Cryptography Engineering

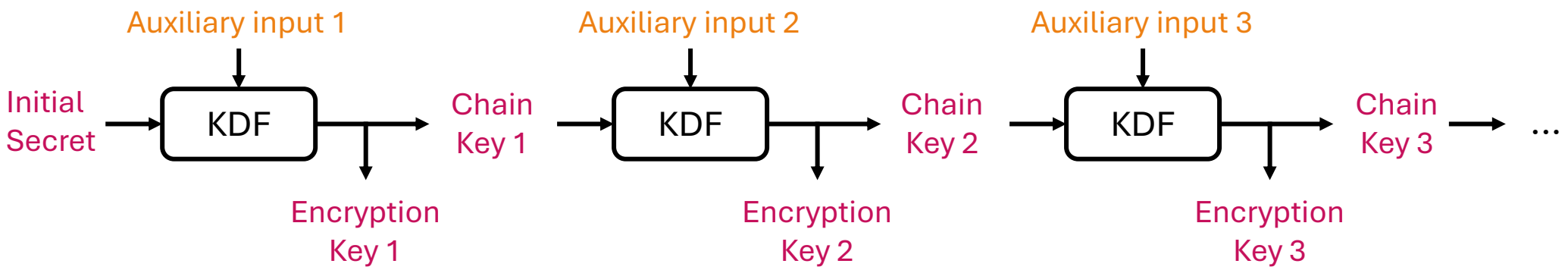
- Lecture 5 (Nov 20, 2024)

- Today's notes:
  - Symmetric-key Ratchet (Continue)
  - Forward/Backward Secrecy
  - Diffie-Hellman Ratchet

- No homework

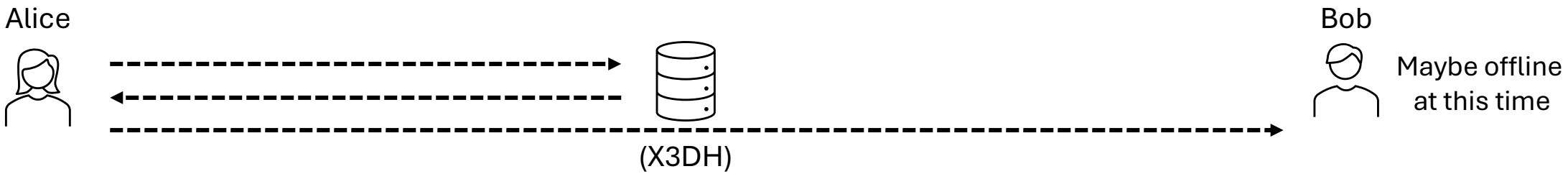
Symmetric-key Ratchet

- KDF chain
  - KDF: Key derivation function



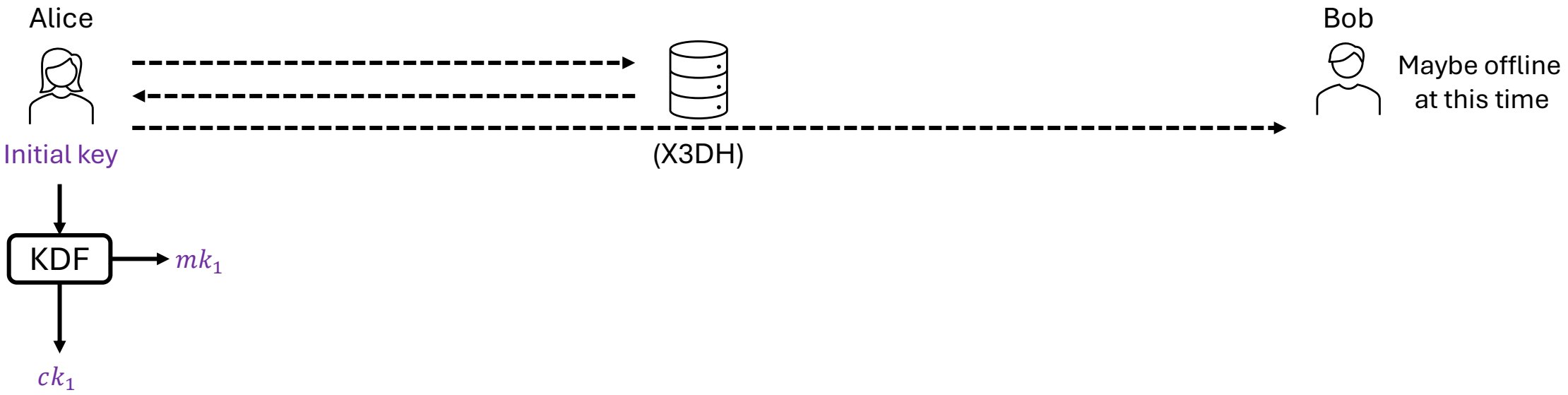
Symmetric-key Ratchet

- A toy example of instant messaging using symmetric-key ratchet



Symmetric-key Ratchet

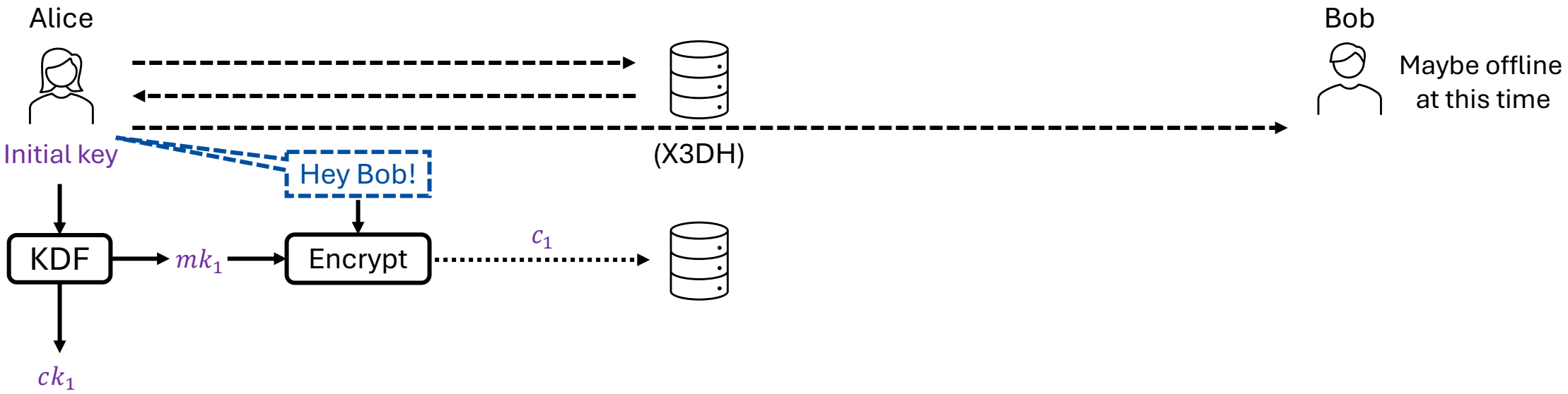
- A toy example of instant messaging using symmetric-key ratchet



*We ignore the auxiliary input to KDF

Symmetric-key Ratchet

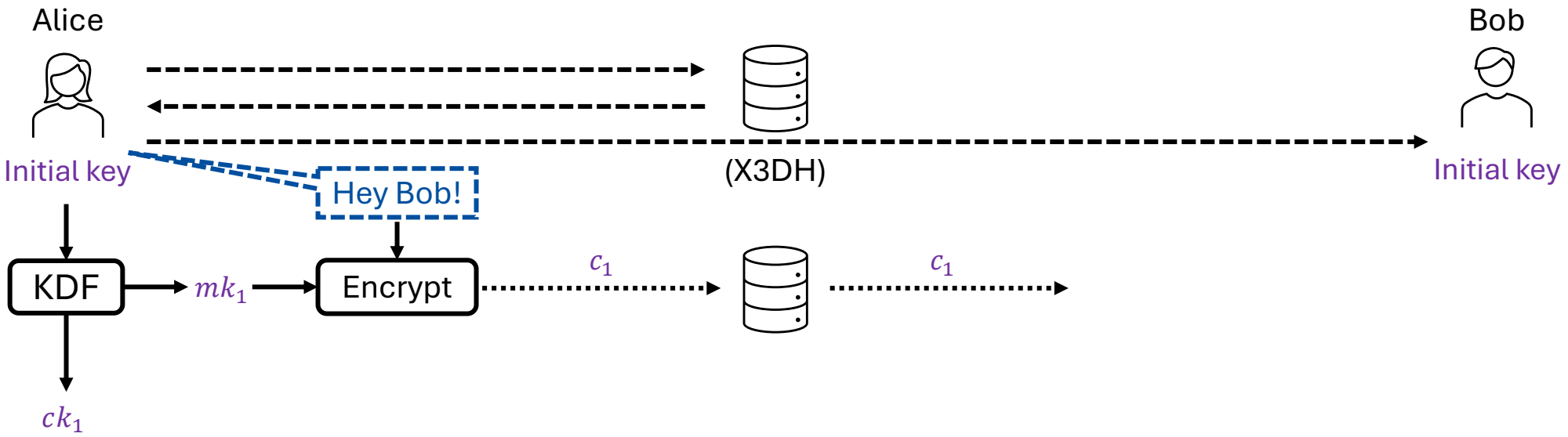
- A toy example of instant messaging using symmetric-key ratchet



*We ignore the auxiliary input to KDF

Symmetric-key Ratchet

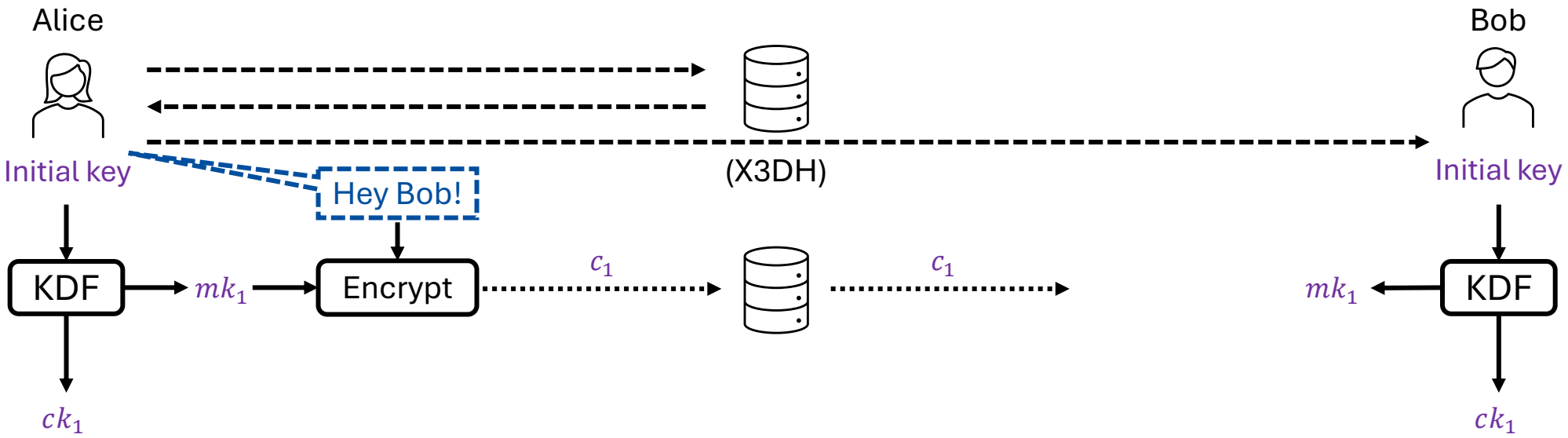
- A toy example of instant messaging using symmetric-key ratchet



*We ignore the auxiliary input to KDF

Symmetric-key Ratchet

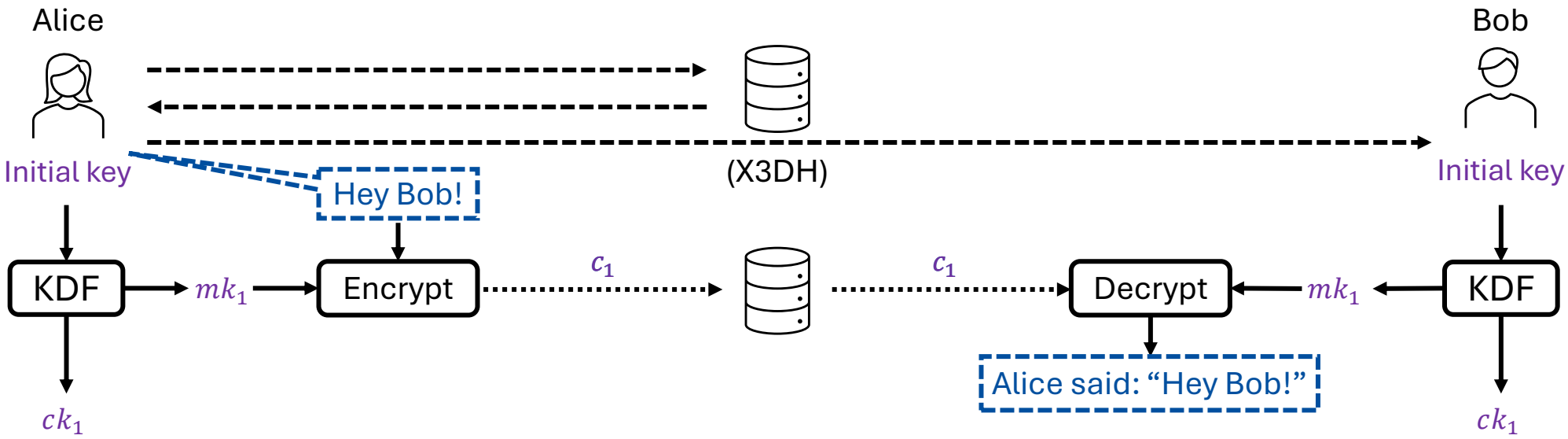
- A toy example of instant messaging using symmetric-key ratchet



