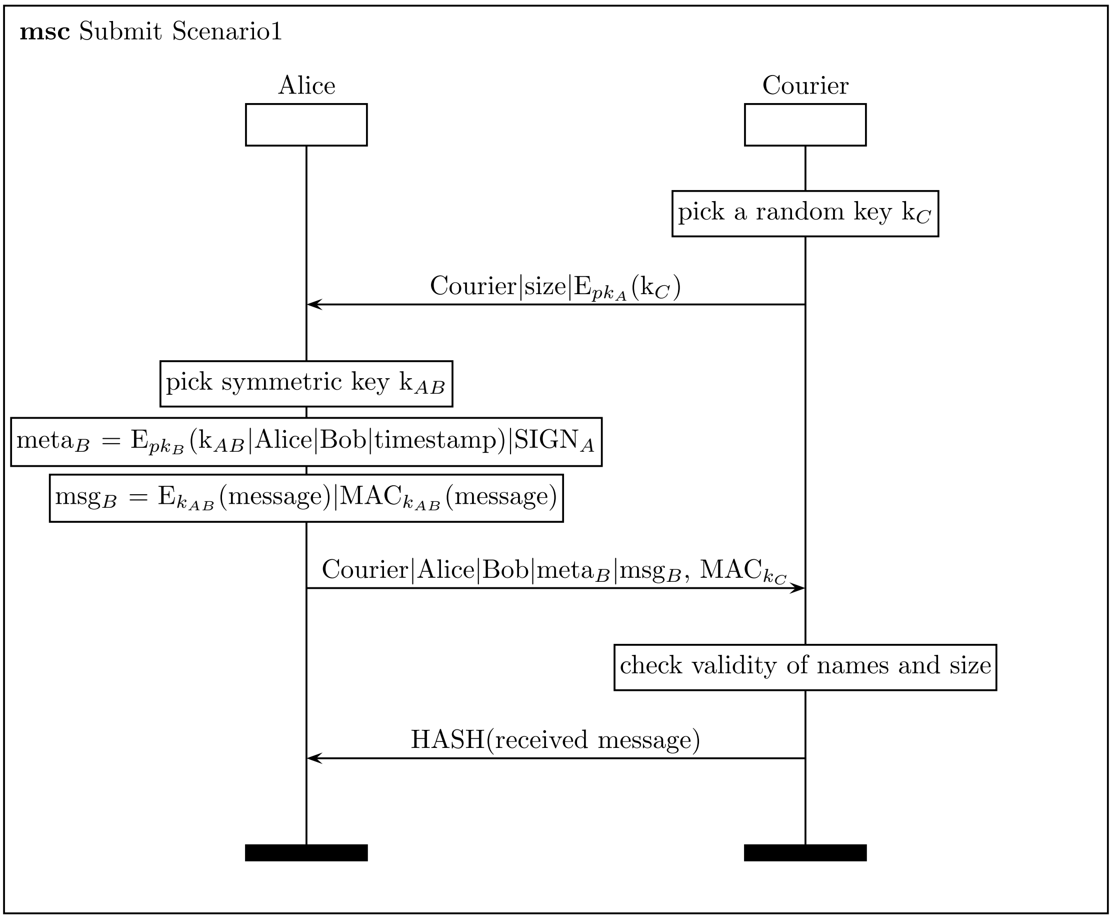
Global Assumptions

- 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
pick a random key k_C
Courier|size|$E_{pk_A}(k_C)$
pick symmetric key k_{AB}
$meta_B = E_{pk_B}(k_{AB}|Alice|Bob|timestamp)|SIGN_A$
$msg_B = E_{k_{AB}}(message)|MAC_{k_{AB}}(message)$
Courier|Alice|Bob|$meta_B$|msg_B, MAC_{k_C}
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

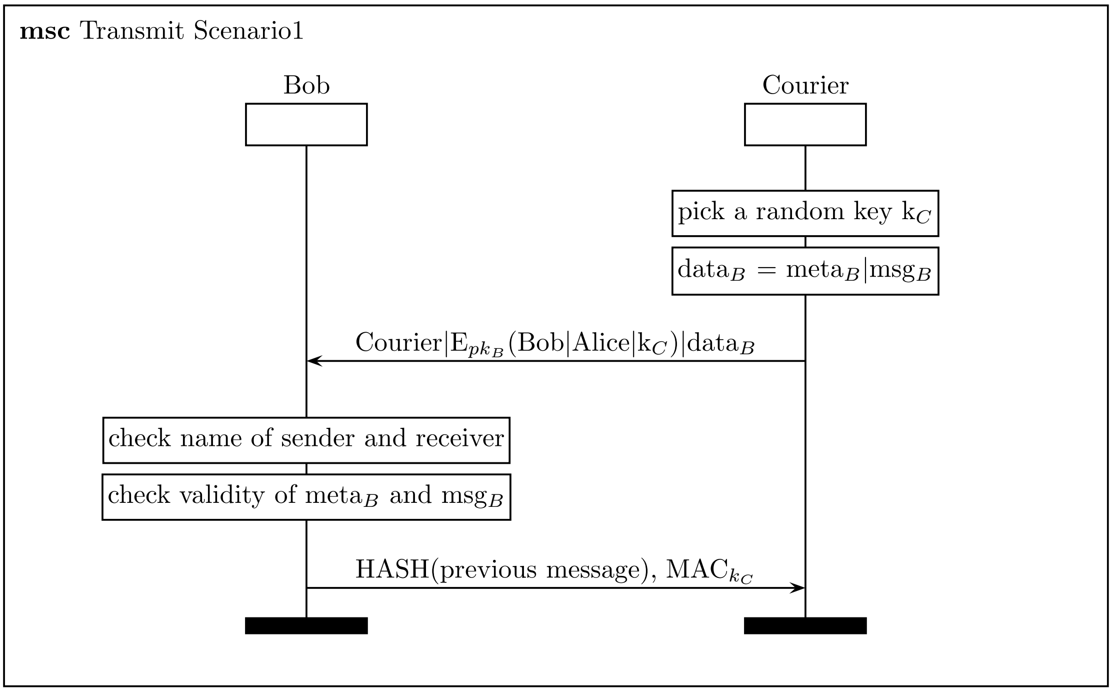
Alice:

