CAN304
Computer Systems Security

Lecture 5. Security Protocol

Week 5: 2022-03-25, 14:00-16:00, Friday

Jie Zhang
Department of Communications and Networking
Email: jie.zhang01@xjtlu.edu.cn
Office: EE522

Review

- Encryption
- Message authentication code
- Digital signature
- DH key agreement

- Symmetric cryptography
- Asymmetric cryptography

Outline

- Designing security protocols
- Key establishment protocols
- Other applications of security protocols

Learning objectives

- Understand key establishment protocols.
- Be able to design security protocols for given scenarios.

1. Designing security protocols

Security protocols

- A series of steps involving two or more parties designed to accomplish a task with suitable security
- Sequence is important
- Different protocols assume different levels of trust between participants

Participants in security protocols

Alice

Bob

And the bad guys

Eve

Who only listens passively

And sometimes Alice or Bob might cheat

Mallory

Who is actively malicious

Trusted arbitrator

Trent

- A disinterested third party trusted by all legitimate participants
- Arbitrators often simplify protocols, but add overhead and may limit applicability

Goals of security protocols

- Each protocol is intended to achieve some very particular goal
  - Like setting up a key between two parties
- Protocols may only be suitable for that particular purpose
- Important secondary goal is minimalism
  - Fewest possible messages
  - Least possible data
  - Least possible computation

Examples

- Payment protocol
- Smart lock protocol

2. Key establishment protocols

- Key transport protocols
- Key exchange protocols

Key establishment protocols

- Often, we want a different encryption key for each communication session
- How do we get those keys to the participants?
  - Securely
  - Quickly
  - Even if they've never communicated before

Key establishment protocols

- Key establishment is a cryptographic mechanism that provides two or more parties communicating over an open network with a shared secret key.

- Category
  - Key transport protocols. In a key transport protocol, the shared secret key is created by one party and securely transmitted to the second party.
  - Key agreement (or exchange) protocols. In a key exchange protocol, both parties contribute information which is used to derive the shared secret key.

2.1 Key transport protocols

Key transport with private-key cryptography

- Alice and Bob want to talk securely with a new key
- They both trust Trent
  - Assume Alice & Bob each share a key with Trent
- How do Alice and Bob get a shared key?

Key transport with private-key cryptography

K_A

Alice

K_B

Bob

K_A K_B

Trent

Protocol

- Initially, Alice and Trent share K_A, Bob and Trent share K_B

- Key establishment
  1. Alice - > Trent: Alice requests for a session key with Bob
  2. Trent - > Alice: $\{K_S\}_{K_A}, \{K_S\}_{K_B}$
  3. Alice: $K_S \leftarrow \{K_S\}_{K_A}$

     Alice - > Bob: $\{K_S\}_{K_B}$
  4. Bob: $K_S \leftarrow \{K_S\}_{K_B}$

What has the protocol achieved?

- Alice and Bob both have a new session key
- The session key was transmitted using keys known only to Alice and Bob
- Both Alice and Bob know that Trent participated

- But there are vulnerabilities
- What if the initial request was grabbed by Mallory?

The man-in-the-middle attack

K_A

Alice

K_M

Mallory

K_B

Bob

K_M

K_A K_B

Trent

The man-in-the-middle attack

- 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: $\mathrm{K}_S' \leftarrow \{K_S'\}_{K_B}$

