CAN304 W5

Designing security protocols

Security protocols：涉及两方或多方的一系列步骤，旨在以适当的安全性完成任务，这些步骤的顺序很重要。不同的协议假设参与者之间的信任程度不同。

例如：Payment protocol，Smart lock protocol。

Goals of security protocols

- 每个协议都旨在实现一些非常特殊的目标，协议可能只适用于该特定目的
- 重要的次要目标是极简主义
    - Fewest possible messages，Least possible data，Least possible computation

Trusted arbitrator：所有合法参与者都信任的无私的第三方。Arbitrators 通常会简化协议，但会增加开销，并可能限制适用性

Key establishment protocols

通常，我们希望每个通信会话使用不同的加密密钥。但是，我们如何将这些钥匙交给参与者？

Key establishment 是一种加密机制，它为通过开放网络进行通信的两方或多方提供共享密钥。

类别：

- Key transport protocols
    - 在密钥传输协议中，共享密钥由一方创建并安全地传输给另一方
- Key agreement (or exchange) protocols
    - 在密钥交换协议中，双方都提供用于派生共享密钥的信息

Key transport protocols

Key transport with private-key cryptography

Alice 和 Bob 想用新密钥安全地交谈，他们都信任 arbitrators Trent，并假设 Alice 和 Bob 各自与 Trent 共享一个密钥。

最初，Alice 和 Trent 分享 K_A，Bob 和 Trent 分享 K_B。

Key establishment：

1. Alice -> Trent：Alice 请求和 Bob 的会话密钥

2. Trent -> Alice：$\{K_S\}_{K_A}$，$\{K_S\}_{K_B}$ (Trent 生成密钥 K_S，并分别使用 K_A 和 K_B 加密 K_S，然后把加密后的内容发给 Alice)
3. Alice：$K_S \leftarrow \{K_S\}_{K_A}$ (Alice 使用 K_A 解密得到 K_S)

   Alice -> Bob：$\{K_S\}_{K_B}$ (Alice 把剩下的内容发给 Bob)
4. Bob：$K_S \leftarrow \{K_S\}_{K_B}$ (Bob 使用 K_B 解密得到 K_S，K_S 即他们交流所使用的密钥)

What has the protocol achieved?

- Alice 和 Bob 都有一个新的会话密钥，会话密钥是使用只有 Alice 和 Bob 知道的密钥传输的，Alice 和 Bob 都知道 Trent 参加了
- 不过这有一个漏洞

The man-in-the-middle attack

假如有一个攻击者 Mallory，他也是 Trent 的合法用户。当 Alice 索求和 Bob 的密钥时，Mallory 把信息改成 Alice 和 Mallory 的密钥，然后他就会和 Alice 共享一个密钥；之后 Mallory 再索求和 Bob 的密钥，然后他就和 Bob 也共享密钥；这样 Alice 和 Bob 以为他们在交流，实际上他们的信息经过了 Mallory 的中转，Mallory 可以窃听 Alice 和 Bob 的对话。

- Initially, Alice and Trent share K_A, Bob and Trent share K_B, Mallory and Trent share K_M
  1. Alice - > Trent: Alice requests for a session key with Bob
  2. Malory replaces the message as following

     Malory - > Trent: Alice requests for a session key with Malory
  3. Trent - > Alice: $\{K_S\}_{K_A}$, $\{K_S\}_{K_M}$
  4. Alice: $K_S \leftarrow \{K_S\}_{K_A}$

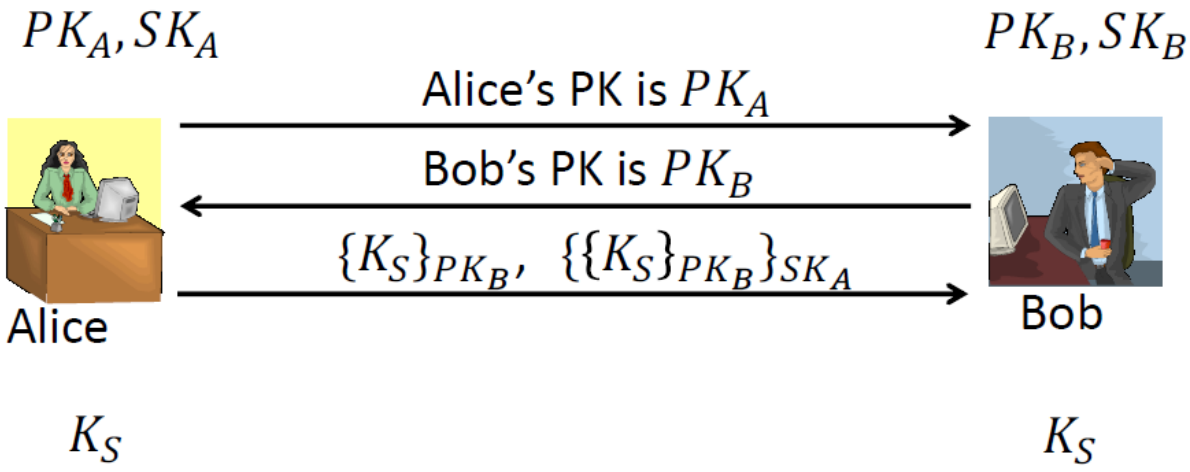
     Alice - > Bob: $\{K_S\}_{K_M}$
  5. Malory: $K_S \leftarrow \{K_S\}_{K_M}$
  6. Malory -> Trent: Mallory requests for a session key with Bob
  7. Trent - > Malory: $\{K_S'\}_{K_M}$, $\{K_S'\}_{K_B}$
  8. Malory: $K_S' \leftarrow \{K_S'\}_{K_M}$, and replace message in step 4 with $\{K_S'\}_{K_B}$
  9. Bob: $K_S' \leftarrow \{K_S'\}_{K_B}$

