Cryptography Engineering

- Lecture 6 (Nov 27, 2024)

- Today's notes:
  - Double Ratchet Algorithm
  - Signal Secure Messaging Protocol
  - Introduction to Password Login

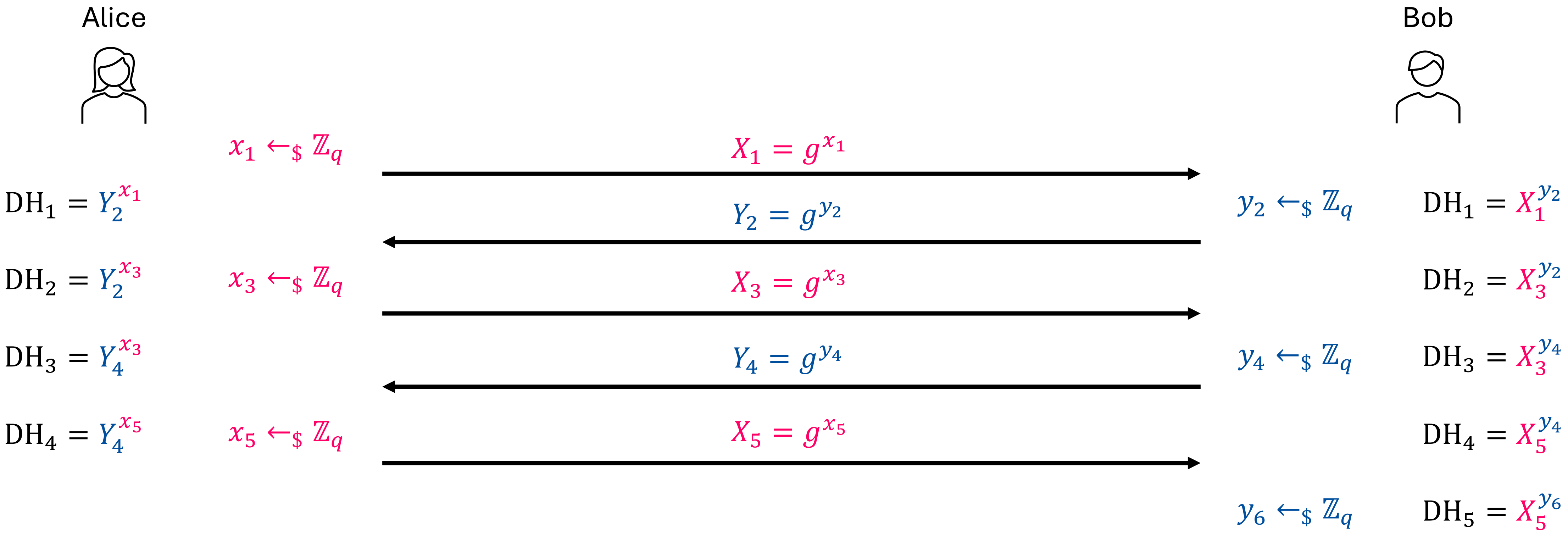
- No homework

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*...



Alice

Bob

$x_1 \leftarrow_{\$} \mathbb{Z}_q$ $\xrightarrow{X_1 = g^{x_1}}$

$\mathrm{DH}_1 = Y_2^{x_1}$ $\xleftarrow{Y_2 = g^{y_2}}$ $y_2 \leftarrow_{\$} \mathbb{Z}_q$ $\mathrm{DH}_1 = X_1^{y_2}$

$\mathrm{DH}_2 = Y_2^{x_3}$ $x_3 \leftarrow_{\$} \mathbb{Z}_q$ $\xrightarrow{X_3 = g^{x_3}}$ $\mathrm{DH}_2 = X_3^{y_2}$

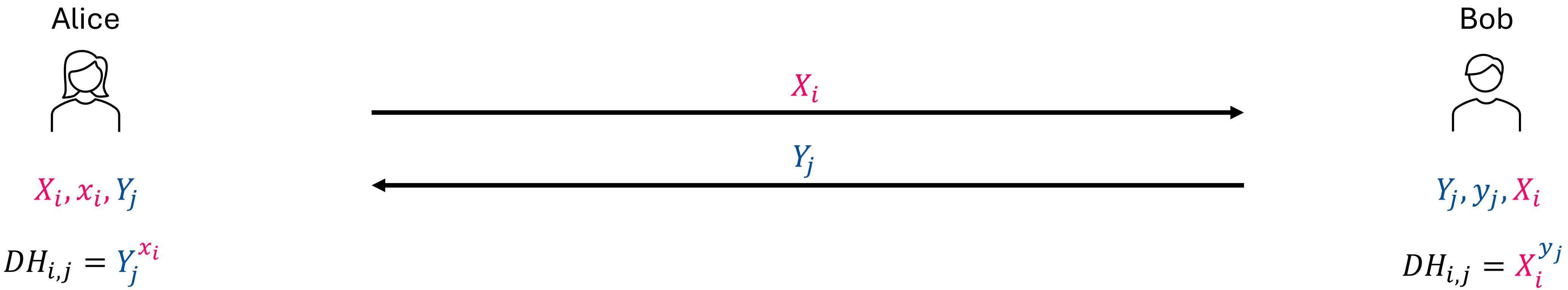
$\mathrm{DH}_3 = Y_4^{x_3}$ $\xleftarrow{Y_4 = g^{y_4}}$ $y_4 \leftarrow_{\$} \mathbb{Z}_q$ $\mathrm{DH}_3 = X_3^{y_4}$

$\mathrm{DH}_4 = Y_4^{x_5}$ $x_5 \leftarrow_{\$} \mathbb{Z}_q$ $\xrightarrow{X_5 = g^{x_5}}$ $\mathrm{DH}_4 = X_5^{y_4}$

$y_6 \leftarrow_{\$} \mathbb{Z}_q$ $\mathrm{DH}_5 = X_5^{y_6}$

Double Ratchet – DH Ratchet

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



Alice

Bob

X_i

Y_j

X_i, x_i, Y_j

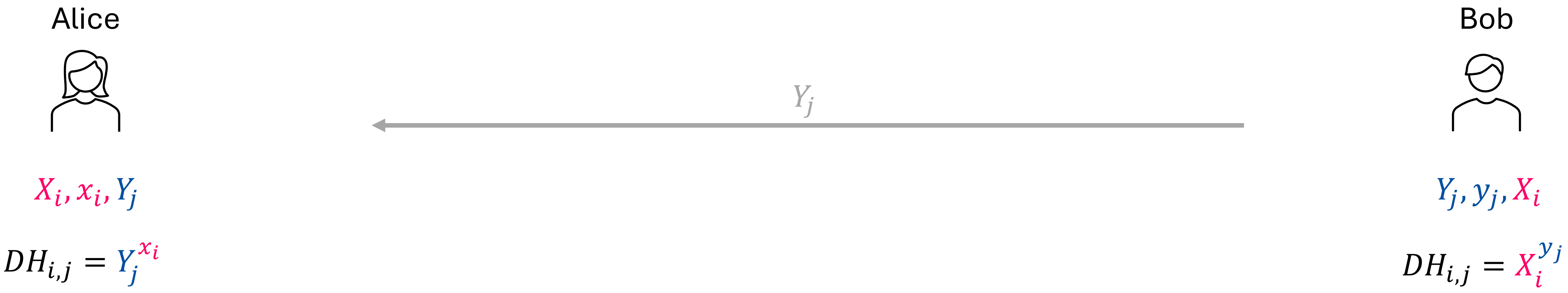
$DH_{i,j} = Y_j^{x_i}$

Y_j, y_j, X_i

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

Double Ratchet – DH Ratchet

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



Alice

Bob

Y_j

X_i, x_i, Y_j

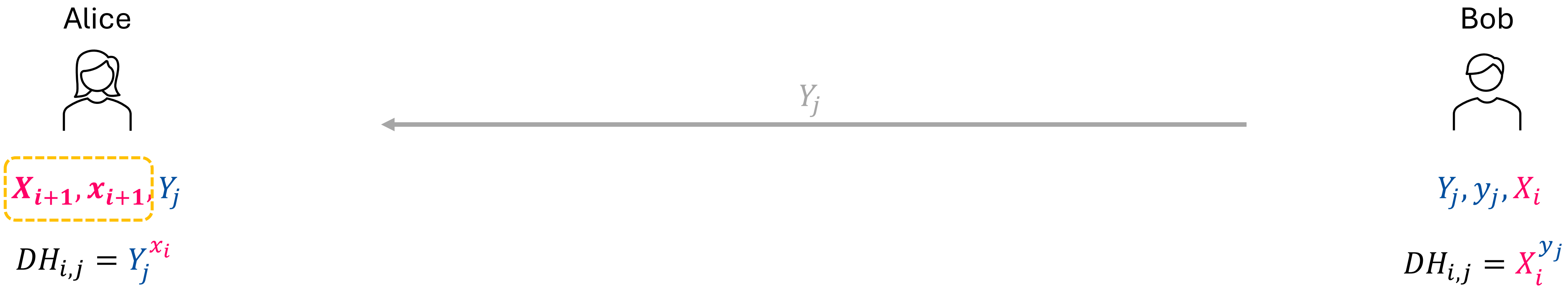
$DH_{i,j} = Y_j^{x_i}$

Y_j, y_j, X_i

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

Double Ratchet – DH Ratchet

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



Alice

Y_j

Bob

$\boxed{\mathbf{X_{i+1}}, \mathbf{x_{i+1}},} Y_j$

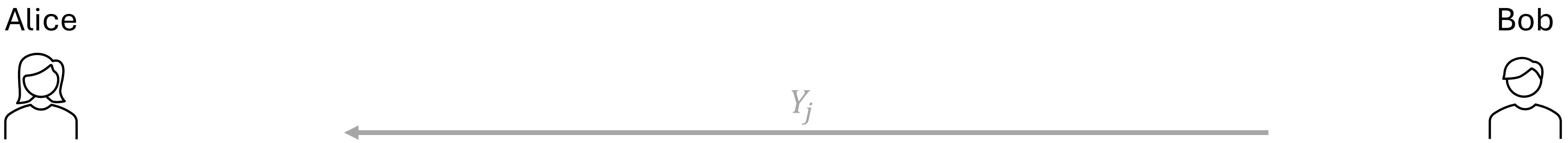
$DH_{i,j} = Y_j^{x_i}$

Y_j, y_j, X_i

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

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

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

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

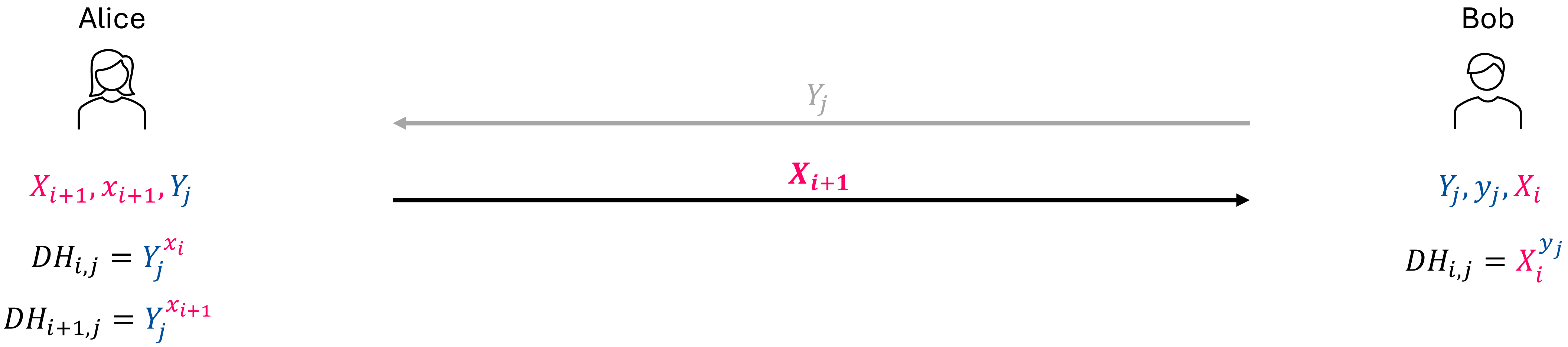
Bob:

Y_j, y_j, X_i

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

Double Ratchet – DH Ratchet

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



Alice

Bob

Y_j

$\boldsymbol{X_{i+1}}$

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

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

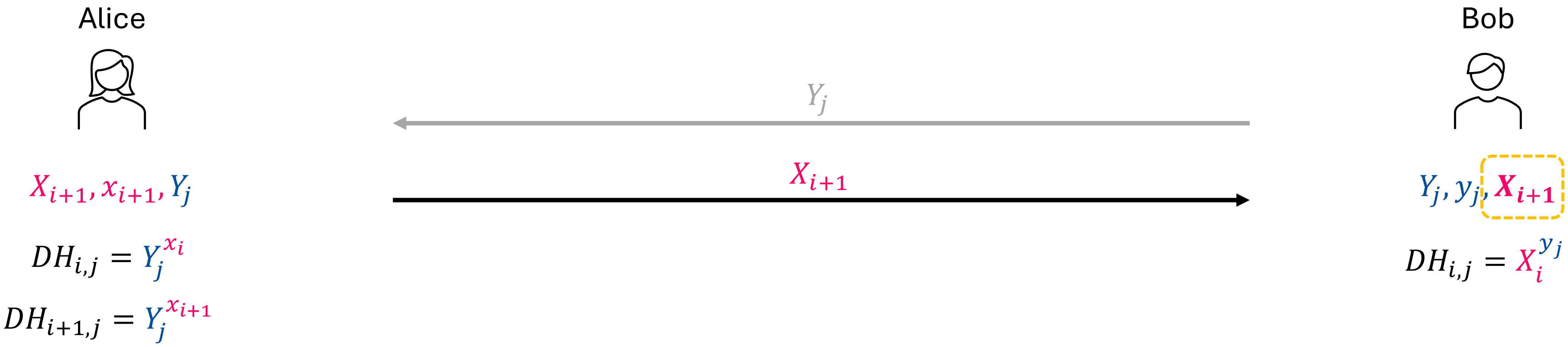
$DH_{i+1,j} = Y_j^{x_{i+1}}$

Y_j, y_j, X_i

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

Double Ratchet – DH Ratchet

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



Alice

Bob

Y_j

X_{i+1}

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

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

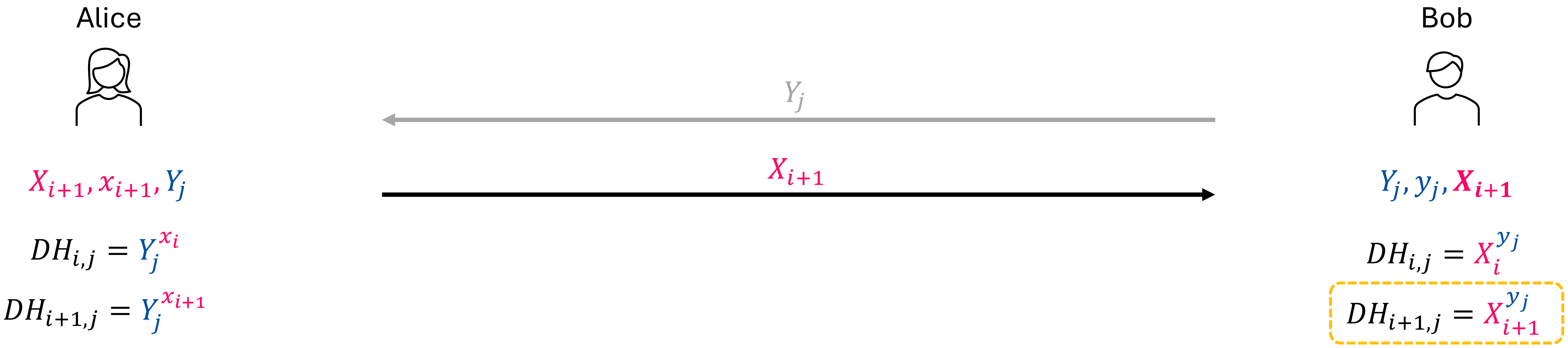
$DH_{i+1,j} = Y_j^{x_{i+1}}$

$Y_j, y_j, \boxed{\boldsymbol{X_{i+1}}}$

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

Double Ratchet – DH Ratchet

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



Alice

Bob

Y_j

X_{i+1}

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

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

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

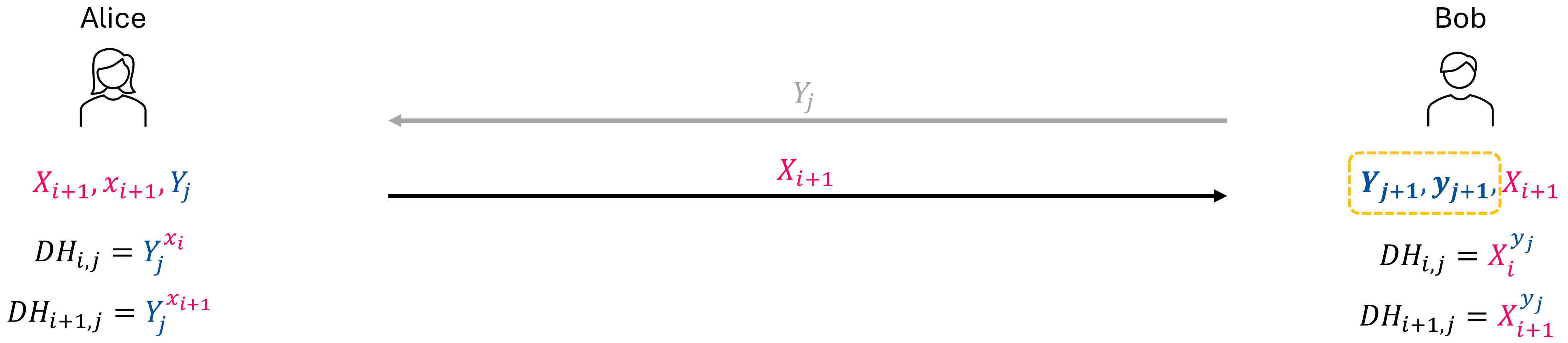
$Y_j, y_j, \boldsymbol{X_{i+1}}$

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

$\boxed{DH_{i+1,j} = X_{i+1}^{y_j}}$

Double Ratchet – DH Ratchet

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



Alice

Bob

Y_j

X_{i+1}

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

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

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

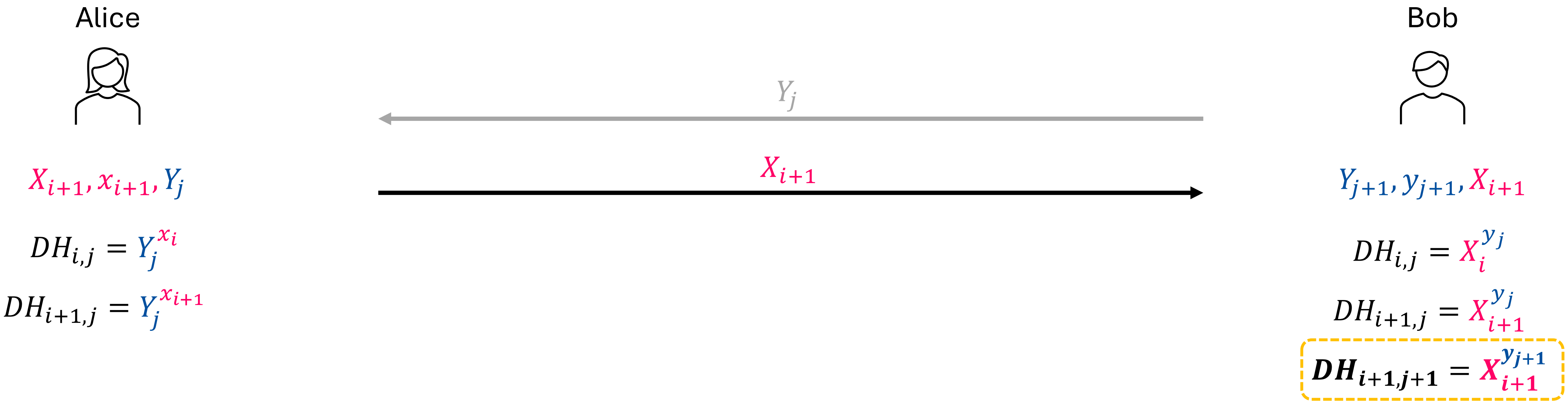
$\boxed{\boldsymbol{Y_{j+1}, y_{j+1},}} X_{i+1}$

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

$DH_{i+1,j} = X_{i+1}^{y_j}$

Double Ratchet – DH Ratchet

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



Alice

Bob

Y_j

X_{i+1}

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

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

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

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

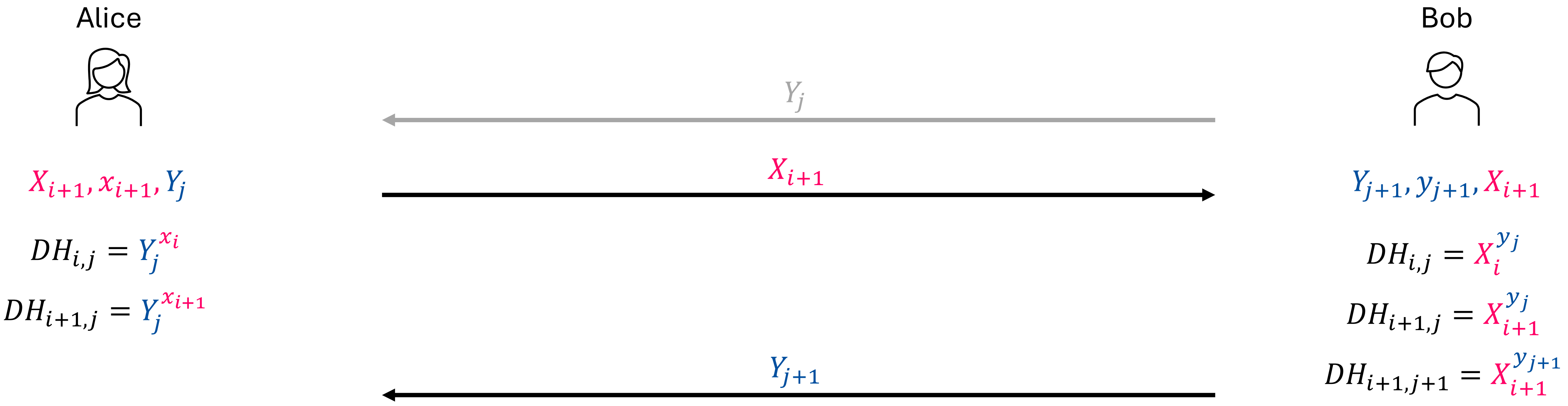
$DH_{i,j} = X_i^{y_j}$

$DH_{i+1,j} = X_{i+1}^{y_j}$

$\boxed{\boldsymbol{DH_{i+1,j+1} = X_{i+1}^{y_{j+1}}}}$

Double Ratchet – DH Ratchet

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



Alice

Bob

Y_j

X_{i+1}

Y_{j+1}

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

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

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

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

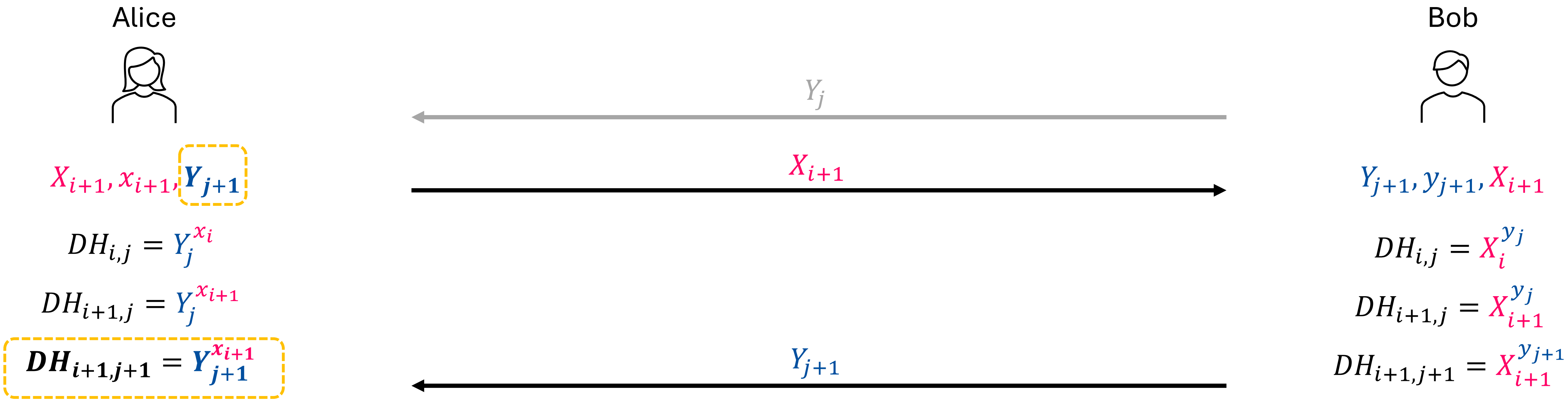
$DH_{i,j} = X_i^{y_j}$

$DH_{i+1,j} = X_{i+1}^{y_j}$

$DH_{i+1,j+1} = X_{i+1}^{y_{j+1}}$

Double Ratchet – DH Ratchet

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



Y_j

$X_{i+1}, x_{i+1}, \boxed{\boldsymbol{Y_{j+1}}}$

X_{i+1}

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

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

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

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

$DH_{i+1,j} = X_{i+1}^{y_j}$

$\boxed{\boldsymbol{DH_{i+1,j+1} = Y_{j+1}^{x_{i+1}}}}$

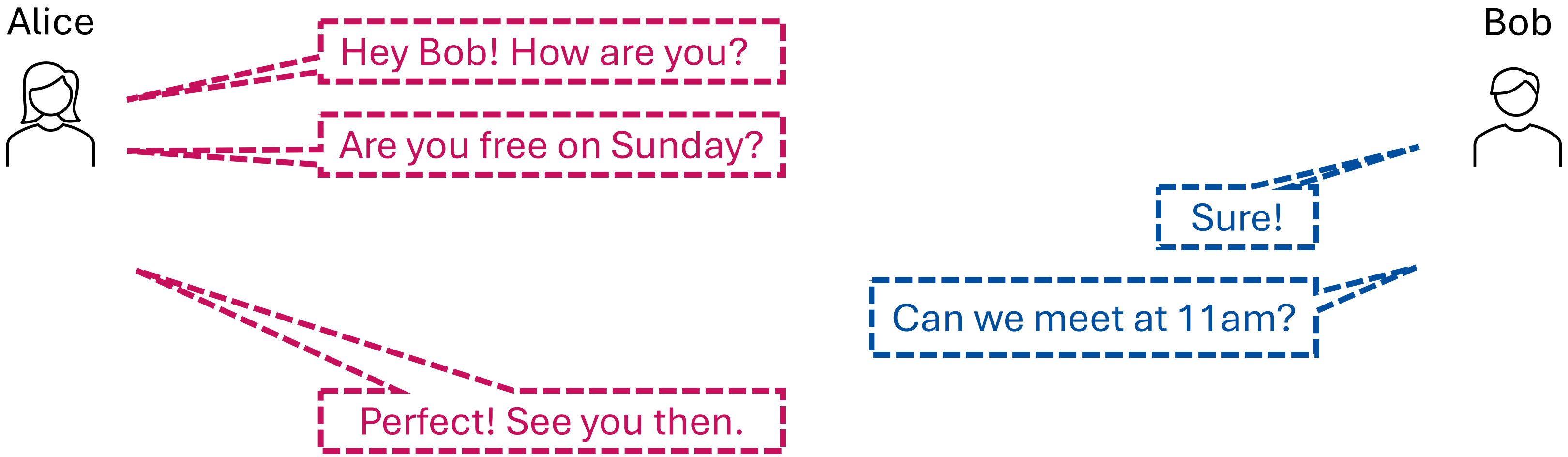
Y_{j+1}

$DH_{i+1,j+1} = X_{i+1}^{y_{j+1}}$

Double Ratchet

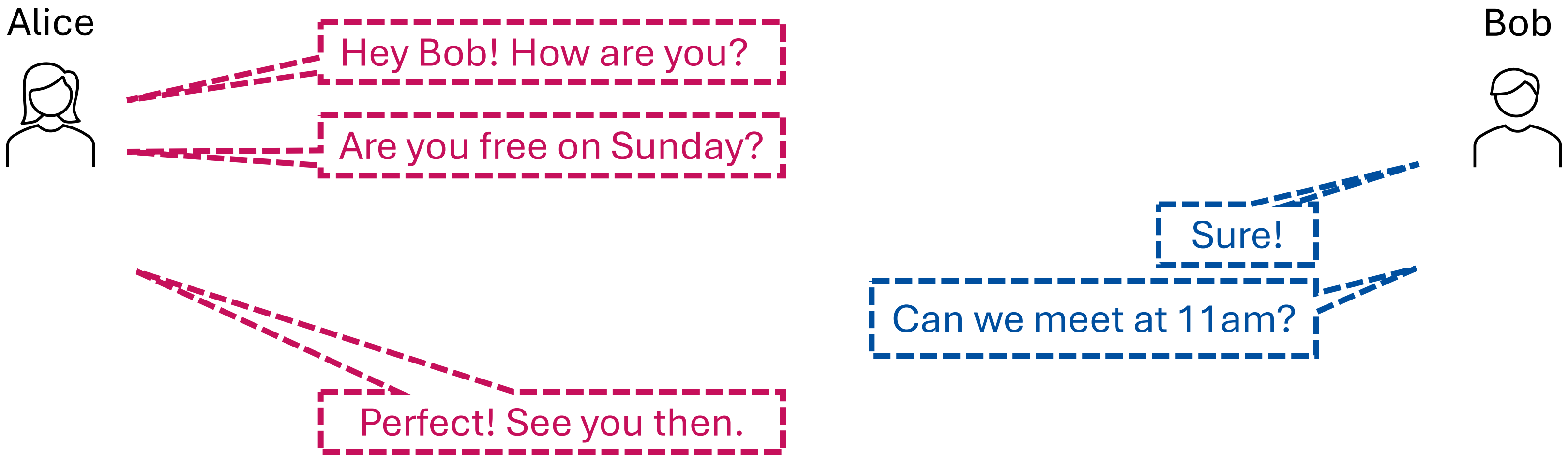
- The main idea: Symmetric-key Ratchet + Diffie-Hellman Ratchet
  - When a party sends messages (**before** its peer party replies): Use Symmetric-key Ratchet...
  - When the peer party replies: Use Diffie-Hellman Ratchet to update the key...

- Example:



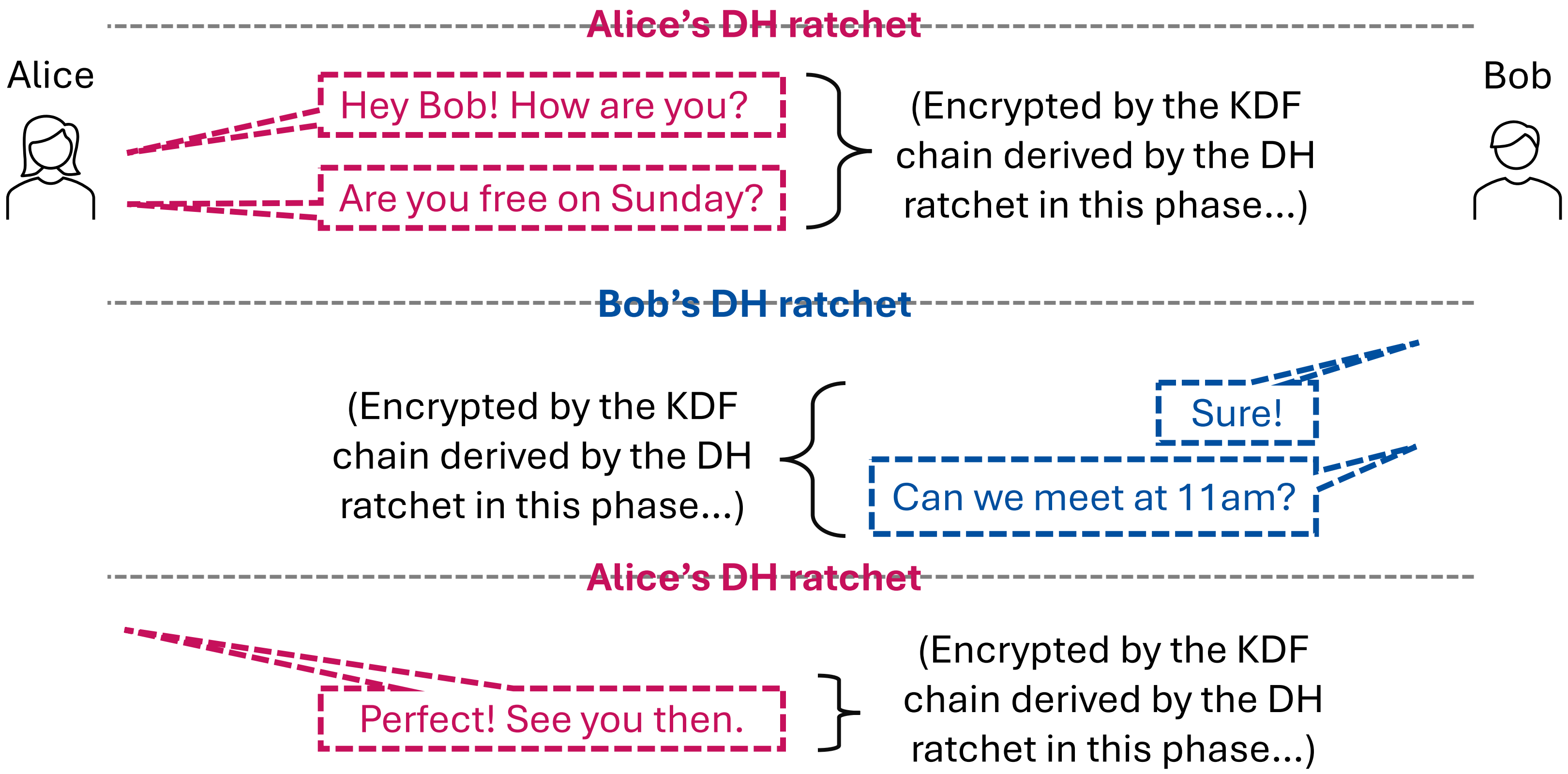
Double Ratchet

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



Double Ratchet

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



Alice's DH ratchet

Alice

Bob

Hey Bob! How are you?

