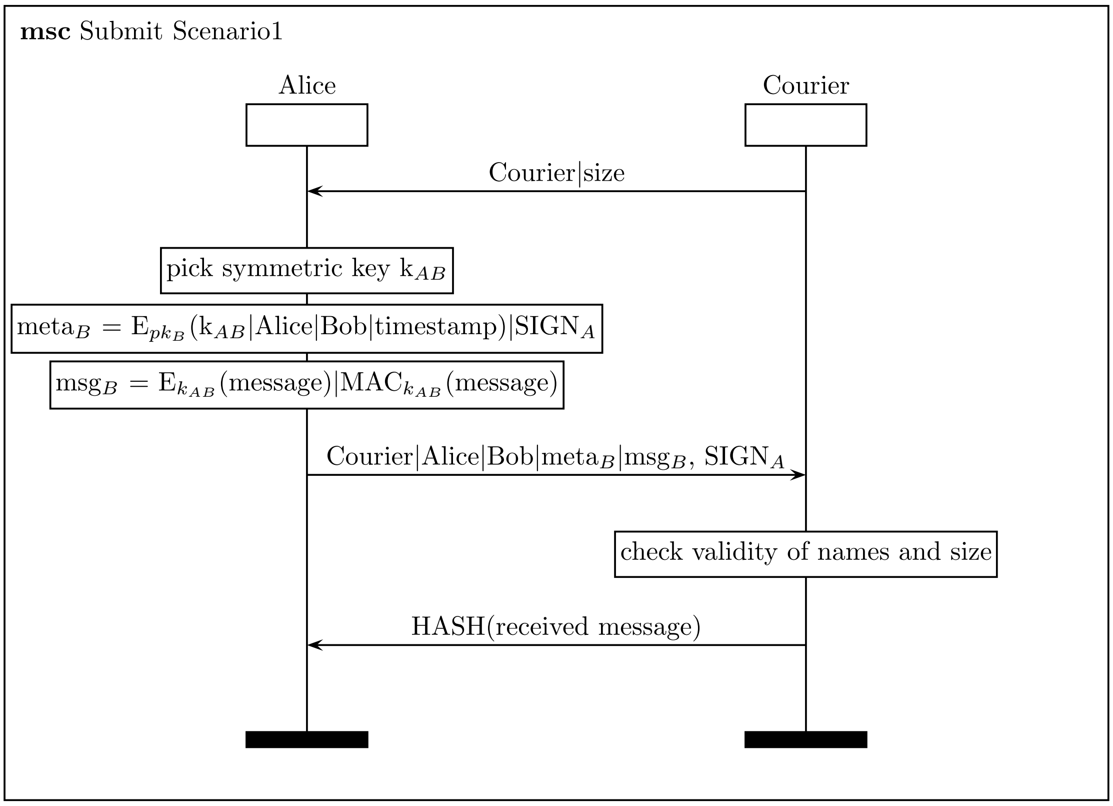
Global Assumptions

- knowledge of any public key is known by any entity in the protocol
- the cost for courier transferred from one side to the other side is considered very high

Scenario 1: Unilateral Authenticated

Submit Scenario1


msc Submit Scenario1
Alice
Courier
Courier|size
pick symmetric key k_{AB}
$\text{meta}_B = E_{pk_B}(k_{AB}|\text{Alice}|\text{Bob}|\text{timestamp})|\text{SIGN}_A$
$\text{msg}_B = E_{k_{AB}}(\text{message})|\text{MAC}_{k_{AB}}(\text{message})$
$\text{Courier}|\text{Alice}|\text{Bob}|\text{meta}_B|\text{msg}_B, \text{SIGN}_A$
check validity of names and size
HASH(received message)

Preconditions

Alice:

- Alice holds a unique asymmetric key sk_A which corresponds to its public key pk_A

Courier:

- None

Postconditions

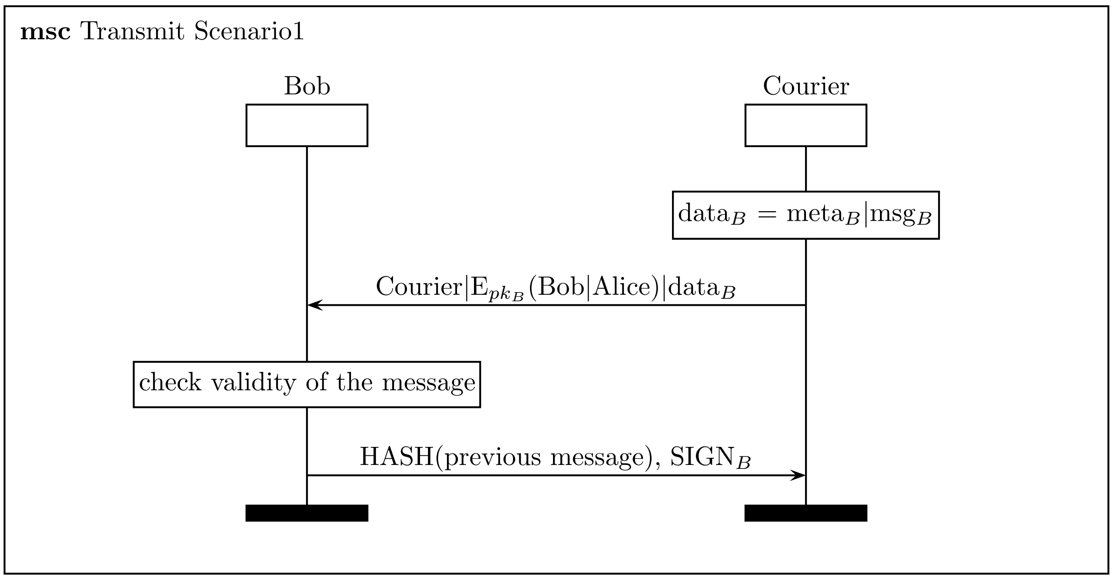
Alice:

- Alice knows all the message is successfully sent to someone
- Alice doesn't know the identity of the receiver
- Alice doesn't know whether the message will be eventually deliver to Bob

Courier:

- Courier knows the integrity of the message is preserved
- Courier knows the authenticity of origin of Alice's messages

Transmit Scenario1


msc Transmit Scenario1
Bob
Courier
data$_B$ = meta$_B$|msg$_B$
Courier|E$_{pk_B}$(Bob|Alice)|data$_B$
check validity of the message
HASH(previous message), SIGN$_B$

where:
meta$_B$ = E$_{pk_B}$(k$_{AB}$|Alice|Bob|timestamp)|SIGN$_A$
msg$_B$ = E$_{k_{AB}}$(message)|MAC$_{k_{AB}}$(message)

Preconditions

Bob:

- Bob holds a unique asymmetric key sk$_B$ which corresponds to its public key pk$_B$

Courier:

- Courier has messages for Bob

Postconditions

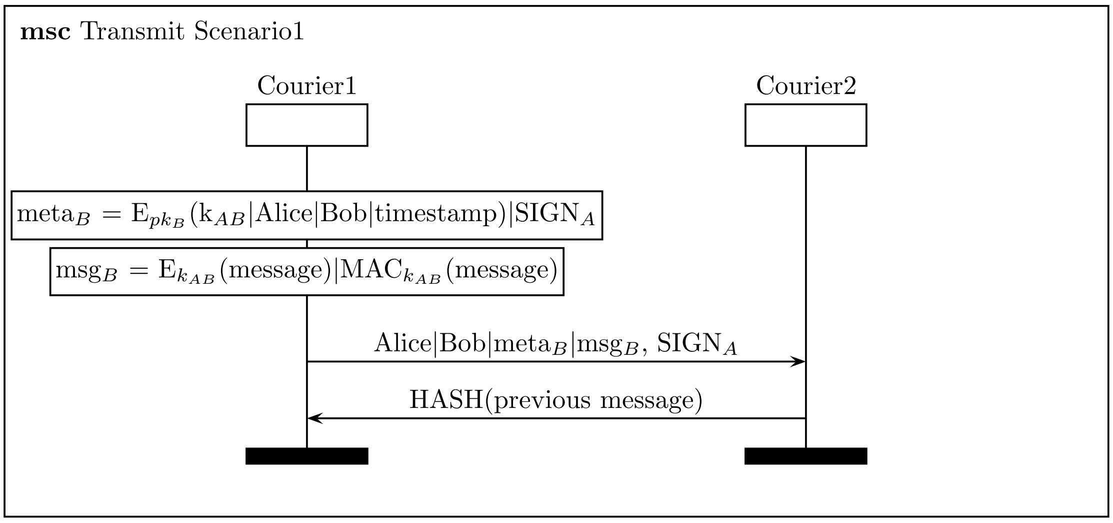
Bob:

- Bob accepts the message
- Bob knows the authenticity of origin of meta$_B$ and msg$_B$
- Bob doesn't know the identity of the message sender
- Bob doesn't know whether he receives the right message that the sender intend to send

Courier:

- if response is valid, Courier knows Bob has successfully received and accepted the message
- if response is not valid, Courier doesn't know if Bob receives the message
  it doesn't necessarily mean Bob doesn't get the message. But Courier has to resend the message again until it gets the valid response

Transmit Scenario1


msc Transmit Scenario1
Courier1
Courier2
$\mathrm{meta}_B = \mathrm{E}_{pk_B}(\mathrm{k}_{AB}|\mathrm{Alice}|\mathrm{Bob}|\mathrm{timestamp})|\mathrm{SIGN}_A$
$\mathrm{msg}_B = \mathrm{E}_{k_{AB}}(\mathrm{message})|\mathrm{MAC}_{k_{AB}}(\mathrm{message})$
$\mathrm{Alice}|\mathrm{Bob}|\mathrm{meta}_B|\mathrm{msg}_B$, SIGN_A
HASH(previous message)

Preconditions

Courier1:

- Courier1 holds the message from Alice to Bob
- Courier1 knows the name of Courier2 who is the next message carrier, so it approaches Courier2 actively
- Courier1 doesn't know Courier2's identity so any device can claim itself as Courier2

Courier2:

- Courier2 doesn't know Courier1's identity so any device can claim itself as Courier1

Postconditions

Courier1:

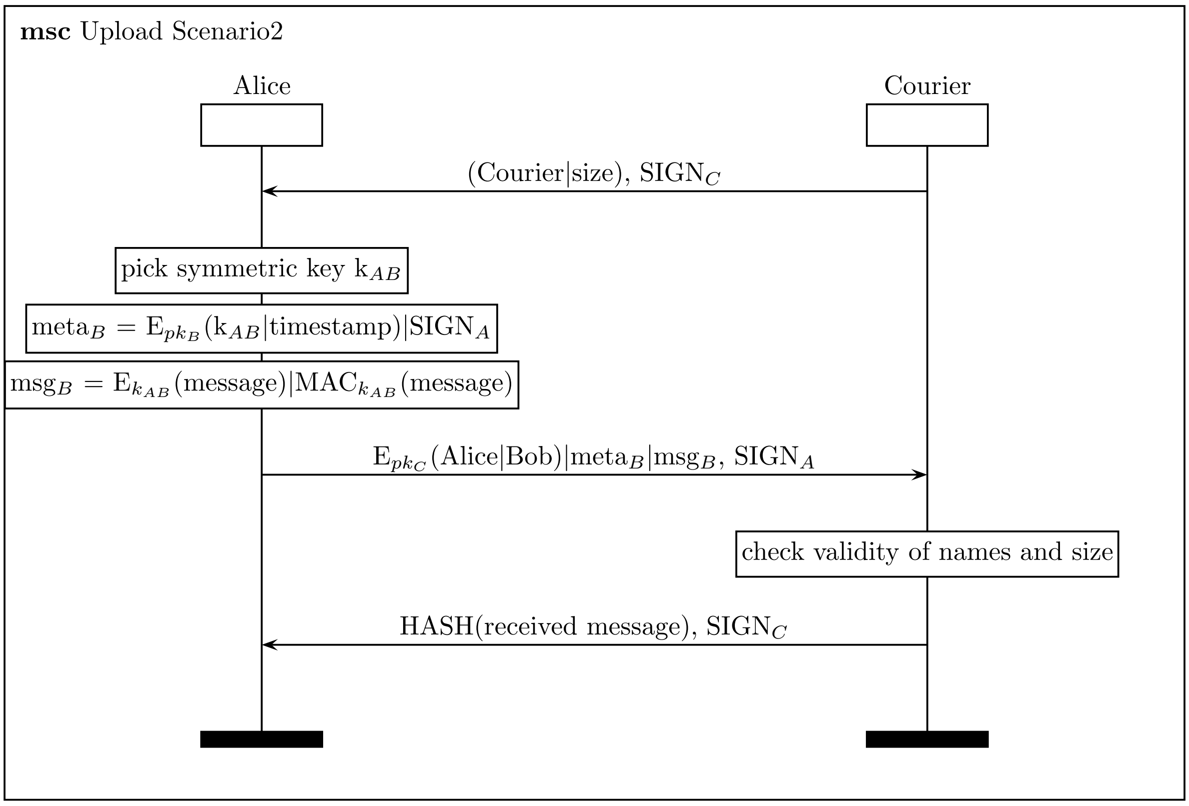
- Courier1 knows the message has been successfully sent to someone
- Courier1 doesn't know the identity of receiver

Courier2:

- Courier2 knows the authenticity of origin of the message
- Courier2 knows the original sender is Alice and recipient is Bob
- Courier2 dosen't know the identity of the sender
- Courier2 doesn't know the validity of the message

Scenario 2: Bilateral Authenticated

Upload Scenario2


msc Upload Scenario2
Alice
Courier
(Courier|size), SIGN_C
pick symmetric key k_{AB}
$\text{meta}_B = E_{pk_B}(k_{AB}|\text{timestamp})|\text{SIGN}_A$
$\text{msg}_B = E_{k_{AB}}(\text{message})|\text{MAC}_{k_{AB}}(\text{message})$
$E_{pk_C}(\text{Alice}|\text{Bob})|\text{meta}_B|\text{msg}_B$, SIGN_A
check validity of names and size
HASH(received message), SIGN_C

Preconditions

Alice:

- Alice holds a unique asymmetric key sk_A which corresponds to its public key pk_A
- the message recipient is specified (here Bob)
- Alice may/may not have messages to send

Courier:

- Courier holds a unique asymmetric key sk_C which corresponds to its public key pk_C
- Courier knows Alice is the potential message sender and approach her actively
- if Courier is empty, it should periodically check whether Alice has message to send

Postconditions

Alice:

- if response is valid, Alice knows all the message has been successfully sent to Courier
- if response is not valid, Alice doesn't know whether message has been successfully sent
  it doesn't necessarily mean Courier didn't successfully receive the message, it might. But Alice has to keep current message for future potential resending
- Alice doesn't know whether the message will be eventually deliver to Bob