Defeating the man-in-the-middle

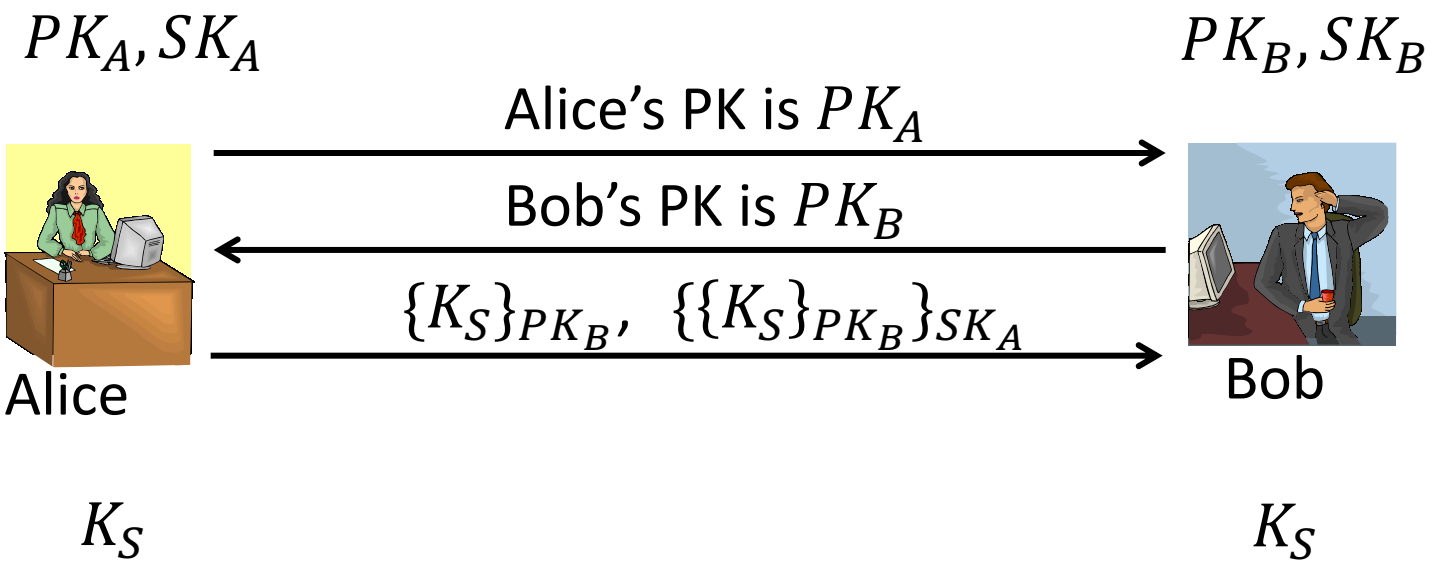
- Problems:
  - Lacking integrity

- Minor changes can fix that
  - using MAC
  - including timestamps/counter and identity in the messages

Key transport with public key cryptography

- With no trusted arbitrator
- Alice sends Bob her public key
- Bob sends Alice his public key
- Alice generates a session key and sends it to Bob encrypted with his public key, signed with her private key
- Bob decrypts Alice's message with his private key
- Encrypt session with shared session key

Basic key transport using PK



Bob verifies the message came from Alice

Bob extracts the key from the message

Protocol

1. Alice $\rightarrow$ Bob: PK_A
2. Bob $\rightarrow$ Alice: PK_B
3. Alice $\rightarrow$ 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

PK_A, SK_A

Alice

PK_M, SK_M

Mallory

PK_B, SK_B

Bob

Man-in-the-middle attack

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

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

Solutions

- PKI: public-key infrastructure
  - CA (certificate authority) issues public-key certificates
- Identity-based cryptography
  - the identity is (or can be used to derive) the public-key