- 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
pick a random key k_C
$data_B = meta_B|msg_B$
$Courier|E_{pk_B}(Bob|Alice|k_C)|data_B$
check name of sender and receiver
check validity of $meta_B$ and msg_B
HASH(previous message), MAC_{k_C}

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:

- None

Postconditions

Bob:

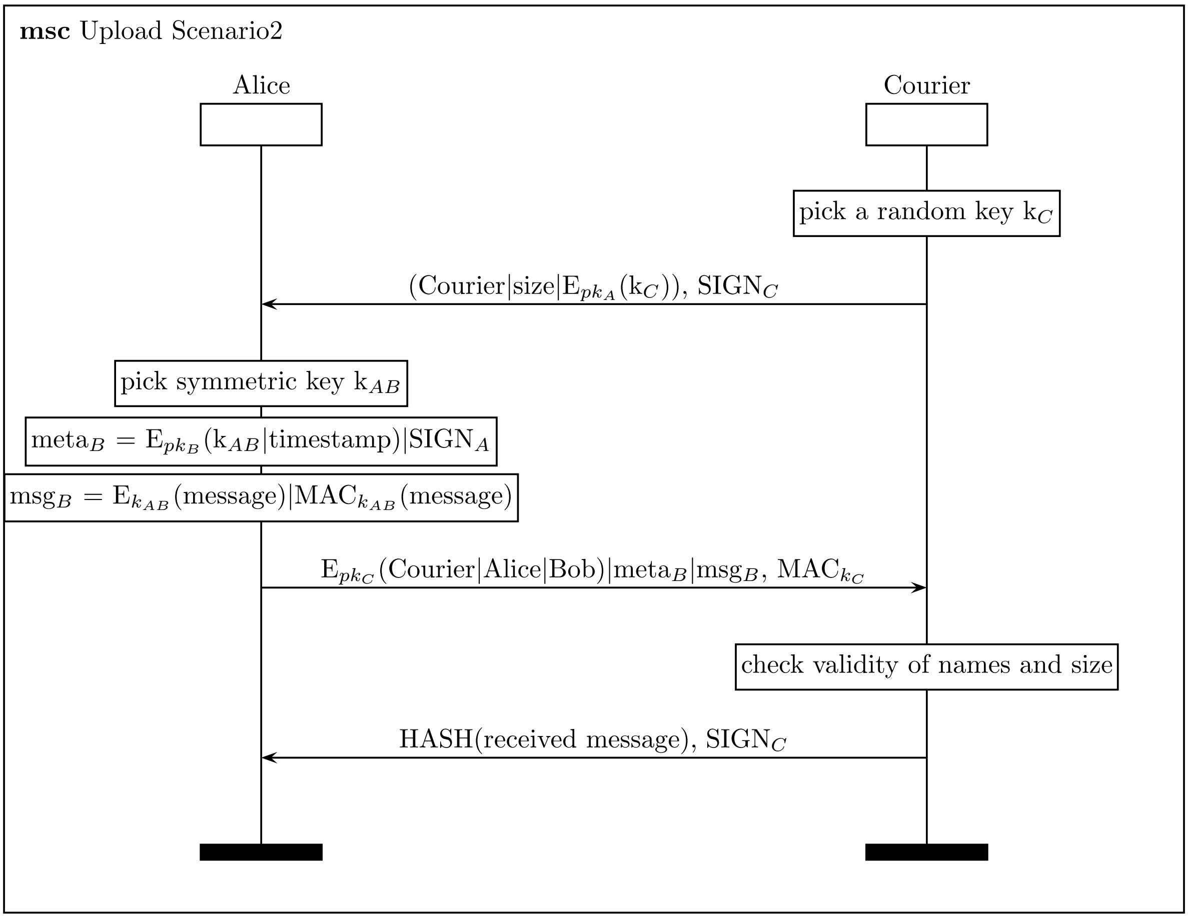
- Bob accepts the message
- Bob doesn’t know the identity of the message sender
- Bob doesn’t know whether the message has been intercepted

Courier:

- Courier knows Bob has successfully received and accepted the message

Scenario 2: Bilateral Authenticated

Upload Scenario2


msc Upload Scenario2
Alice
Courier
pick a random key k_C
(Courier|size|$E_{pk_A}(k_C)$), $SIGN_C$
pick symmetric key k_{AB}
$meta_B = E_{pk_B}(k_{AB}|timestamp)|SIGN_A$
$msg_B = E_{k_{AB}}(message)|MAC_{k_{AB}}(message)$
$E_{pk_C}(Courier|Alice|Bob)|meta_B|msg_B$, MAC_{k_C}
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

Courier:

- Courier holds a unique asymmetric key sk_C which corresponds to its public key pk_C

Postconditions

Alice:

- Alice knows all the message has been successfully received by Courier
- Alice doesn't know whether the message will be eventually deliver to Bob

Courier:

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

Transmit Scenario2

Exatly same with Download Scenario1

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