*We ignore the auxiliary input to KDF

Symmetric-key Ratchet

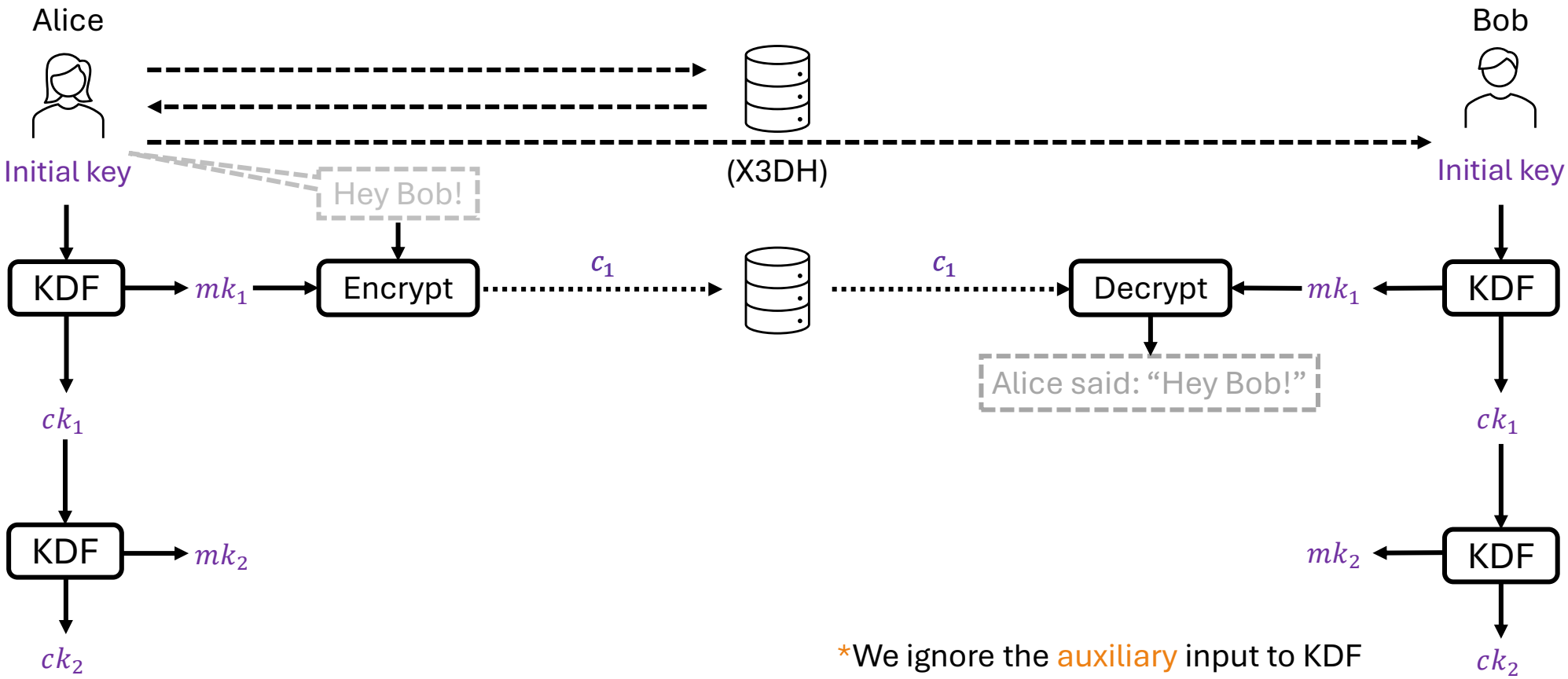
- A toy example of instant messaging using symmetric-key ratchet



*We ignore the auxiliary input to KDF

Symmetric-key Ratchet

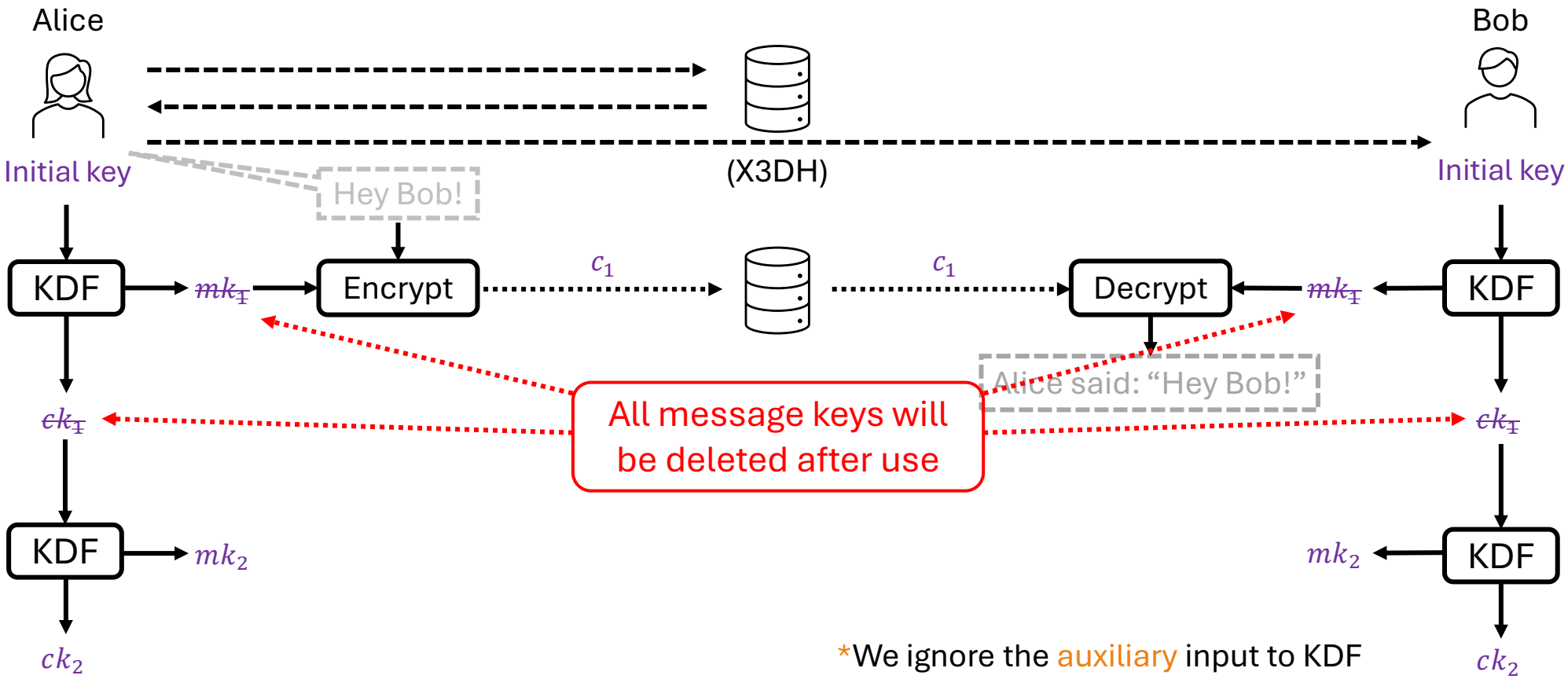
- A toy example of instant messaging using symmetric-key ratchet



*We ignore the auxiliary input to KDF

Symmetric-key Ratchet

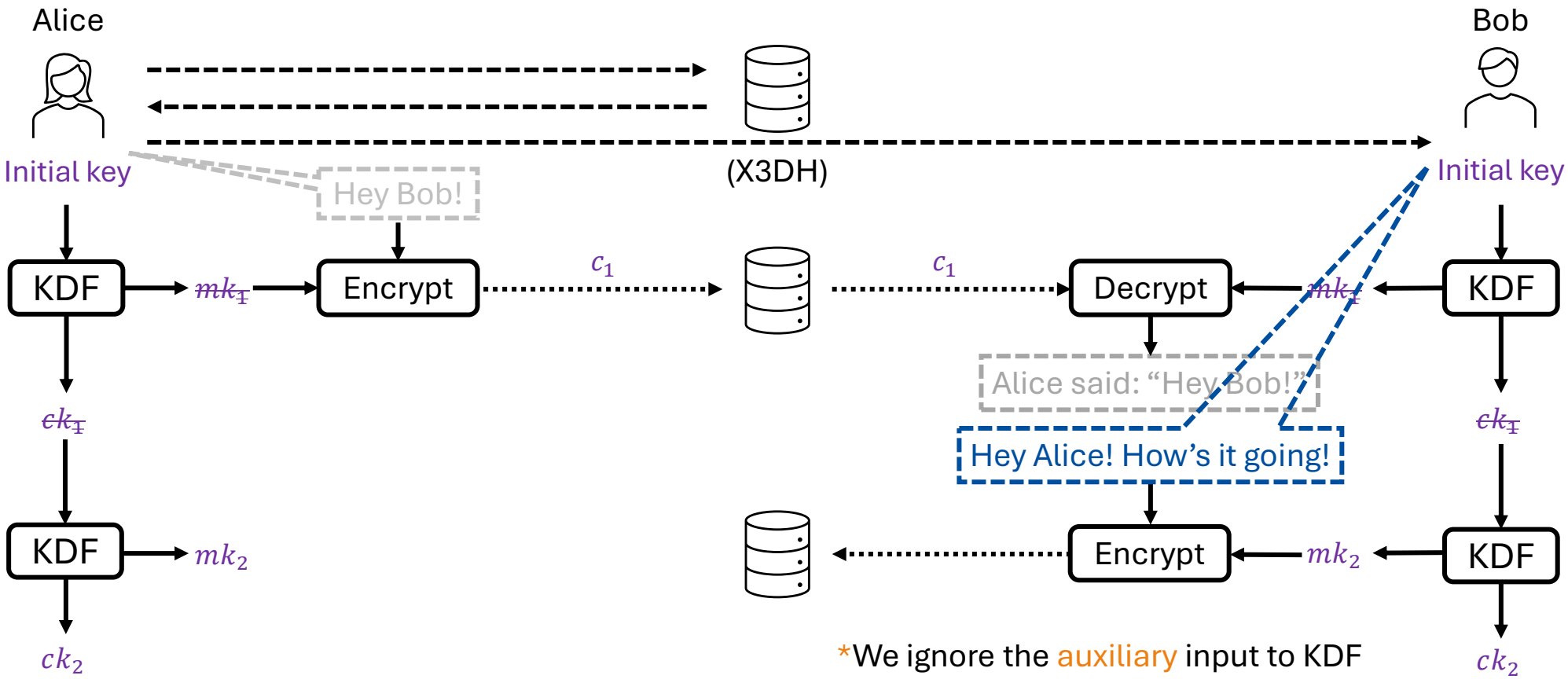
- A toy example of instant messaging using symmetric-key ratchet



