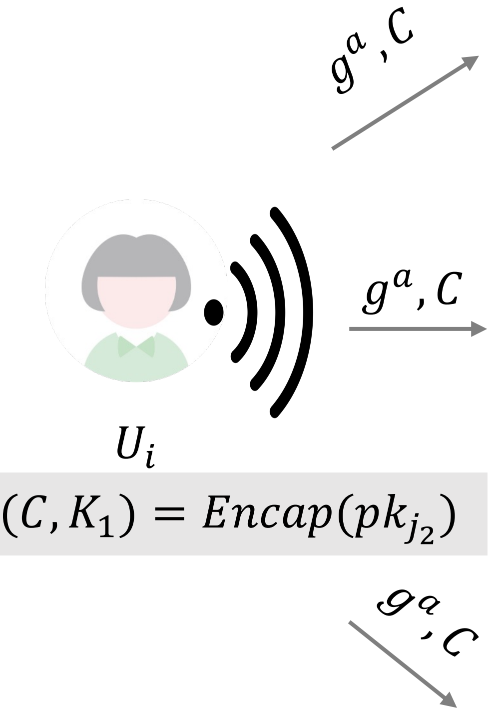
The Requirements for KEM

Key Encapsulation Mechanism (KEM):

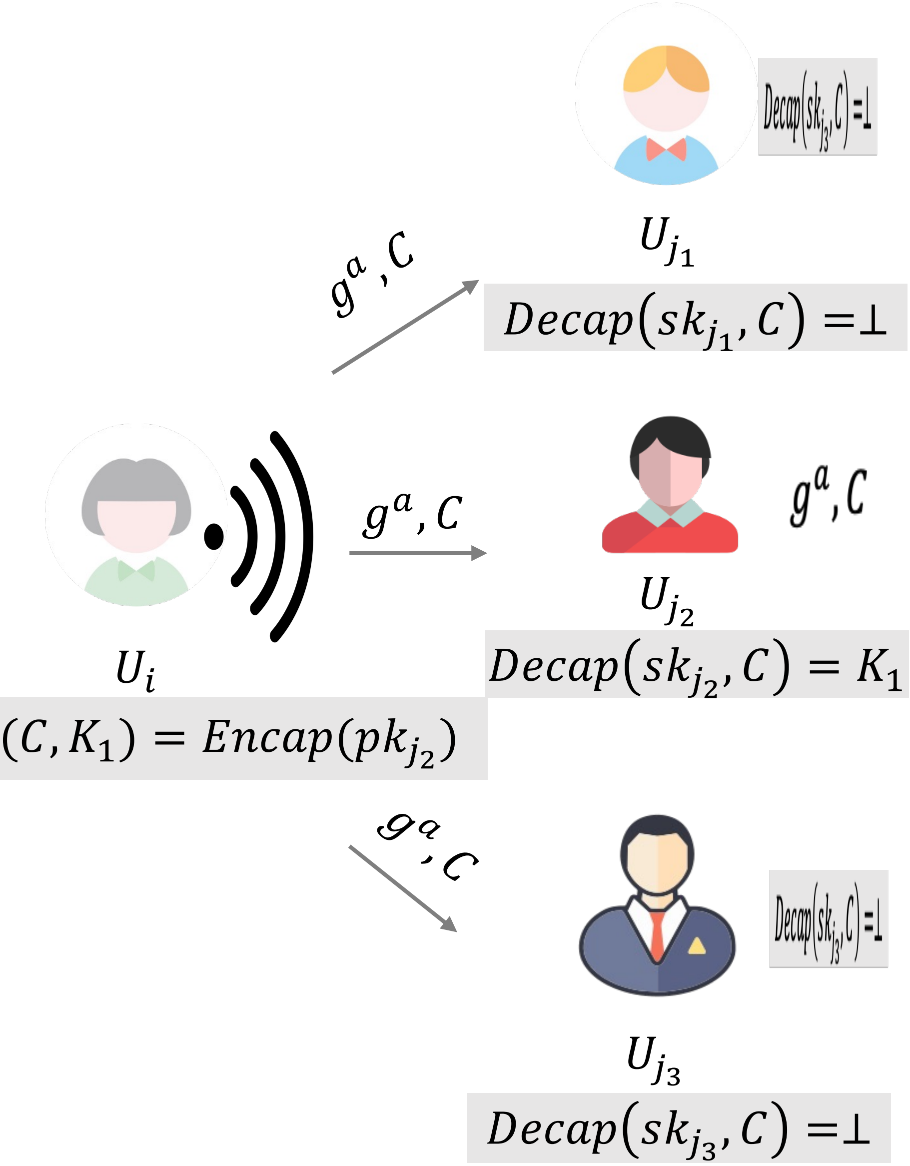
- $\mathrm{KEM.Gen}(1^\lambda; r) \to (pk, sk)$
- $\mathrm{KEM.Encap}(pk; r) \to (C, K)$
- $\mathrm{KEM.Decap}(sk, C) \to K'$

- Anonymity: the adversary cannot distinguish whether the ciphertext C is generated by pk_0 or pk_1
- Robustness: If a ciphertext is generated by pk_b, then $\mathrm{Decap}(sk_{1-b}, C) = \perp$ with overwhelming probability.