2.2 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

h_1

h_2

$k_1 = xh_2$

$x \leftarrow Z_q$

$h_1 = xG$

$k_2 = yh_1$

$y \leftarrow Z_q$

$h_2 = yG$

Man-in-the-middle to basic ECDH

$$E, G, q$$

$k_1 = xh$ $\quad$ $k_3 = ch_1, k_4 = ch_2$ $\quad$ $k_2 = yh$

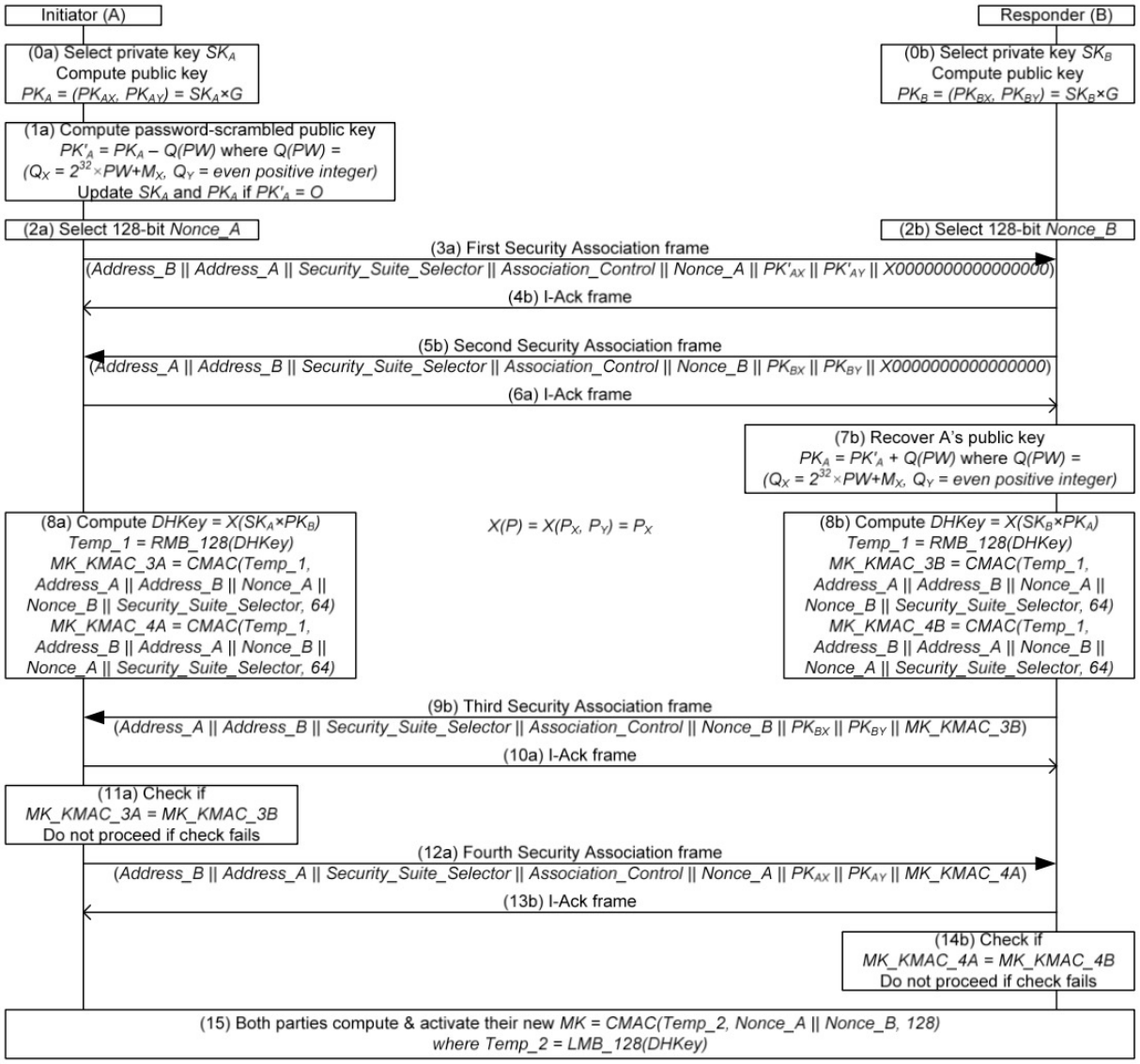
$x \leftarrow Z_q$ $\quad$ $c \leftarrow Z_q$ $\quad$ $y \leftarrow Z_q$

$h_1 = xG$ $\quad$ $h = cG$ $\quad$ $h_2 = yG$

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

ECDH-based AKE example: PW-AKE

- IEEE 802.15.6 Password Authenticated Association



Abstracted version:

- Initialization:
  - A, B share group parameters and password PW
- Key exchange:
  1. A: generates SK_A, PK_A, computes $PK'_A = PK_A - PW$, generates N_A