*We ignore the auxiliary input to KDF

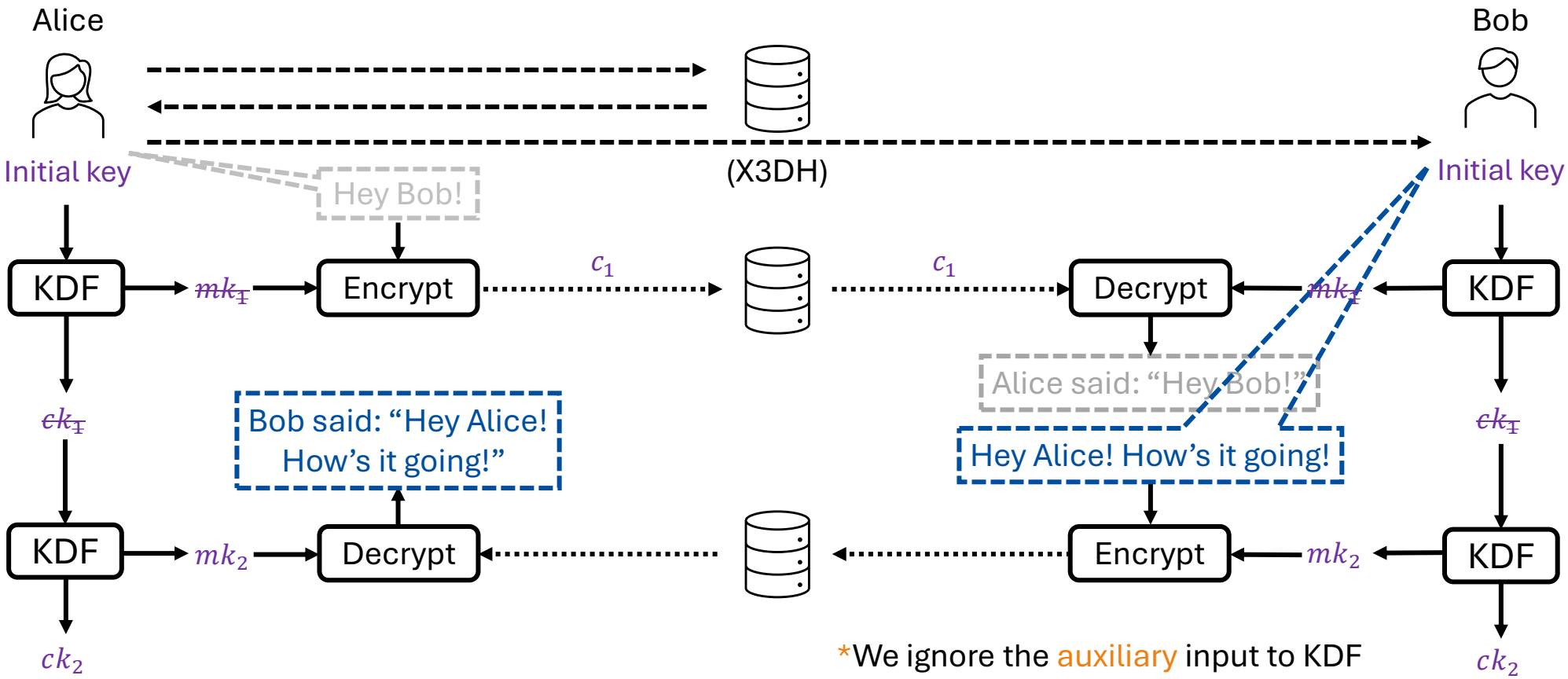
Symmetric-key Ratchet

- A toy example of instant messaging using symmetric-key ratchet



Symmetric-key Ratchet

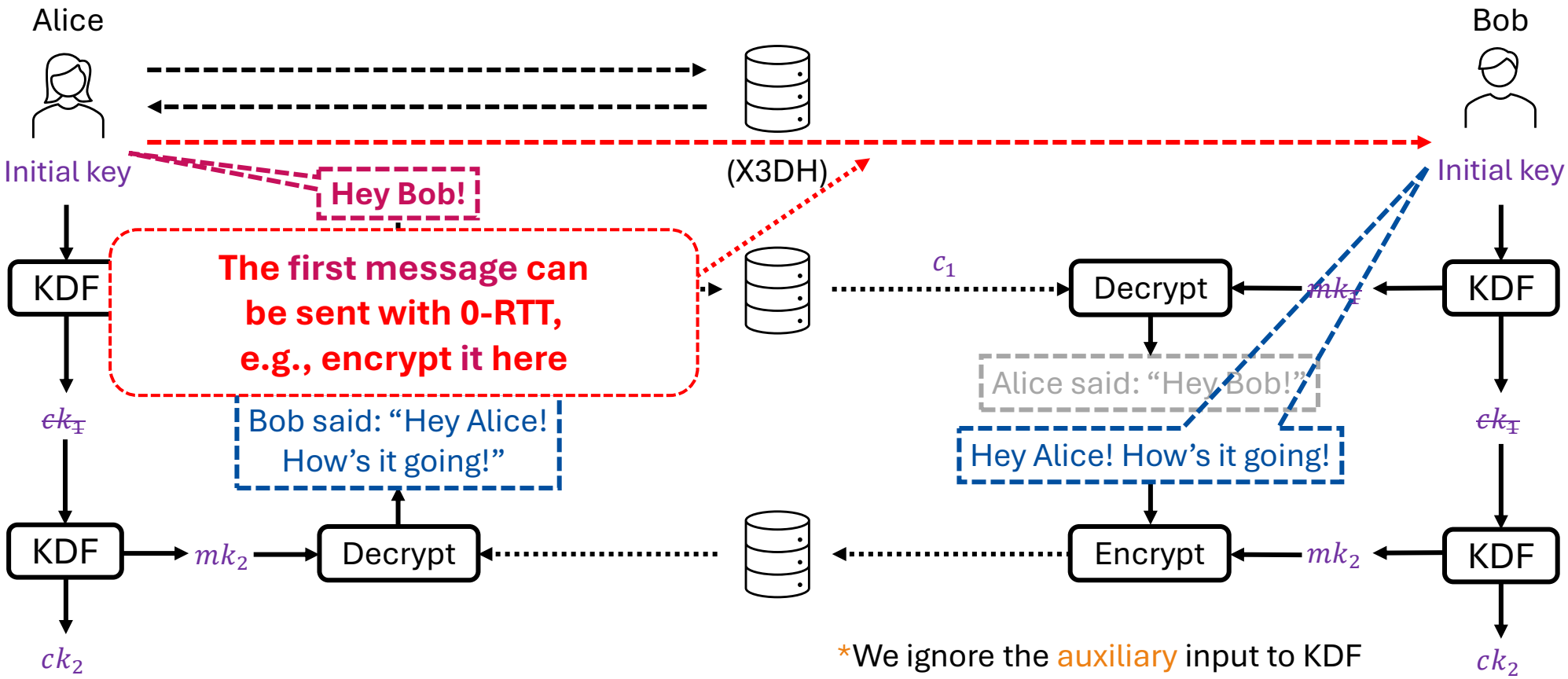
- A toy example of instant messaging using symmetric-key ratchet



*We ignore the auxiliary input to KDF

Symmetric-key Ratchet

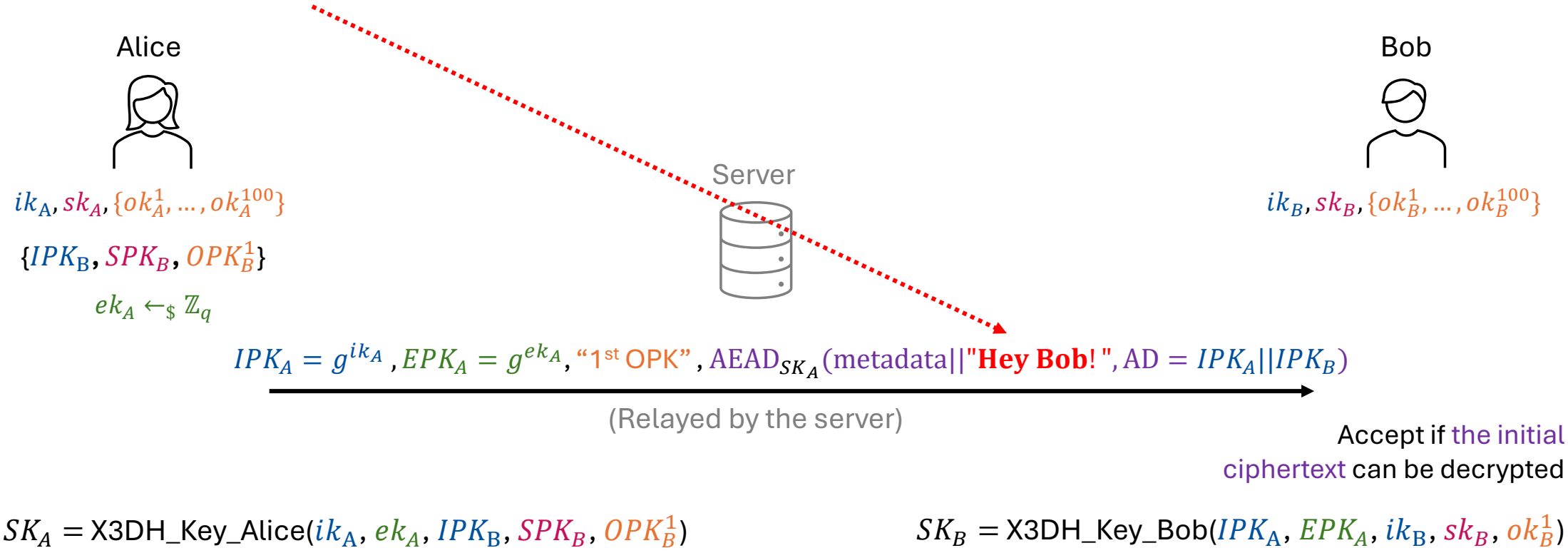
- A toy example of instant messaging using symmetric-key ratchet



