Man in the middle and replay attacks

Crypto basics: symmetric key

- Alice (A) and Bob (B) have a pre-shared key K. Only they have K
- A encrypts a message M to generate cipher text C using K. We denote this as

$$C = \{M\}_K$$

- B decrypts using K^{-1}

$$M = \{C\}_{K^{-1}}$$

- Example: *Substitution ciphers*. Attacks?

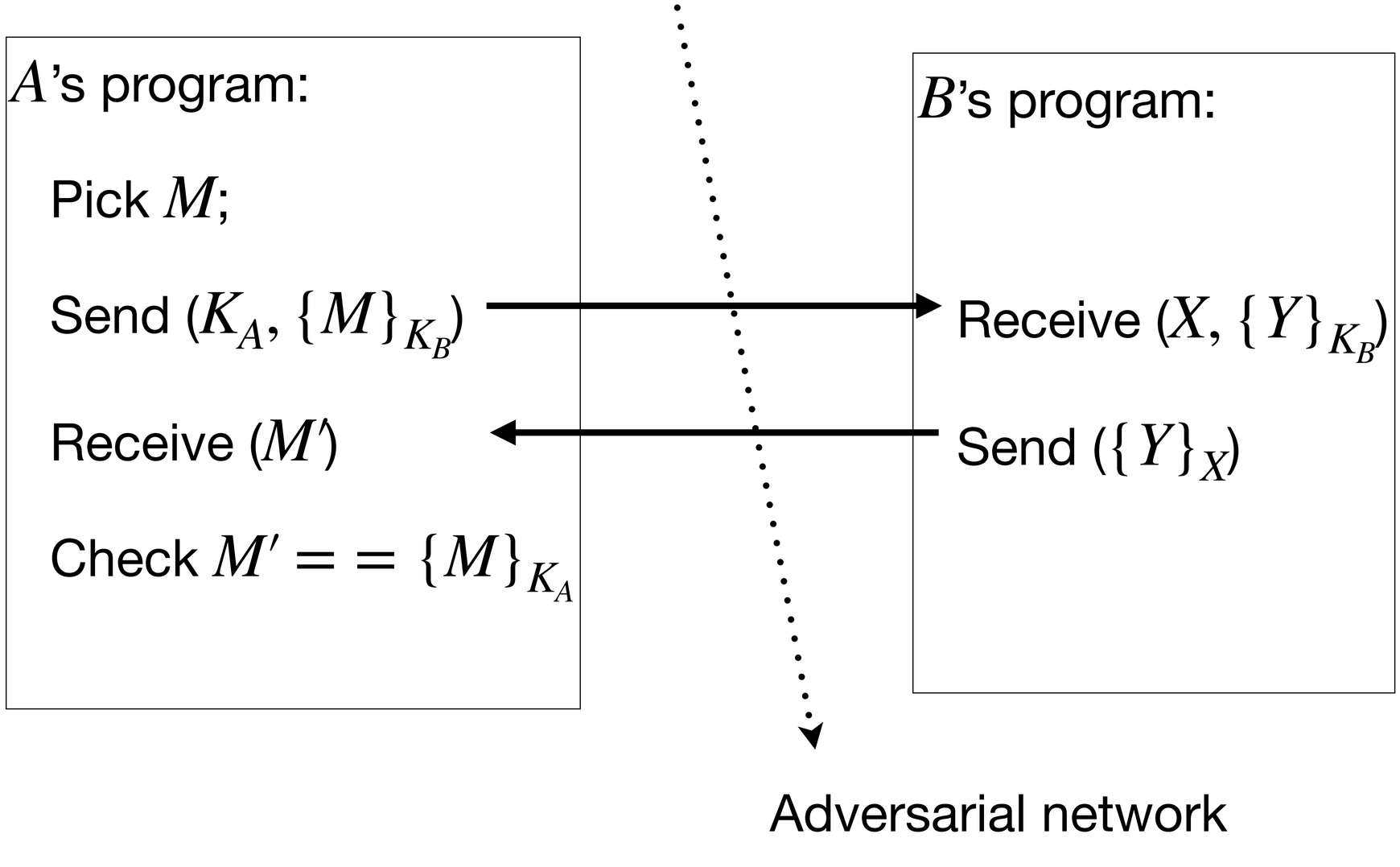
Diffie-Hellman key exchange

- A and B agree on some public parameters
- A and B independently choose their secret keys K_A and K_B respectively
- Using the public parameters, Alice and Bob independently apply a special transformation to their secret and send the transformed result (public keys) to the other party
- Upon receiving the other party's public key, Alice and Bob independently apply another transformation using the other party's public key and their own secret. This process ensures that both arrive at the same final value.
- Secure?

Crypto basic: public key cryptography

- Both A and B have public-secret key pairs (K_A, K_A^{-1}) and (K_B, K_B^{-1})
- K_A and K_B are public information, K_A^{-1} and K_B^{-1} are secret info of A and B
- For both, $C = \{M\}_K \iff M = \{C\}_{K^{-1}}$
- To **encrypt a message** M for B, A sends $C = \{M\}_{K_B}$. Only B can decrypt with $M = \{C\}_{K_B^{-1}}$
- To **sign a message** M, A computes $M' = \{M\}_{K_A^{-1}}$ and sends (M, M'). Anybody can verify $\{M'\}_{K_A} = M$.
- A can combine the above two to send a signed and encrypted message to B (**figure out how and submit by EOD**)

A crypto protocol



Secure?

