

# Cryptography Engineering

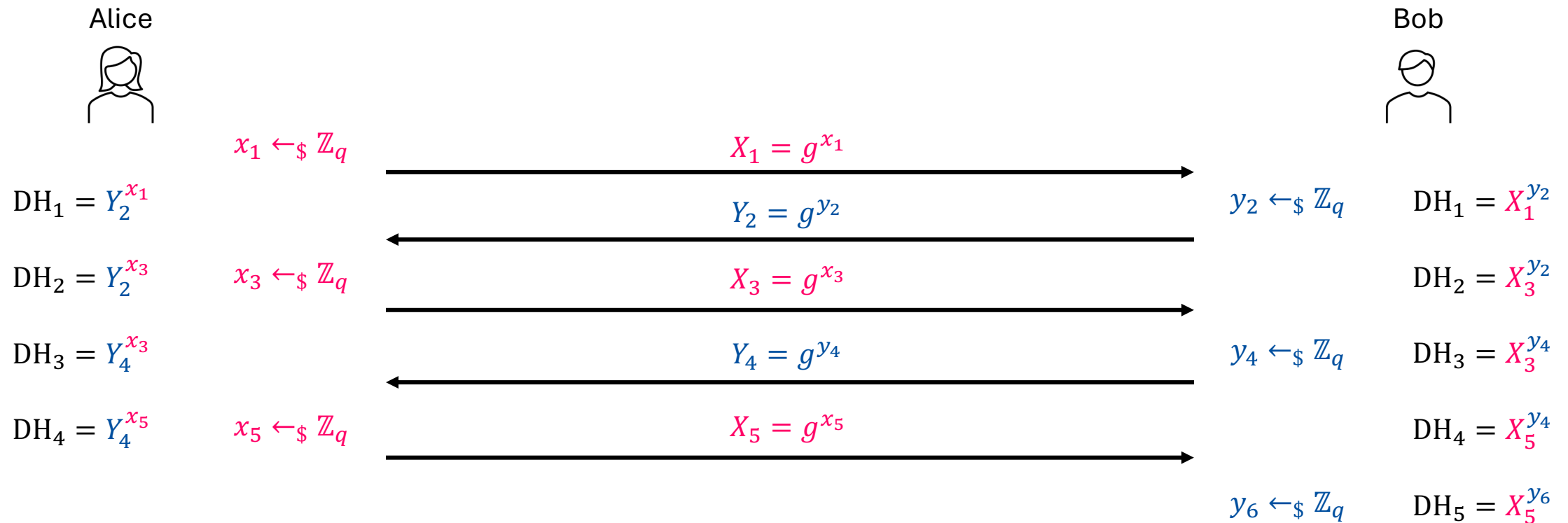
- Lecture 6 (Nov 27, 2024)
- Today's notes:
  - Double Ratchet Algorithm
  - Signal Secure Messaging Protocol
- No homework

# Double Ratchet

- The main idea: Symmetric-key Ratchet + **Diffie-Hellman Ratchet**

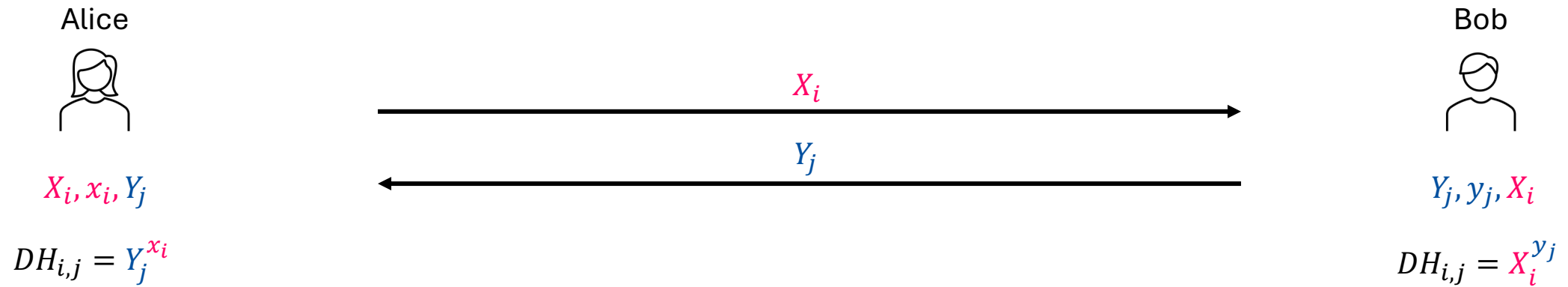
# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...



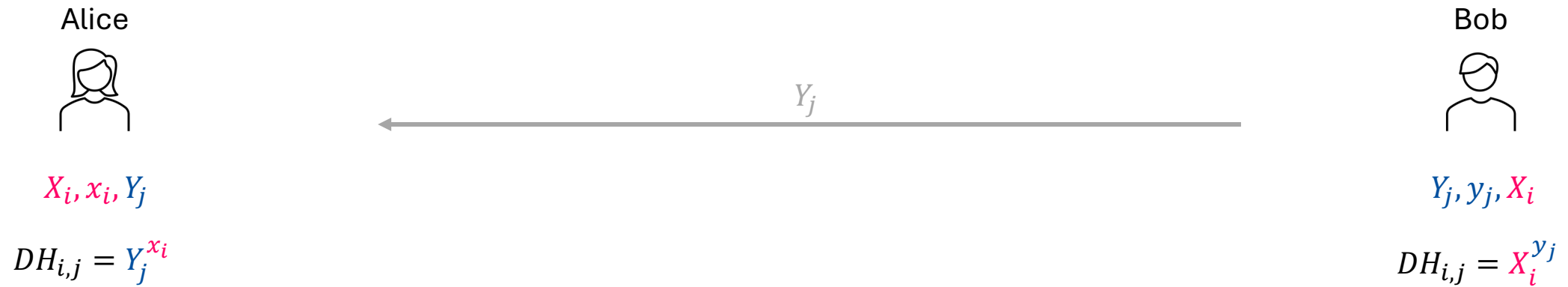
# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...



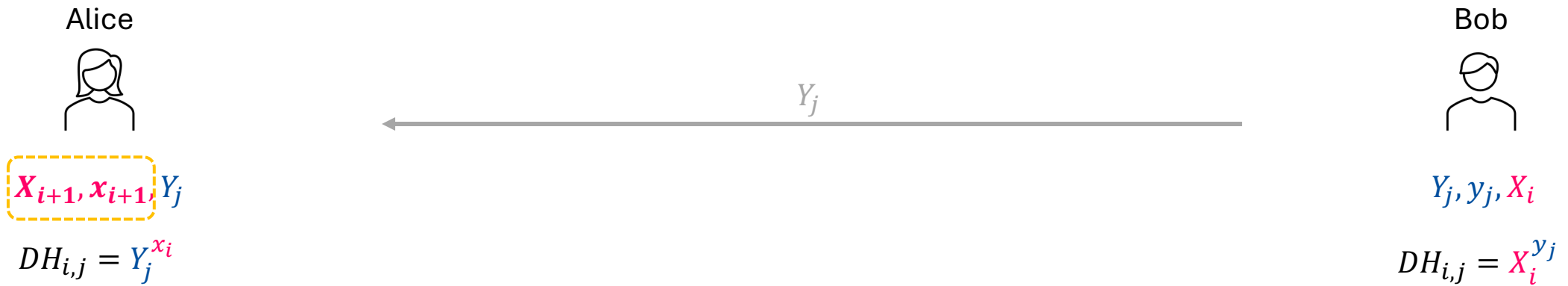
# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...



# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...



# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...

Alice



$X_{i+1}, x_{i+1}, Y_j$

$$DH_{i,j} = Y_j^{x_i}$$

$$DH_{i+1,j} = Y_j^{x_{i+1}}$$

Bob



$Y_j, y_j, X_i$

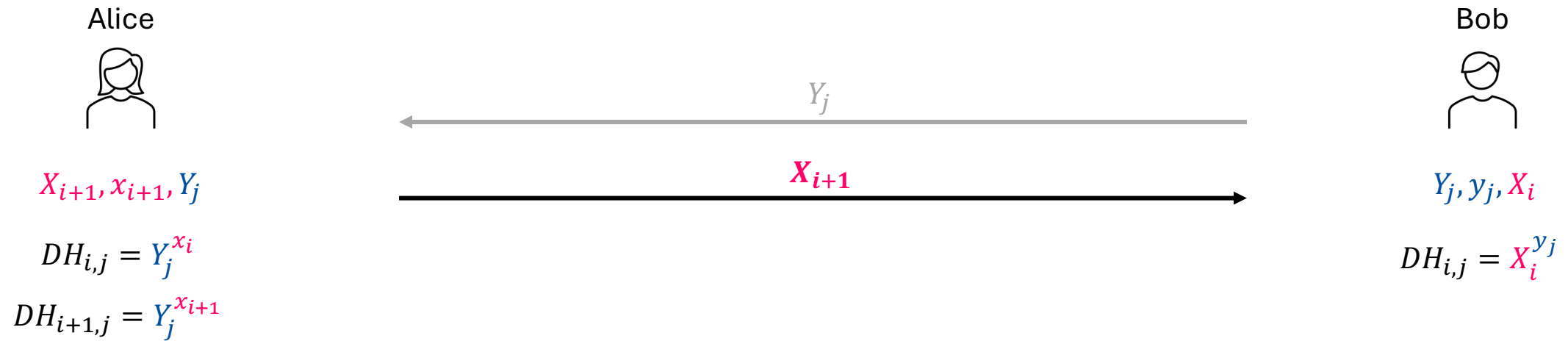
$$DH_{i,j} = X_i^{y_j}$$

$Y_j$



# Double Ratchet – DH Ratchet

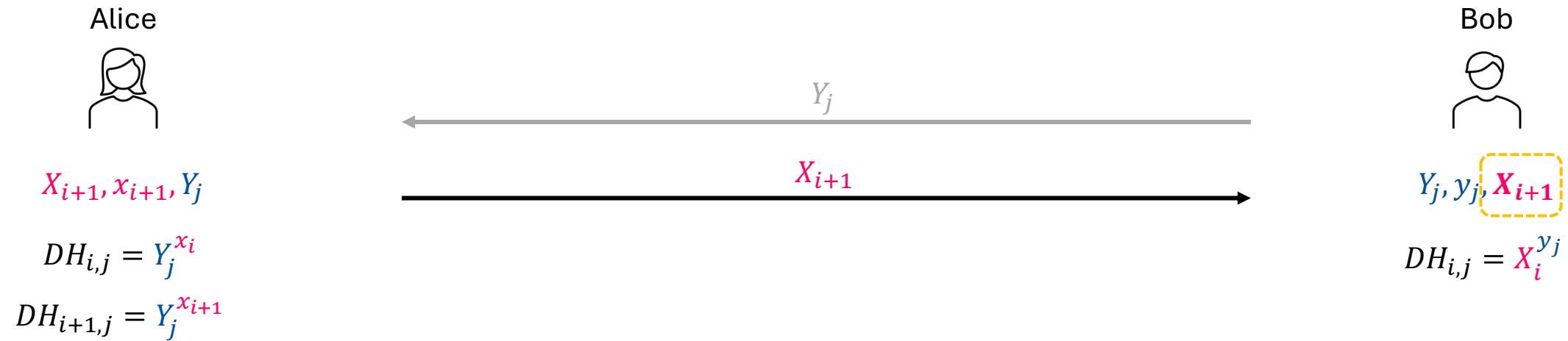
- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...





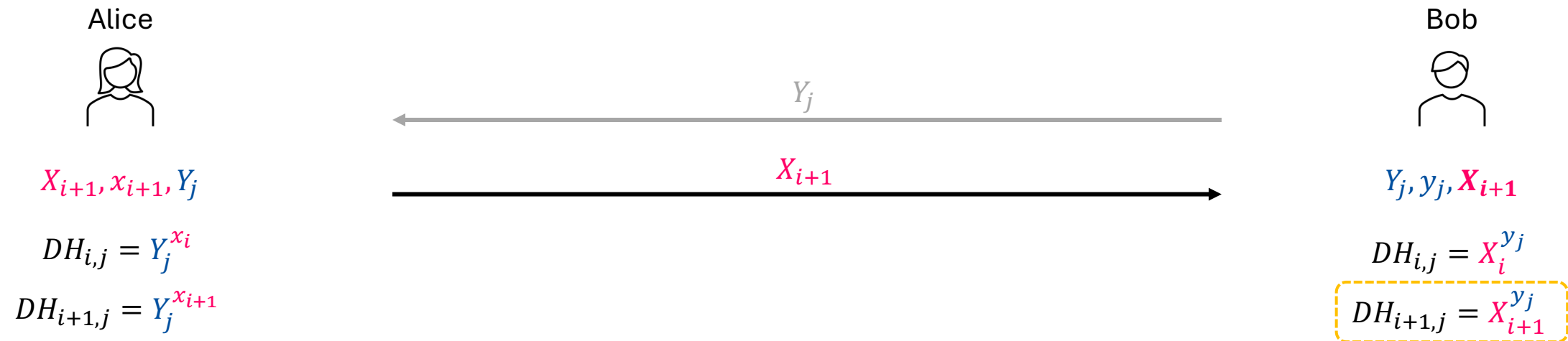
# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...