Are you free on Sunday?

(Encrypted by the KDF chain derived by the DH ratchet in this phase...)

Bob's DH ratchet

Sure!

Can we meet at 11am?

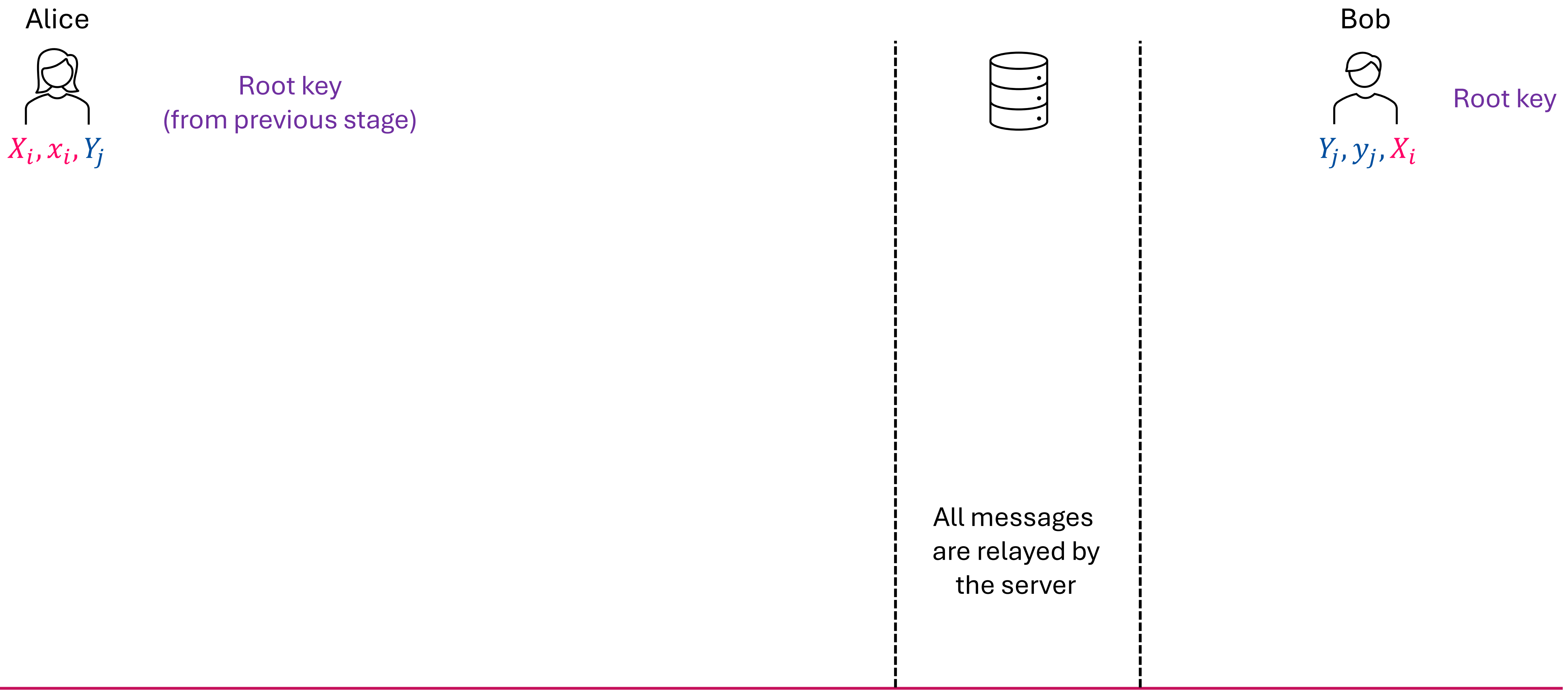
(Encrypted by the KDF chain derived by the DH ratchet in this phase...)

Alice's DH ratchet

Perfect! See you then.

(Encrypted by the KDF chain derived by the DH ratchet in this phase...)

Double Ratchet



Alice

X_i, x_i, Y_j

Root key
(from previous stage)

All messages
are relayed by
the server

Bob

Y_j, y_j, X_i

Root key

Double Ratchet

Alice

Root key
(from previous stage)

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

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

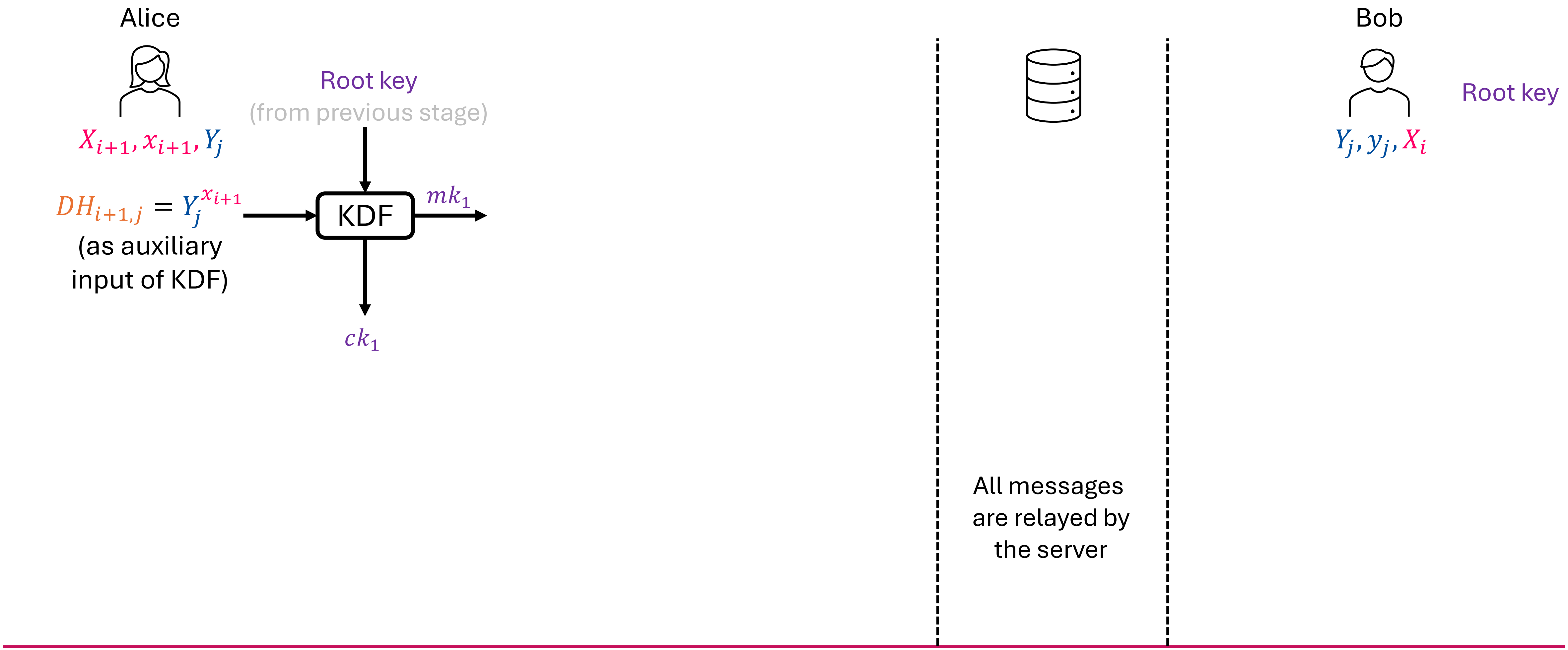
All messages
are relayed by
the server

Bob

Root key

Y_j, y_j, X_i

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)

KDF

mk_1

ck_1

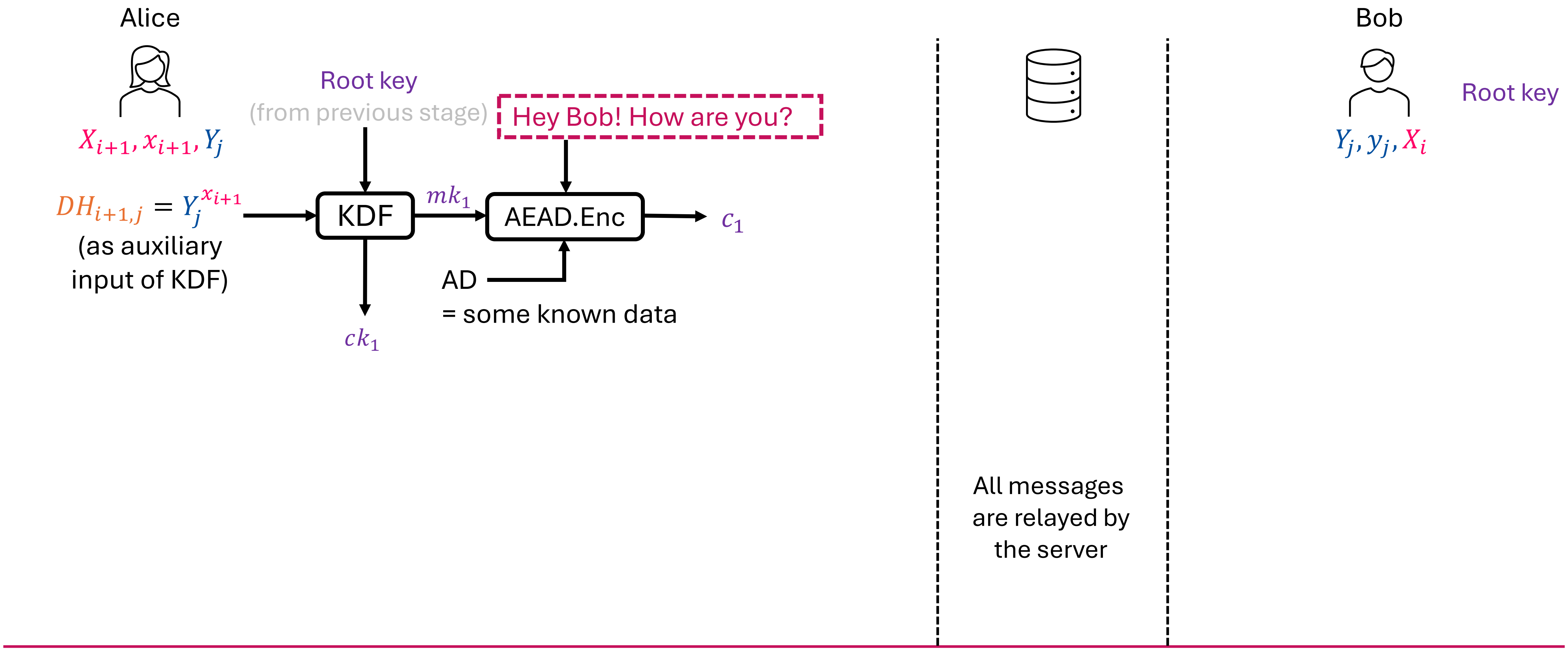
All messages are relayed by the server

Bob

Root key

Y_j, y_j, X_i

Double Ratchet



Alice

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

Root key

(from previous stage)

Hey Bob! How are you?

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

(as auxiliary input of KDF)

KDF

mk_1

AEAD.Enc

c_1

ck_1

AD

= some known data

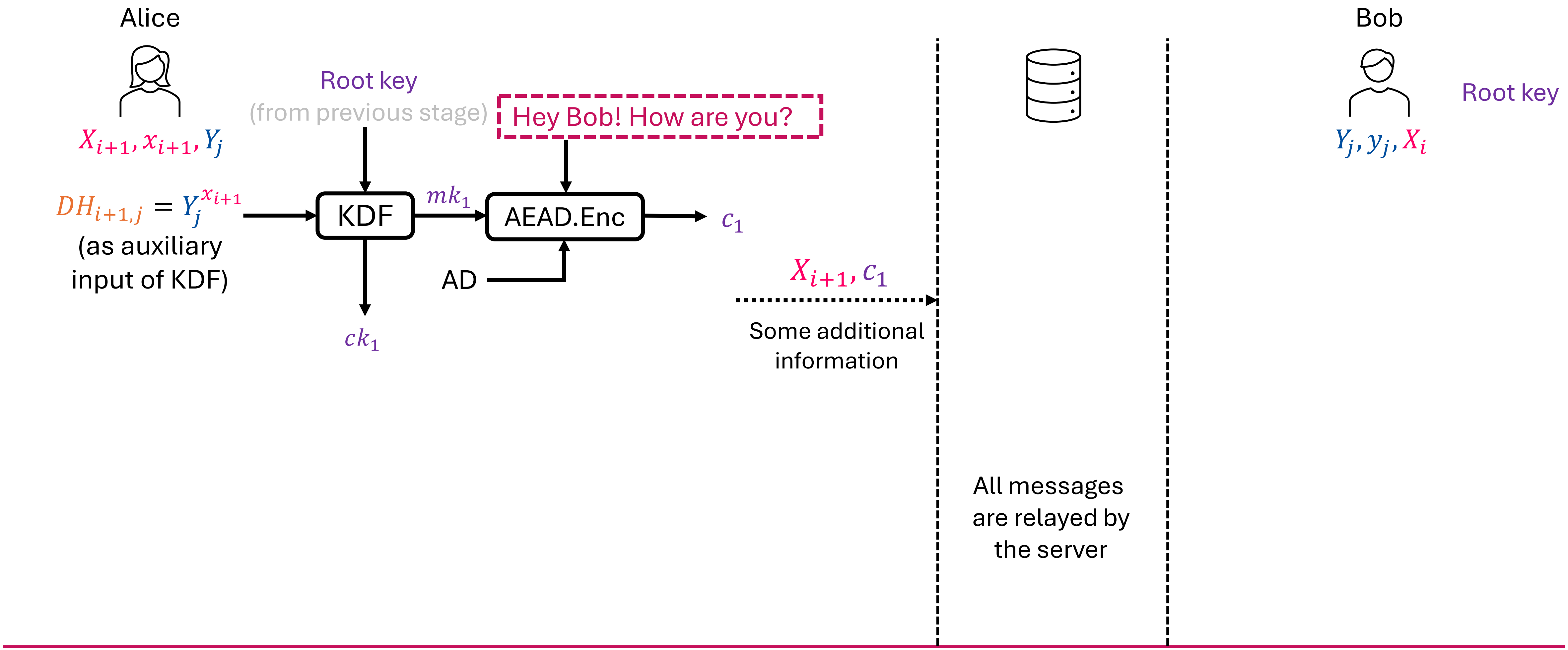
All messages are relayed by the server

Bob

Root key

Y_j, y_j, X_i

Double Ratchet



Alice

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

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

Root key (from previous stage)

Hey Bob! How are you?