     $\mathrm{A} \rightarrow \mathrm{B}: B, A, N_A, PK'_A$
  2. B: generates SK_B, PK_B, generates N_B

     $\mathrm{B} \rightarrow \mathrm{A}: A, B, N_B, PK_B$
  3. computes $PK_A = PK'_A + PW$, computes shared secret $K = SK_B \cdot PK_A$, $t_B = MAC(K_x, A, B, N_A, N_B)$

     $\mathrm{B} \rightarrow \mathrm{A}: B, A, N_B, PK_B, t_B$
  4. A: computes shared secret $K = SK_A \cdot PK_B$, verifies t_B, computes $t_{\mathrm{A}} = MAC(K_x, B, A, N_B, N_A)$

     $\mathrm{A} \rightarrow \mathrm{B}: B, A, N_A, PK_A, t_A$
  5. B: verifyt_A

Zhang, J., Huang, X., Craig, P., Marshall, A. and Liu, D., 2016. An improved protocol for the password authenticated association of IEEE 802.15. 6 standard that alleviates computational burden on the node. *Symmetry*, *8*(11), p.131.

3. Other applications of security protocols

Smart lock system

Suppose in unlock process, the lock only support short-range bluetooth connection for security purpose.

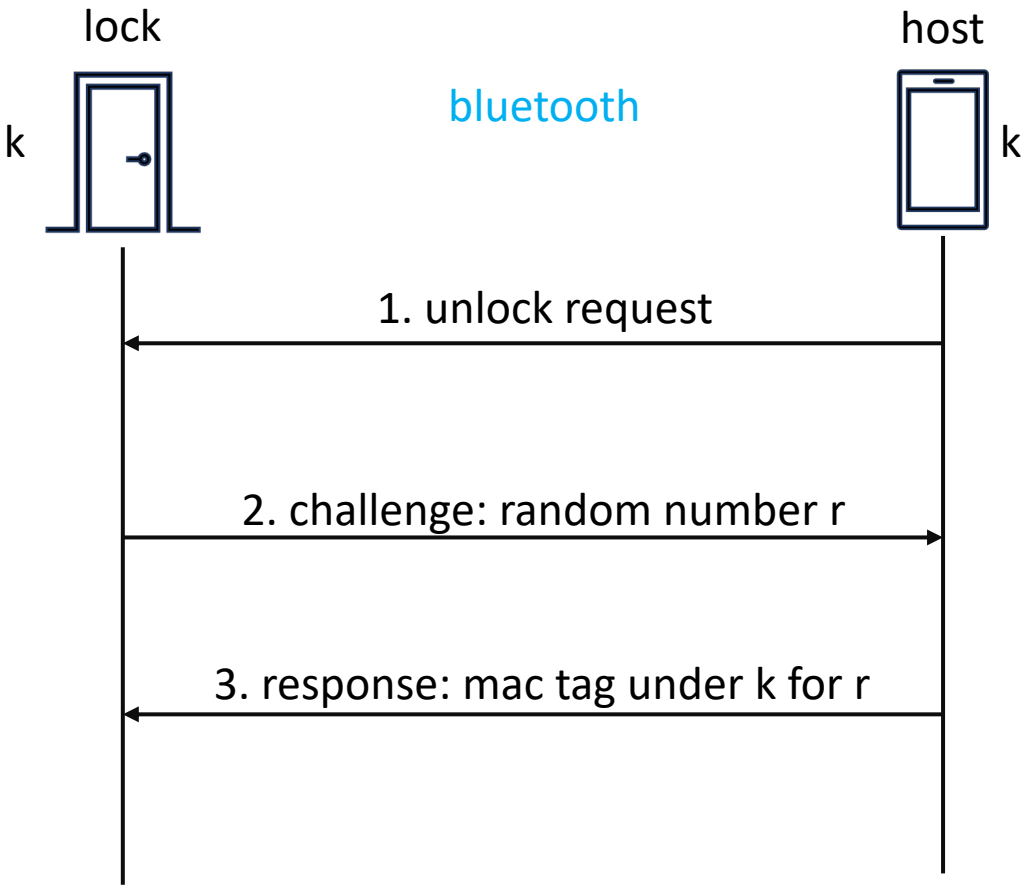
- Task/purpose
  - When receive a “unlock” request from the host, unlock
  - When receive a “unlock” request from host authorized guest, unlock
  - Othervise, keep locked
- Security goals
  - Integrity: the request if organiated from the host and not changed improperly