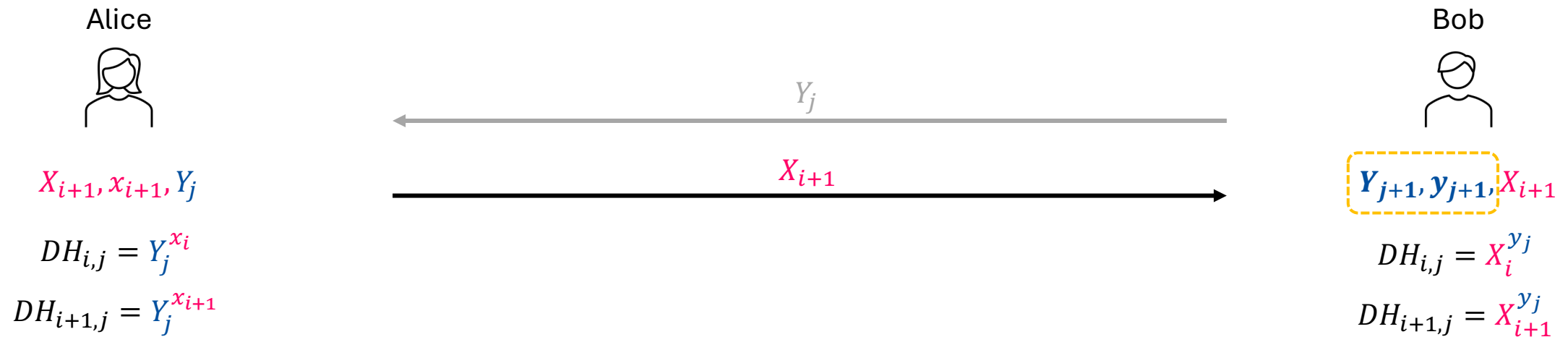
# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...



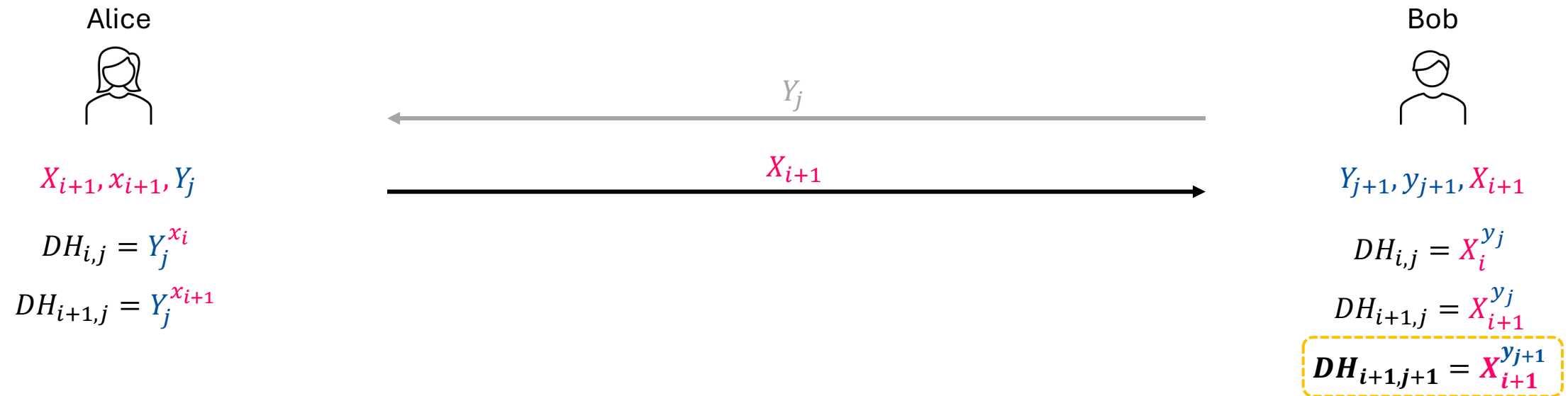
# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...



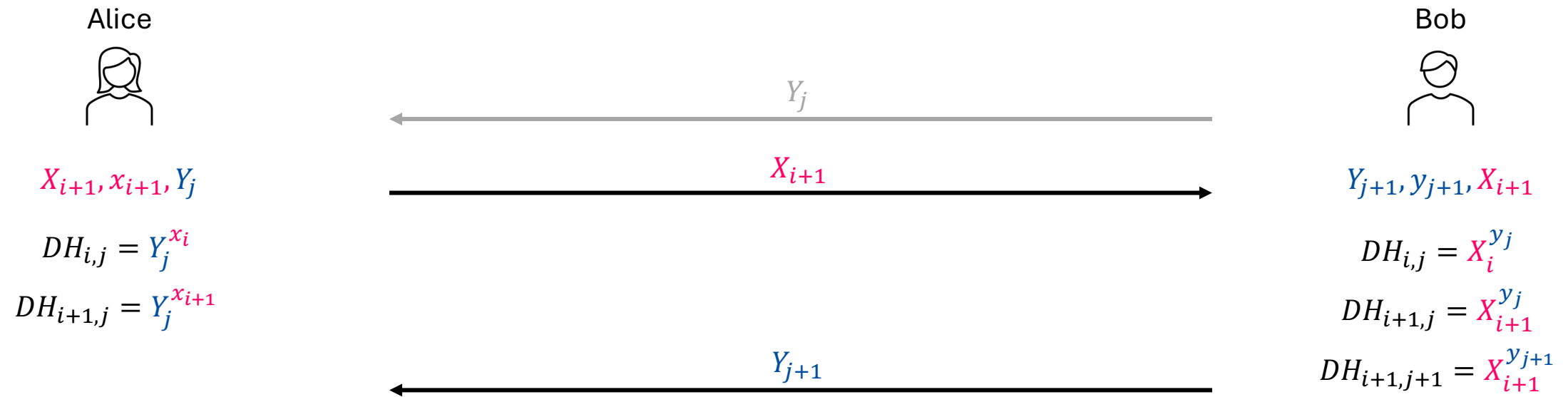
# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...



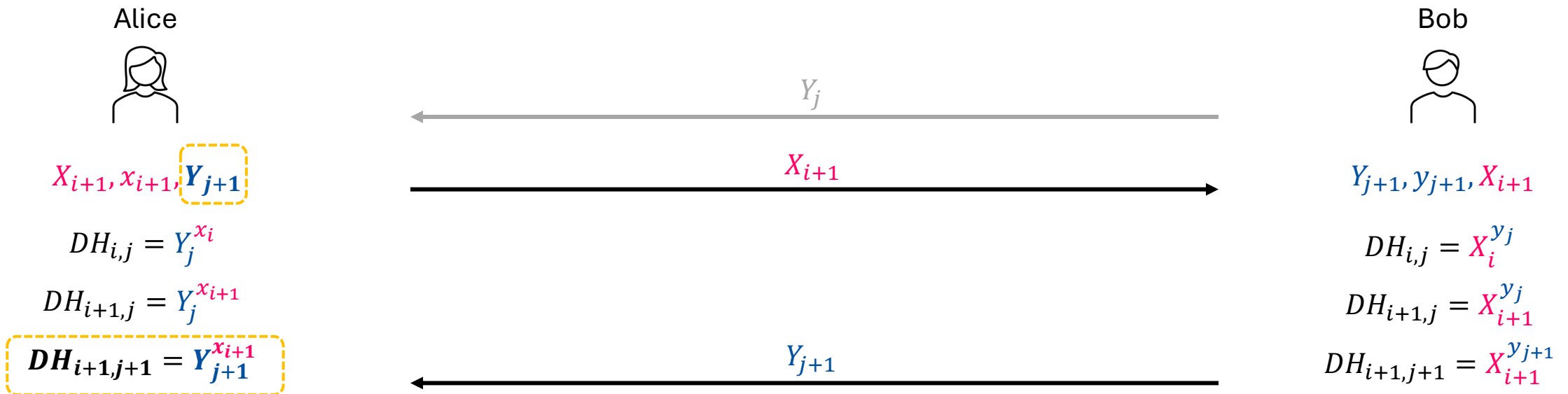
# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...



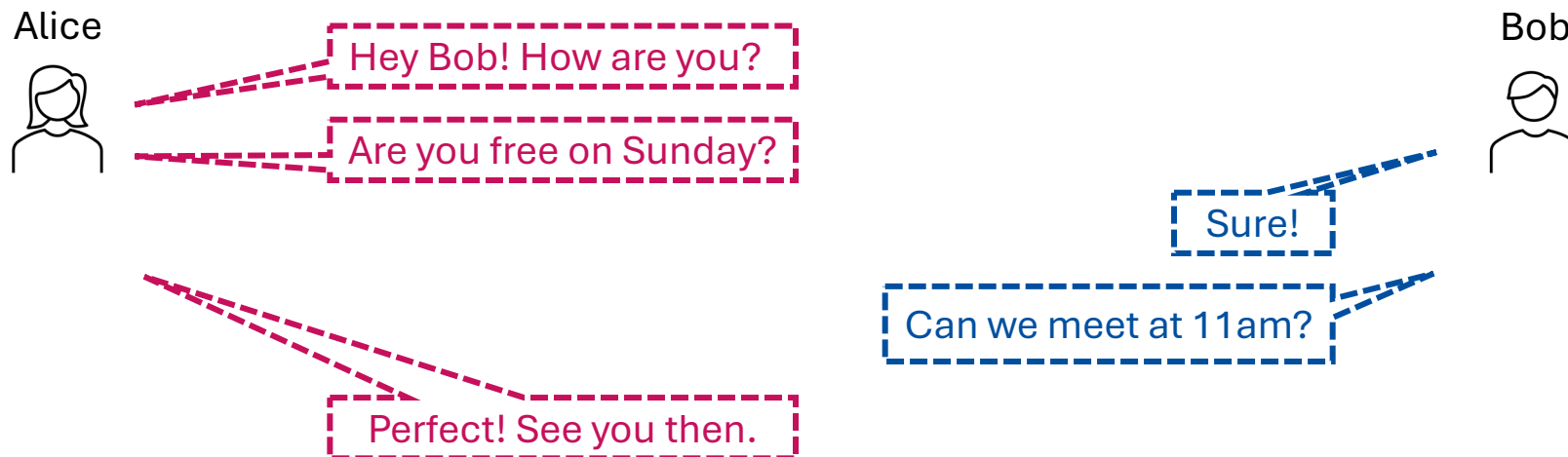
# Double Ratchet – DH Ratchet

- Main idea of DH Ratchet: Running DHKE continuously with *rotating ephemeral keys*...



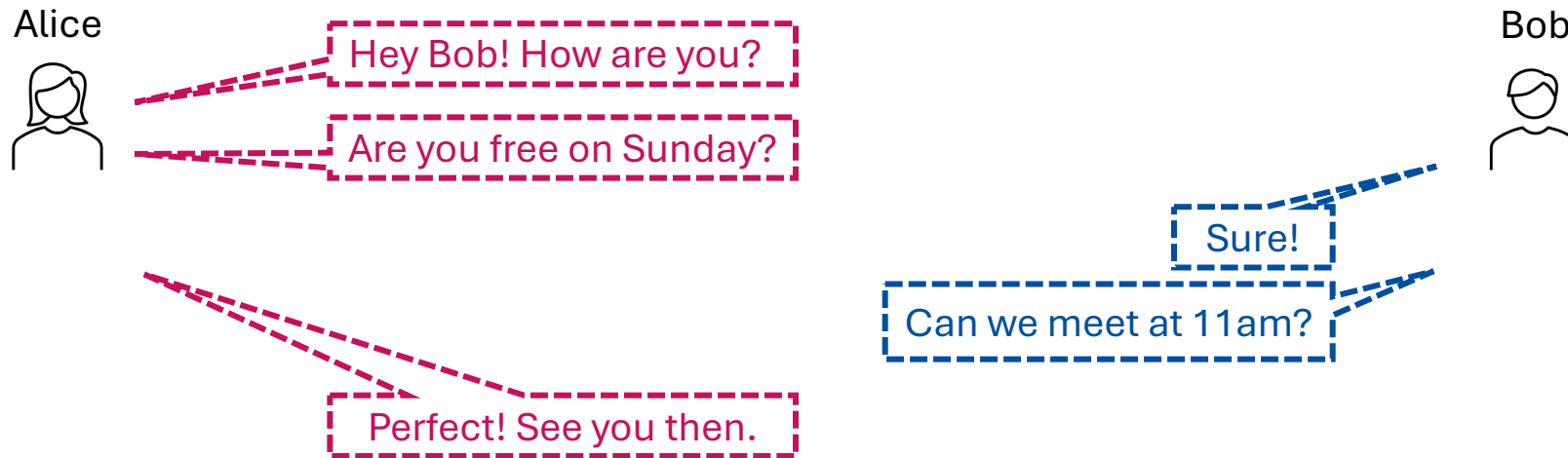
# Double Ratchet

- The main idea: Symmetric-key Ratchet + Diffie-Hellman Ratchet
  - When a party sends messages (**before** its peer party replies): Use Symmetric-key Ratchet...
  - When the peer party replies: Use Diffie-Hellman Ratchet to update the key...
- Example:



# Double Ratchet

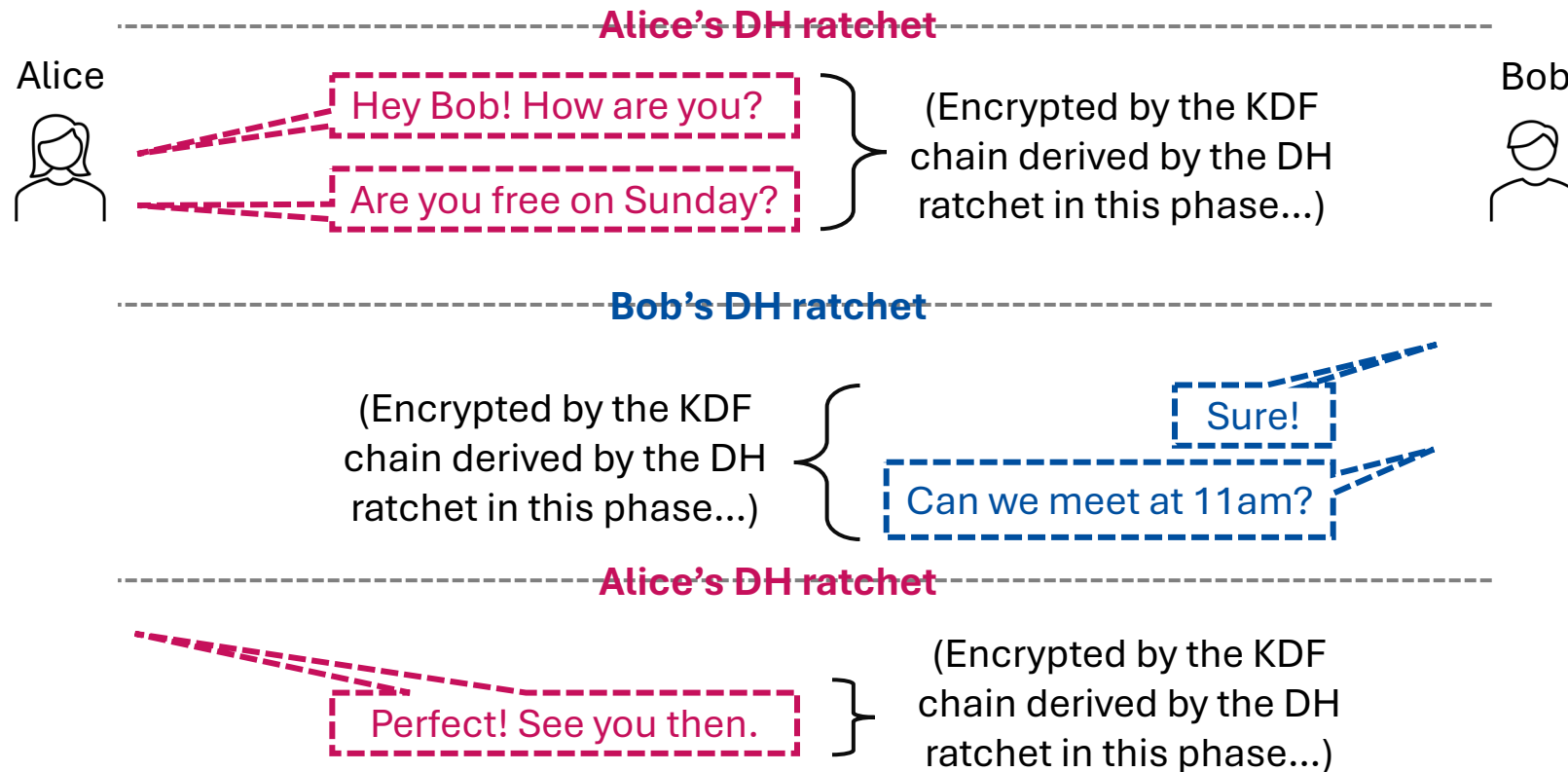
- The main idea: Symmetric-key Ratchet + Diffie-Hellman Ratchet





# Double Ratchet

- The main idea: Symmetric-key Ratchet + Diffie-Hellman Ratchet



# Double Ratchet

Alice



$X_i, x_i, Y_j$

Root key  
(from previous stage)

Bob



$Y_j, y_j, X_i$

Root key



All messages  
are relayed by  
the server

# Double Ratchet

Alice



$X_{i+1}, x_{i+1}, Y_j$

Root key  
(from previous stage)

$DH_{i+1,j} = Y_j^{x_{i+1}}$   
(as auxiliary  
input of KDF)



All messages  
are relayed by  
the server

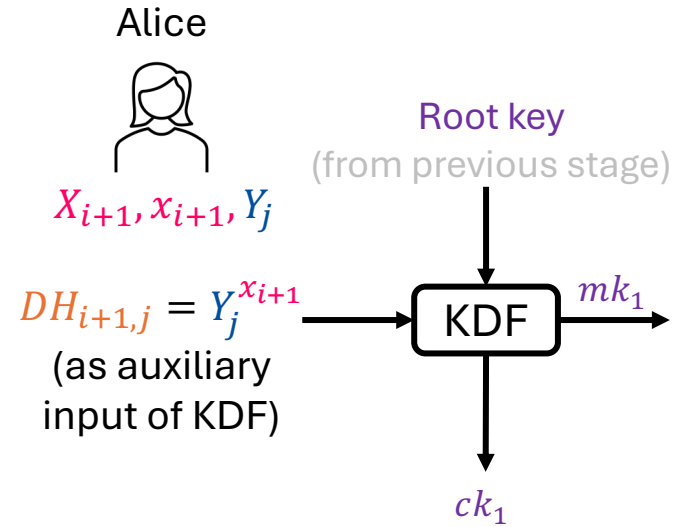
Bob



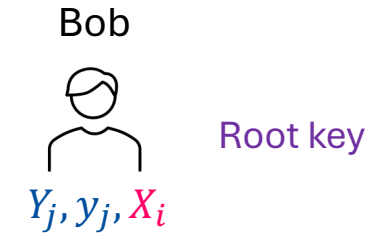
$Y_j, y_j, X_i$

Root key

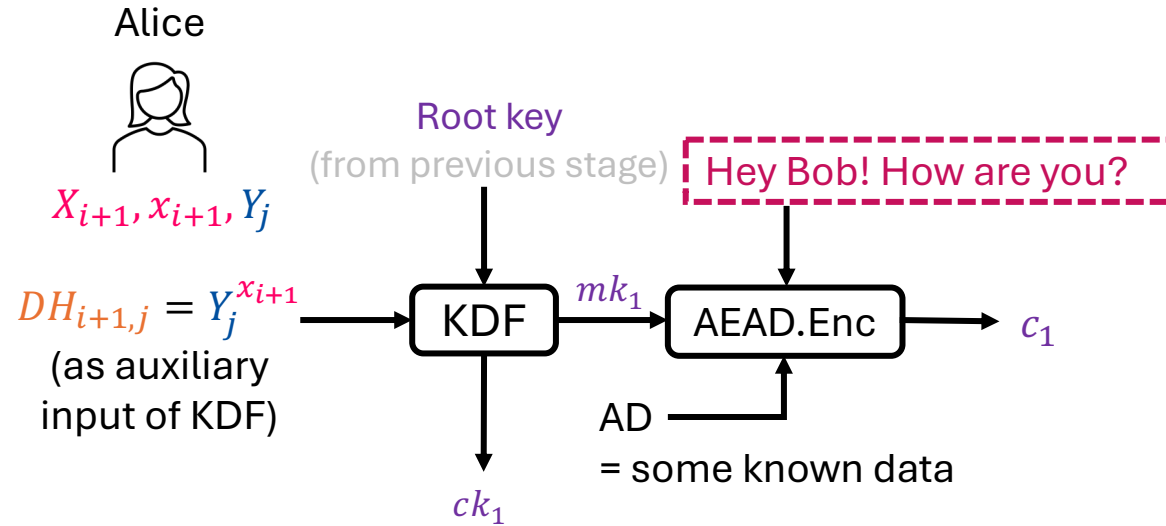
# Double Ratchet



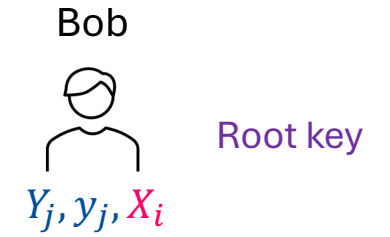
All messages  
are relayed by  
the server



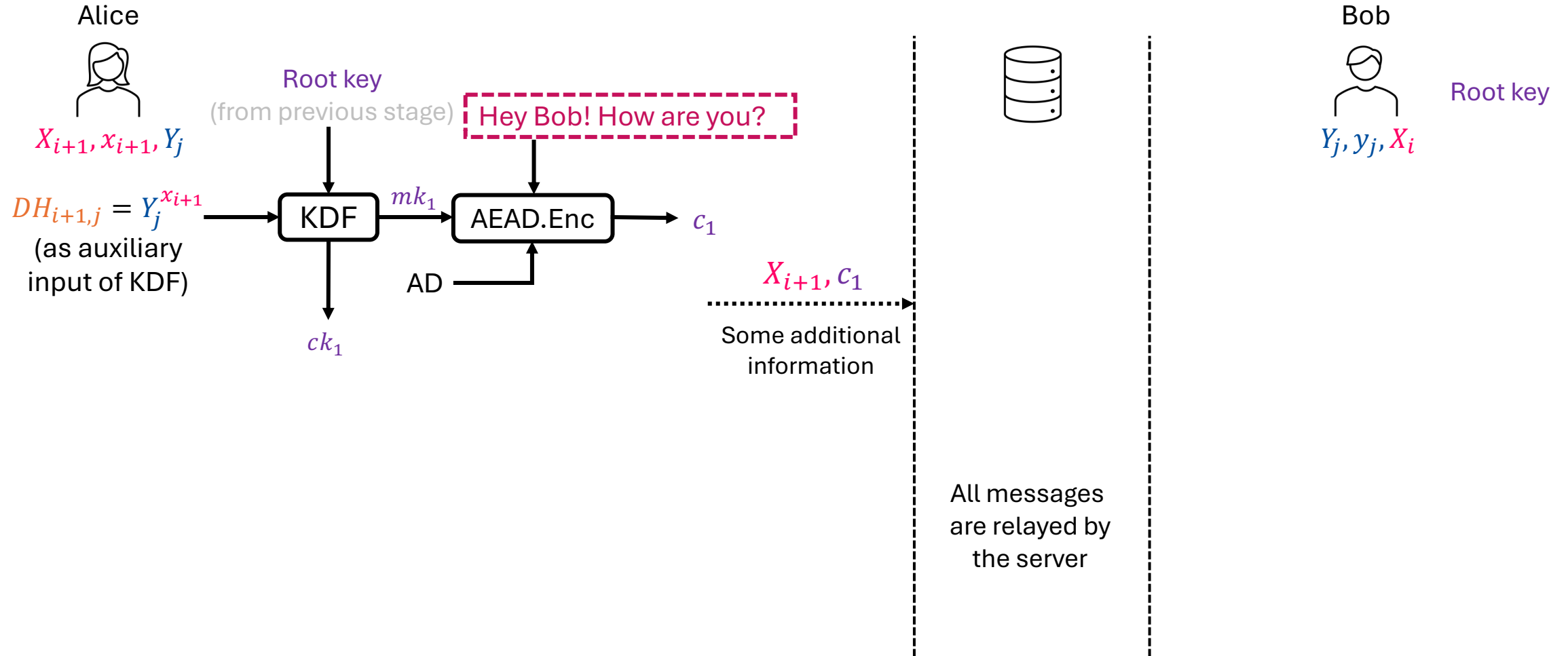
# Double Ratchet



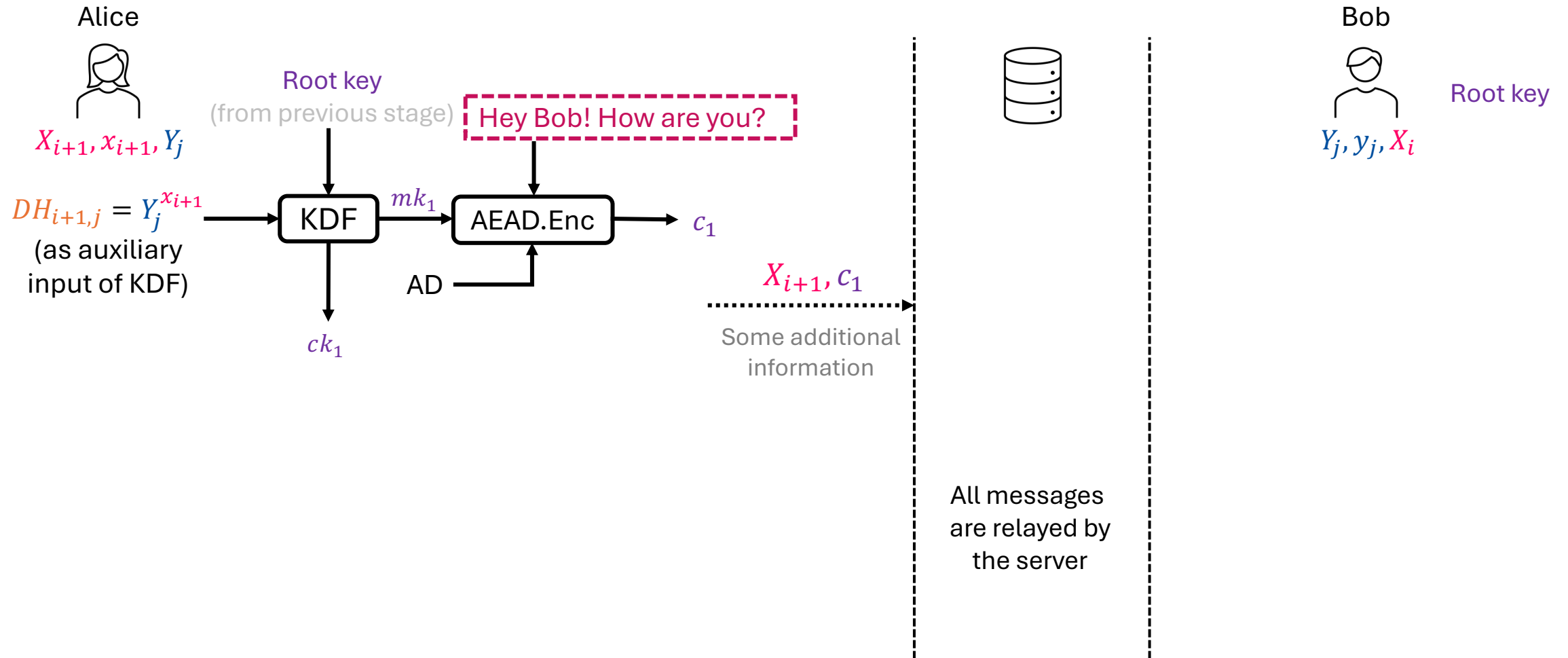
All messages  
are relayed by  
the server