KDF

mk_1

AEAD.Enc

AD

c_1

ck_1

X_{i+1}, c_1

Some additional information

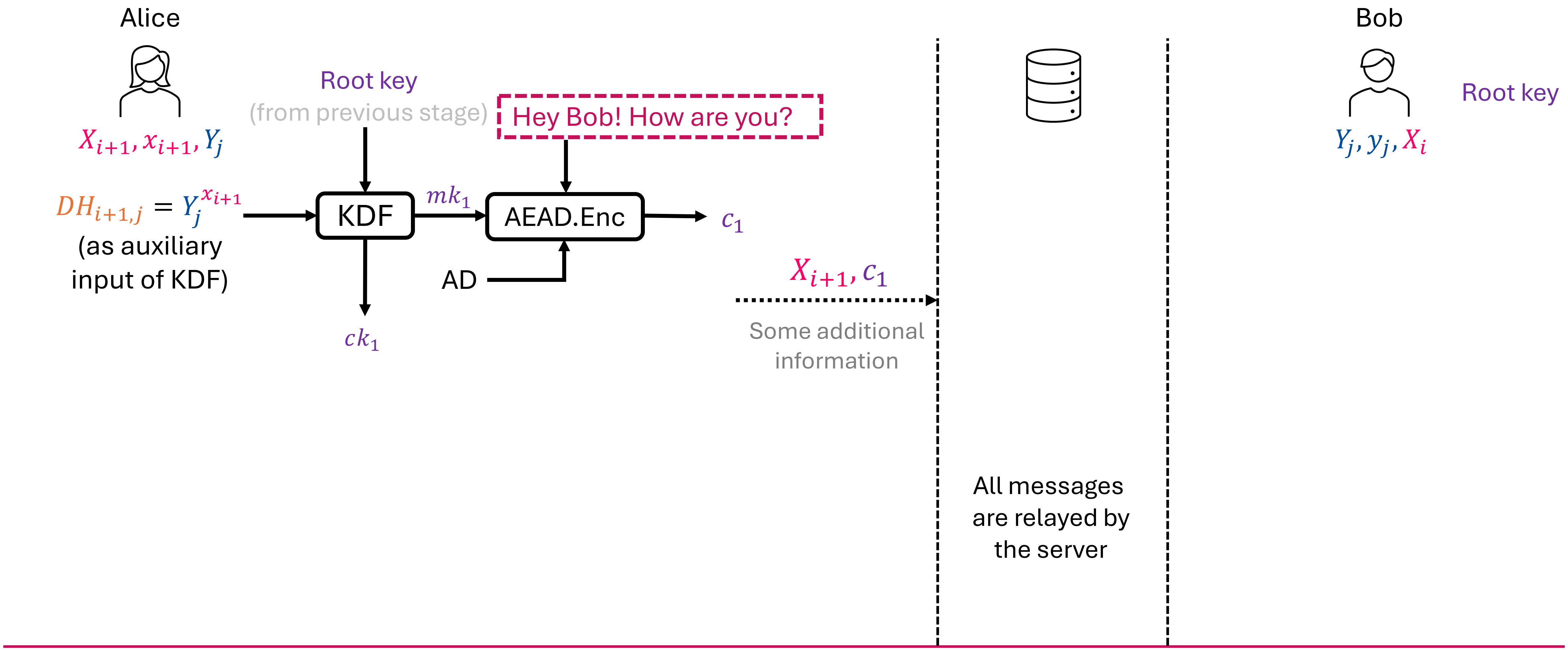
All messages are relayed by the server

Bob

Root key

Y_j, y_j, X_i

Double Ratchet



Alice

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

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

Root key (from previous stage)

Hey Bob! How are you?

KDF

mk_1

AEAD.Enc

AD

ck_1

c_1

X_{i+1}, c_1

Some additional information

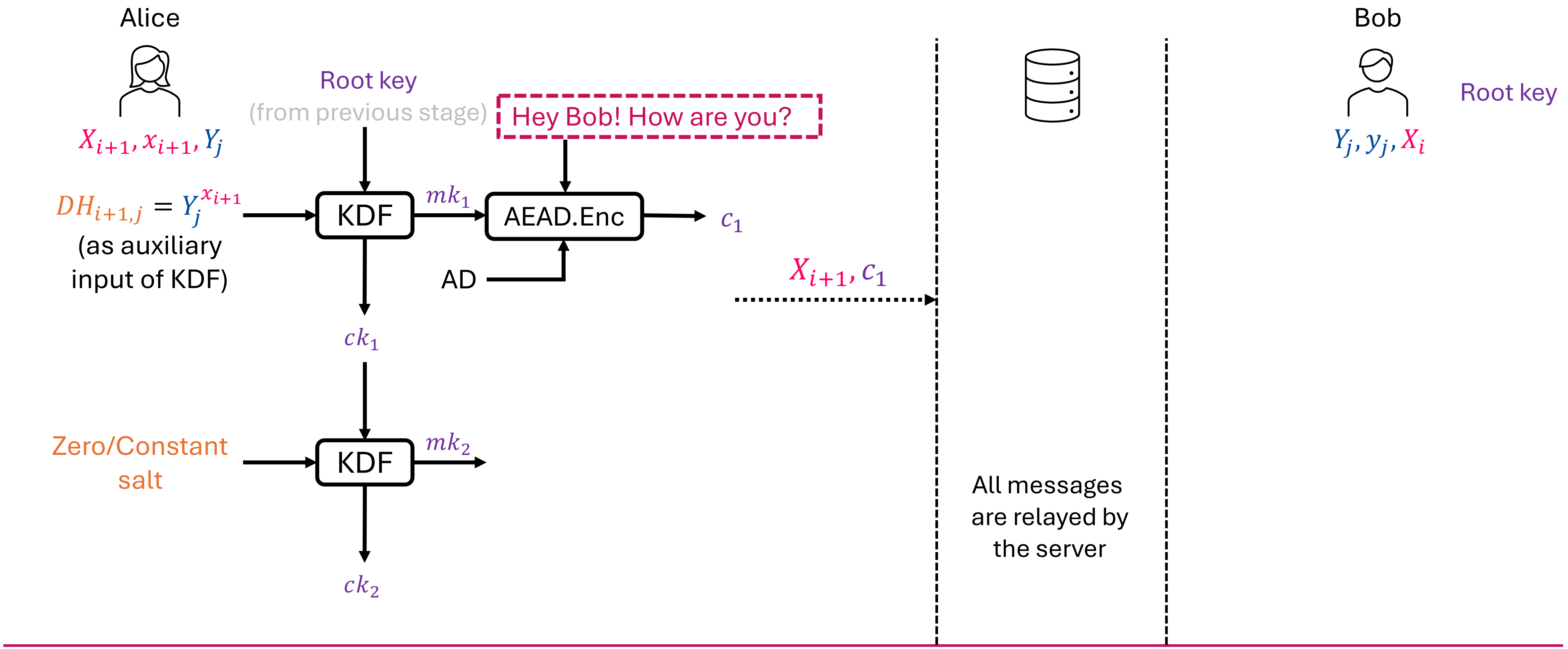
All messages are relayed by the server

Bob

Root key

Y_j, y_j, X_i

Double Ratchet



Alice

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

Root key

(from previous stage)

Hey Bob! How are you?

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

(as auxiliary input of KDF)

KDF

mk_1

AEAD.Enc

c_1

AD

ck_1

Zero/Constant salt

KDF

mk_2

ck_2

X_{i+1}, c_1

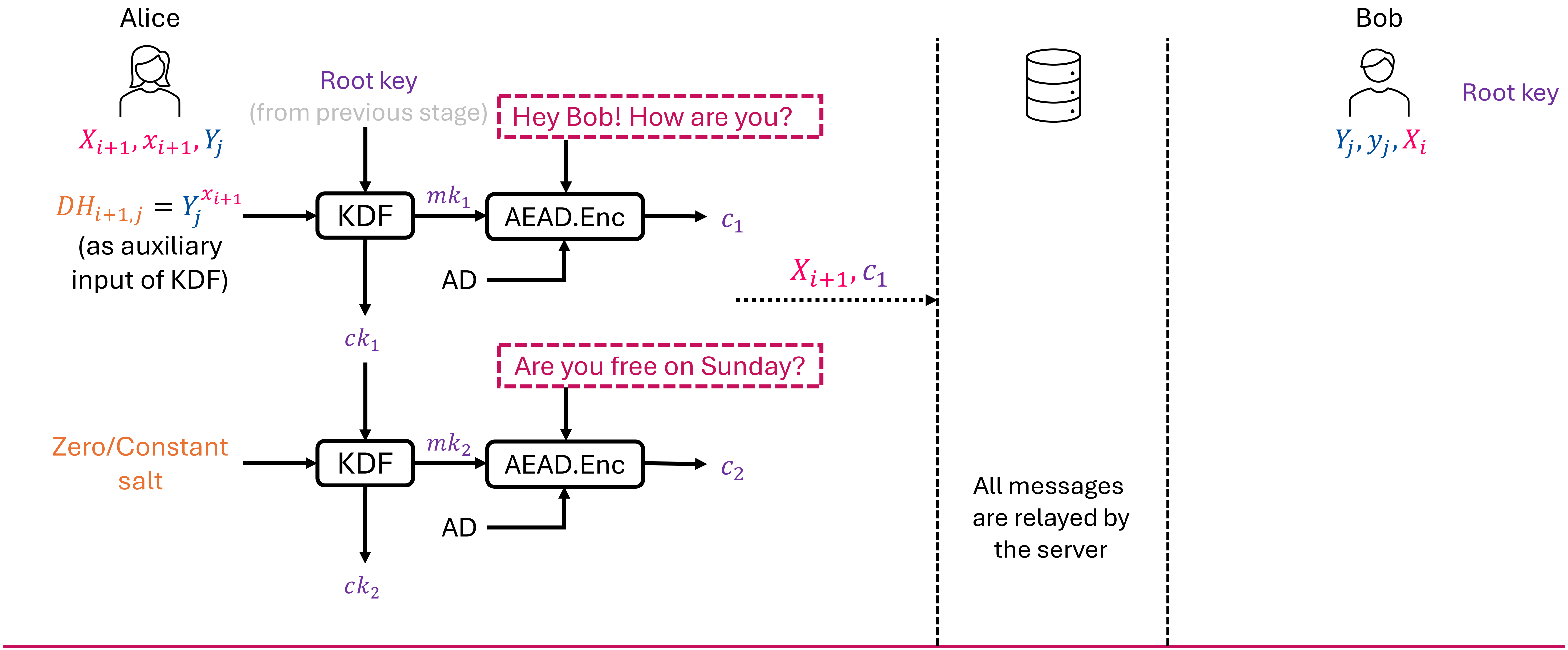
All messages are relayed by the server

Bob

Y_j, y_j, X_i

Root key

Double Ratchet



Alice

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

Root key
(from previous stage)

Hey Bob! How are you?

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

KDF

mk_1

AEAD.Enc

c_1

AD

ck_1

X_{i+1}, c_1

Are you free on Sunday?