Design 1: using private-key tools

- Initially, the lock and the host (phone) share a secret key
- Protocol I: host unlock
- Protocol II: guest unlock

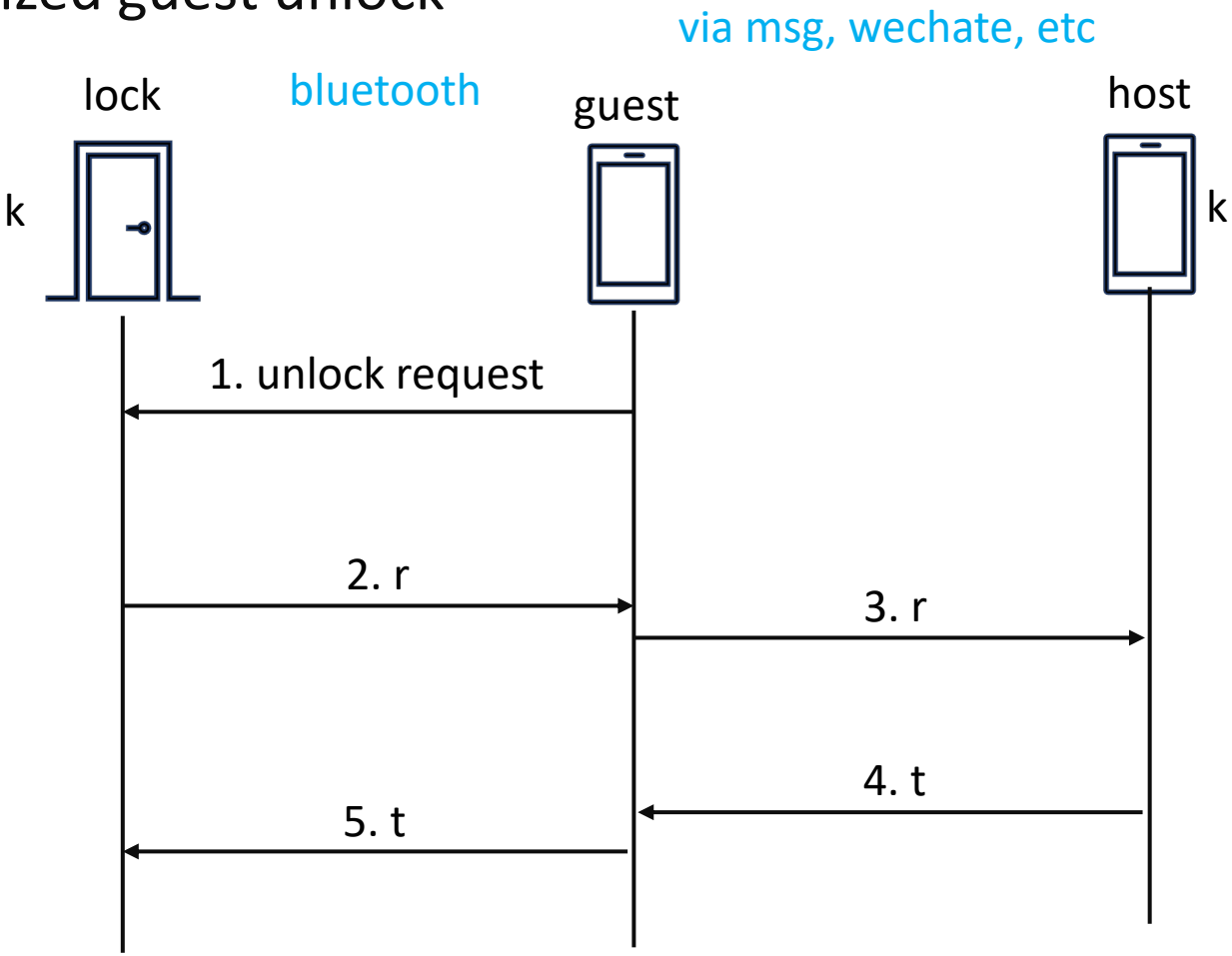
Protocol I