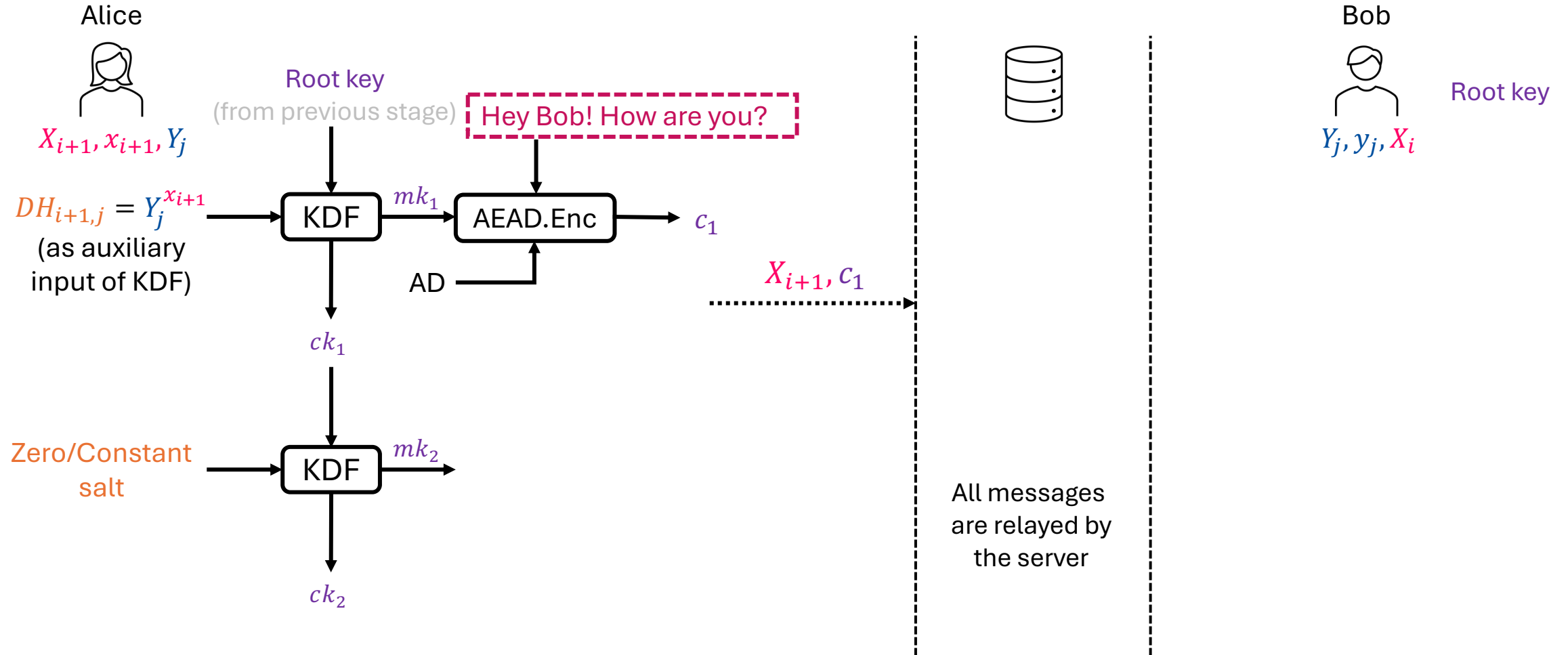
# Double Ratchet



# Double Ratchet

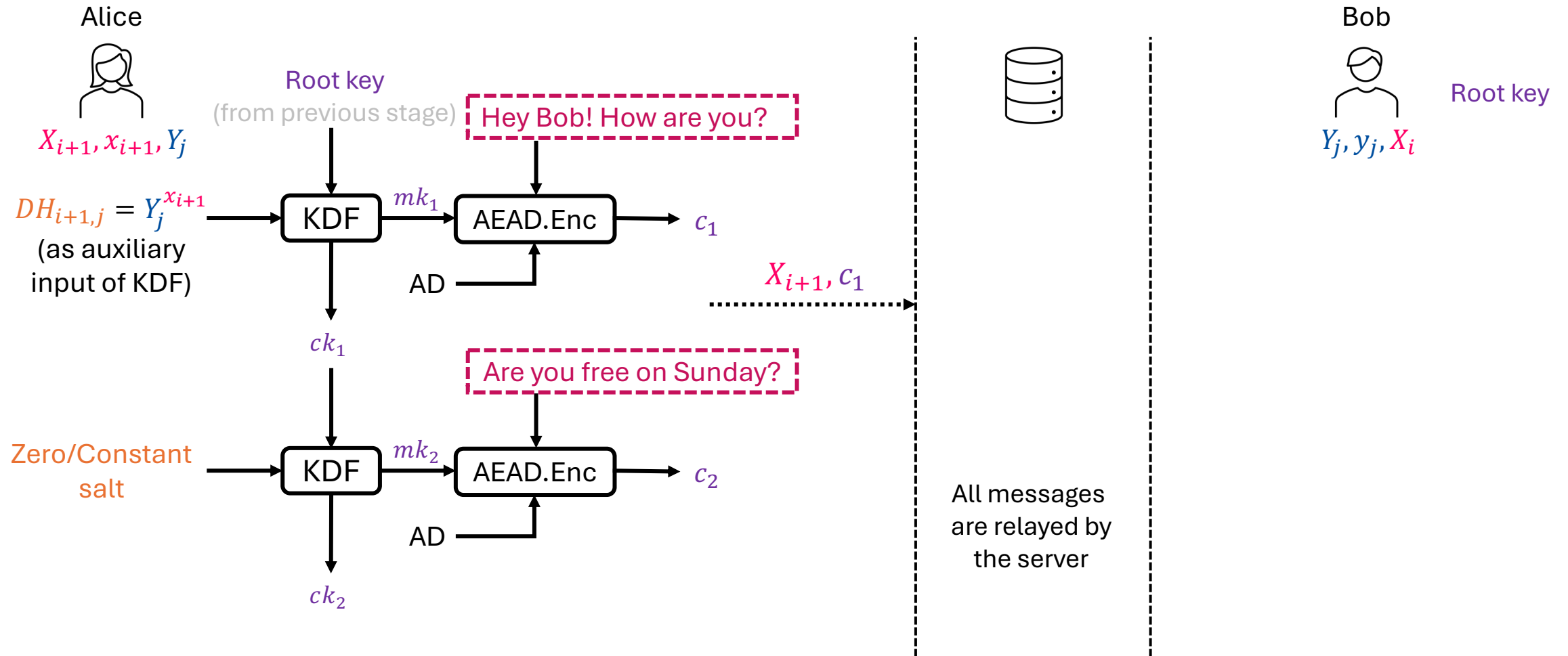


# Double Ratchet

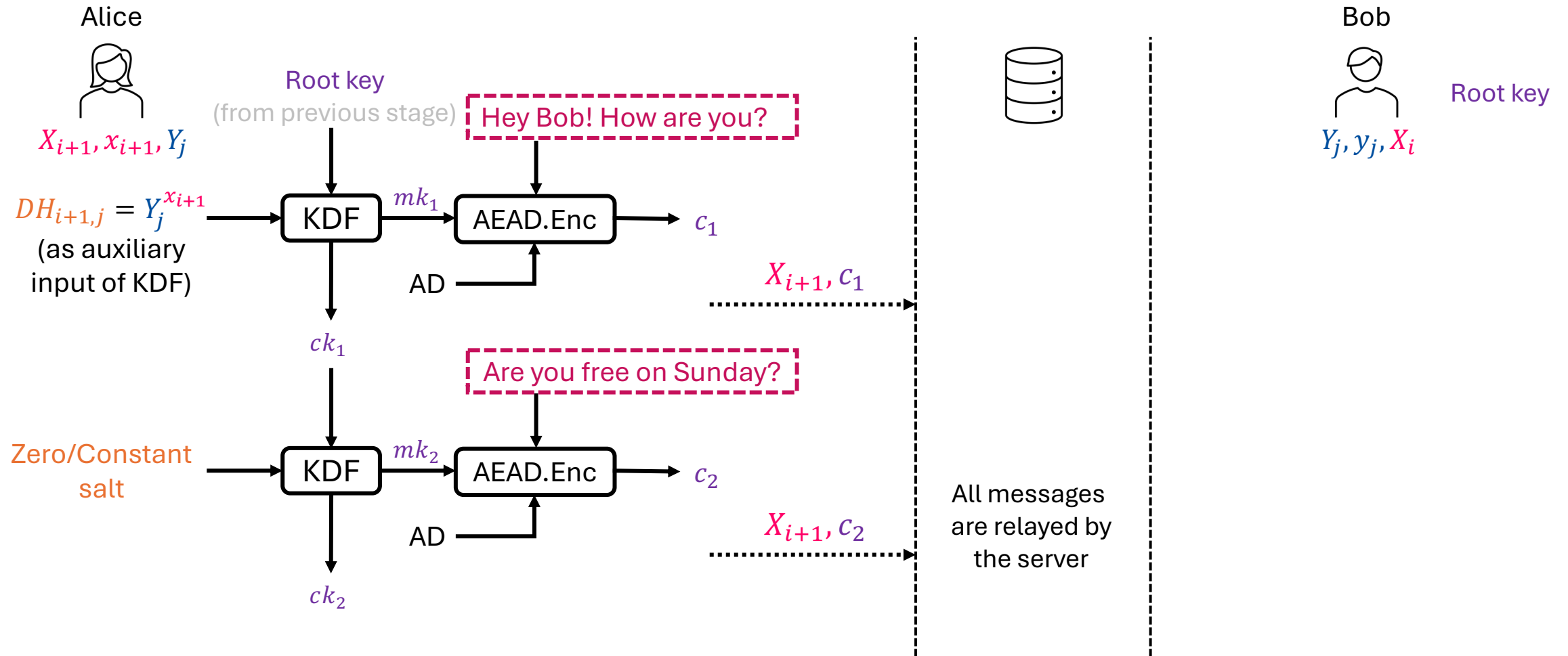




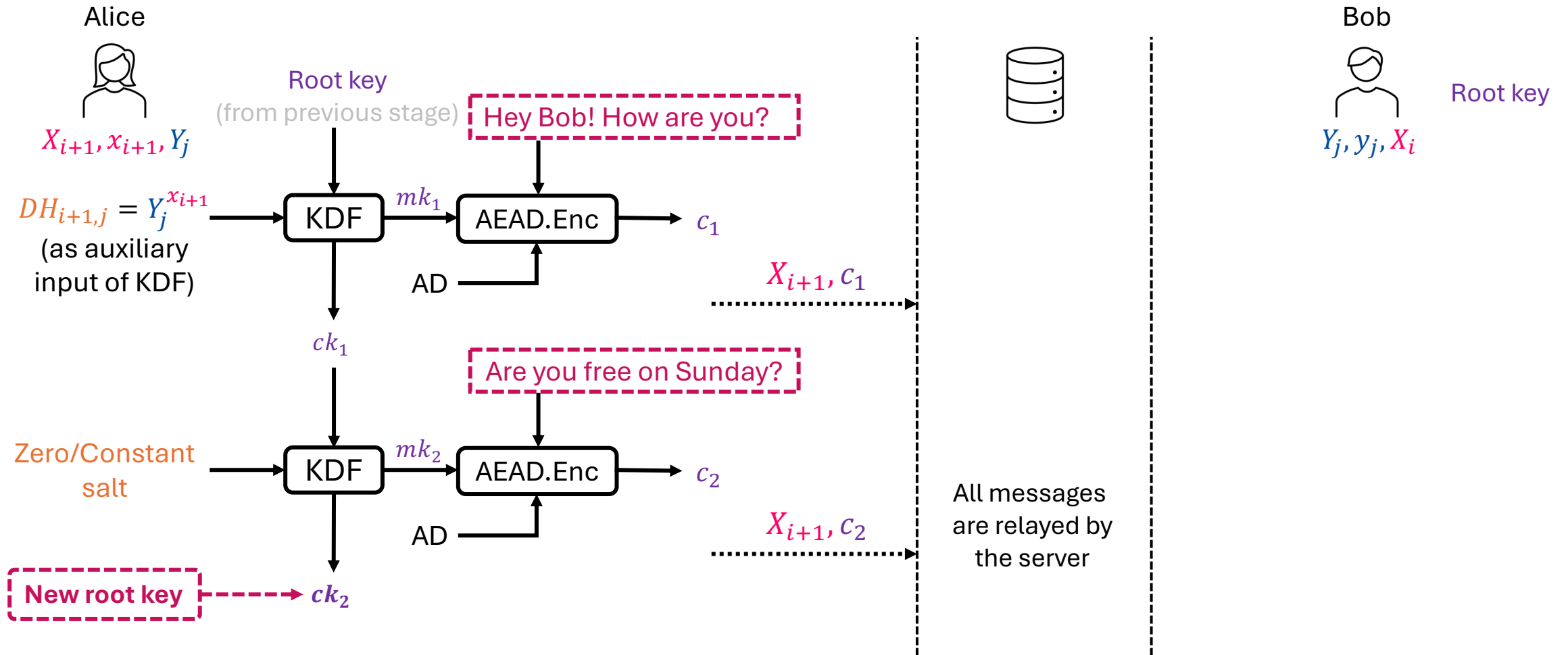
# Double Ratchet



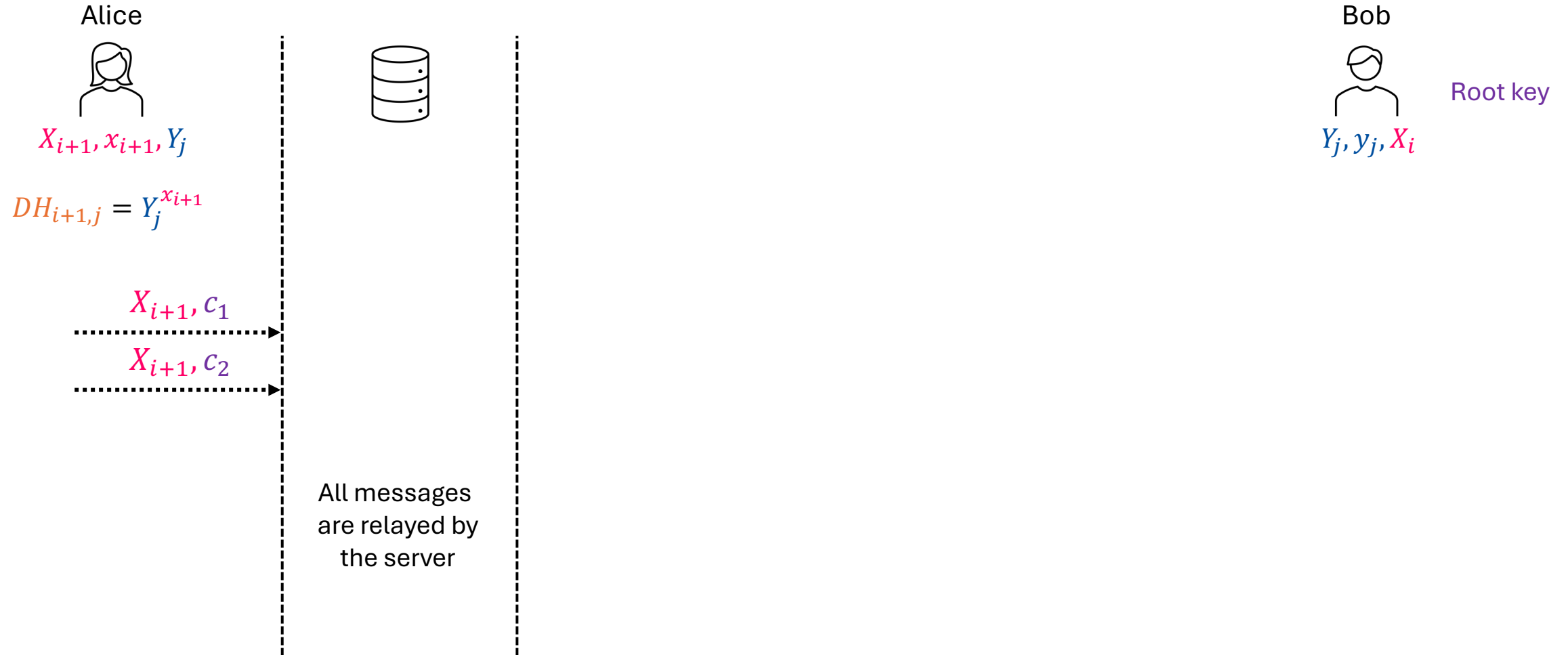
# Double Ratchet



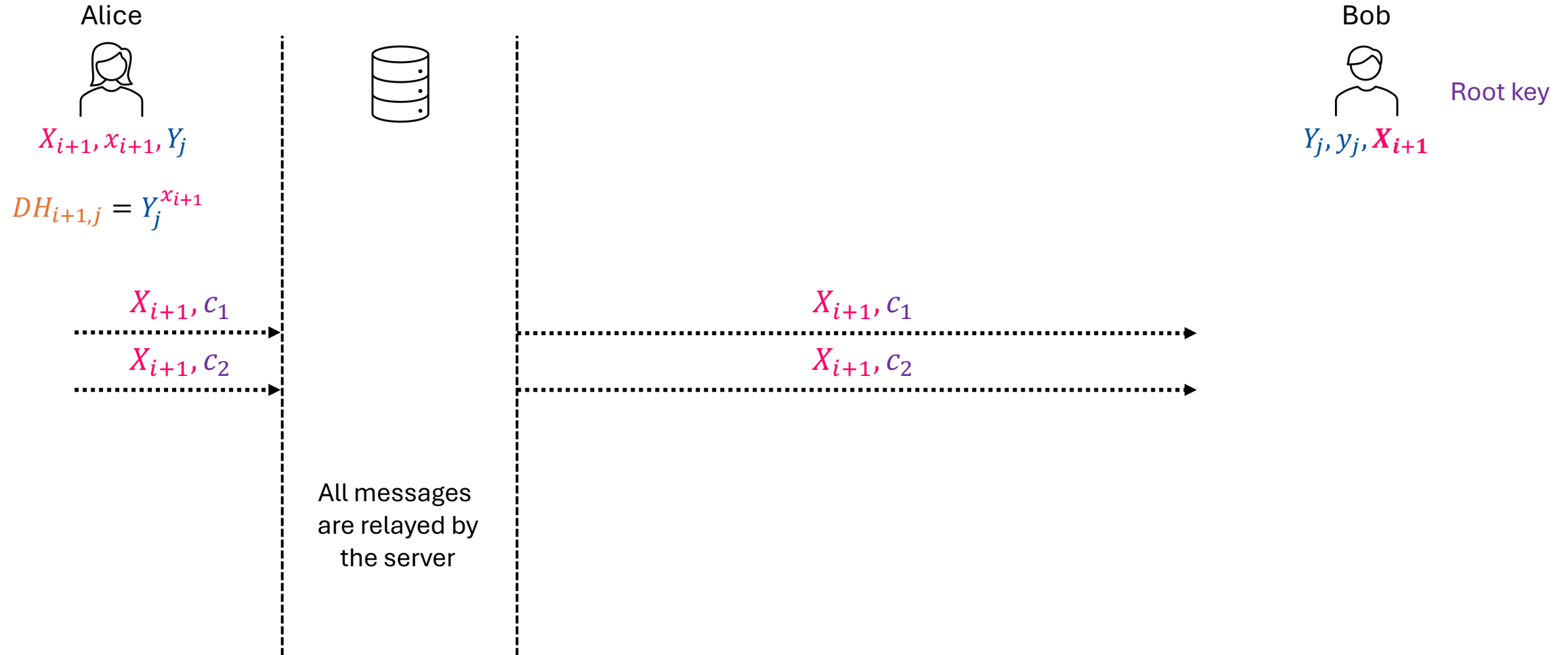