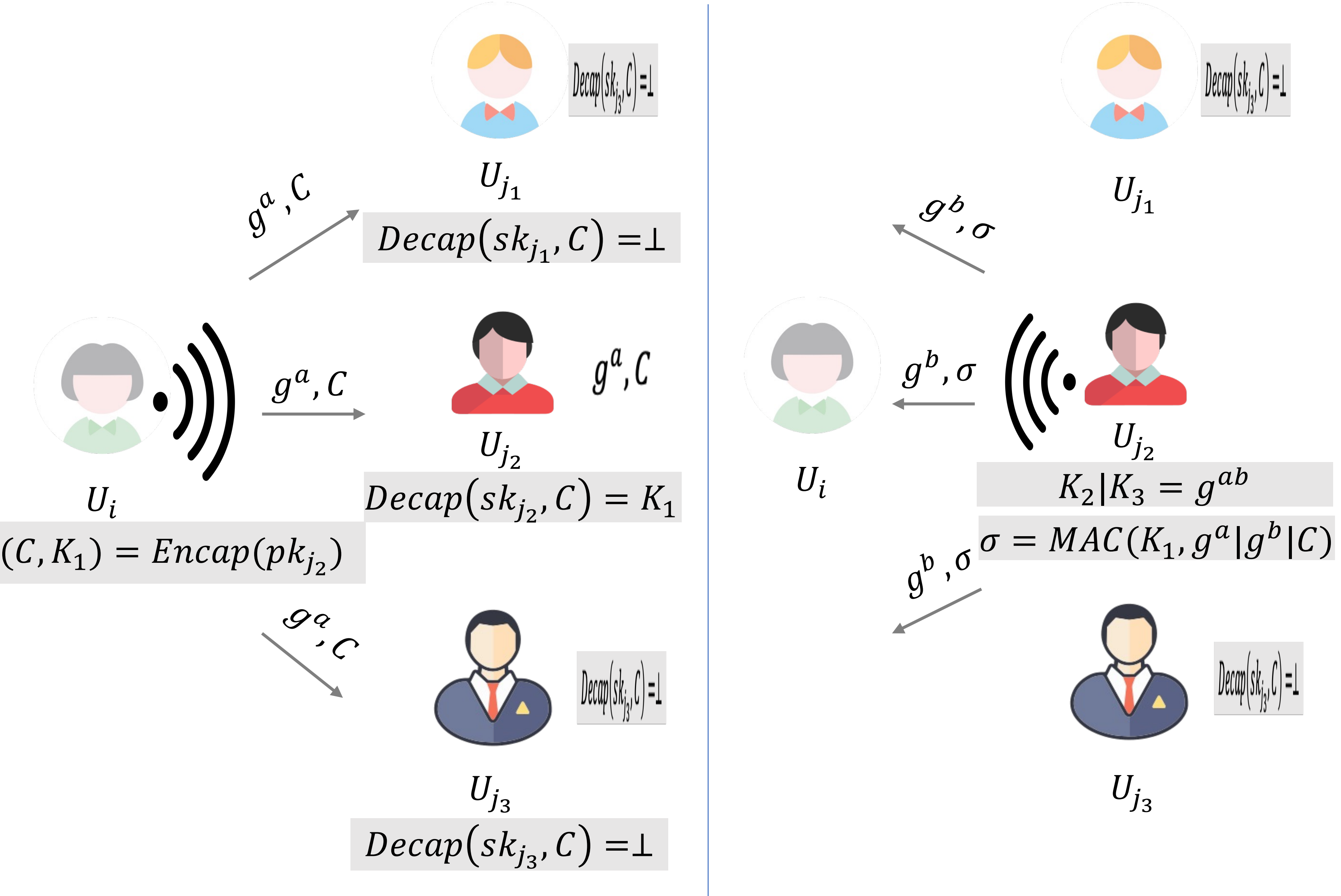
Our Construction: The First Round