- Host unlock



Protocol II

- Authorized guest unlock



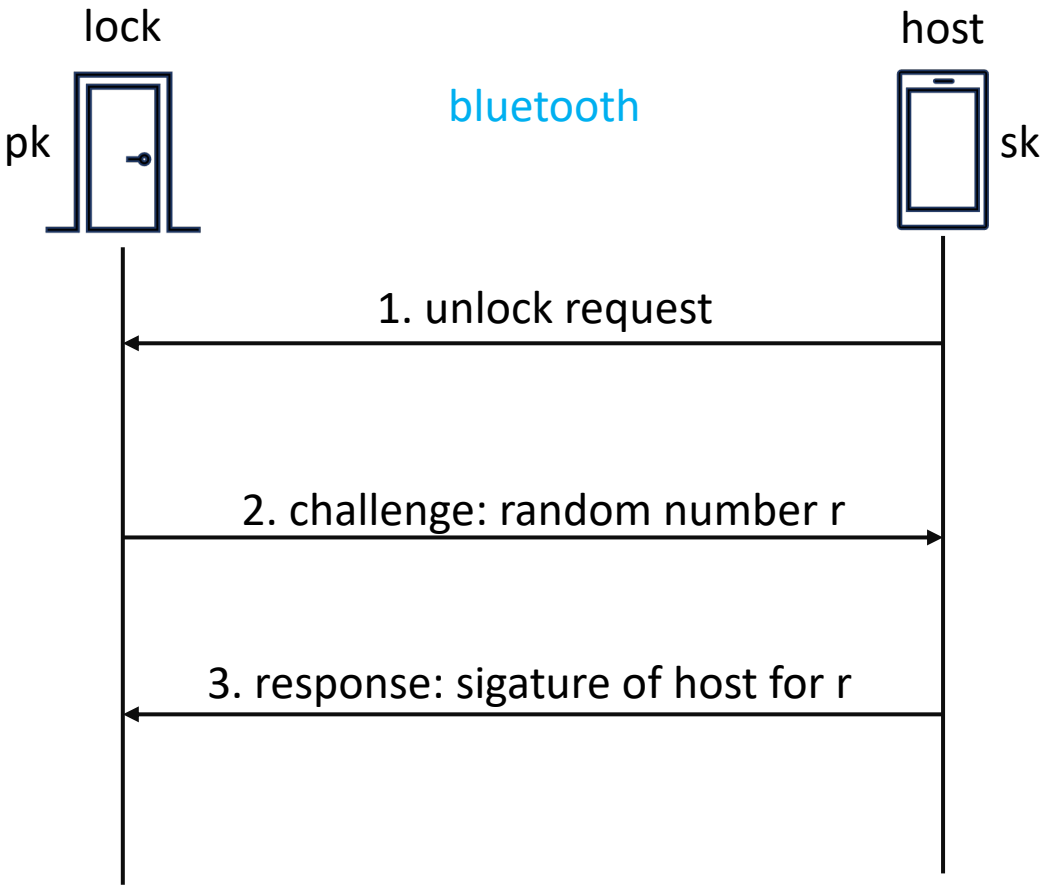
In implementation, the guest even neednot set up any app speciliazed for this smart lock system. How?