这个问题是缺乏完整性 (lacking integrity) 导致的，可以通过使用 MAC (包括 counter 和 identity) 解决。

Key transport with public key cryptography

此问题中，没有受信任的 arbitrator，Alice 和 Bob 互相发送他们的公钥。

Alice 生成一个会话密钥并将其发送给 Bob，用 Bob 的公钥加密，用Alice 的私钥签名；Bob 验证签名，并用他的私钥解密 Alice 的消息，然后使用共享会话密钥加密会话。



1. Alice → Bob: PK_A
2. Bob → Alice: PK_B
3. Alice → Bob: $c = \{K_S\}_{PK_B}, \sigma_A = \{c\}_{SK_A}$
4. Bob: verifies σ_A and c; extract K_S from c

Man-in-the-middle attack

攻击者 Mallory 也有一对公钥和私钥，在 Alice 和 Bob 交换公钥时，Mallory 把自己的公钥发给双方，之后 Alice 和 Bob 以为他们在交流，实际上他们的信息经过了 Mallory 的中转，Mallory 可以窃听 Alice 和 Bob 的对话。

1. Alice → Bob: PK_A
2. Mallory → Bob: PK_M
3. Bob → Alice: PK_B
4. Mallory → Bob: PK_M
5. Alice → Bob: $c = \{K_S\}_{PK_M}, \sigma_A = \{c\}_{SK_A}$
6. Mallory extract extract K_S from c

   Mallory → Bob: $c' = \{K_S\}_{PK_B}, \sigma_A{}' = \{c'\}_{SK_M}$
7. Bob: verifies $\sigma_A{}'$ and c'; extract K_S from c'

这个问题可以通过以下方法解决：

- PKI: public-key infrastructure
  - CA (certificate authority) issues public-key certificates，证书中会包含验证身份的信息
- Identity-based cryptography
  - identity 是（或可用于生成）公钥

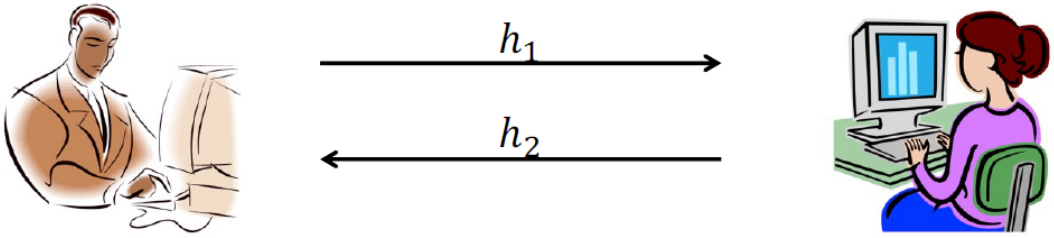
Key exchange protocols

Basic ECDH

- E: elliptic curve group
- q: prime, order of E
- G: generator of E

ECDDH problem: Given xG, yG, distinguish xyG from a uniform group element

$$E, G, q$$



$k_1 = xh_2$

$x \leftarrow Z_q$

$h_1 = xG$

$k_2 = yh_1$

$y \leftarrow Z_q$

$h_2 = yG$

通过 ECDDH 来共享密钥，最后 k1 和 k2 是相等的 (因为 $xyG = yxG$)。

Man-in-the-middle to basic ECDH

$$E, G, q$$

$k_1 = xh$

$x \leftarrow Z_q$

$h_1 = xG$

$k_3 = ch_1, k_4 = ch_2$

$c \leftarrow Z_q$

$h = cG$

$k_2 = yh$

$y \leftarrow Z_q$

$h_2 = yG$

- Reason:
  - Lack of authentication
- Solution:
  - Introduce authentication in the protocol.