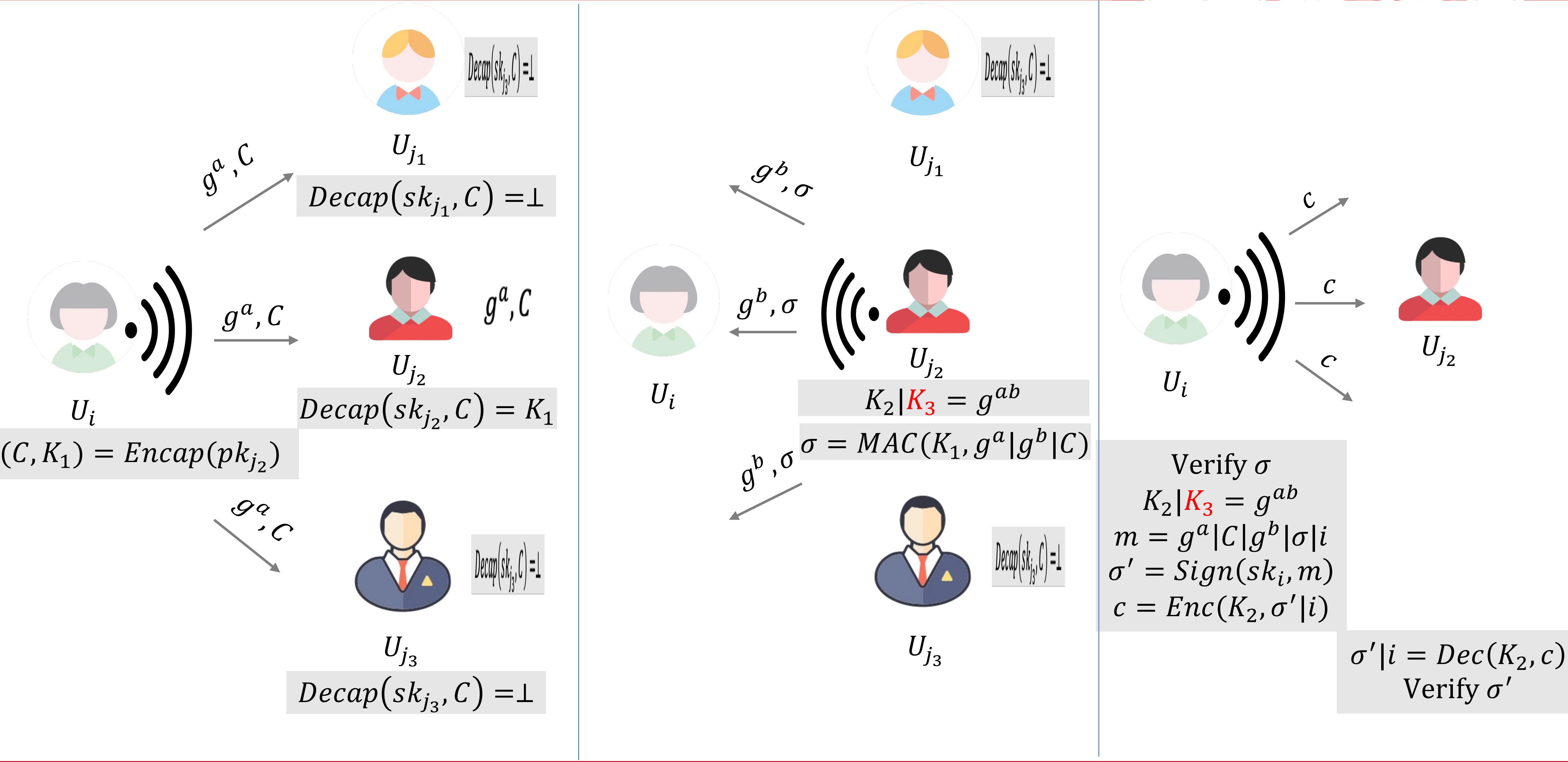
Our Construction: The First Round



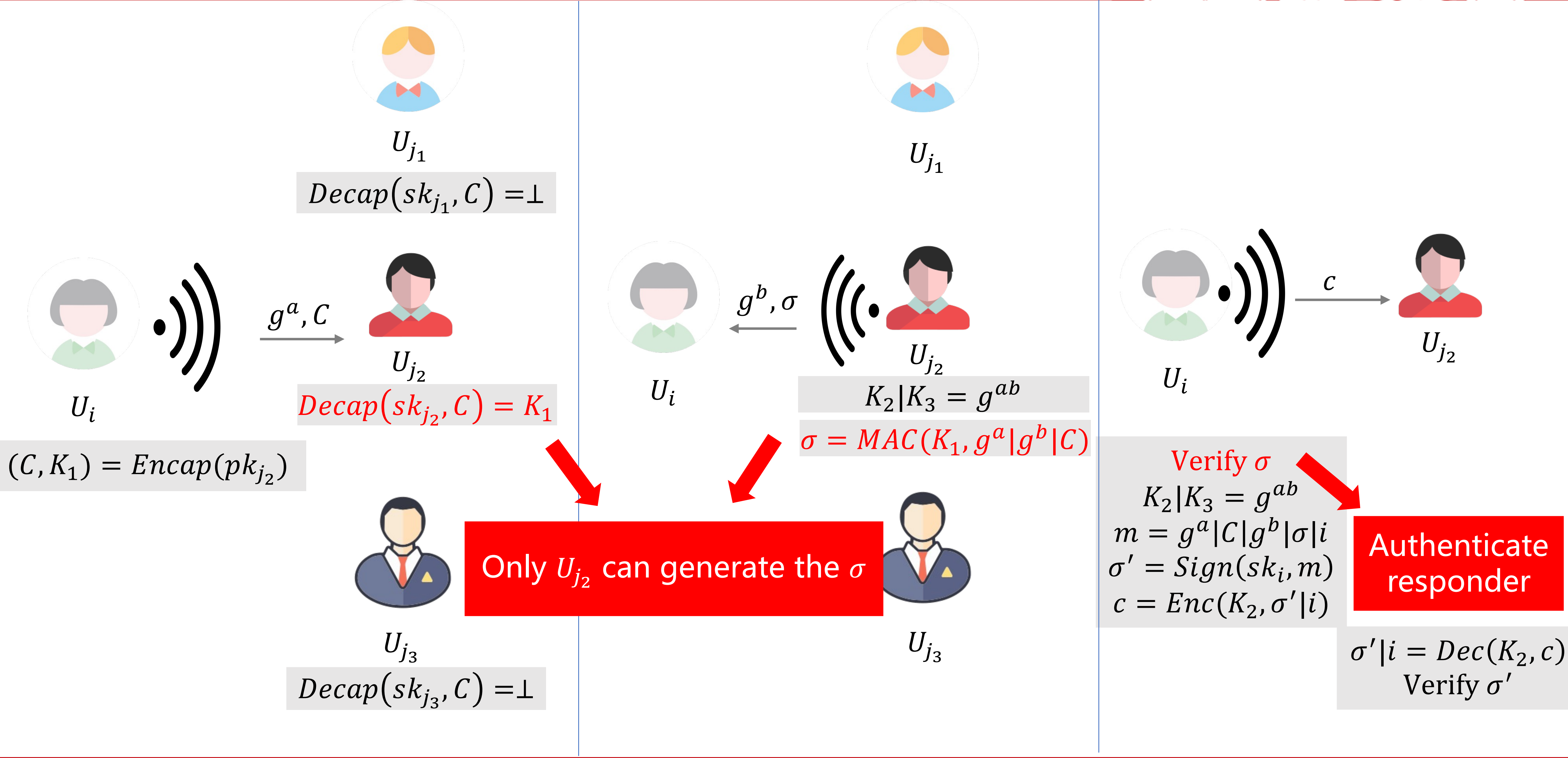