Zero/Constant salt

KDF

mk_2

AEAD.Enc

c_2

AD

ck_2

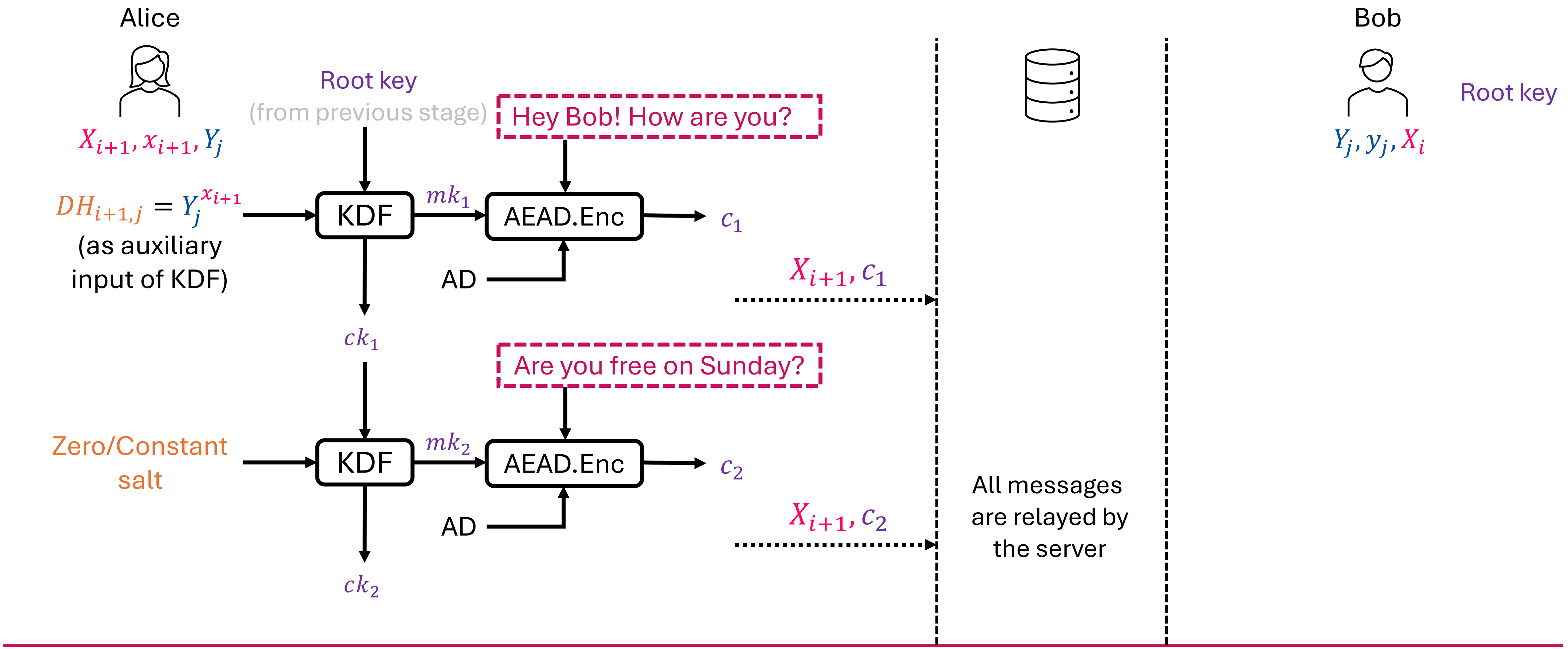
All messages are relayed by the server

Bob

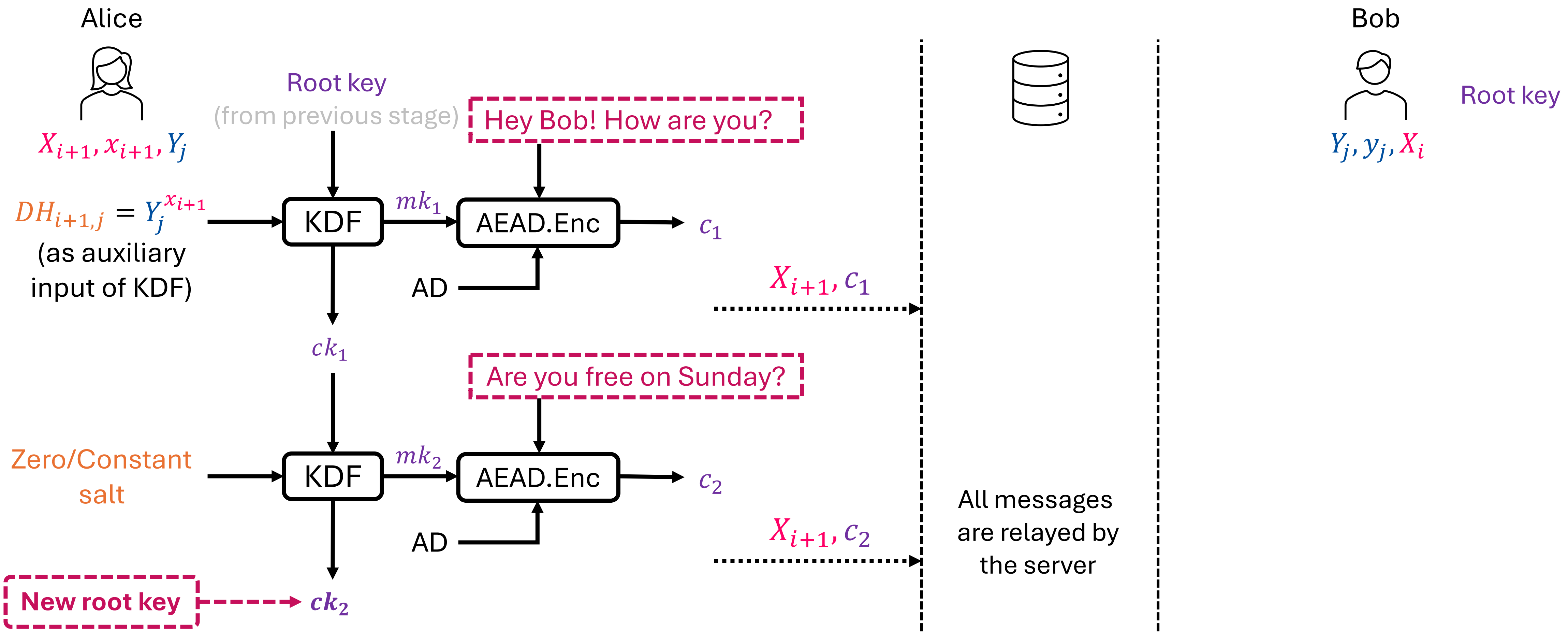
Root key

Y_j, y_j, X_i

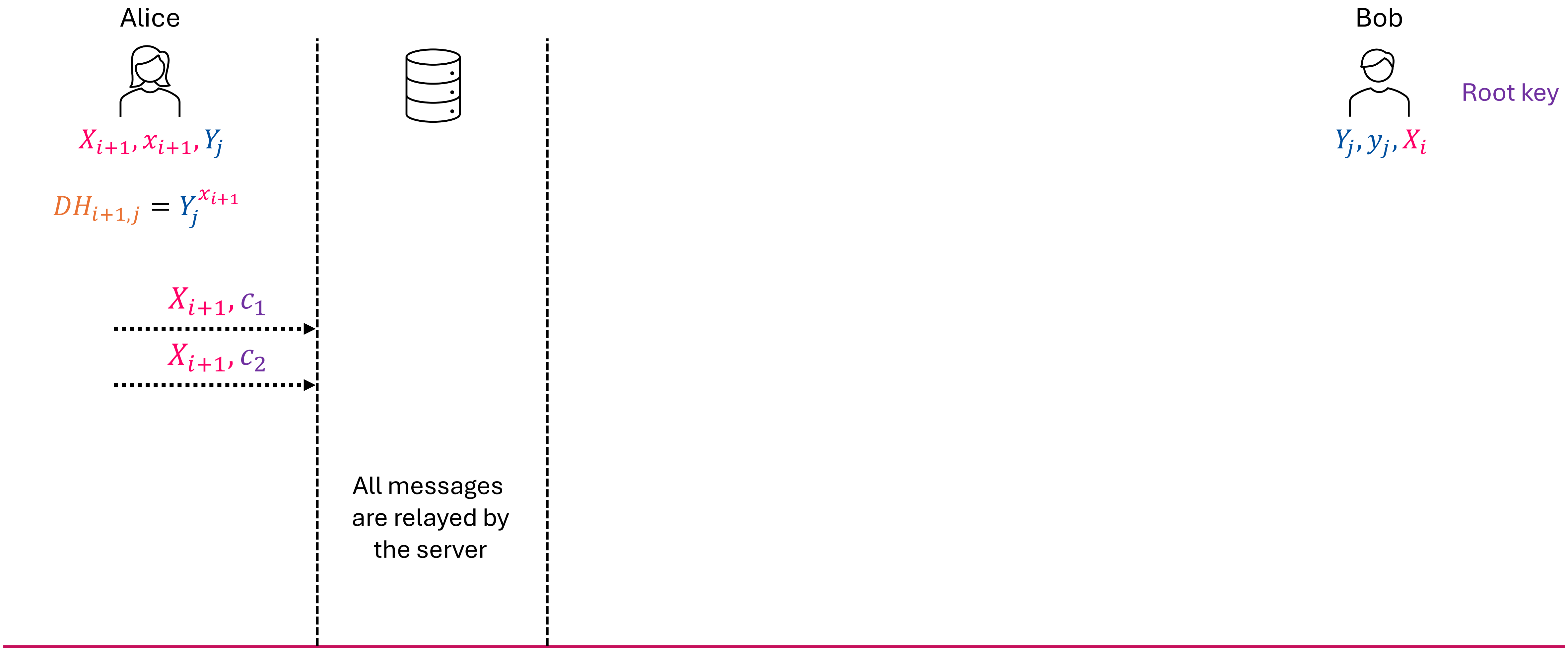
Double Ratchet



Double Ratchet



Double Ratchet



Alice

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

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

X_{i+1}, c_1

X_{i+1}, c_2

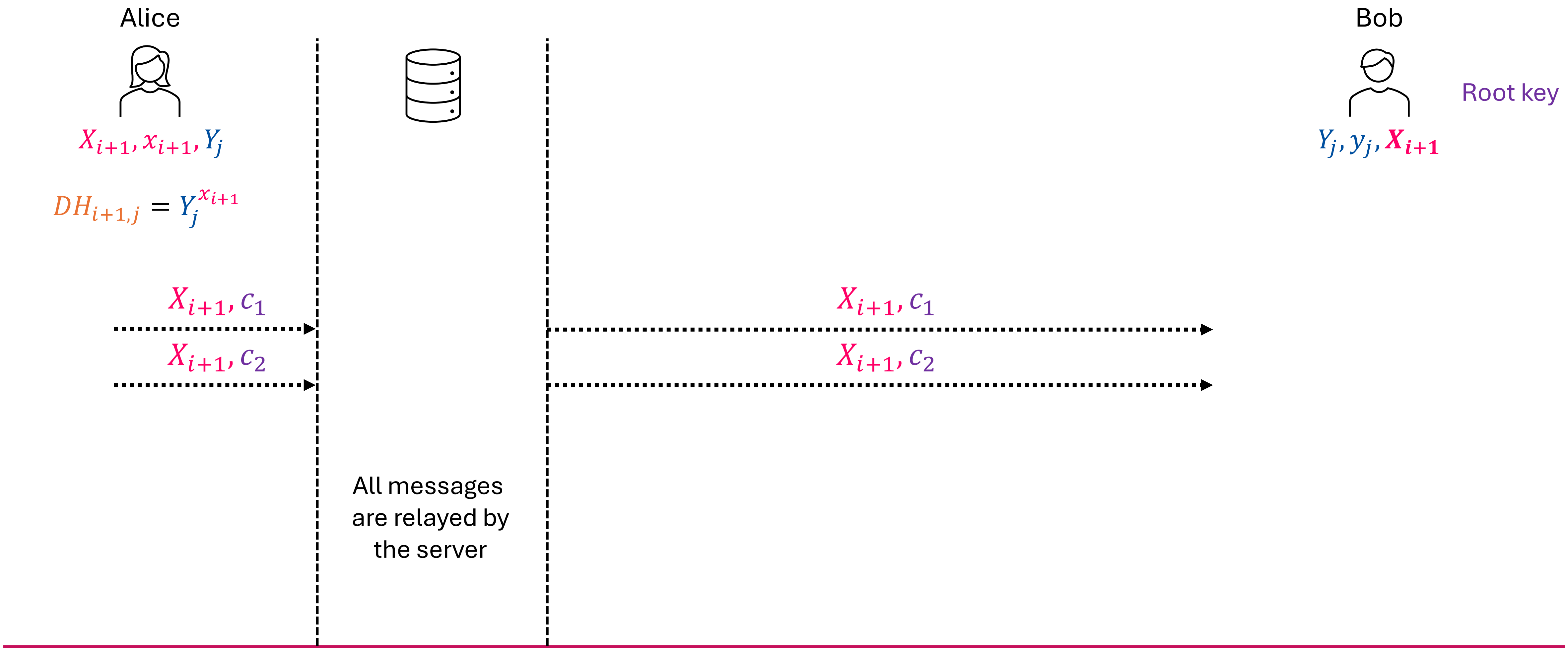
All messages are relayed by the server

Bob

Root key

Y_j, y_j, X_i

Double Ratchet



Alice

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

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

X_{i+1}, c_1

X_{i+1}, c_2

All messages are relayed by the server

X_{i+1}, c_1

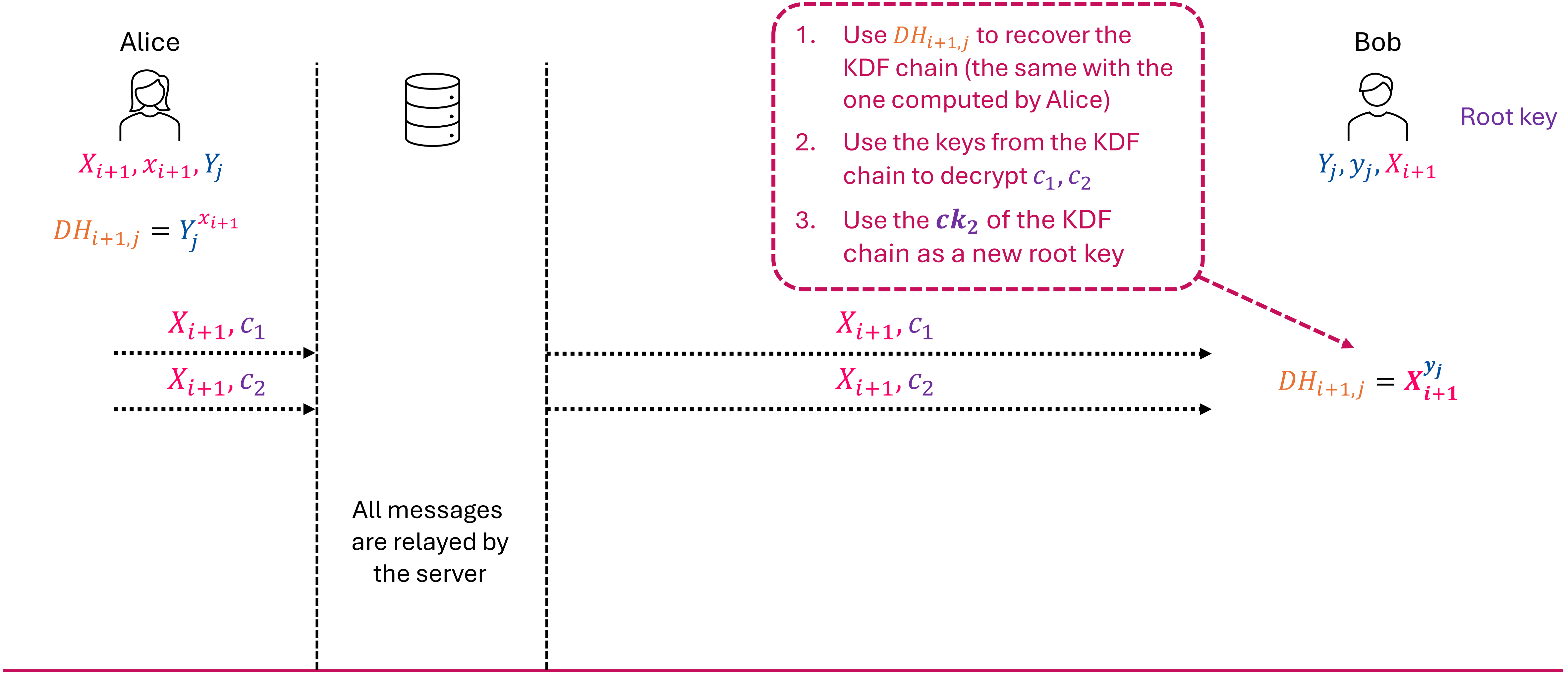
X_{i+1}, c_2

Bob

Root key

$Y_j, y_j, \boldsymbol{X_{i+1}}$

Double Ratchet



Alice

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

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

X_{i+1}, c_1

X_{i+1}, c_2

All messages are relayed by the server

X_{i+1}, c_1

X_{i+1}, c_2

1. Use $DH_{i+1,j}$ to recover the KDF chain (the same with the one computed by Alice)
2. Use the keys from the KDF chain to decrypt c_1, c_2
3. Use the $\boldsymbol{ck_2}$ of the KDF chain as a new root key

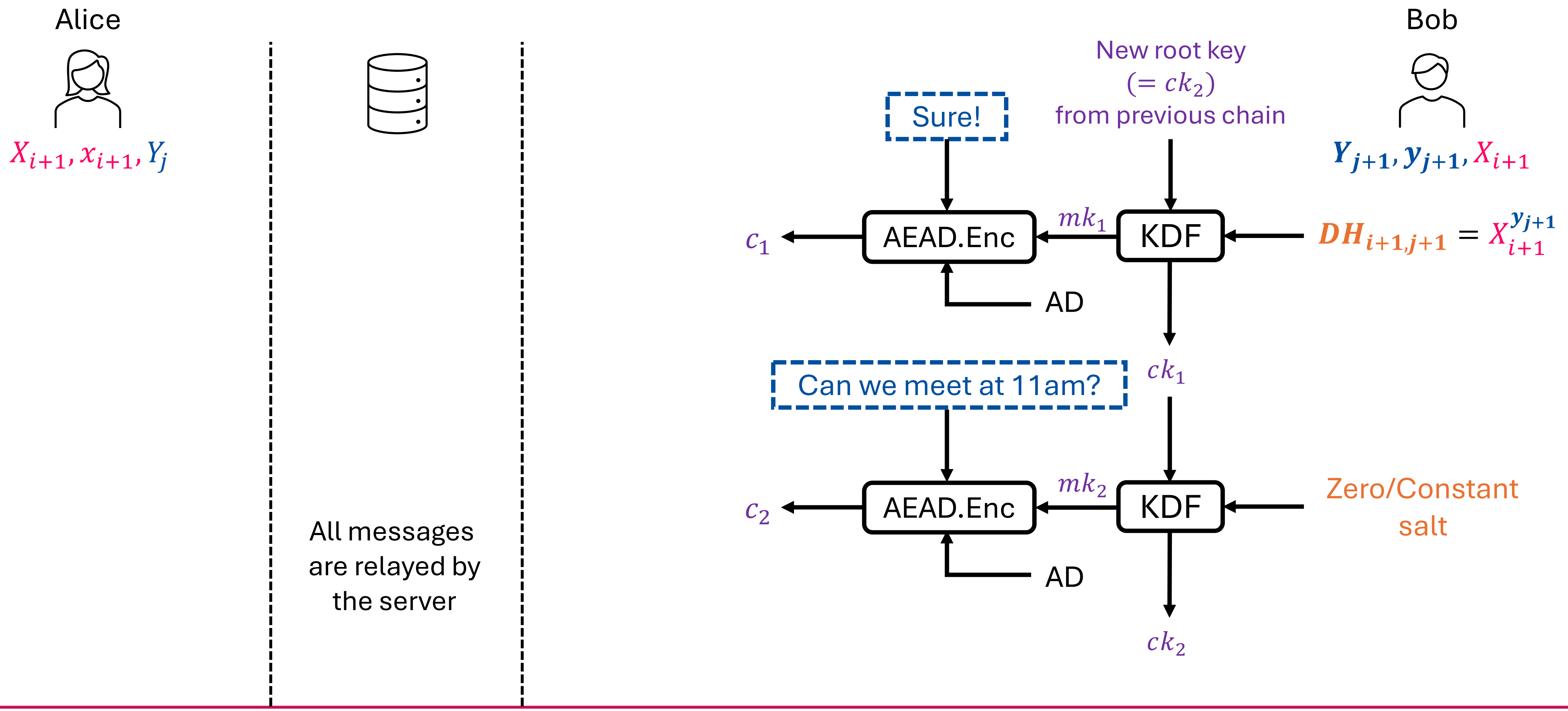
Bob

Root key

Y_j, y_j, X_{i+1}

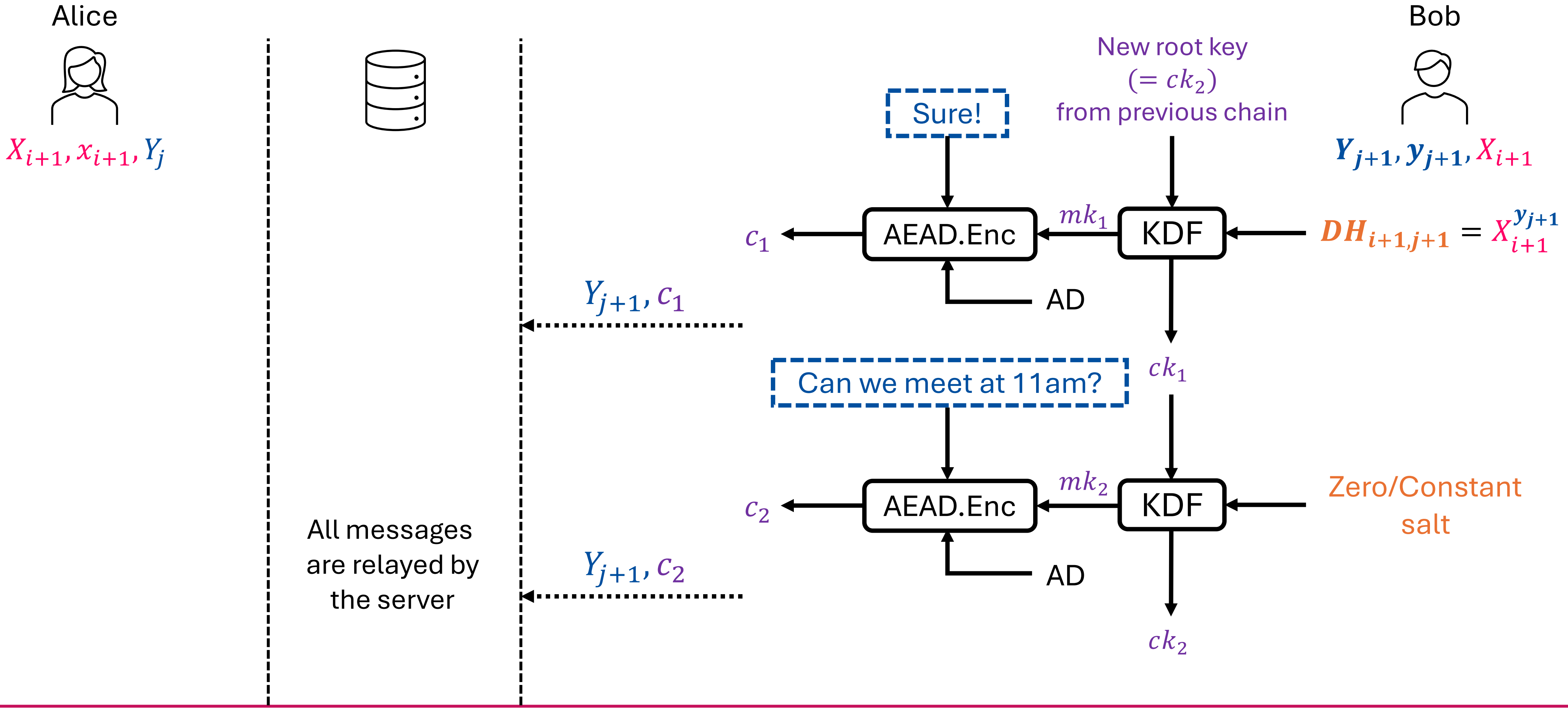
$DH_{i+1,j} = \boldsymbol{X_{i+1}^{y_j}}$