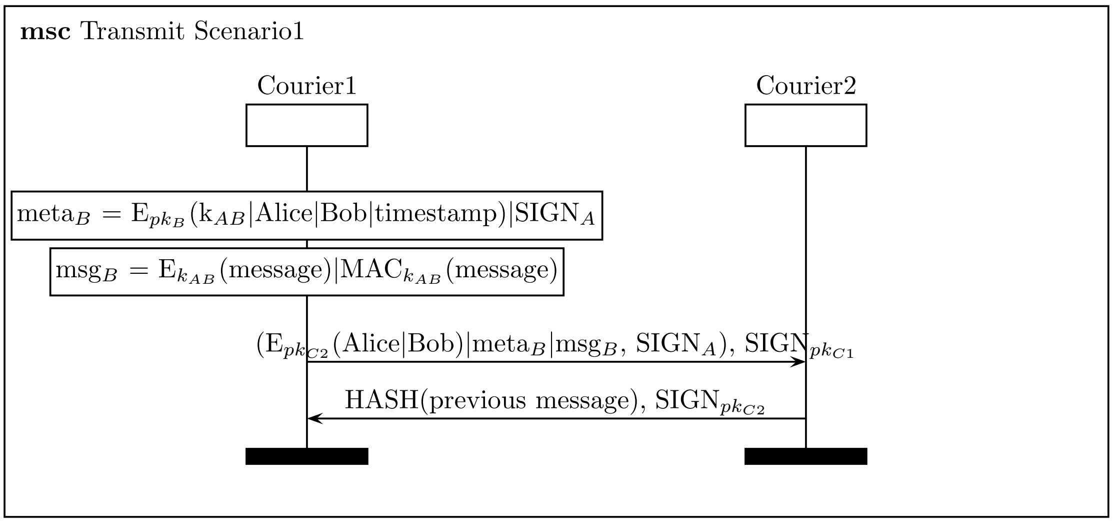
Courier:

- Courier successfully receives all Alice's messages
- Courier knows the authenticity of origin of Alice's messages
- Courier doesn't know the content of meta$_B$ and msg$_B$

Download Scenario2

Exatly same with Download Scenario1

Trasmit Scenario2



msc Transmit Scenario1
Courier1
Courier2
meta$_B$ = E$_{pk_B}$(k$_{AB}$|Alice|Bob|timestamp)|SIGN$_A$
msg$_B$ = E$_{k_{AB}}$(message)|MAC$_{k_{AB}}$(message)
(E$_{pk_{C2}}$(Alice|Bob)|meta$_B$|msg$_B$, SIGN$_A$), SIGN$_{pk_{C1}}$
HASH(previous message), SIGN$_{pk_{C2}}$

Preconditions

Courier1:

- Courier1 holds a unique asymmetric key sk$_{pk_{C1}}$ which corresponds to its public key pk$_{pk_{C1}}$
- Courier1 knows it is Courier2 who is the next message carrier, so it approaches Courier2 actively

Courier2:

- Courier2 holds a unique asymmetric key sk$_{pk_{C2}}$ which corresponds to its public key pk$_{pk_{C2}}$

Postconditions

Courier1:

- if response is valid, Courier1 knows the message has been successfully sent to Courier2

- if response is not valid, Courier1 doesn't know whether Courier2 has successfully received the message
  It doesn't necessarily mean Courier2 didn't get message correctly, but Courier1 has to resend the message until it gets valid response

Courier2:

- Courier2 knows the authenticity of origin of the message
- Courier2 knows the original sender is Alice and recipient is Bob
- Courier2 knows the message sender is Courier1 and the message integrity is preserved

Scenario 3: No Authentication

Considering the fact that anyone can pretend to be Alice and send fake message to Courier, Courier will never know which message is come from real Alice. So the Courier has to wait infinitely long before start transporting or it will be very likely that it carries all invalid messages after transporting. The potential cost is unacceptable and this method is not considered.

Pros and Cons

Scenario 1:

- simpler for communication and easier to implement
- inefficient for Alice
  because she doesn't know which is the real responsible courier, she has to response for every request and send the message to infinite number of potential couriers
- most of the messages exchanged are unprotected including the recipient name
- in Transmit Scenario, Intruder can pretend to be Courier1 and send Courier2 fake messages. Although the message won't be accepted by Bob, it may prevent Courier2 from getting real message from Courier1. And the cost of that could be huge.

Scenario 2: Vice Versa