Our Construction: The Second Round



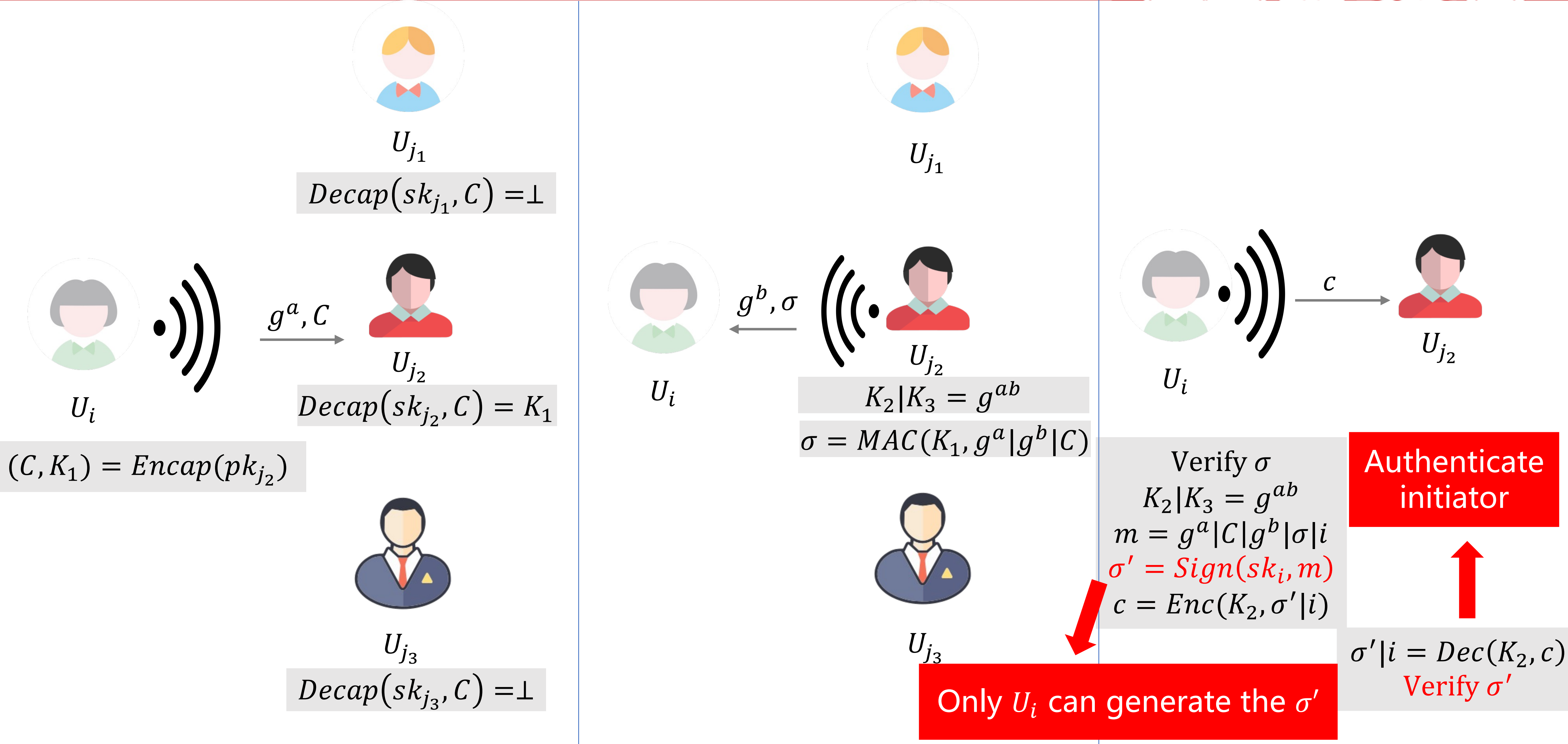