*We ignore the auxiliary input to KDF

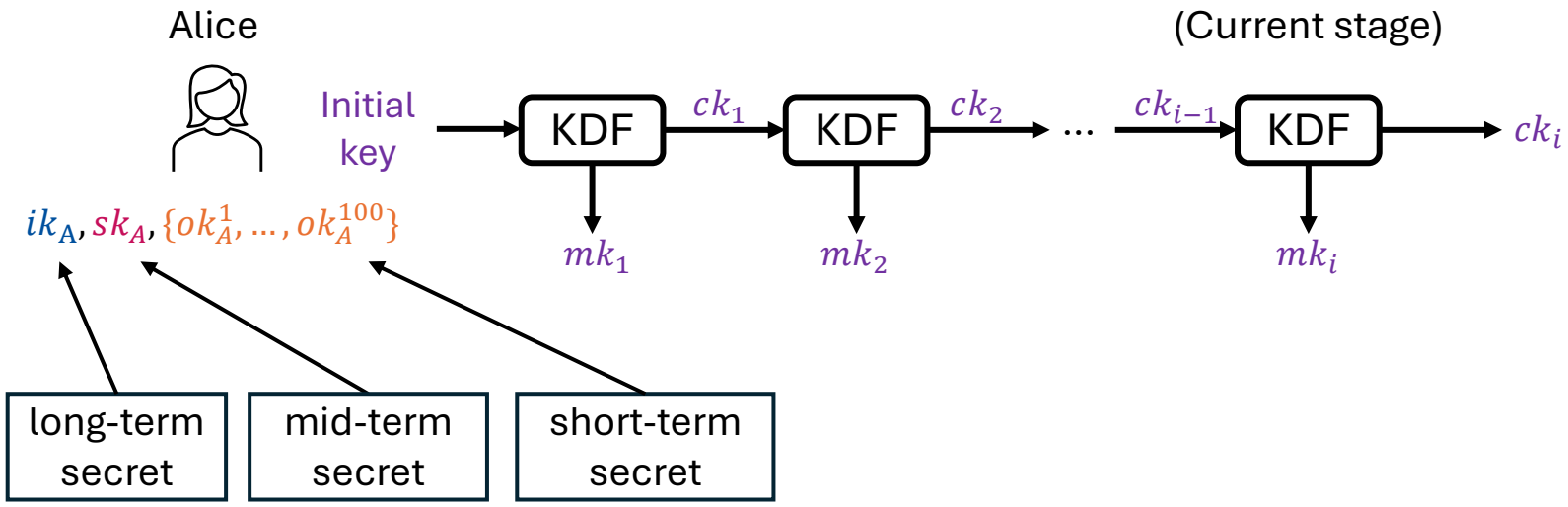
Symmetric-key Ratchet

- Make the first message 0-RTT (Zero Round Time Trip)…



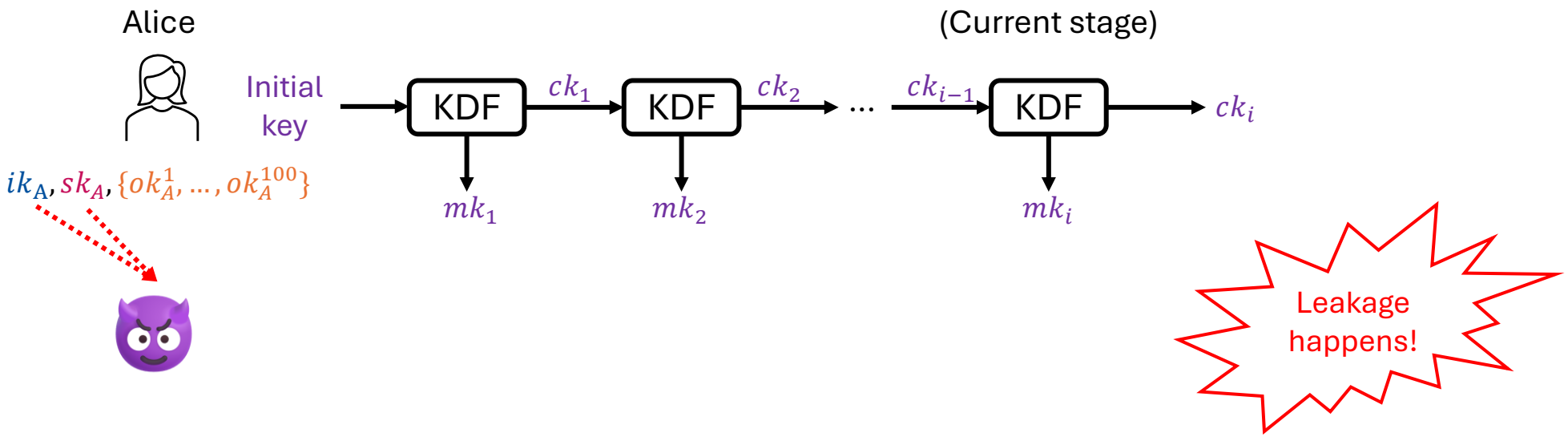
Forward Secrecy

- Long-term secret keys are compromised, but past communication remains secure…



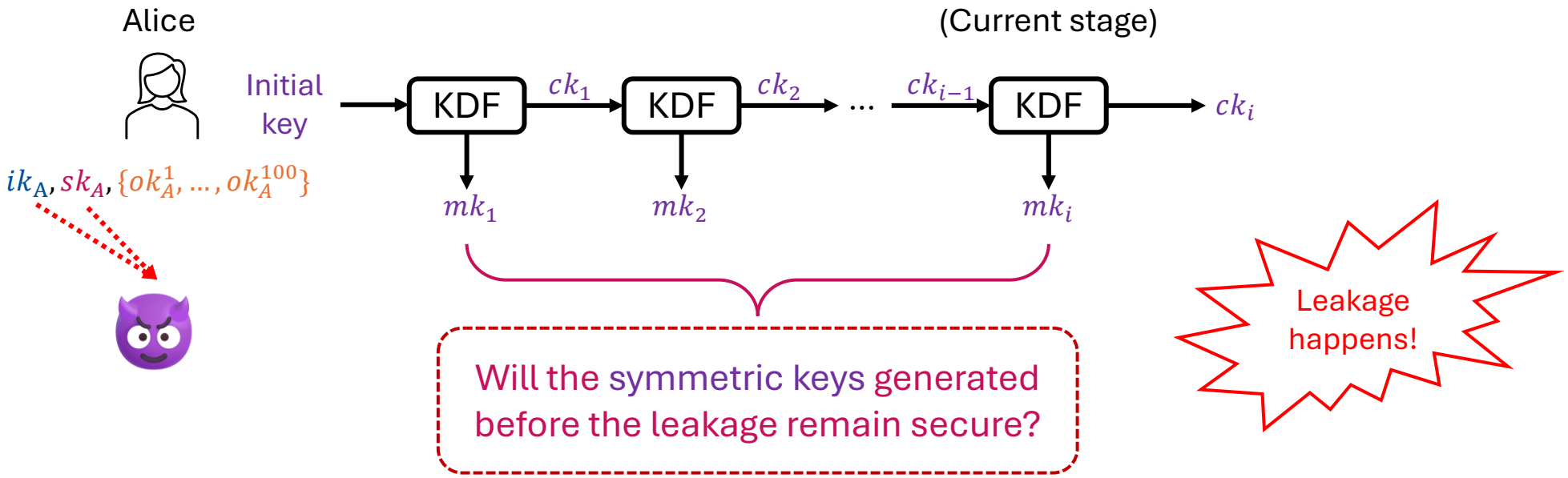
Forward Secrecy

- Long-term secret keys are compromised, but past communication remains secure…



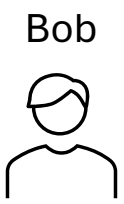
Forward Secrecy

- Long-term secret keys are compromised, but past communication remains secure…



Forward Secrecy

- Recall: How the X3DH protocol computes a shared secret…



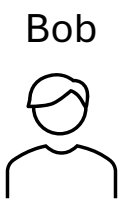
$ik_B, sk_B, \{ok_B^1, \dots, ok_B^{100}\}$

$SK_A = \text{X3DH\_Key\_Alice}(ik_A, ek_A, IPK_B, SPK_B, OPK_B)$

1. $DH_1 = SPK_B^{ik_A}$
2. $DH_2 = IPK_B^{ek_A}$
3. $DH_3 = SPK_B^{ek_A}$
4. $DH_4 = (OPK_B)^{ek_A}$
5. $SK_A = \text{KDF}(DH_1, DH_2, DH_3, DH_4)$

Forward Secrecy

- Recall: How the X3DH protocol computes a shared secret…



$ik_B, sk_B, \{ok_B^1, \ldots, ok_B^{100}\}$

$SK_A = \text{X3DH\_Key\_Alice}(ik_A, ek_A, IPK_B, SPK_B, OPK_B)$

1. $DH_1 = SPK_B^{ik_A}$
2. $DH_2 = IPK_B^{ek_A}$
3. $DH_3 = SPK_B^{ek_A}$
4. $DH_4 = (OPK_B)^{ek_A}$
5. $SK_A = \text{KDF}(DH_1, DH_2, DH_3, DH_4)$

ek_A is not a long-term secret

Forward Secrecy

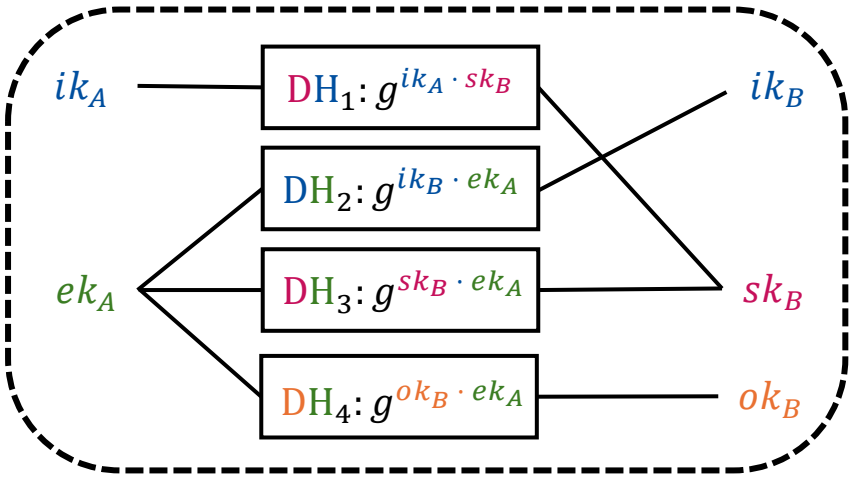
- Recall: How the X3DH protocol computes a shared secret…

Alice

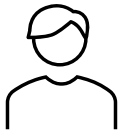
Leakage!

$SK_A = \text{X3DH\_Key\_Alice}(ik_A, ek_A, IPK_B, SPK_B, OPK_B)$

1. $DH_1 = SPK_B^{ik_A}$
2. $DH_2 = IPK_B^{ek_A}$
3. $DH_3 = SPK_B^{ek_A}$
4. $DH_4 = (OPK_B)^{ek_A}$
5. $SK_A = \text{KDF}(DH_1, DH_2, DH_3, DH_4)$



Bob



$ik_B, sk_B, \{ok_B^1, \ldots, ok_B^{100}\}$

Forward Secrecy

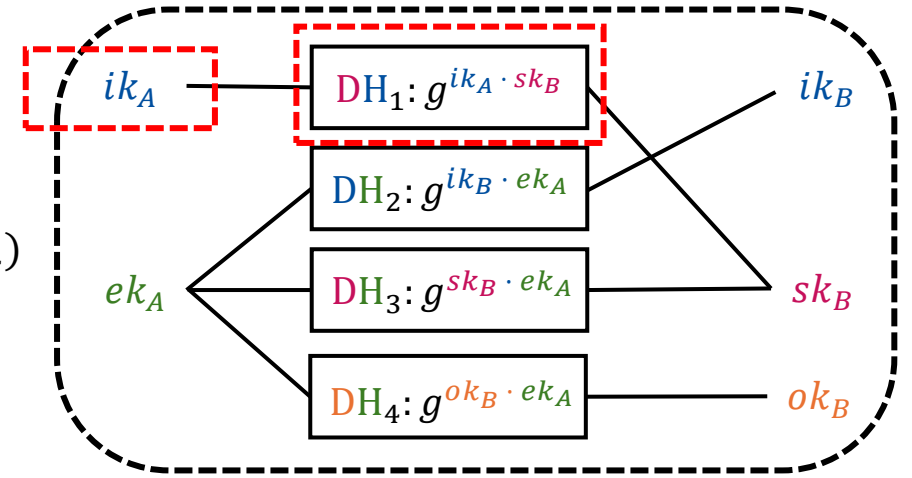
- Recall: How the X3DH protocol computes a shared secret…

Alice

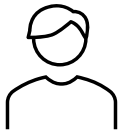
Leakage!

$SK_A = \text{X3DH\_Key\_Alice}(ik_A, ek_A, IPK_B, SPK_B, OPK_B)$

1. $\text{DH}_1 = SPK_B^{ik_A}$
2. $\text{DH}_2 = IPK_B^{ek_A}$
3. $\text{DH}_3 = SPK_B^{ek_A}$
4. $\text{DH}_4 = (OPK_B)^{ek_A}$
5. $SK_A = \text{KDF}(\text{DH}_1, \text{DH}_2, \text{DH}_3, \text{DH}_4)$



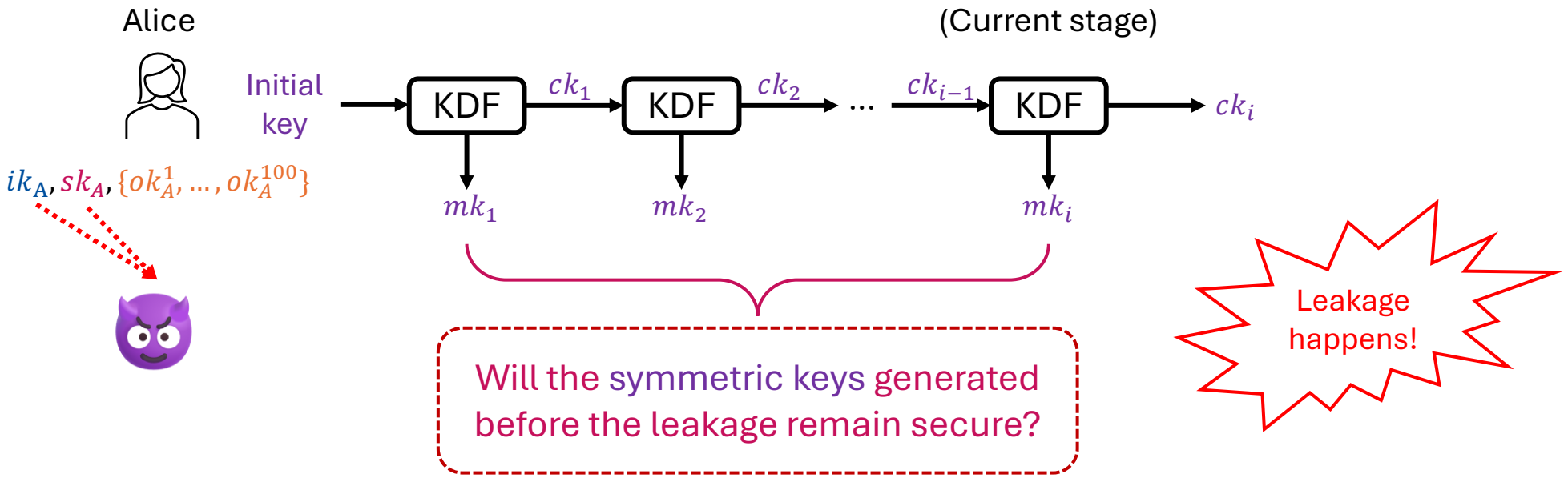
Bob



$ik_B, sk_B, \{ok_B^1, \ldots, ok_B^{100}\}$

Forward Secrecy

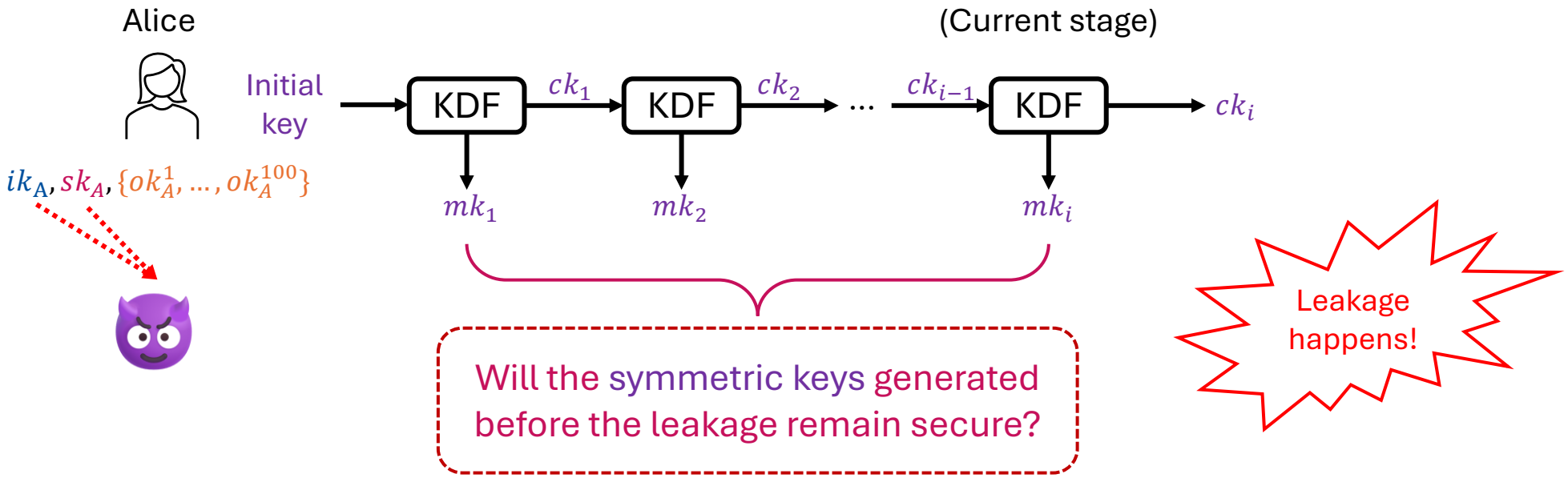
- Long-term secret keys are compromised, but past communication remains secure…