Design 2: using public-key tools

- Initially, the lock stores the public key of the host
- Protocol I: host unlock
- Protocol II: guest unlock

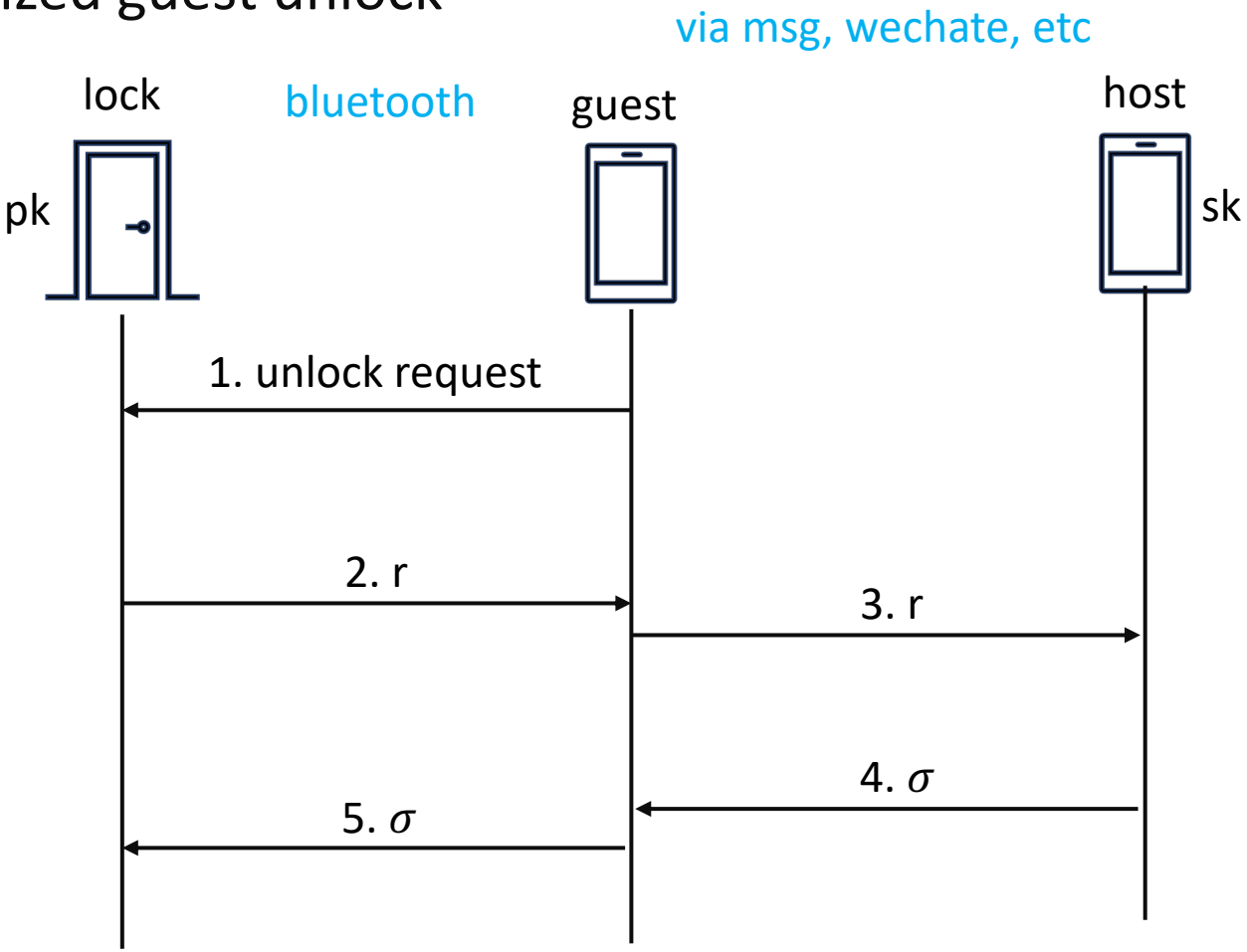
Protocol I

- Host unlock



Protocol II

- Authorized guest unlock



Finally, how to evaluate a security protocol?

- Matheticatics proof
- Formal methods: BAN, GNY logics
  - Further reading paper
- Tools
  - CSP Casper: https://www.cs.ox.ac.uk/gavin.lowe/Security/Casper/

Summary

- Designing security protocols
- Key establishment protocols
  - key transport
  - key agreement
- Other applications of security protocols
  - A smart lock system