Double Ratchet

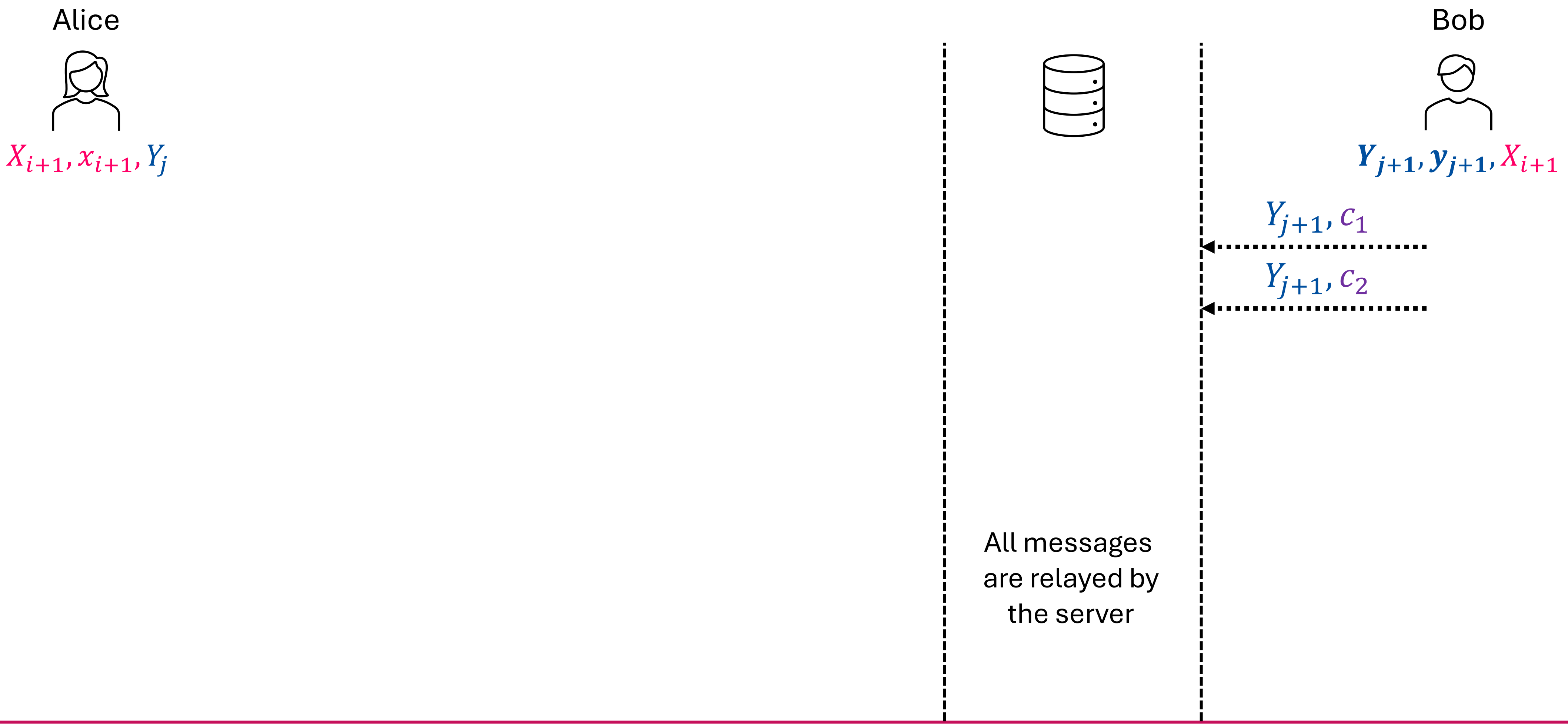


All messages are relayed by the server

Double Ratchet



Double Ratchet



Alice

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

Bob

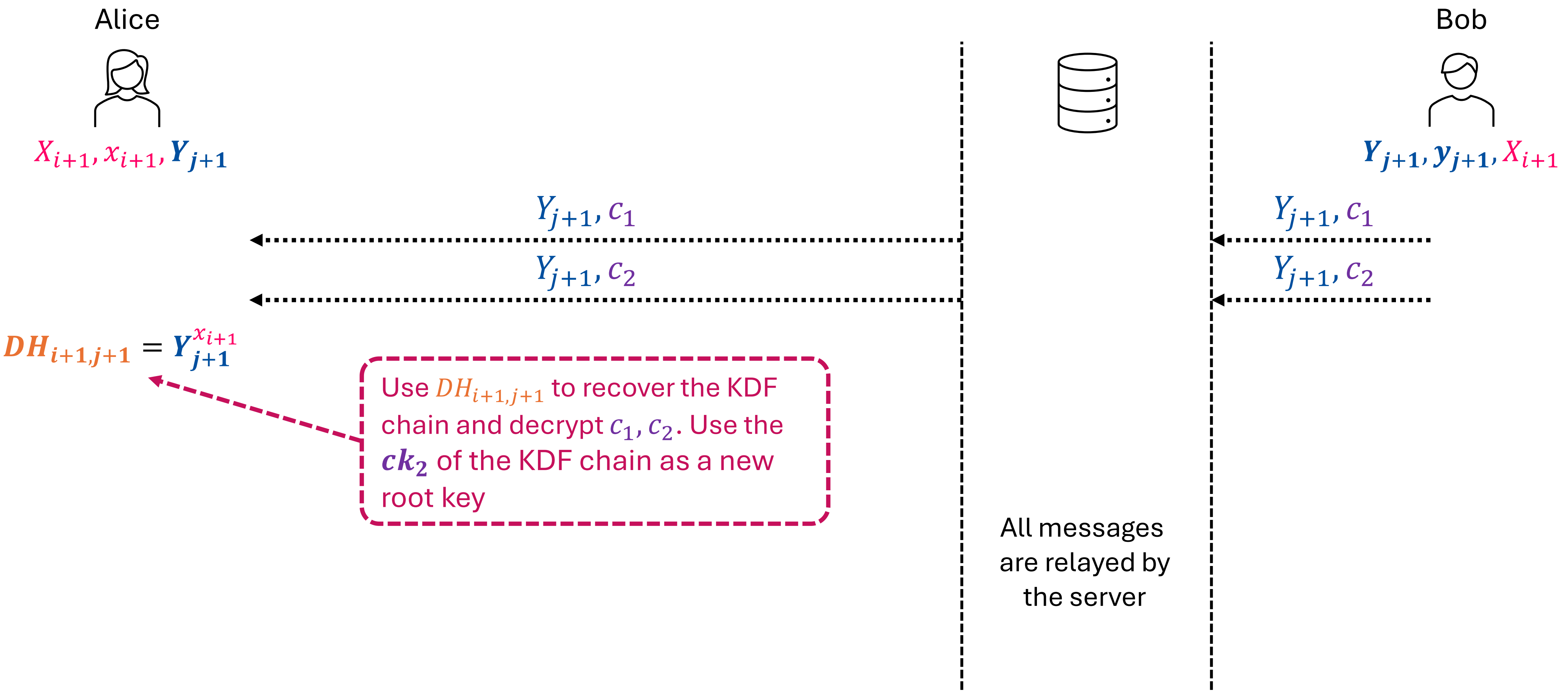
$\boldsymbol{Y_{j+1}}, \boldsymbol{y_{j+1}}, X_{i+1}$

Y_{j+1}, c_1

Y_{j+1}, c_2

All messages are relayed by the server

Double Ratchet



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

Use $DH_{i+1,j+1}$ to recover the KDF chain and decrypt c_1, c_2. Use the $\boldsymbol{ck_2}$ of the KDF chain as a new root key

All messages are relayed by the server

Double Ratchet

Alice

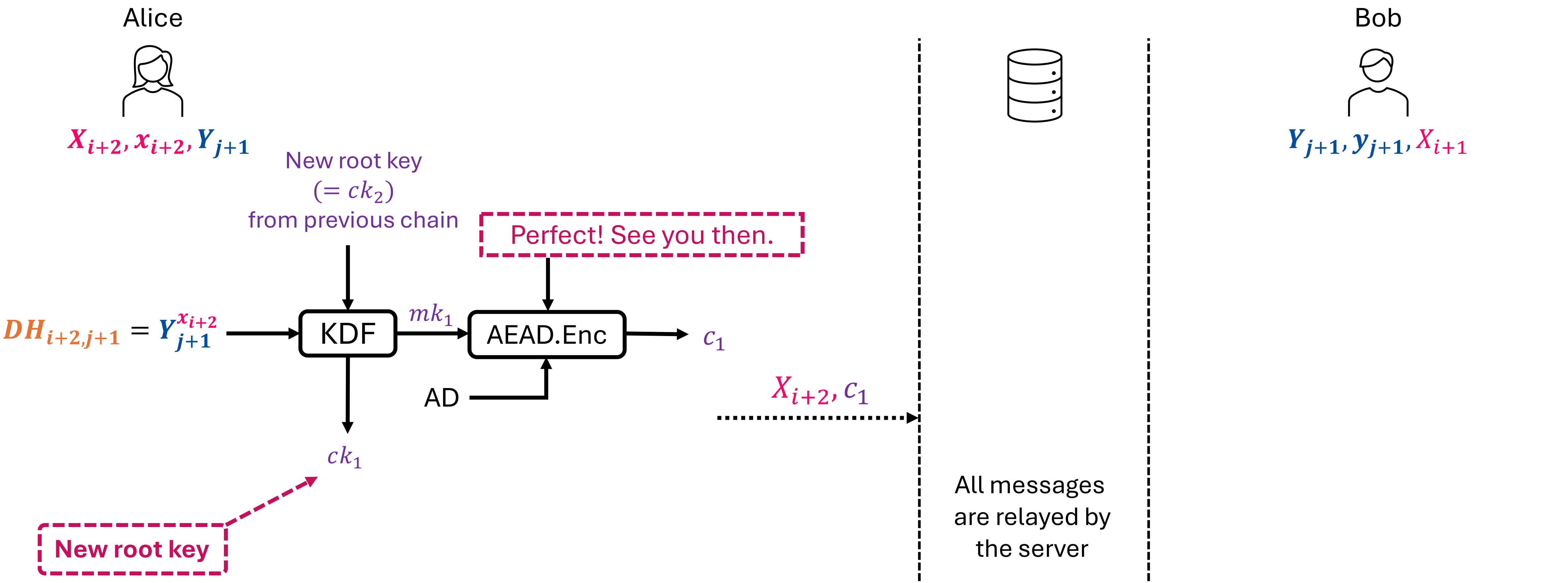
$X_{i+2}, x_{i+2}, Y_{j+1}$

Bob

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

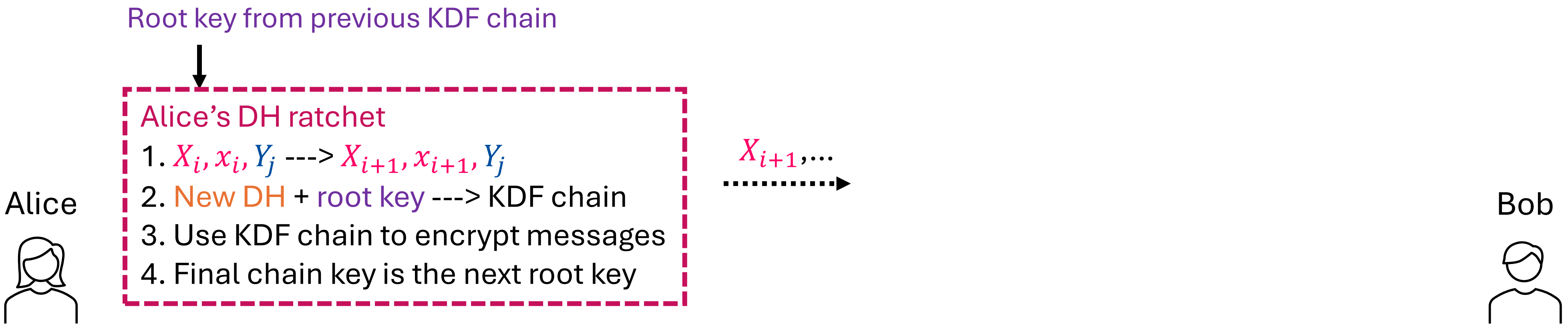
All messages are relayed by the server

Double Ratchet



Double Ratchet

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



Root key from previous KDF chain

Alice

Alice's DH ratchet

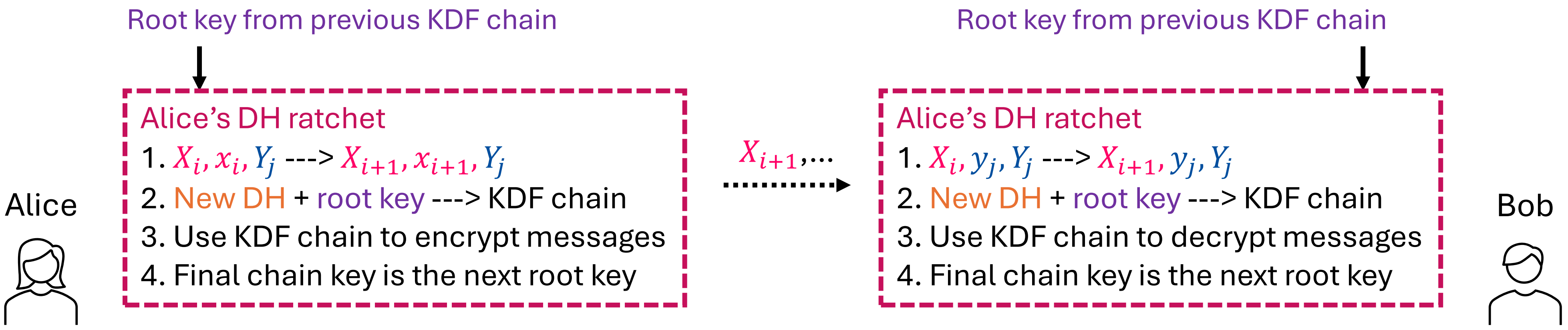
1. X_i, x_i, Y_j ---> X_{i+1}, x_{i+1}, Y_j
2. New DH + root key ---> KDF chain
3. Use KDF chain to encrypt messages
4. Final chain key is the next root key