Initial_key (of X3DH)= $\mathrm{KDF}(\mathrm{DH}_1, \mathrm{DH}_2, \mathrm{DH}_3, \mathrm{DH}_4)$

Forward Secrecy

- Long-term secret keys are compromised, but past communication remains secure…



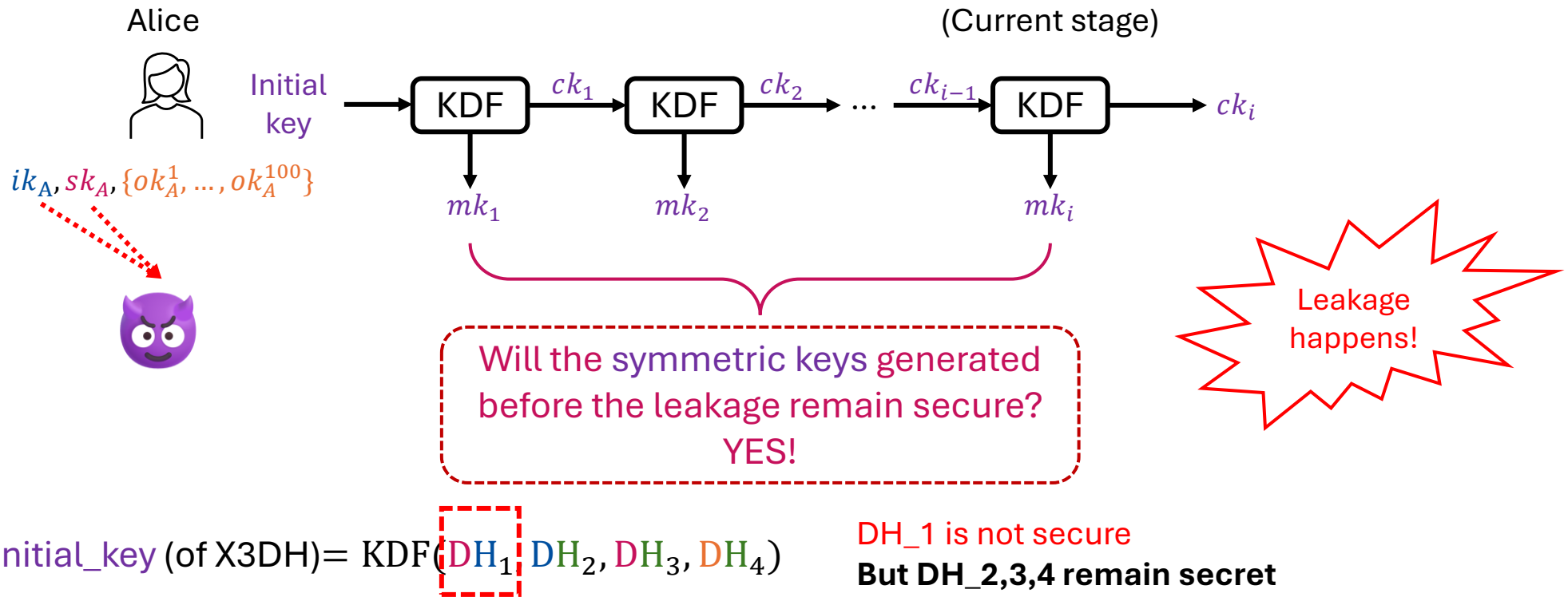
Initial_key (of X3DH)= $\mathrm{KDF}(\mathrm{DH}_1, \mathrm{DH}_2, \mathrm{DH}_3, \mathrm{DH}_4)$

DH_1 is not secure

But DH_2,3,4 remain secret

Forward Secrecy

- Long-term secret keys are compromised, but past communication remains secure…



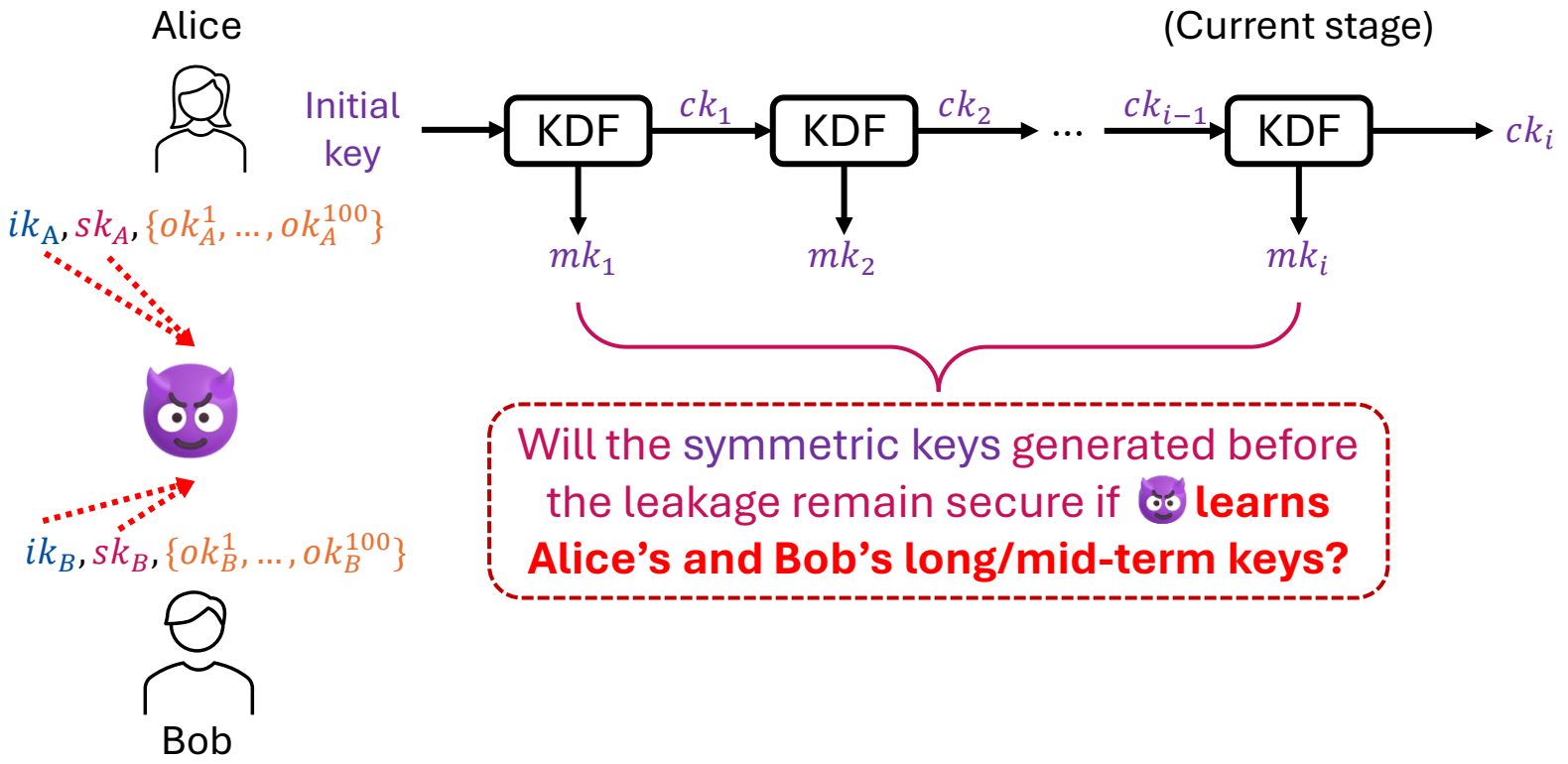
Initial_key (of X3DH)= $\mathrm{KDF}(\mathrm{DH}_1, \mathrm{DH}_2, \mathrm{DH}_3, \mathrm{DH}_4)$