Other applications of security protocols

A smart lock system

假设在开锁过程中，为了安全起见，锁只支持短距离蓝牙连接。

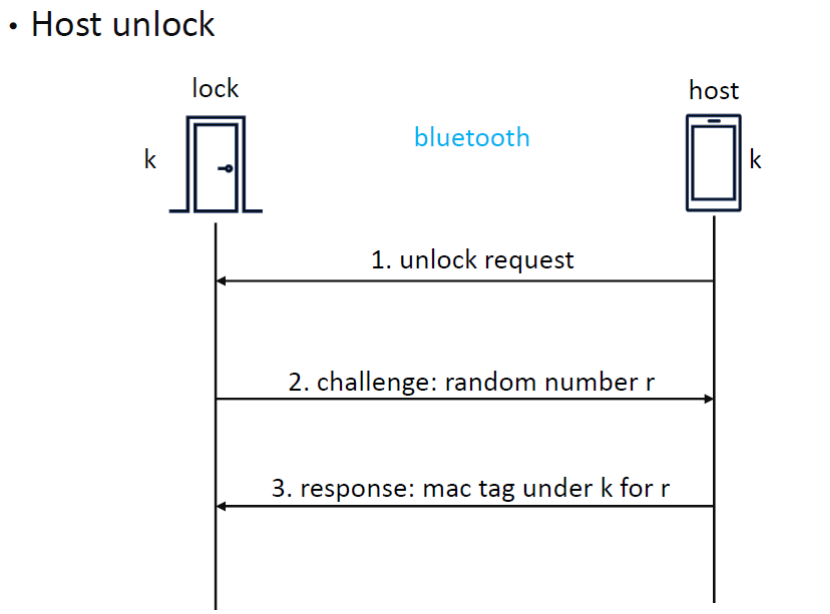
- Task/purpose
  - 当收到主机 (host) 的"解锁"请求时，解锁
  - 当收到来自主人授权客人 (guest) 的"解锁"请求时，解锁
  - 否则，保持锁定
- Security goals
  - Integrity: the request if organiated from the host and not changed improperly

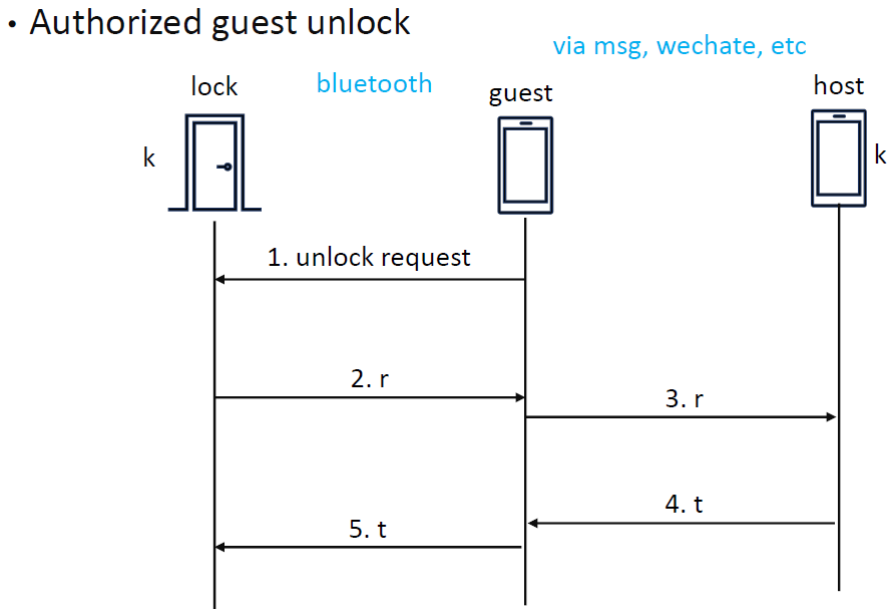
Design 1: using private-key tools

最初，锁和主机（手机）共享一个密钥 k。

Protocol I: host unlock；Protocol II: guest unlock。

主要思路是 host 根据密钥和随机数 r 生成 MAC tag，然后锁验证 tag。





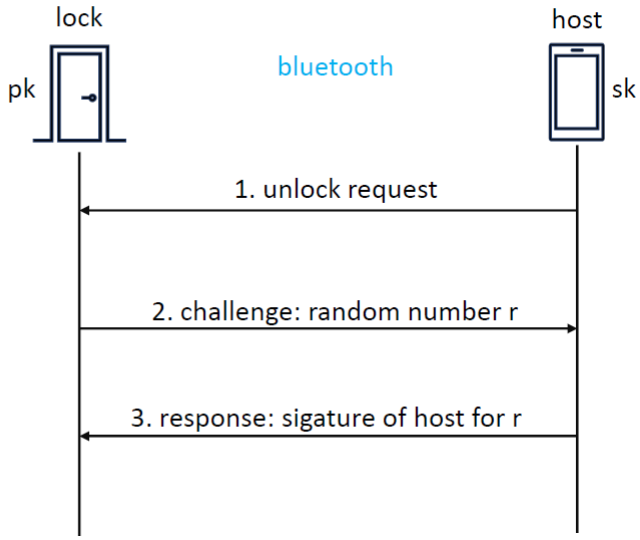
Design 2: using public-key tools

最初，锁存储主机的公钥。

Protocol I: host unlock；Protocol II: guest unlock。

主要思路是 host 根据私钥和随机数 r 生成签名，然后锁用公钥验证签名。

- Host unlock



- Authorized guest unlock