Our Construction: The Third Round



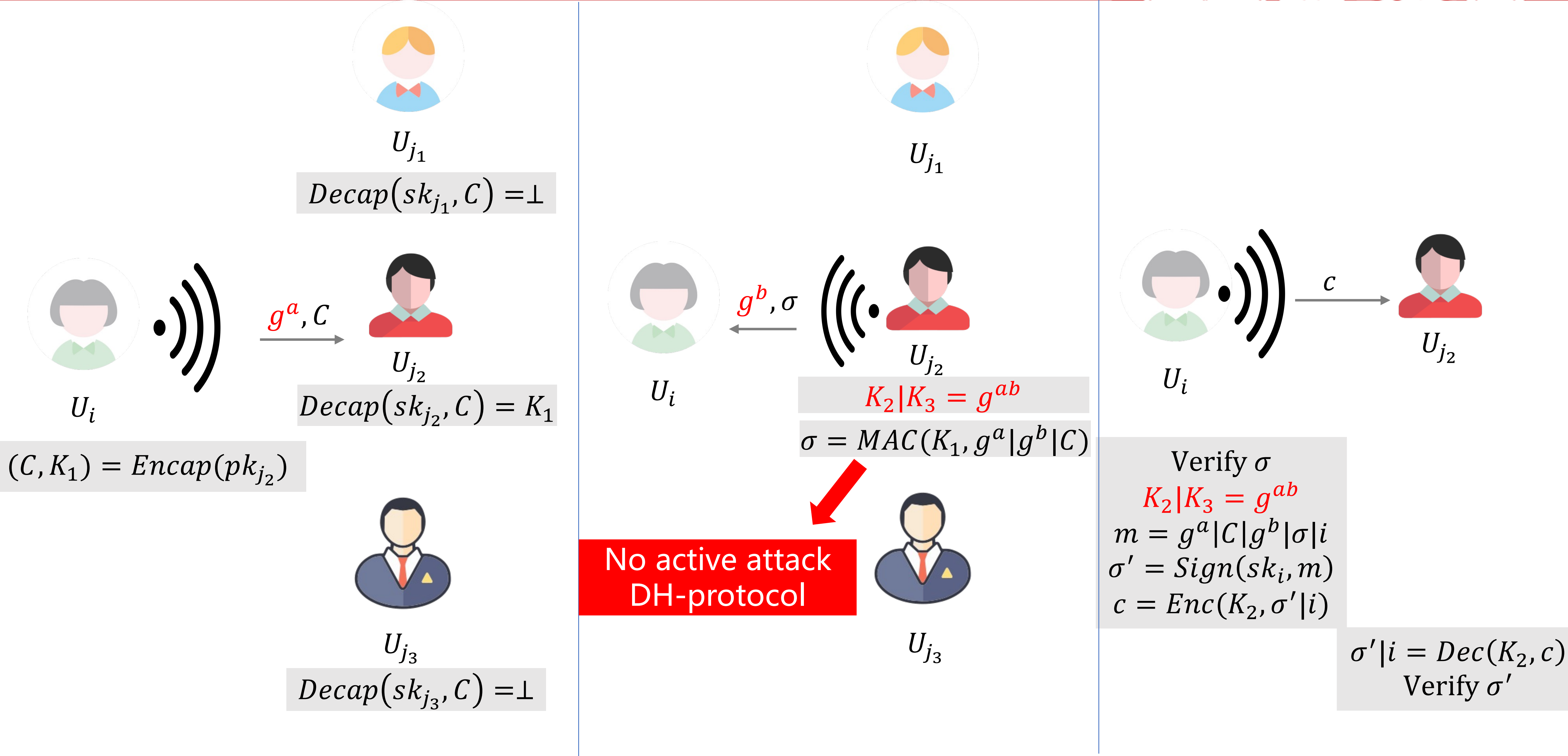
Authentication