DH_1 is not secure

But DH_2,3,4 remain secret

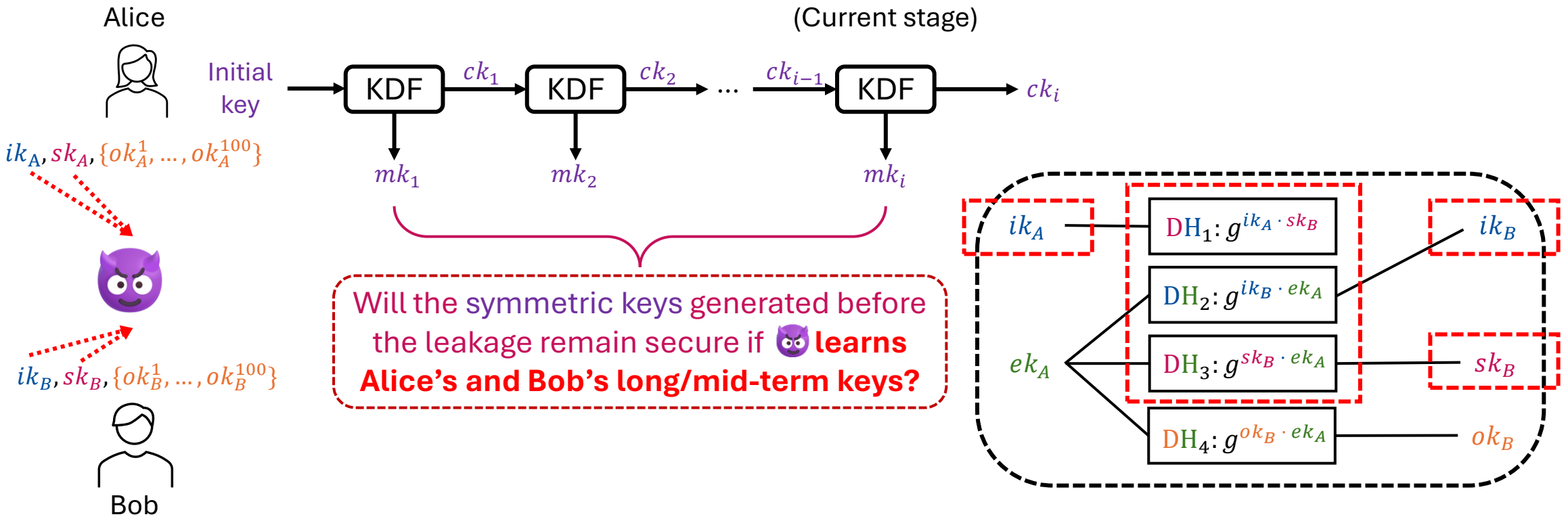
Forward Secrecy

- Long-term secret keys are compromised, but past communication remains secure…

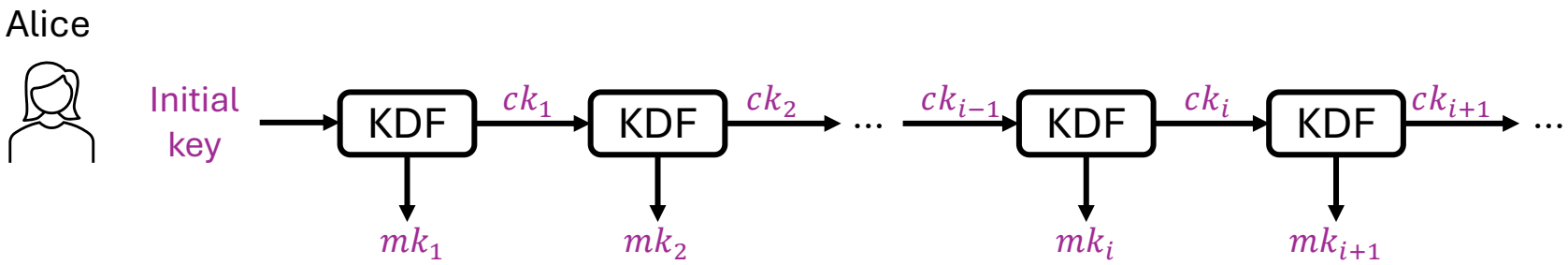


Forward Secrecy

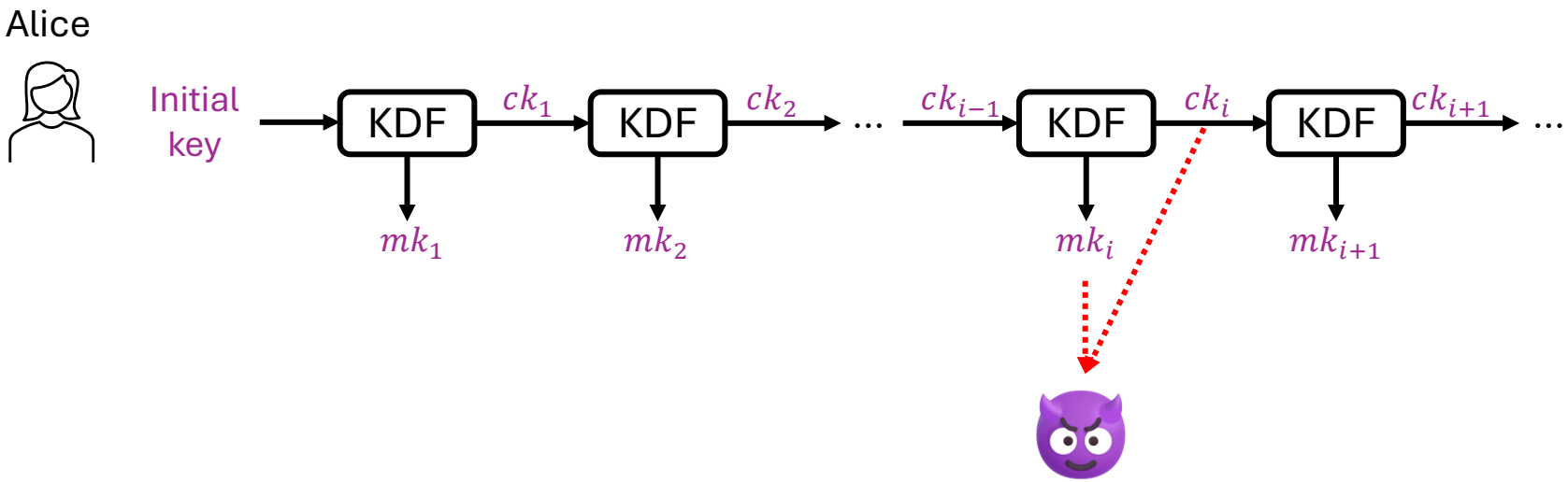
- Long-term secret keys are compromised, but past communication remains secure…



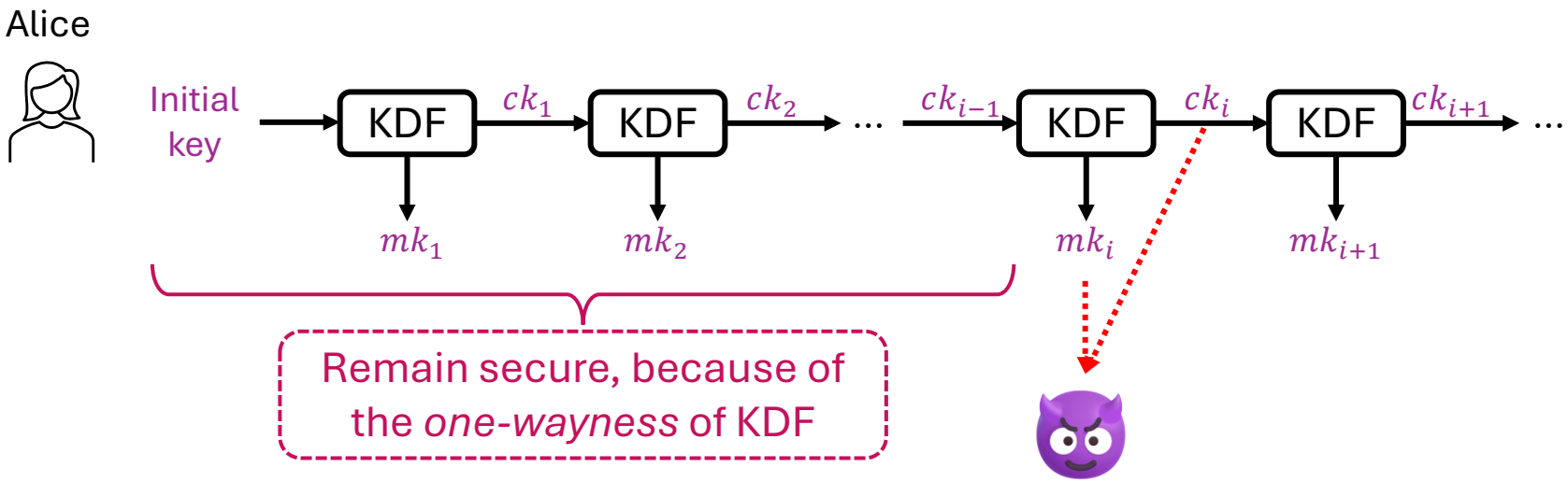
Backward Secrecy



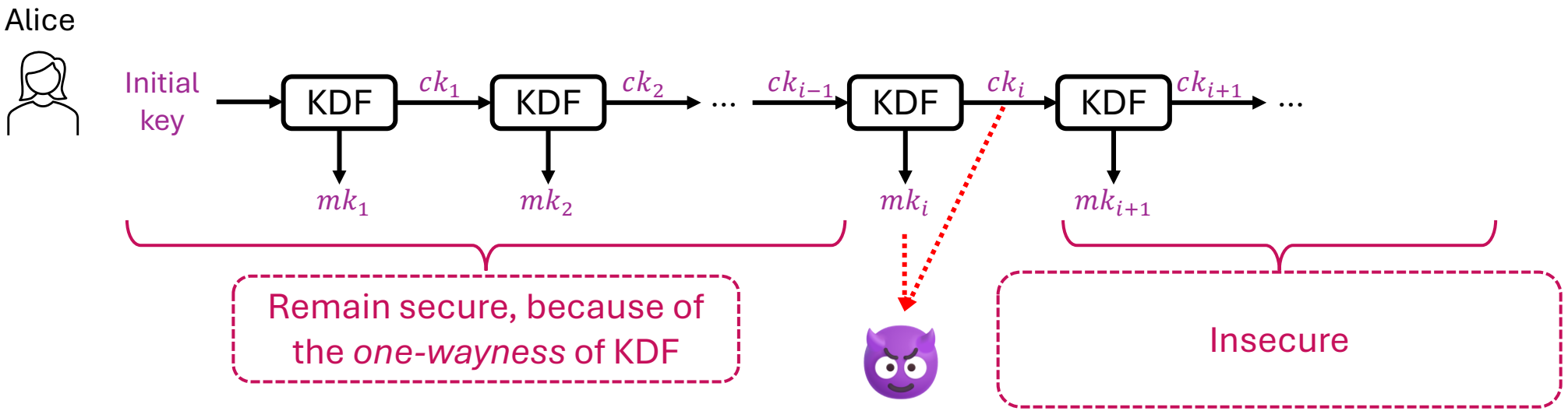
Backward Secrecy



Backward Secrecy

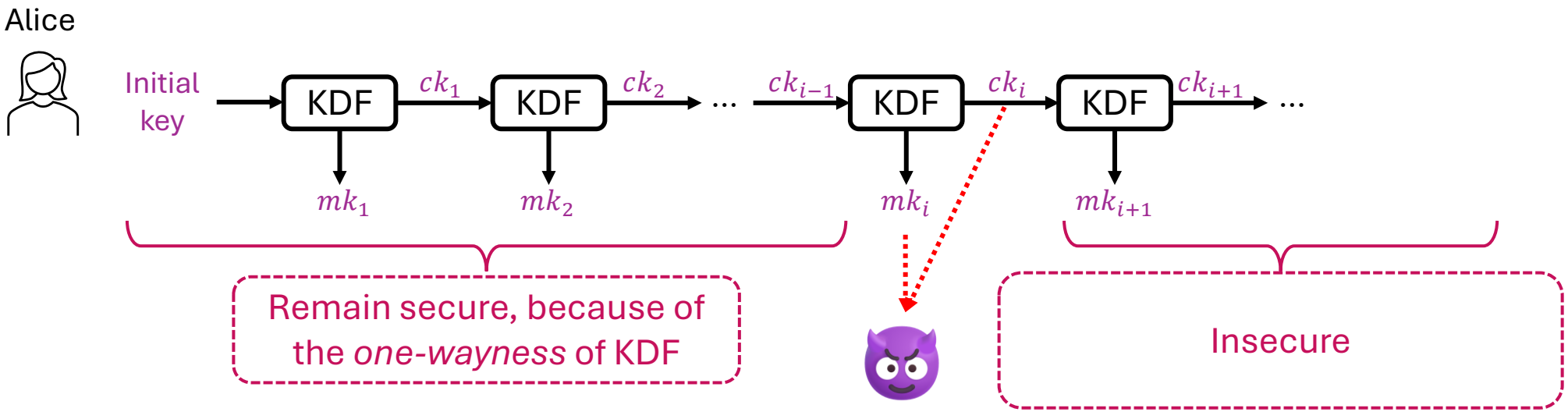


Backward Secrecy



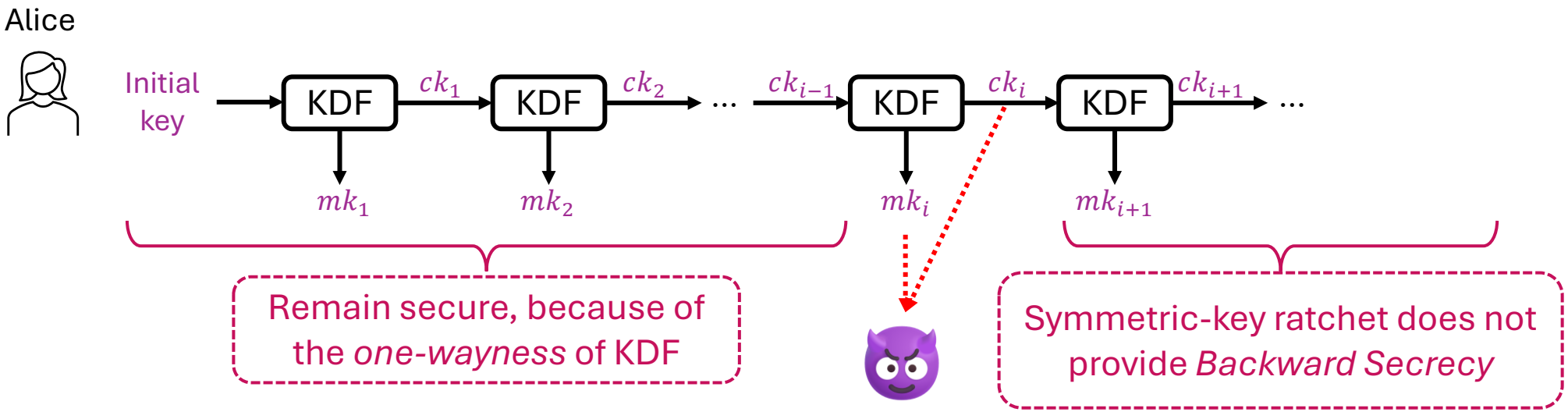
Backward Secrecy

- Future communication remains secure even if a current session key is compromised



Backward Secrecy

- Future communication remains secure even if a current session key is compromised



Diffie-Hellman Ratchet

- X3DH + Symmetric-key Ratchet
  - X3DH provides *Forward Secrecy*
  - Current session key compromises does not lead to the compromise of previous session keys
    - (by the one-wayness of KDF in Symmetric-key Ratchet)
  - No Backward Secrecy

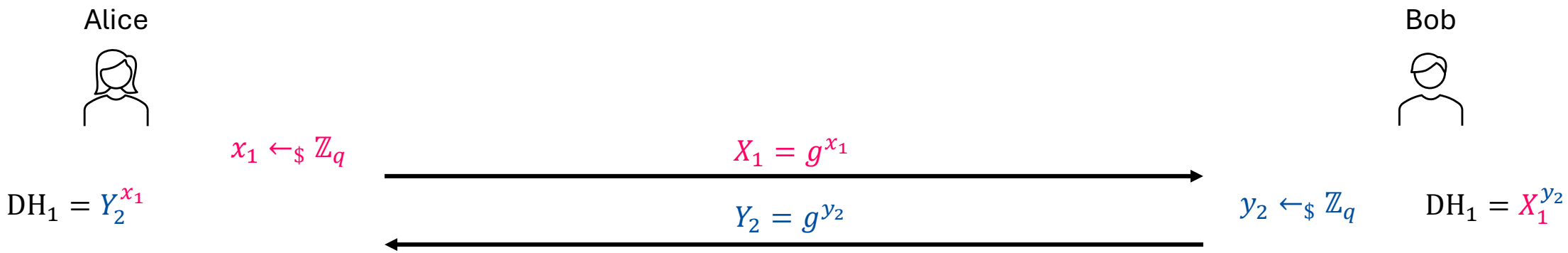
Diffie-Hellman Ratchet

- X3DH + Symmetric-key Ratchet
  - X3DH provides *Forward Secrecy*
  - Current session key compromises does not lead to the compromise of previous session keys
    - (by the one-wayness of KDF in Symmetric-key Ratchet)
  - No Backward Secrecy