$X_{i+1}, \dots$

Bob

Double Ratchet

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



Root key from previous KDF chain

Alice's DH ratchet

1. X_i, x_i, Y_j ---> X_{i+1}, x_{i+1}, Y_j
2. New DH + root key ---> KDF chain
3. Use KDF chain to encrypt messages
4. Final chain key is the next root key

$X_{i+1}, \ldots$

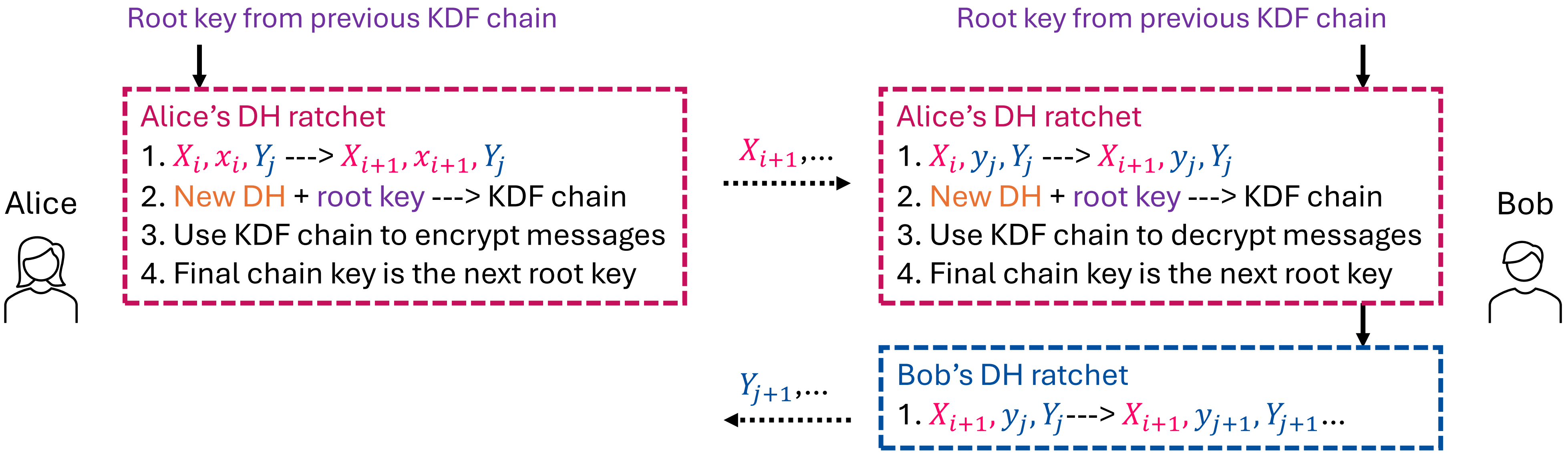
Root key from previous KDF chain

Alice's DH ratchet

1. X_i, y_j, Y_j ---> X_{i+1}, y_j, Y_j
2. New DH + root key ---> KDF chain
3. Use KDF chain to decrypt messages
4. Final chain key is the next root key

Double Ratchet

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



Root key from previous KDF chain

Alice

Alice's DH ratchet

1. X_i, x_i, Y_j ---> X_{i+1}, x_{i+1}, Y_j
2. New DH + root key ---> KDF chain
3. Use KDF chain to encrypt messages
4. Final chain key is the next root key

$X_{i+1}, \dots$

Root key from previous KDF chain

Alice's DH ratchet

1. X_i, y_j, Y_j ---> X_{i+1}, y_j, Y_j
2. New DH + root key ---> KDF chain
3. Use KDF chain to decrypt messages
4. Final chain key is the next root key

Bob

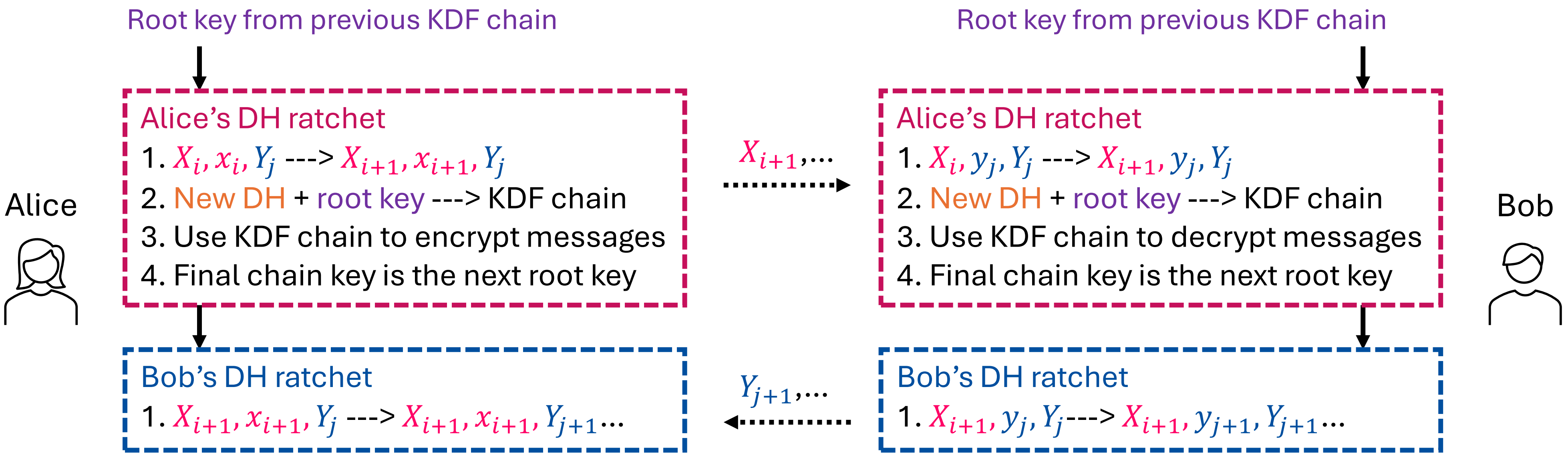
Bob's DH ratchet

1. X_{i+1}, y_j, Y_j---> $X_{i+1}, y_{j+1}, Y_{j+1} \dots$

$Y_{j+1}, \dots$

Double Ratchet

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



Root key from previous KDF chain

Alice

Alice’s DH ratchet

1. X_i, x_i, Y_j ---> X_{i+1}, x_{i+1}, Y_j
2. New DH + root key ---> KDF chain
3. Use KDF chain to encrypt messages
4. Final chain key is the next root key

Bob’s DH ratchet

1. X_{i+1}, x_{i+1}, Y_j ---> $X_{i+1}, x_{i+1}, Y_{j+1}\ldots$

$X_{i+1},\ldots$

$Y_{j+1},\ldots$

Root key from previous KDF chain

Bob

Alice’s DH ratchet

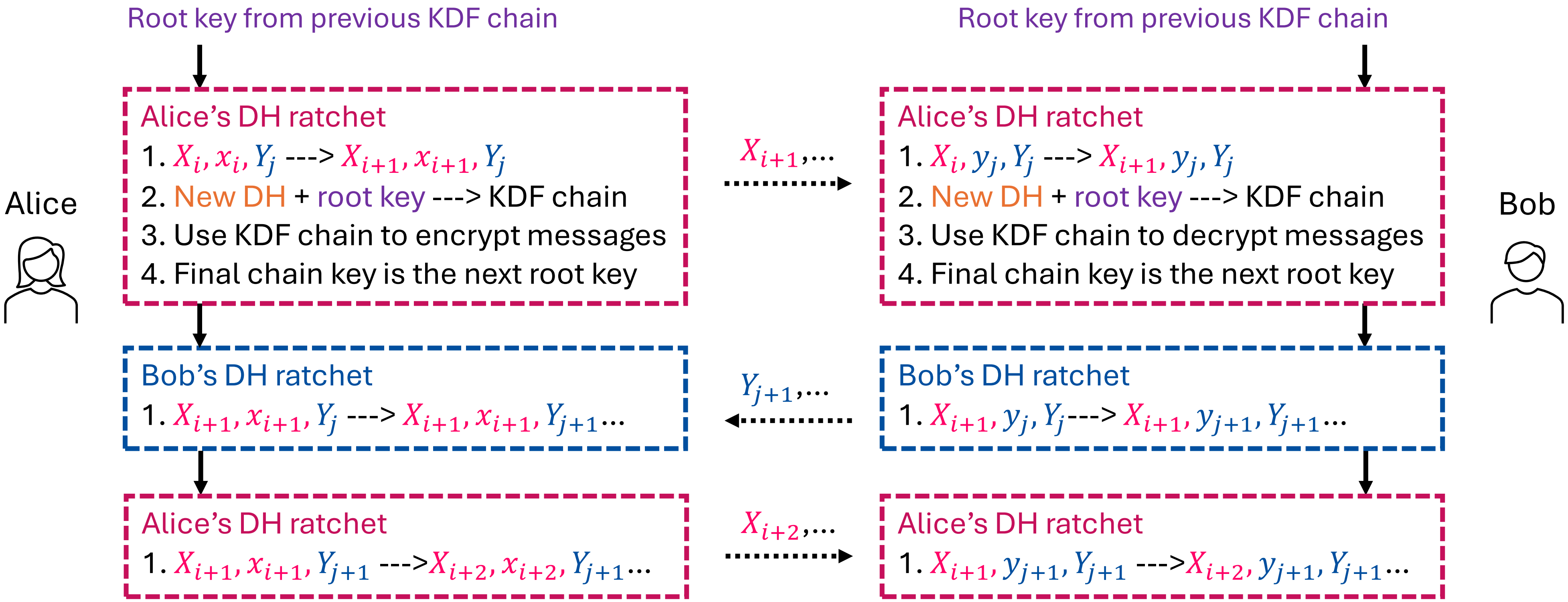
1. X_i, y_j, Y_j ---> X_{i+1}, y_j, Y_j
2. New DH + root key ---> KDF chain
3. Use KDF chain to decrypt messages
4. Final chain key is the next root key

Bob’s DH ratchet

1. X_{i+1}, y_j, Y_j---> $X_{i+1}, y_{j+1}, Y_{j+1}\ldots$

Double Ratchet

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



Root key from previous KDF chain

Alice

Alice's DH ratchet

1. X_i, x_i, Y_j ---> X_{i+1}, x_{i+1}, Y_j
2. New DH + root key ---> KDF chain
3. Use KDF chain to encrypt messages
4. Final chain key is the next root key

Bob's DH ratchet

1. X_{i+1}, x_{i+1}, Y_j ---> $X_{i+1}, x_{i+1}, Y_{j+1}\ldots$

Alice's DH ratchet

1. $X_{i+1}, x_{i+1}, Y_{j+1}$ --->$X_{i+2}, x_{i+2}, Y_{j+1}\ldots$

$X_{i+1},\ldots$

$Y_{j+1},\ldots$

$X_{i+2},\ldots$

Root key from previous KDF chain

Bob

Alice's DH ratchet

1. X_i, y_j, Y_j ---> X_{i+1}, y_j, Y_j
2. New DH + root key ---> KDF chain
3. Use KDF chain to decrypt messages
4. Final chain key is the next root key

Bob's DH ratchet

1. X_{i+1}, y_j, Y_j---> $X_{i+1}, y_{j+1}, Y_{j+1}\ldots$

Alice's DH ratchet

1. $X_{i+1}, y_{j+1}, Y_{j+1}$ --->$X_{i+2}, y_{j+1}, Y_{j+1}\ldots$

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:

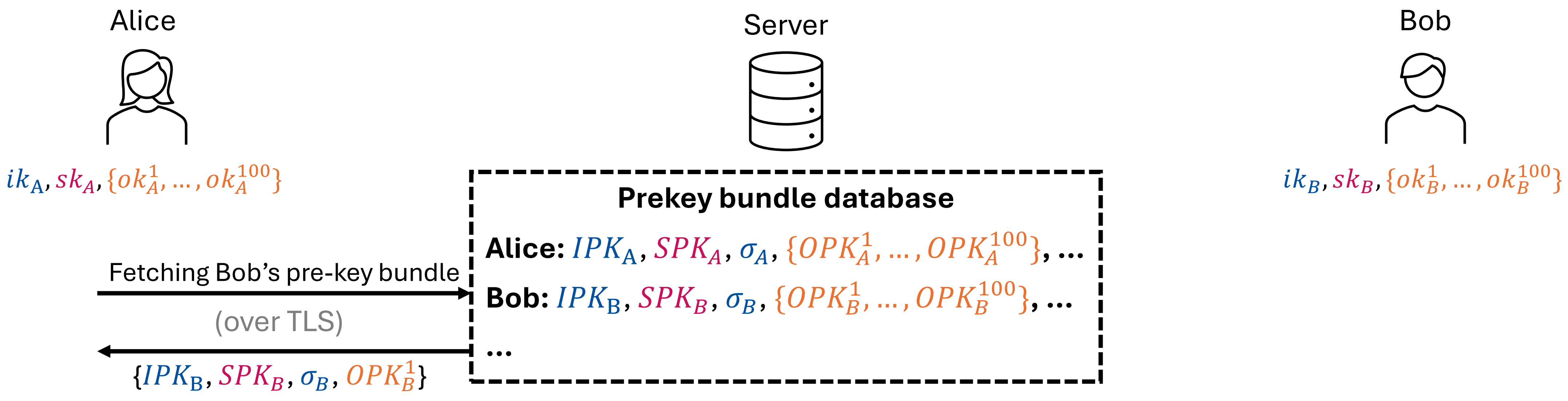
Public parameters: $(\mathbb{G}, g, q)$:

A q-order EC group $\mathbb{G}$ with a generator g

		Alice	Bob
Long-term secret (static)	Identity secret key (IK)	$ik_{\mathrm{A}} \in_{\$} \mathbb{Z}_q$	$ik_{\mathrm{B}} \in_{\$} \mathbb{Z}_q$
	Identity public key (IPK)	$IPK_{\mathrm{A}}(= g^{ik_A})$	IPK_{B}
Mid-term secret (updated periodically)	**Signing secret pre-key (SK)**	$\boldsymbol{sk_{\mathrm{A}}} \in_{\$} \boldsymbol{\mathbb{Z}_q}$	$\boldsymbol{sk_{\mathrm{B}}} \in_{\$} \boldsymbol{\mathbb{Z}_q}$
	Signing public pre-key (SPK)	$\boldsymbol{SPK_{\mathrm{A}}}$	$\boldsymbol{SPK_{\mathrm{B}}}$
Short-term secret (used once)	One-time secret pre-keys (OK)	$\{ok_{\mathrm{A}}^1, ok_A^2, \ldots\} \subseteq_{\$} \mathbb{Z}_q$	$\{ok_{\mathrm{B}}^1, ok_B^2, \ldots\} \subseteq_{\$} \mathbb{Z}_q$
	One-time public pre-keys (OPK)	$(OPK_{\mathrm{A}}^1, OPK_{\mathrm{A}}^2, \ldots)$	$(OPK_{\mathrm{B}}^1, OPK_{\mathrm{B}}^2, \ldots)$

X3DH + Double Ratchet

- Recall of X3DH:



Alice

Server

Bob

$ik_\mathrm{A}, sk_A, \{ok_A^1, \dots, ok_A^{100}\}$

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

Prekey bundle database

Alice: $IPK_\mathrm{A}, SPK_A, \sigma_A, \{OPK_A^1, \dots, OPK_A^{100}\}$**, …**

Bob: $IPK_\mathrm{B}, SPK_B, \sigma_B, \{OPK_B^1, \dots, OPK_B^{100}\}$**, …**

…

Fetching Bob’s pre-key bundle

(over TLS)