# Double Ratchet



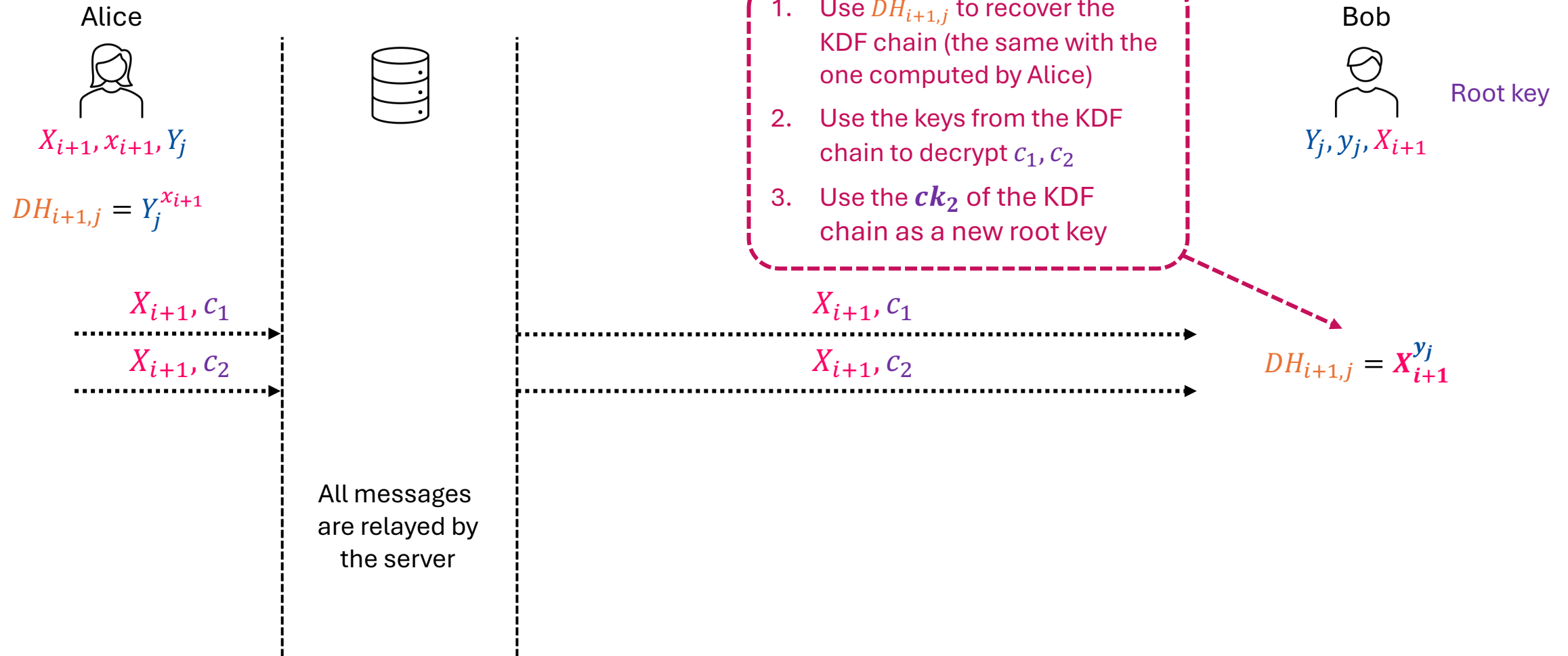
# Double Ratchet



# Double Ratchet



# Double Ratchet

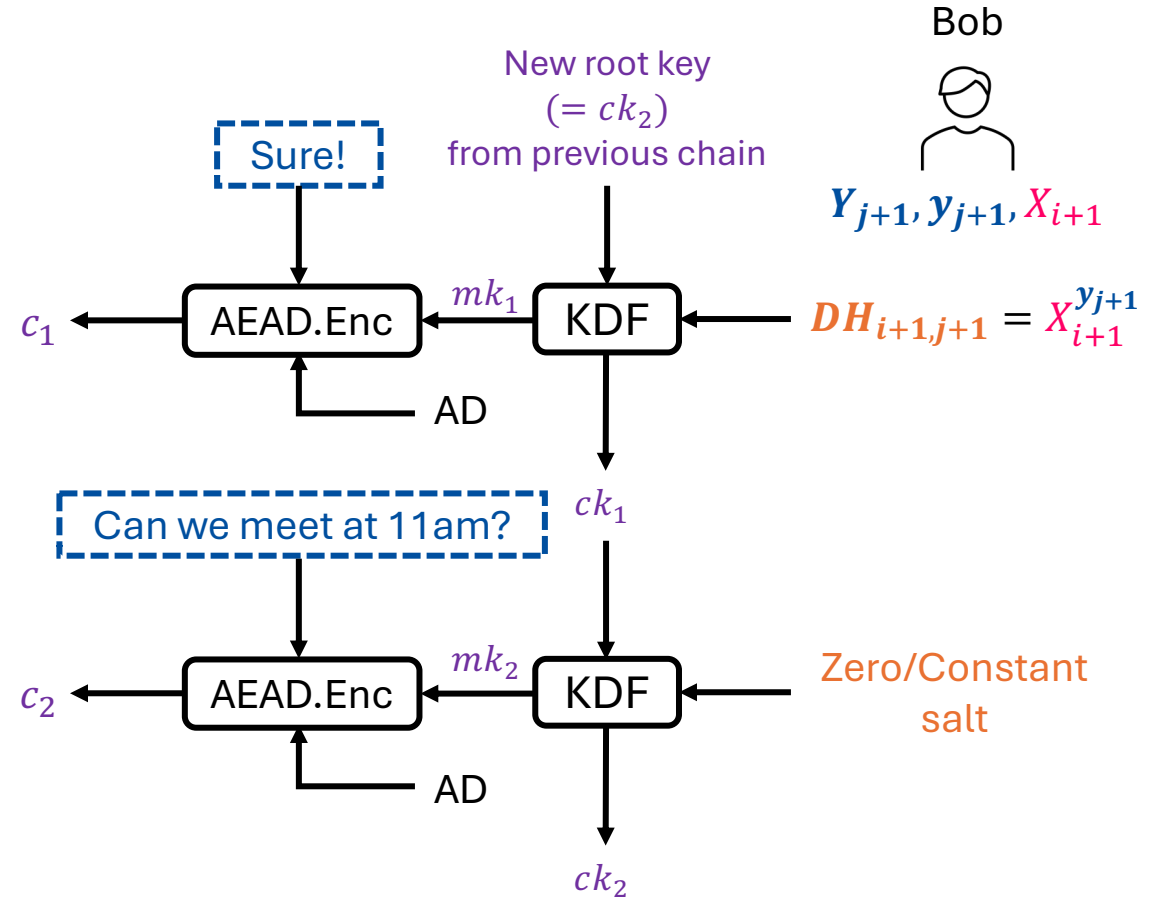


# Double Ratchet

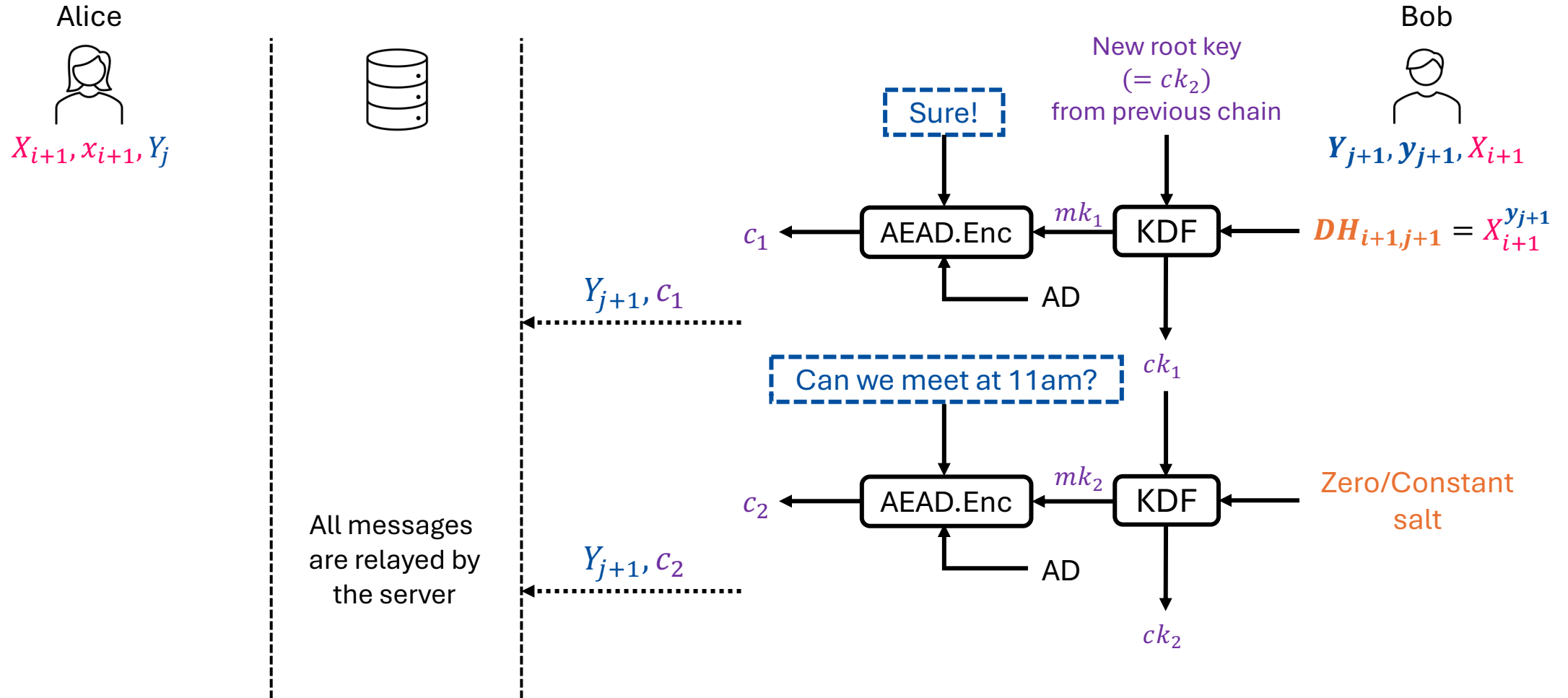
Alice  
 $X_{i+1}, x_{i+1}, Y_j$



All messages  
are relayed by  
the server




# Double Ratchet






# Double Ratchet

Alice  
  
 $X_{i+1}, x_{i+1}, Y_j$