- Solution: Diffie-Hellman Ratchet

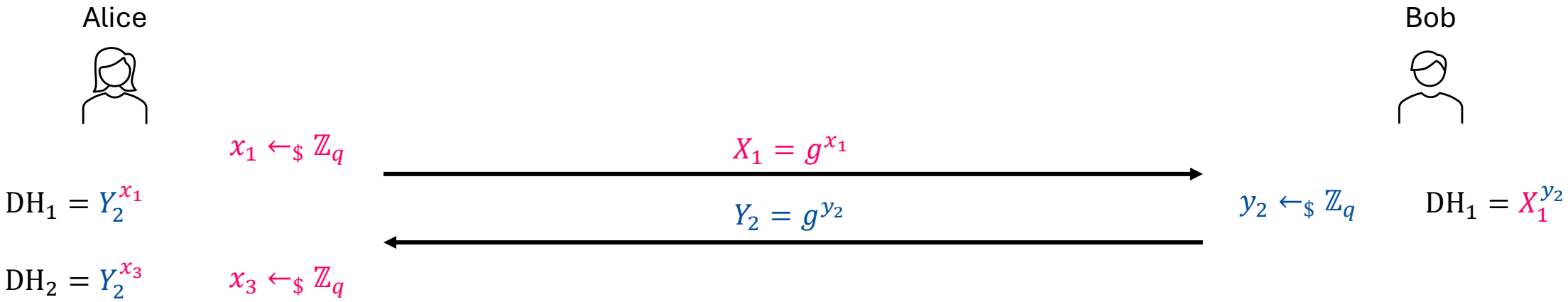
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*…



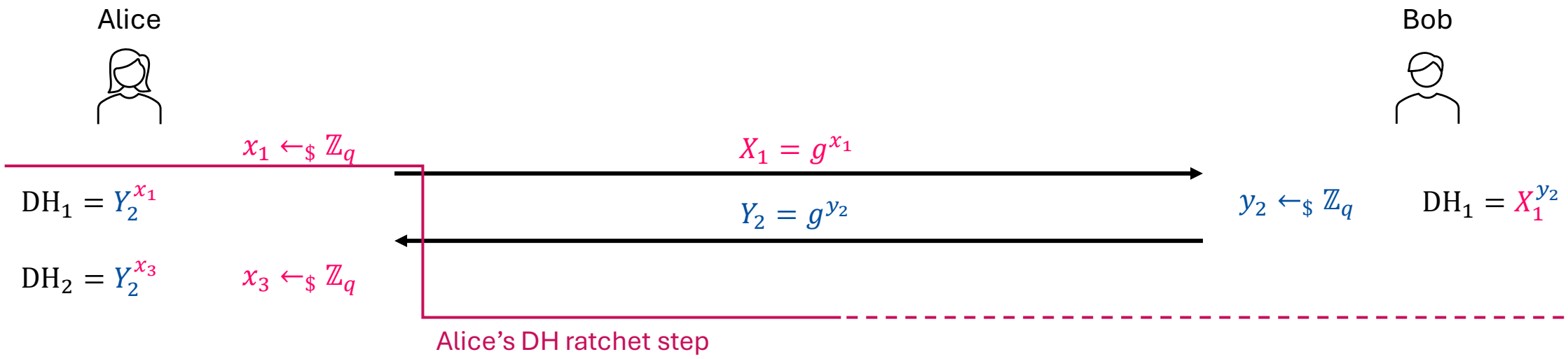
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*...



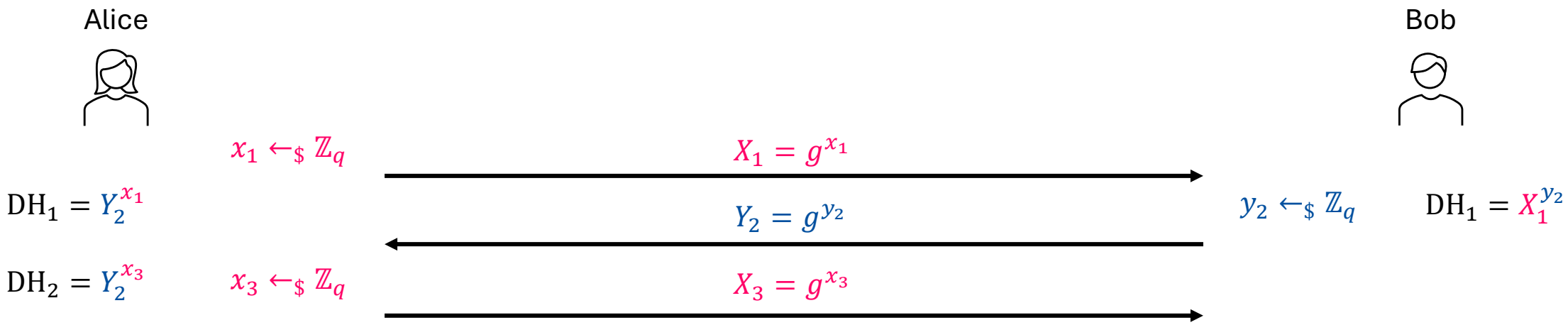
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*...



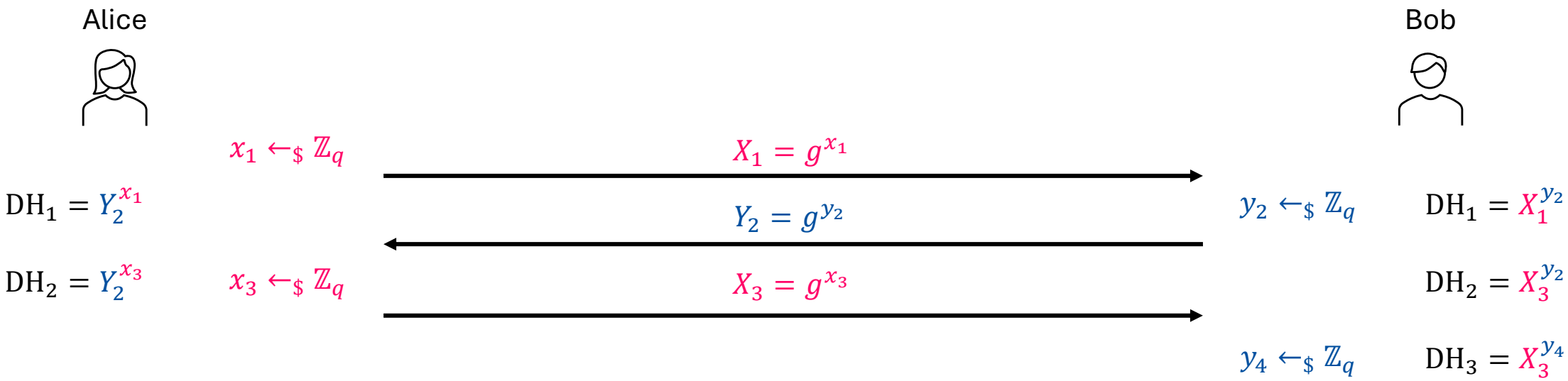
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*...



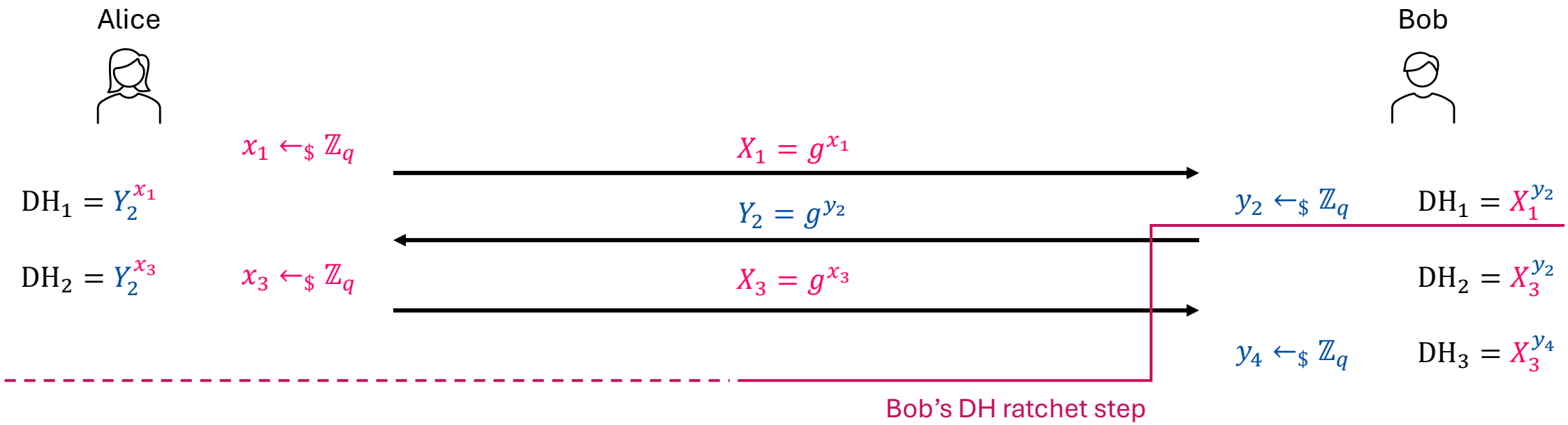
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*...



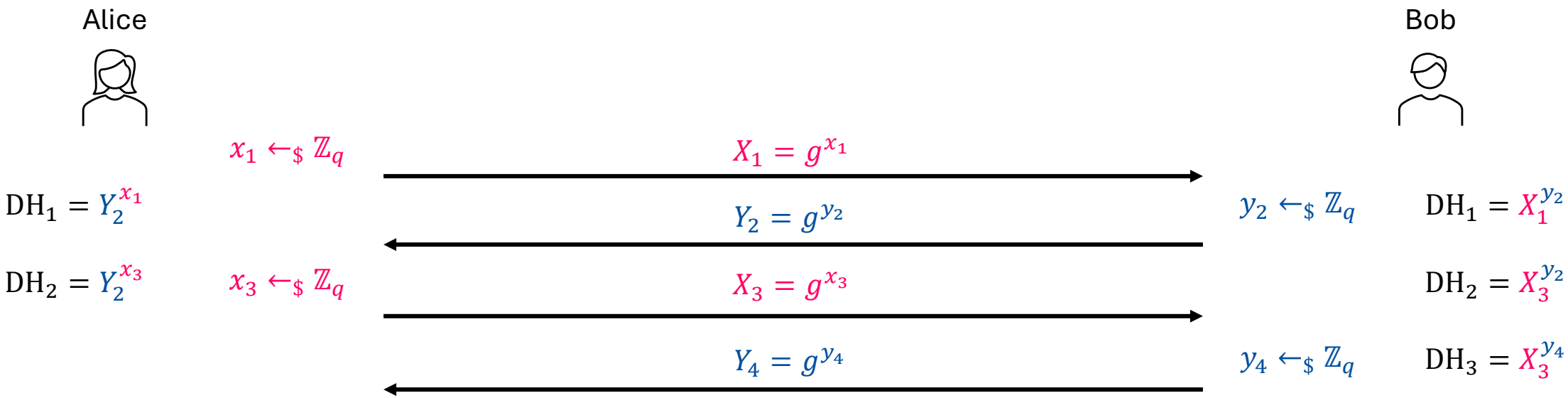
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*…



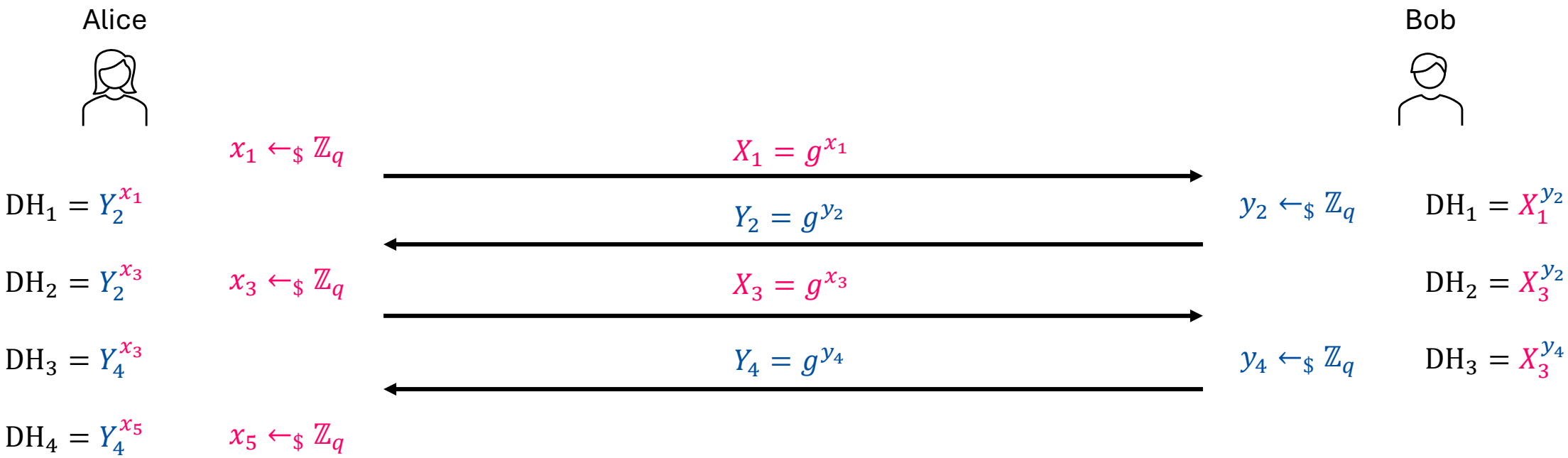
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*...