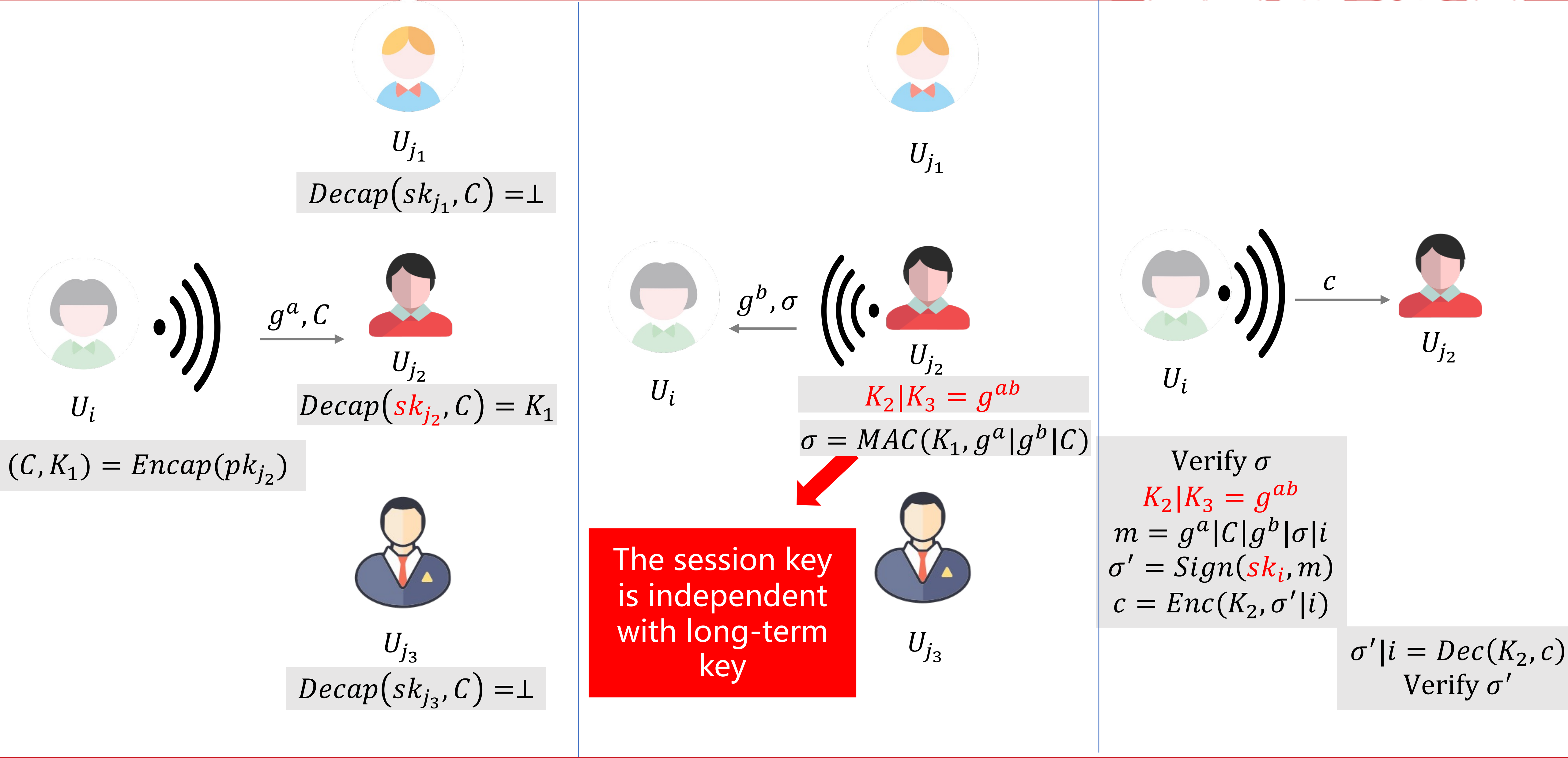
Authentication



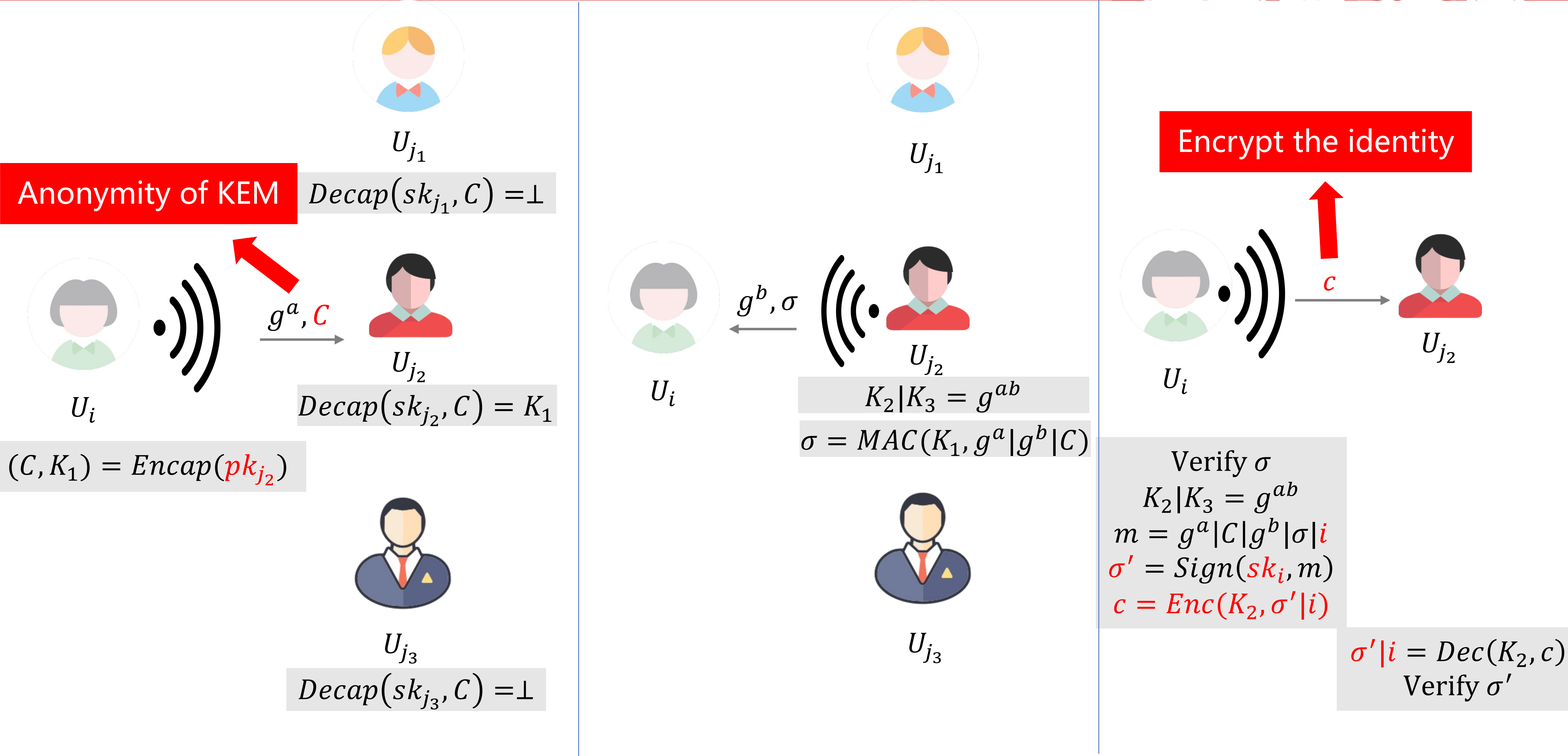
Forward security (Key Pseudo-Randomness)



Forward security (Key Pseudo-Randomness)