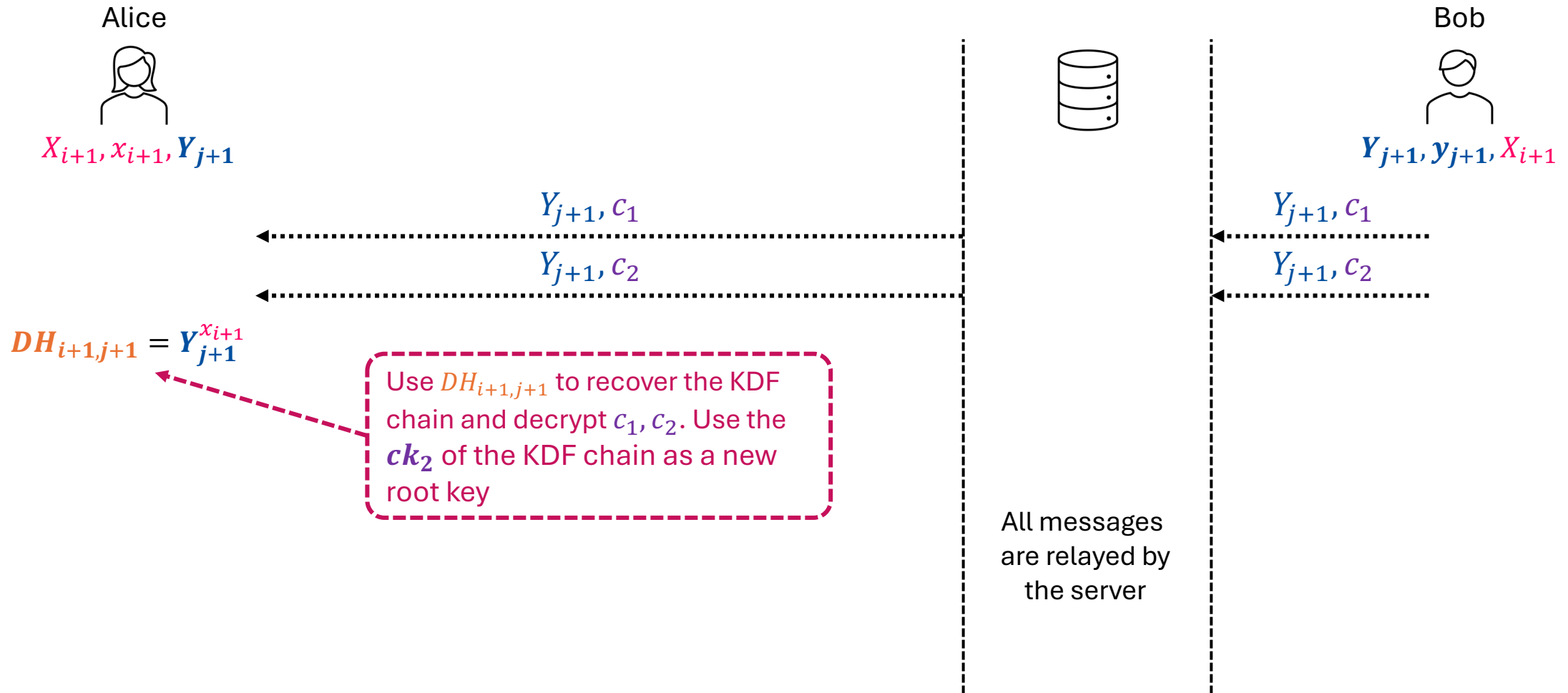
Bob  
  
 $Y_{j+1}, y_{j+1}, X_{i+1}$

$Y_{j+1}, c_1$

$Y_{j+1}, c_2$

All messages  
are relayed by  
the server

# Double Ratchet



# Double Ratchet

Alice



$X_{i+2}, x_{i+2}, Y_{j+1}$

Bob

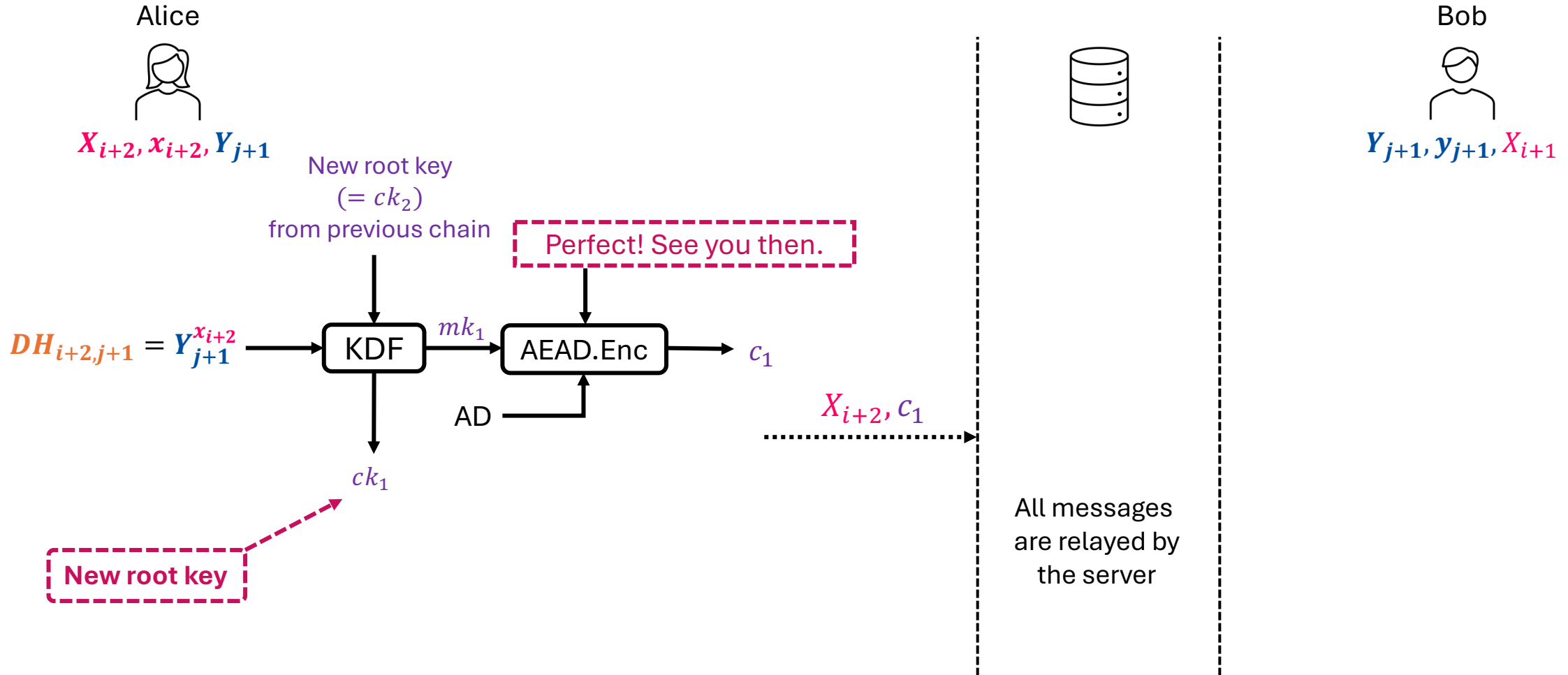


$Y_{j+1}, y_{j+1}, X_{i+1}$



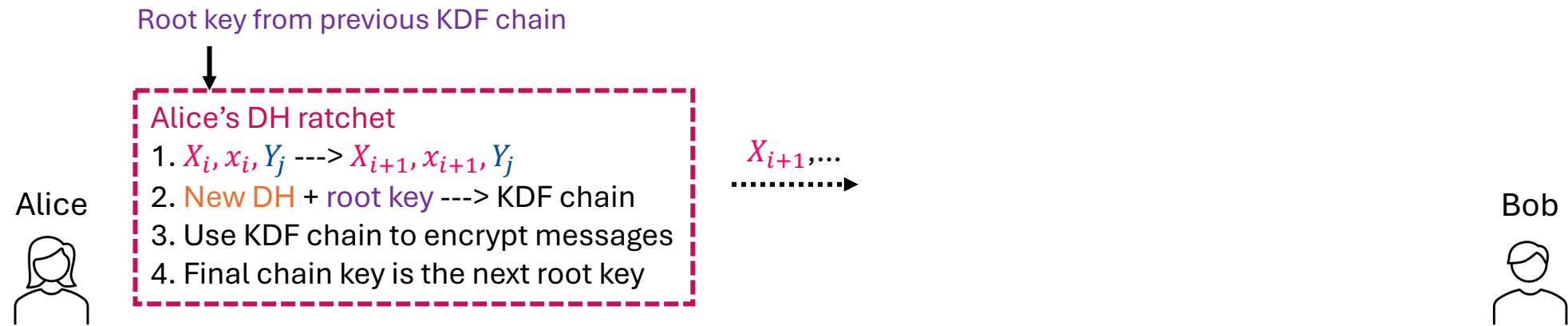
All messages  
are relayed by  
the server