- **After a valid execution, nobody other than A and B should know M**
- Does the above always hold? Assume the crypto is *bulletproof*
- Suppose P is a ***man in the middle***
- A sends $(K_A, \{M\}_{K_B})$
- P captures and sends $(K_P, \{M\}_{K_B})$ to B
- B sends back $\{M\}_{K_P}$. P captures. Gone!
- P sends back $\{M\}_{K_A}$ to A. *A's check passes.*

Certificate Authorities (CA)

- Public keys are obviously vulnerable to MITM attacks.
- A proposed solution is a *trusted third party*, a **CA** (say S).
- S may issue a certificate to each party
- For example, S may issue to A

$$C(A) = \{A, K_A, R_A, E_A\}_{K_S^{-1}}$$

- R_A and E_A usually are access rights and expiry dates.
- **Assignment:** Figure out what are the trusted third party certificates, and how are they stored on your computer/phone/browser?

The Denning-Sacco disaster (1982?)

- The protocol

$$A \longrightarrow S : A, B$$

$$S \longrightarrow A : C(A), C(B)$$

$$A \longrightarrow B : C(A), C(B), \{\{T_A, K_{AB}\}_{K_A^{-1}}\}_{K_B}$$

- Suppose B wants to masquerade as A to C?

The Denning-Sacco disaster (1982?)

- The protocol

$$A \longrightarrow S : A, B$$

$$S \longrightarrow A : C(A), C(B)$$

$$A \longrightarrow B : C(A), C(B), \{\{T_A, K_{AB}\}_{K_A^{-1}}\}_{K_B}$$

- Suppose B wants to masquerade as A to C?
- B gets from $C(C)$ from S, strips off the outer encryption $\{\dots\}_{K_B}$ from item 3
- B makes a bogus third message $B \longrightarrow C : C(A), C(C), \{\{T_A, K_{AB}\}_{K_A^{-1}}\}_{K_C}$
- **Solution?**

A session protocol

Symmetric key

$$A \longrightarrow S : \{T_A, B, K_{AB}\}_{K_{AS}}$$

$$S \longrightarrow B : \{T_S, A, K_{AB}\}_{K_{BS}}$$

- K_{AB} is the session key, which may be valid for T_S duration
- Attack?

A session protocol

Symmetric key

$$A \longrightarrow S : \{T_A, B, K_{AB}\}_{K_{AS}}$$

$$S \longrightarrow B : \{T_S, A, K_{AB}\}_{K_{BS}}$$

- K_{AB} is the session key, which may be valid for T_S duration
- Attack?
- C can prolong the session by sending $\{T_S, A, K_{AB}\}_{K_{BS}}$ to S to get back a $\{T'_S, B, K_{AB}\}_{K_{AS}}$, and send that to get back a $\{T''_S, A, K_{AB}\}_{K_{BS}}$
- C can keep the **replay attack** alive till he can steal a key

The perils of challenge-response

- Consider the following protocol for A logging onto B:

$$A \longrightarrow B : A$$

$$B \longrightarrow A : N_B \text{ (a random challenge)}$$

$$A \longrightarrow B : \{N_B\}_{K_{AS}}$$

$$B \longrightarrow S : \{A, \{N_B\}_{K_{AS}}\}_{K_{BS}}$$

$$S \longrightarrow B : \{N_B\}_{K_{BS}}$$

The perils of challenge-response

- Consider the following protocol (Woo and Lam) for A logging onto B:

$$A \longrightarrow B : A$$

$$B \longrightarrow A : N_B \text{ (a random challenge)}$$

$$A \longrightarrow B : \{N_B\}_{K_{AS}}$$

$$B \longrightarrow S : \{A, \{N_B\}_{K_{AS}}\}_{K_{BS}}$$

$$S \longrightarrow B : \{N_B\}_{K_{BS}}$$

- . The only connection between the last two exchanges is that one shortly follows the other.

Is this correct?

$A \longrightarrow B : A$

$B \longrightarrow A : N_B$ (a random challenge)

$A \longrightarrow B : \{B, N_B\}_{K_{AS}}$

$B \longrightarrow S : \{A, \{B, N_B\}_{K_{AS}}\}_{K_{BS}}$

$S \longrightarrow B : \{N_B, A\}_{K_{BS}}$

Needham-Schroeder protocol

$$A \longrightarrow B : \{N_A, A\}_{K_B}$$
$$B \longrightarrow A : \{N_A, N_B\}_{K_A}$$
$$A \longrightarrow B : \{N_B\}_{K_B}$$

How can C masquerade as A?

Threat model

- Threat actors?
- Adversaries?
- Capabilities of adversaries?
- Trust vs verifiability
- Clear articulation of all trust points