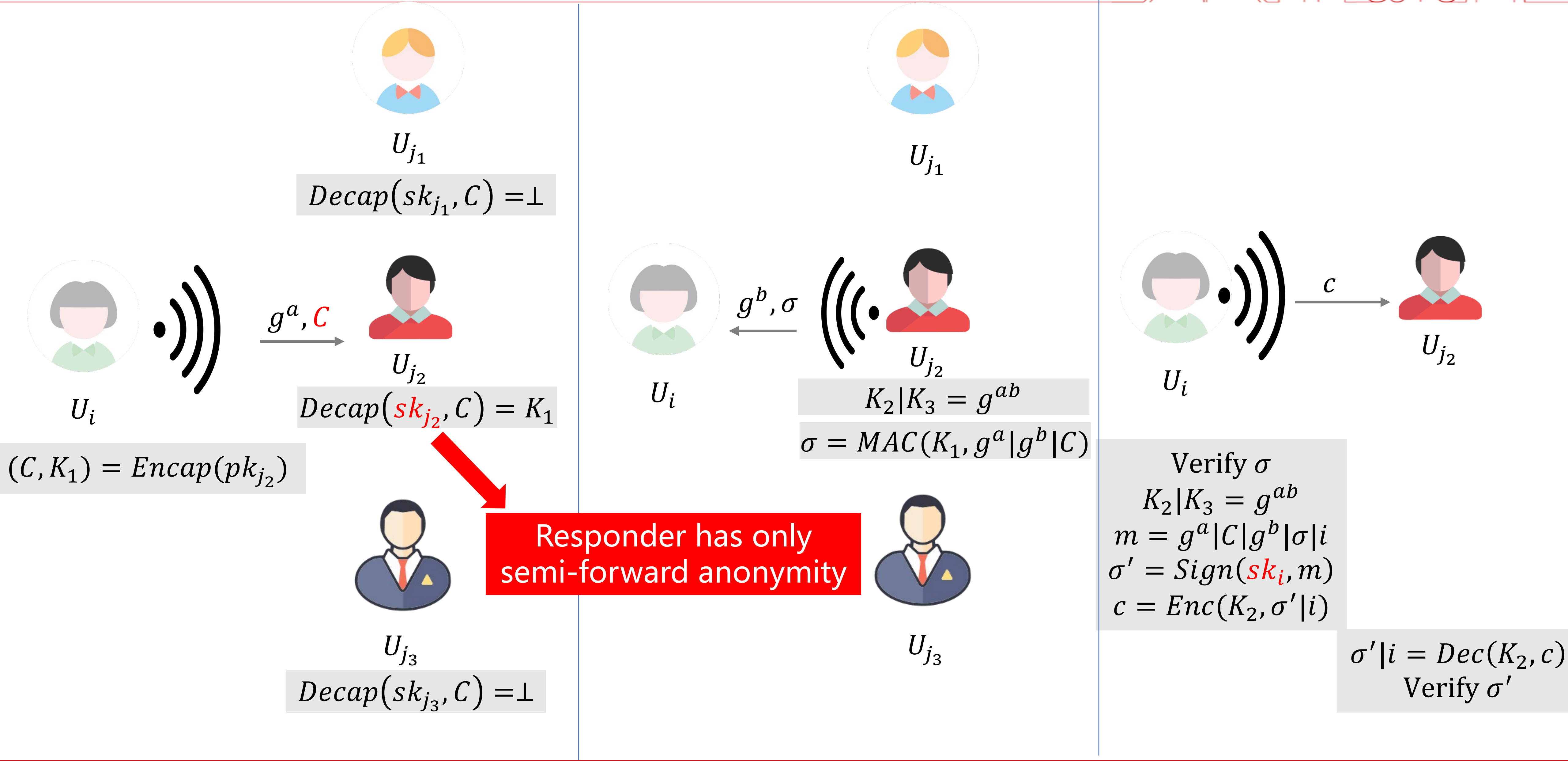


Anonymity



U_{j_1}
Anonymity of KEM
$Decap(sk_{j_1}, C) = \perp$
g^a, C
U_i
U_{j_2}
$Decap(sk_{j_2}, C) = K_1$
$(C, K_1) = Encap(pk_{j_2})$
U_{j_3}
$Decap(sk_{j_3}, C) = \perp$
U_{j_1}
g^b, σ
U_i
U_{j_2}
$K_2|K_3 = g^{ab}$
$\sigma = MAC(K_1, g^a|g^b|C)$
U_{j_3}
Encrypt the identity
c
U_i
U_{j_2}
Verify σ
$K_2|K_3 = g^{ab}$
$m = g^a|C|g^b|\sigma|i$
$\sigma' = Sign(sk_i, m)$
$c = Enc(K_2, \sigma'|i)$
$\sigma'|i = Dec(K_2, c)$
Verify σ'

Semi-forward Anonymity for Responder



Contents

Comparison

PPAKE schemes	Comm	Comp	#	Forward Security	Anonymity		Forward Anonymity CrpI		Forward Anonymity CrpR		Mutual Auth	Std
					I	R	I	R	I	R		
IY22	6	$O(1)$	2	weak	✓	×	✓	×	✓	×	×	✓
SKEME96	16	$O(1)$	3	✓	✓	✓	×	×	×	×	✓	×
SSL20	5μ	$O(\mu)$	4	✓	×	✓	×	✓	×	✓	✓	✓
RSW21	$7\mu - 5$	$O(\mu)$	4	✓	✓	✓	✓	✓	×	×	✓	×
Ours	12	$O(1)$	3	✓	✓	✓	✓	✓	✓	×	✓	✓

Consider all protocols in the user-to-user setting.

μ: number of users in the system.

Thanks to robustness, the computation complexity and communication complexity are independent of μ.

Conclusion

- In this paper, we propose a PPAKE scheme, especially for the broadcast channel and user-to-user setting
- We also give a concrete instantiation based on DDH assumption.
- For more information, please refer to our paper.

  https://eprint.iacr.org/2022/1217.pdf

- An interesting problem:

  Can we construct a PPAKE scheme that satisfied full-forward anonymity in our model?

Thanks! Questions?

References

[IY22] Ren Ishibashi, Kazuki Yoneyama: Post-quantum Anonymous One - Sided Authenticated Key Exchange Without Random Oracles. Public Key Cryptography (2) 2022: 35-65

[SKEME96] Hugo Krawczyk: SKEME: a versatile secure key exchange mechanism for Internet. NDSS 1996: 114-127

[SSL20] Sven Schäge, Jörg Schwenk, Sebastian Lauer: Privacy-Preserving Authenticated Key Exchange and the Case of IKEv2. Public Key Cryptography (2) 2020: 567-596

[RSW21] Sebastian Ramacher, Daniel Slamanig, Andreas Weninger: Privacy-Preserving Authenticated Key Exchange: Stronger Privacy and Generic Constructions. ESORICS (2) 2021: 676-696