# Double Ratchet



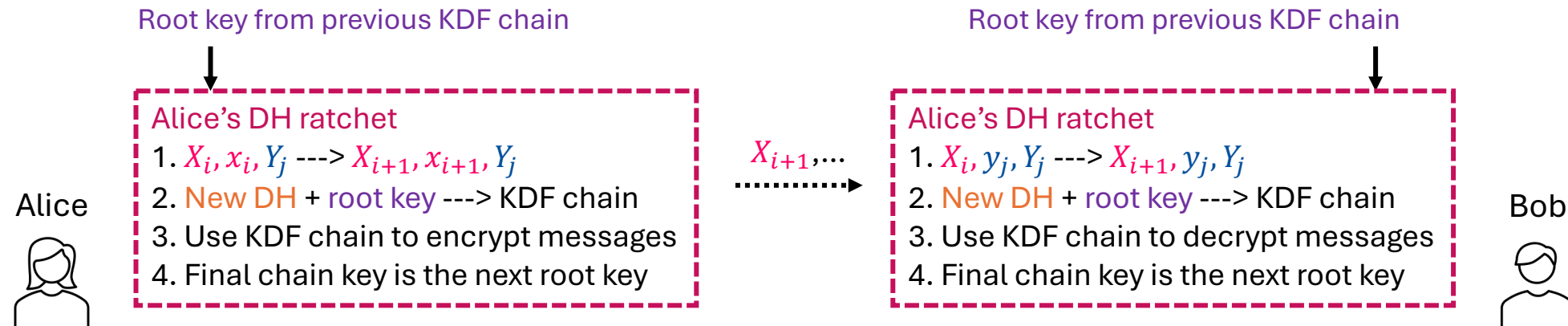
# Double Ratchet

- The main idea: Symmetric-key Ratchet + Diffie-Hellman Ratchet



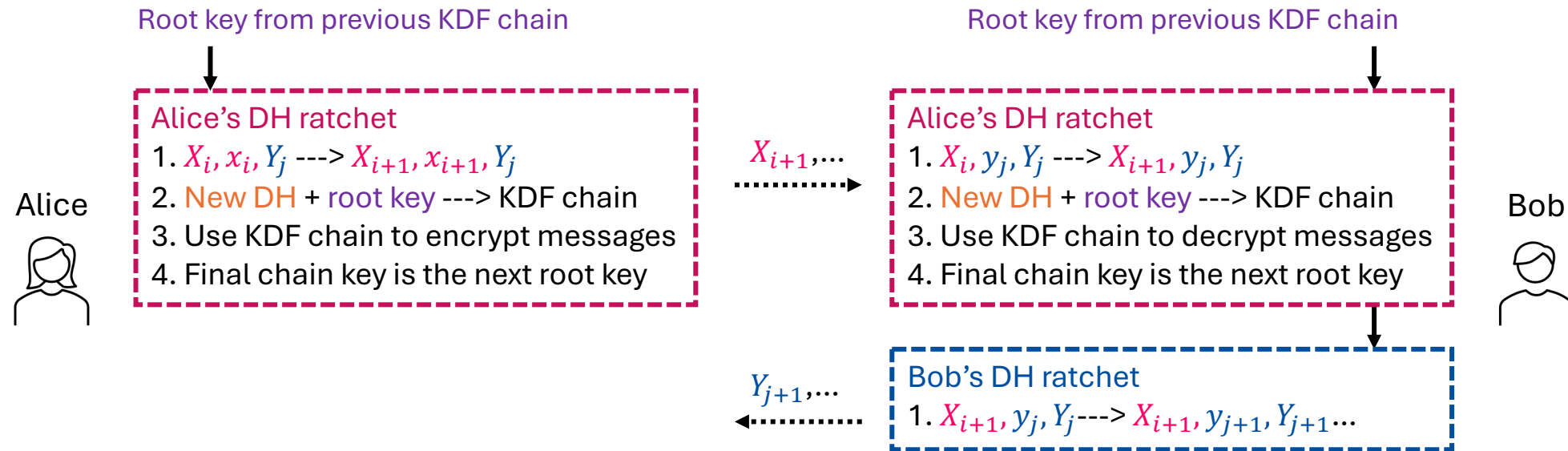
# Double Ratchet

- The main idea: Symmetric-key Ratchet + Diffie-Hellman Ratchet



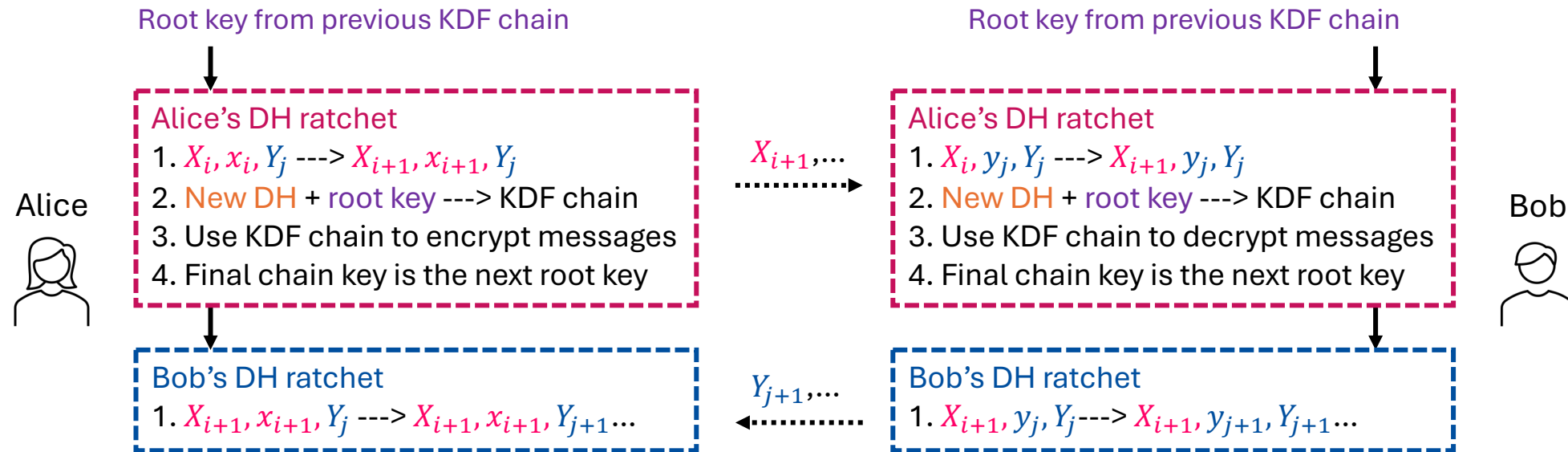
# Double Ratchet

- The main idea: Symmetric-key Ratchet + Diffie-Hellman Ratchet



# Double Ratchet

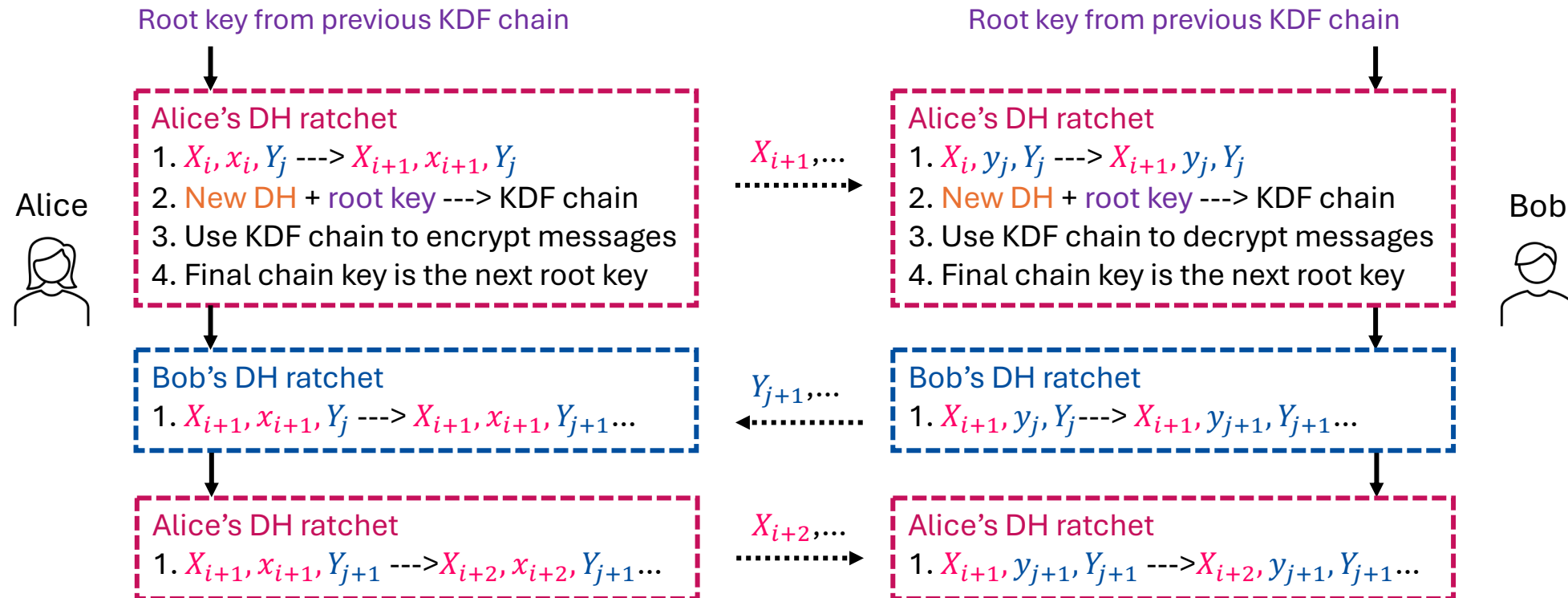
- The main idea: Symmetric-key Ratchet + Diffie-Hellman Ratchet





# Double Ratchet

- The main idea: Symmetric-key Ratchet + Diffie-Hellman Ratchet





# X3DH + Double Ratchet

- Integrate Double Ratchet algorithm with X3DH
  - Use X3DH to bootstrap Double Ratchet
  - The Double Ratchet plays the role of a ‘post-X3DH’ protocol...

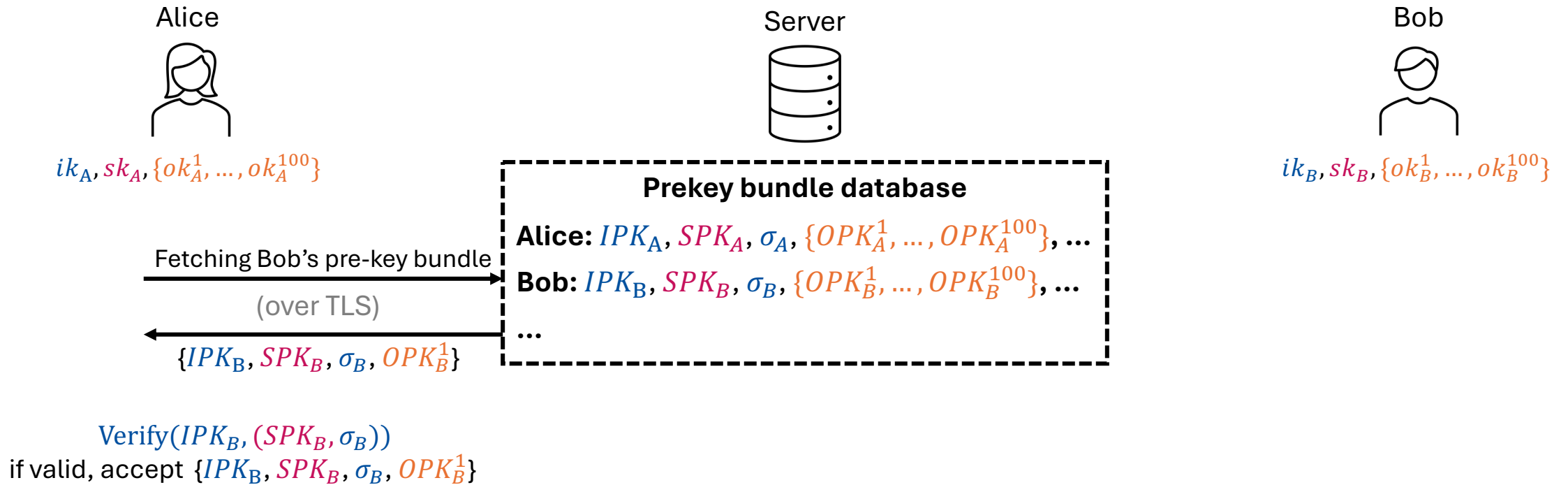
# X3DH + Double Ratchet

- Recall of X3DH:

	Public parameters: $(\mathbb{G}, g, q)$ : A $q$ -order EC group $\mathbb{G}$ with a generator $g$	Alice 	Bob 
Long-term secret (static)	Identity secret key (IK) Identity public key (IPK)	$ik_A \in_{\$} \mathbb{Z}_q$ $IPK_A (= g^{ik_A})$	$ik_B \in_{\$} \mathbb{Z}_q$ $IPK_B$
Mid-term secret (updated periodically)	Signing secret pre-key (SK) Signing public pre-key (SPK)	$sk_A \in_{\$} \mathbb{Z}_q$ $SPK_A$	$sk_B \in_{\$} \mathbb{Z}_q$ $SPK_B$
Short-term secret (used once)	One-time secret pre-keys (OK) One-time public pre-keys (OPK)	$\{ok_A^1, ok_A^2, \dots\} \subseteq_{\$} \mathbb{Z}_q$ $(OPK_A^1, OPK_A^2, \dots)$	$\{ok_B^1, ok_B^2, \dots\} \subseteq_{\$} \mathbb{Z}_q$ $(OPK_B^1, OPK_B^2, \dots)$