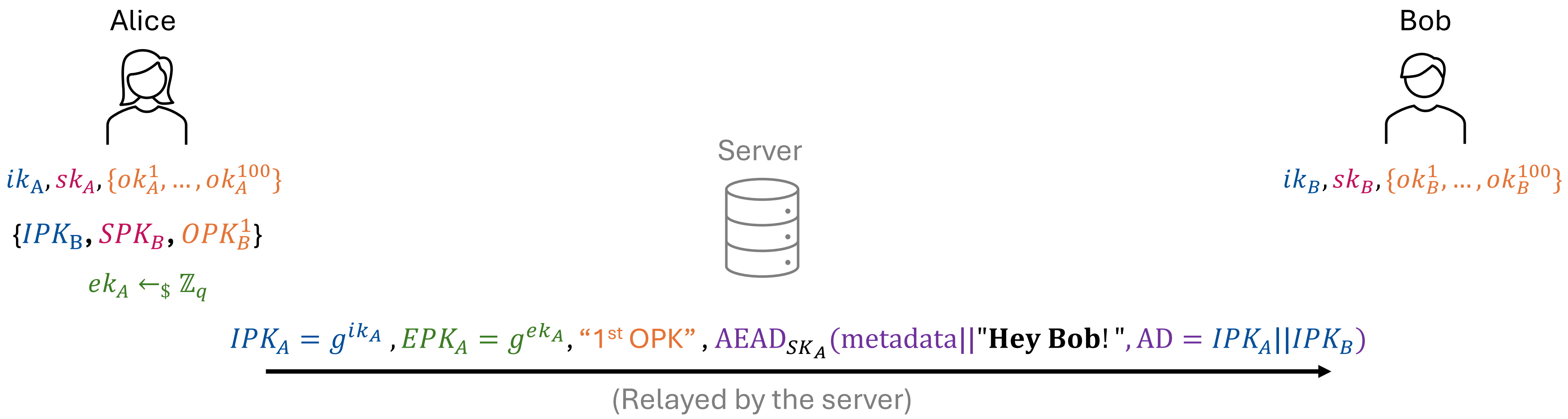
$\{IPK_\mathrm{B}, SPK_B, \sigma_B, OPK_B^1\}$

$\mathrm{Verify}(IPK_B, (SPK_B, \sigma_B))$

if valid, accept $\{IPK_\mathrm{B}, SPK_B, \sigma_B, OPK_B^1\}$

X3DH + Double Ratchet

- Recall of X3DH:



Alice

Bob

Server

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

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

$ek_A \leftarrow_\$ \mathbb{Z}_q$

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

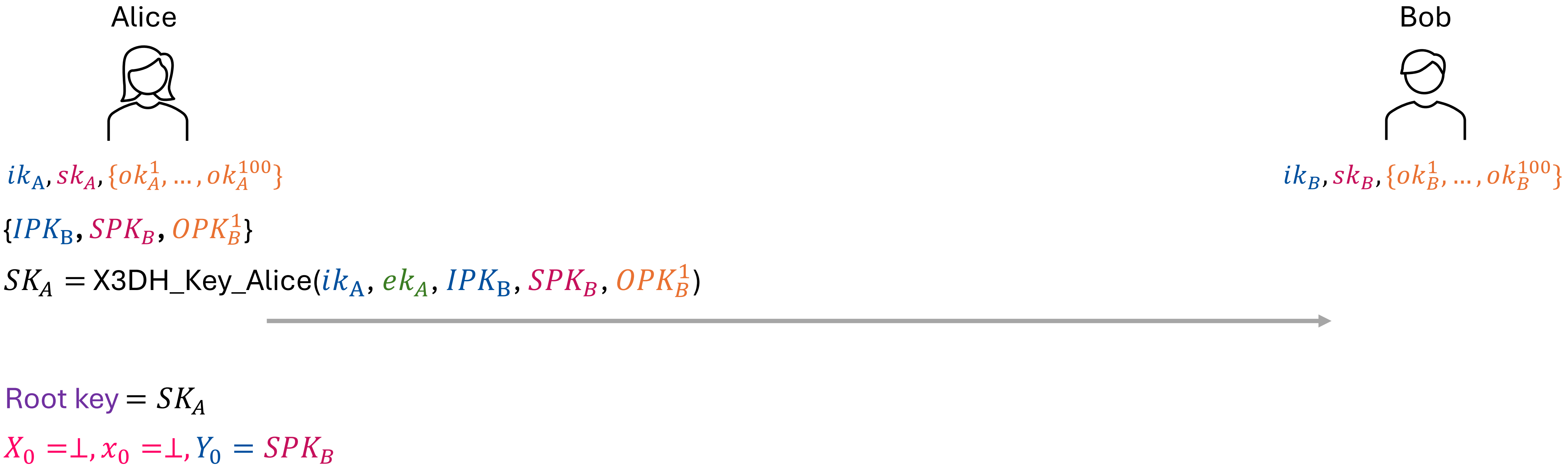
$IPK_A = g^{ik_A}, EPK_A = g^{ek_A}$, "1st OPK", $\mathrm{AEAD}_{SK_A}(\mathrm{metadata}||\text{"}\textbf{Hey Bob}\text{! "}, \mathrm{AD} = IPK_A||IPK_B)$

(Relayed by the server)

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

X3DH + Double Ratchet

- Initialize Double Ratchet using the SK from X3DH



Alice

Bob

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

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

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

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

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 = X3DH_Key_Alice(...)

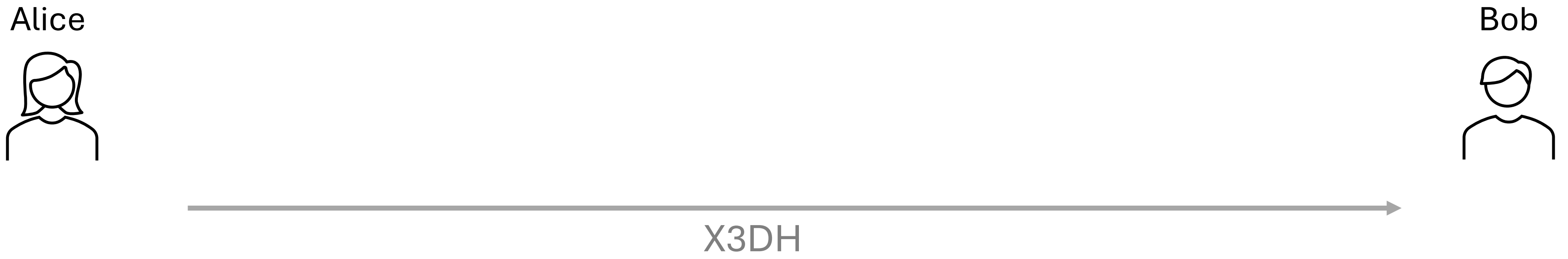
Alice's DH ratchet

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

$SK_A =$ X3DH_Key_Alice(…)

Alice’s DH ratchet

Root key $= SK_A$

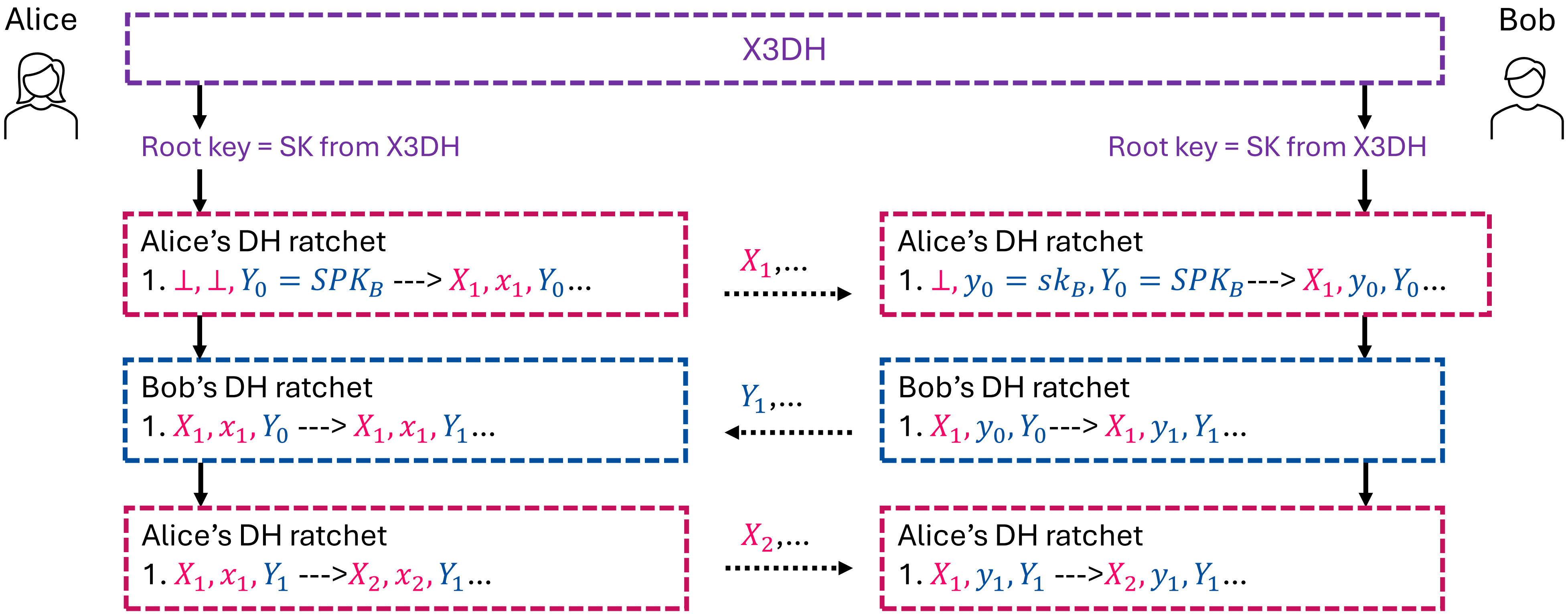
$X_0 = \perp, x_0 = \perp, Y_0 = SPK_B$ (Signing public pre-key of Bob)

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

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

Double Ratcheting

- Initialize Double Ratchet using the SK from X3DH



Alice

Bob

X3DH

Root key = SK from X3DH

Root key = SK from X3DH

Alice's DH ratchet
1. $\perp, \perp, Y_0 = SPK_B$ ---> X_1, x_1, Y_0...

X_1,...

Alice's DH ratchet
1. $\perp, y_0 = sk_B, Y_0 = SPK_B$---> X_1, y_0, Y_0...

Bob's DH ratchet
1. X_1, x_1, Y_0 ---> X_1, x_1, Y_1...

Y_1,...

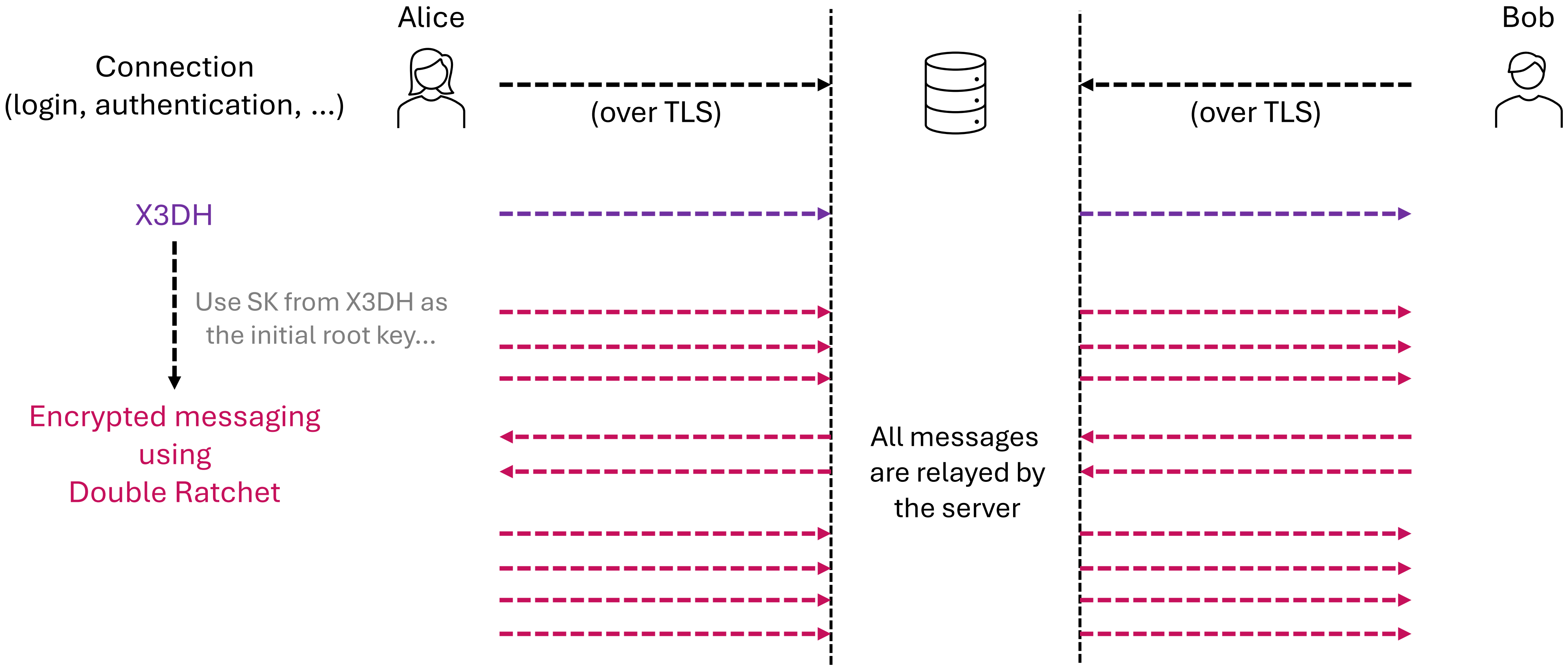
Bob's DH ratchet
1. X_1, y_0, Y_0---> X_1, y_1, Y_1...

Alice's DH ratchet
1. X_1, x_1, Y_1 --->X_2, x_2, Y_1...

X_2,...

Alice's DH ratchet
1. X_1, y_1, Y_1 --->X_2, y_1, Y_1...

Signal Secure Messaging Protocol

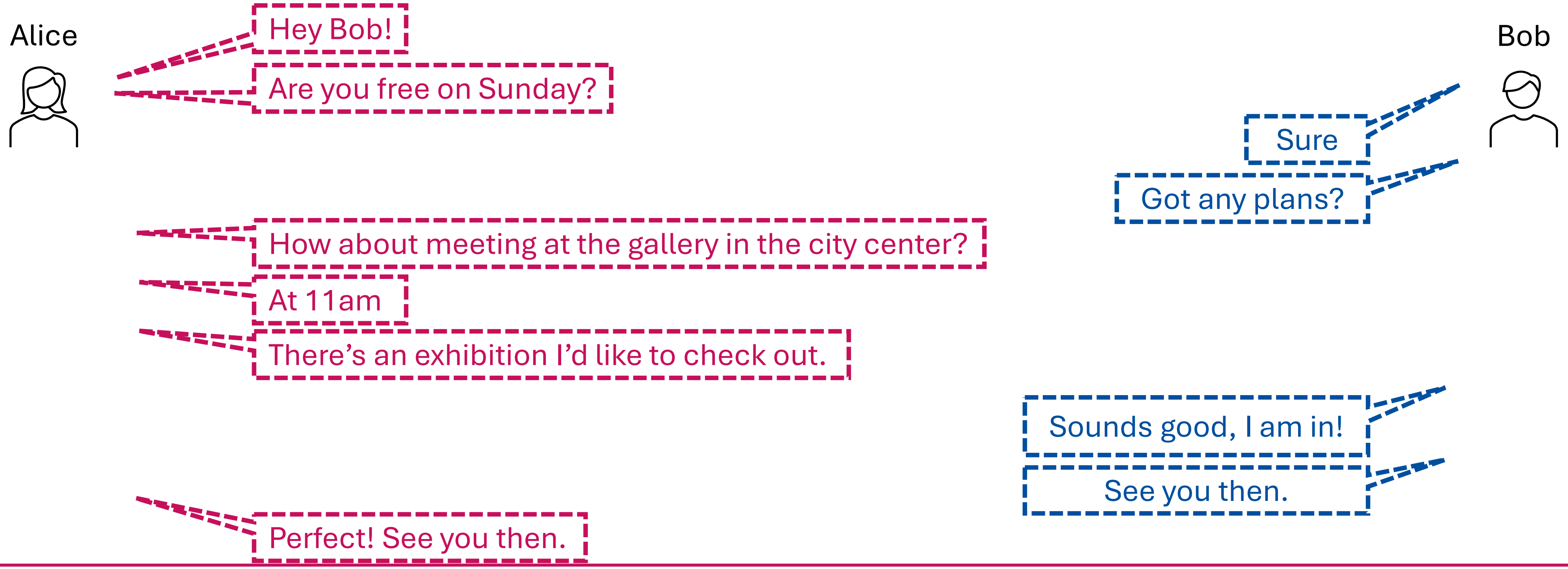


Signal Secure Messaging Protocol

- Some technical details we do not cover:
  - XEdDSA and VXEdDSA:
    - ➢ 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):



Alice

Bob

Alice: Hey Bob!

Alice: Are you free on Sunday?

Bob: Sure

Bob: Got any plans?

Alice: How about meeting at the gallery in the city center?

Alice: At 11am

Alice: There’s an exhibition I’d like to check out.

Bob: Sounds good, I am in!

Bob: See you then.

Alice: Perfect! See you then.

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
  - ➢ …