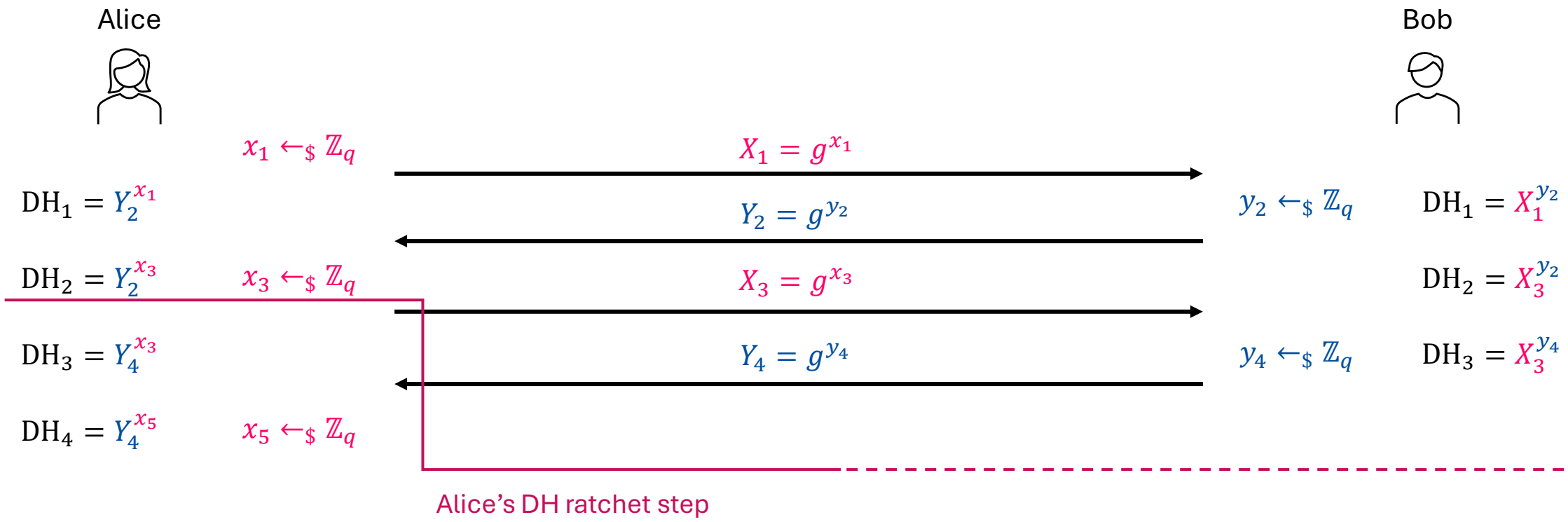
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*...



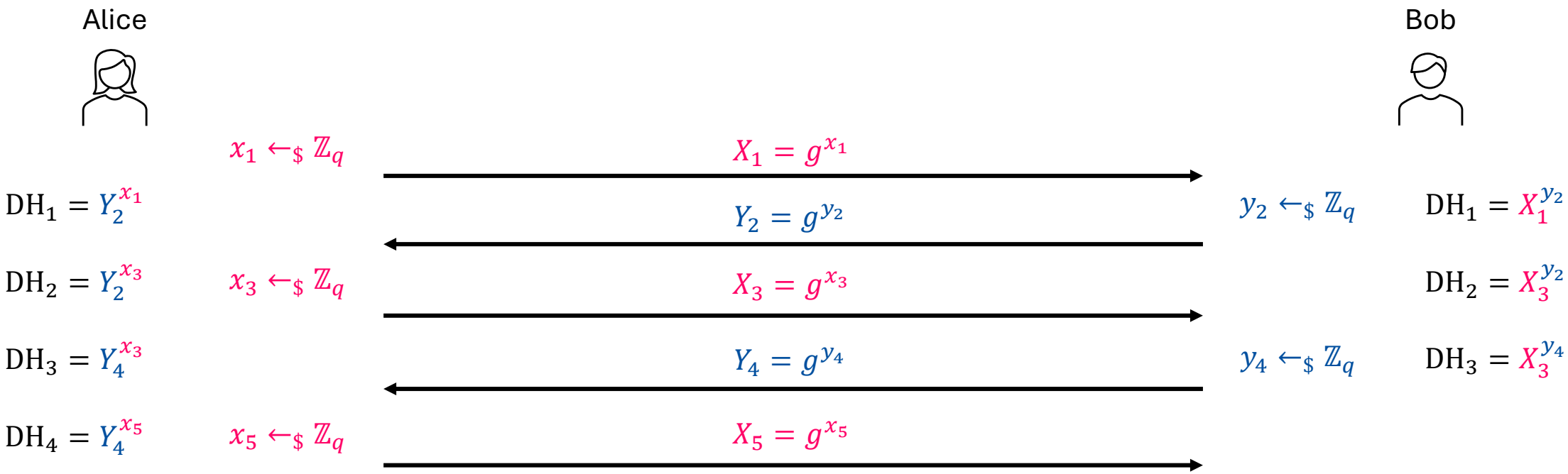
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*...



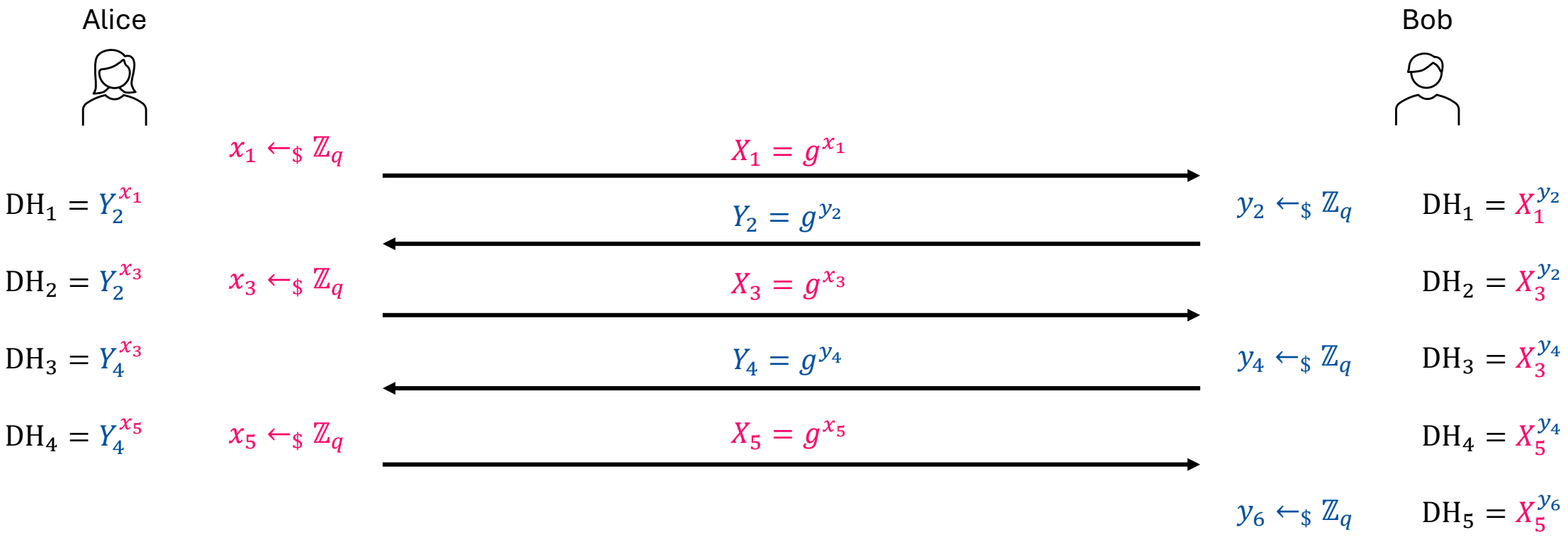
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*...



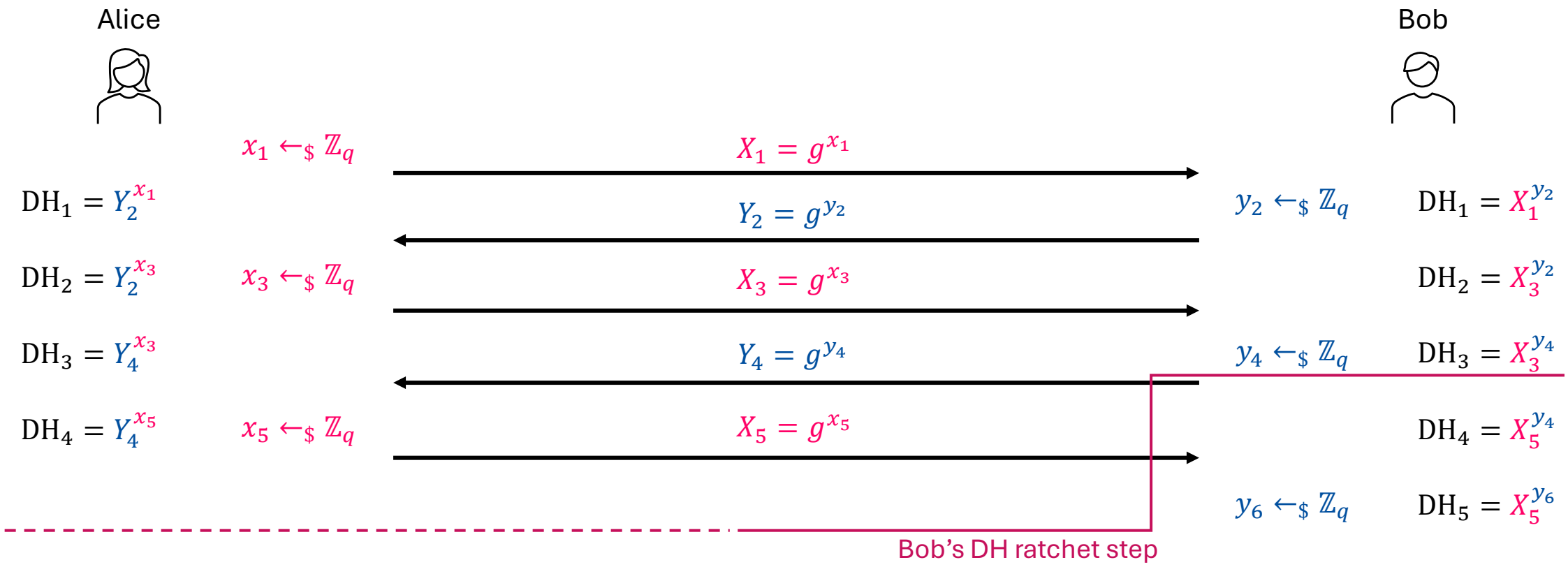
Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*...



Diffie-Hellman Ratchet

- A toy example: Running DHKE continuously with *rotating ephemeral keys*...



Diffie-Hellman Ratchet

- The main idea: Combine Symmetric-key Ratchet and Diffie-Hellman Ratchet
  - DH Ratchet generates fresh shared DH secrets continuously via rotating new ephemeral keys…
  - These fresh DH secrets feed into Symmetric-key Ratchet to add new secret information…

- More details will be explained in the next lecture

Coding Tasks

- Implement the Diffie-Hellman Ratchet algorithm (can be without sockets).

Homework

No Homework

...but the **deadline** for homework in Lectures 1 and 2 is

22.11.2024 at 23:59 (this Friday evening)

Further Reading

- Old news -- *WhatsApp's Signal Protocol integration is now complete:* *https://signal.org/blog/whatsapp-complete/*
- Technical Documentations of Signal: https://signal.org/docs/
- Cohn-Gordon et al's security analysis of Signal: https://eprint.iacr.org/2016/1013