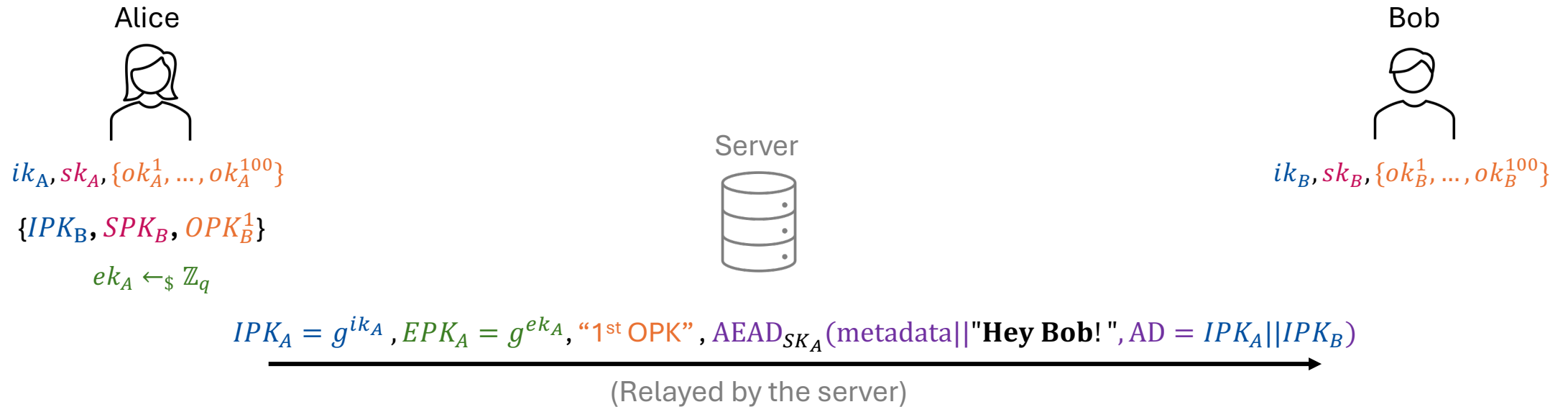
# X3DH + Double Ratchet

- Recall of X3DH:



# X3DH + Double Ratchet

- Recall of X3DH:

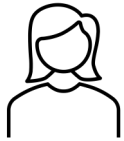


$$SK_A = \text{X3DH\_Key\_Alice}(ik_A, ek_A, IPK_B, SPK_B, OPK_B^1)$$

# X3DH + Double Ratchet

- Initialize Double Ratchet using the SK from X3DH

Alice



$ik_A, sk_A, \{ok_A^1, \dots, ok_A^{100}\}$

$\{IPK_B, SPK_B, OPK_B^1\}$

$SK_A = \text{X3DH\_Key\_Alice}(ik_A, ek_A, IPK_B, SPK_B, OPK_B^1)$

Bob



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

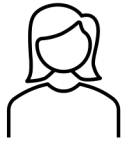
Root key =  $SK_A$

$X_0 = \perp, x_0 = \perp, Y_0 = SPK_B$

# X3DH + Double Ratchet

- Initialize Double Ratchet using the SK from X3DH

Alice



Bob



X3DH

$SK_A = \text{X3DH\_Key\_Alice}(\dots)$

**Alice's DH ratchet**

Root key =  $SK_A$

$X_0 = \perp, x_0 = \perp, Y_0 = SPK_B$  (Signing public pre-key of Bob)

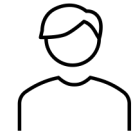
# X3DH + Double Ratchet

- Initialize Double Ratchet using the SK from X3DH

Alice



Bob



X3DH

$$SK_A = \text{X3DH\_Key\_Alice}(\dots)$$

---

## Alice's DH ratchet

$$\text{Root key} = SK_A$$

$$X_0 = \perp, x_0 = \perp, Y_0 = SPK_B \text{ (Signing public pre-key of Bob)}$$

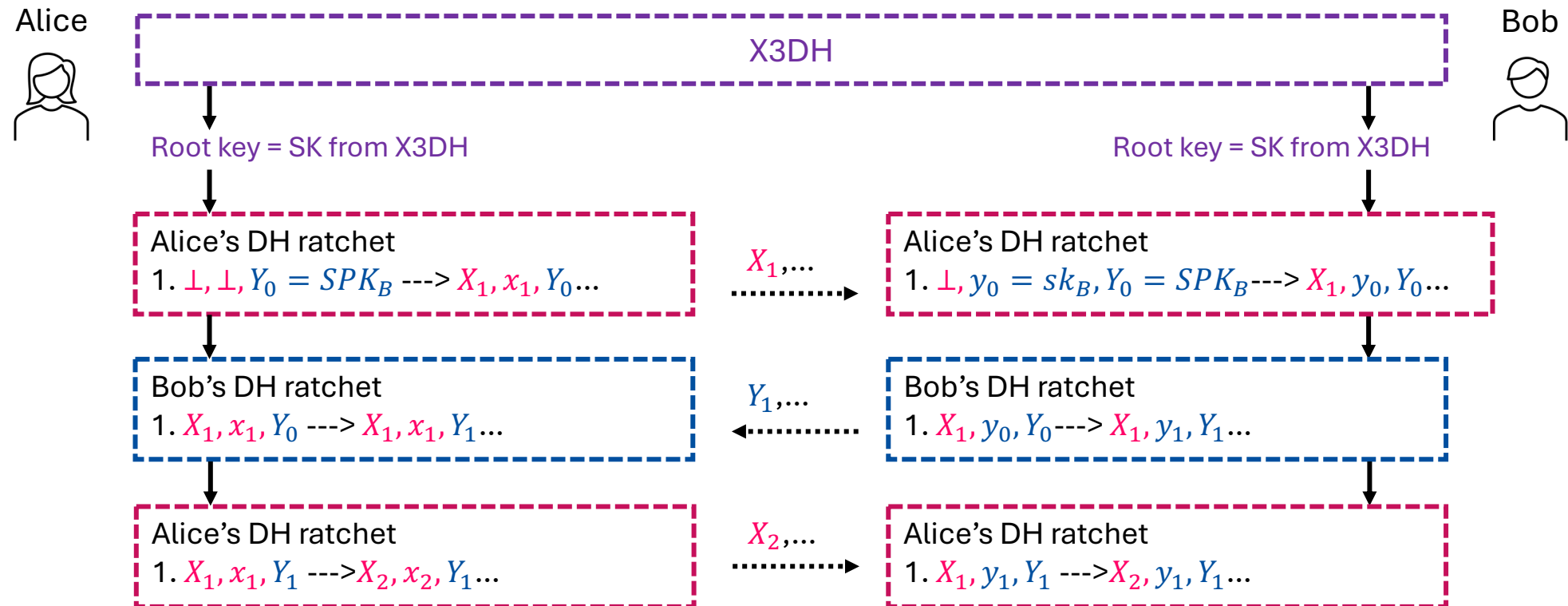
$$X_1 = g^{x_1}, x_1 \leftarrow_{\$} \mathbb{Z}_q, DH_{1,0} = Y_0^{x_1}$$

Use  $DH_{1,0}$  to derive a KDF chain to encrypt messages...

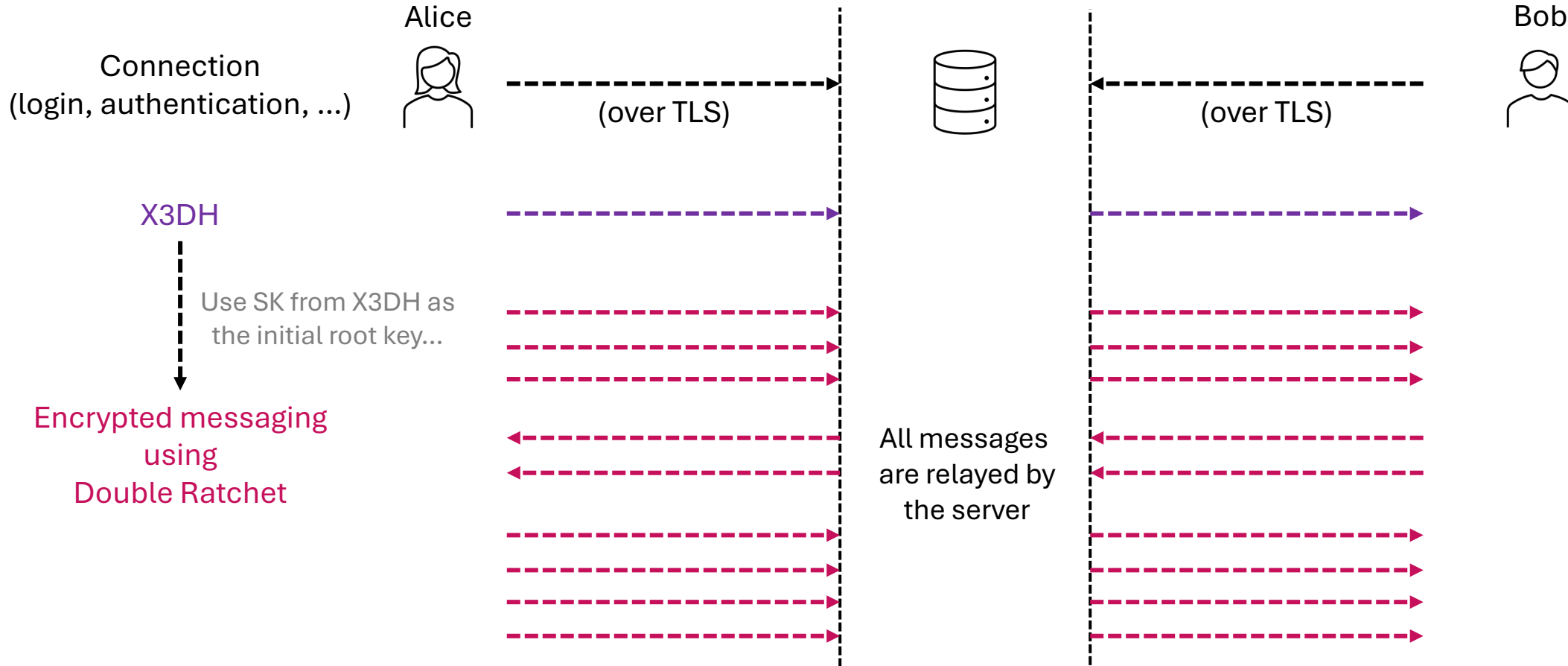


# Double Ratcheting

- Initialize Double Ratchet using the SK from X3DH



# Signal Secure Messaging Protocol

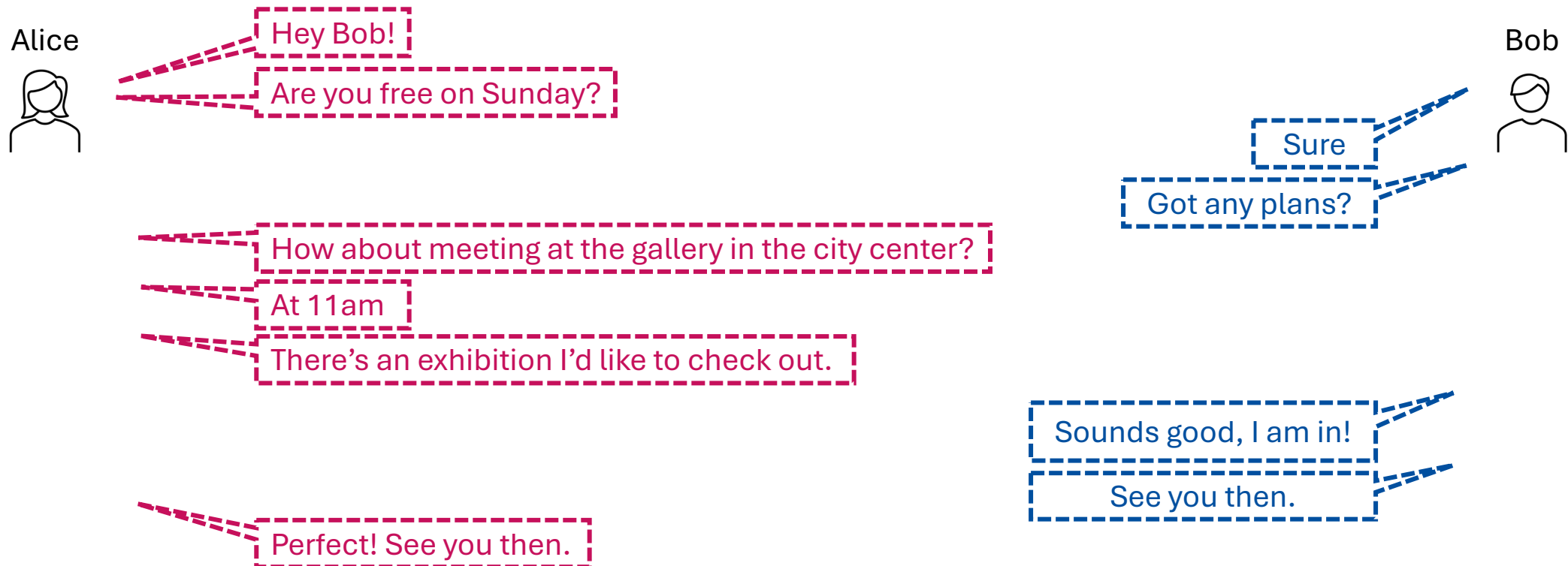


# Signal Secure Messaging Protocol

- Some technical details we do not cover:
  - XEdDSA and VEdDSA:
    - DH key pairs for key exchange and signature...
  - Header encryption:
    - Cannot tell which messages belong to which sessions, or the ordering of messages within a session...
  - Out-of-order messages:
  - Session management and asynchronous settings

# Coding tasks

- (Without sockets) Use X3DH and Double Ratchet to encrypt this conversation (or you can choose other conversations):



# Further Reading

- Technical Documentations of Signal: <https://signal.org/docs/>
- Some research papers of analyzing security of Ratchet algorithms:
  - Bellare et al's work on formalizing ratcheted encryption/key exchange: <https://eprint.iacr.org/2016/1028>
  - Alwen et al's work on formalizing Double Ratchet: <https://eprint.iacr.org/2018/1037>
  - Collins et al's work on Tight security of Double Ratchet: <https://eprint.iacr.org/2024/1625>
  - ...