CSE316
Computer Systems Security

Lab 1. Secure communication for IoT

2020-04-08, 13:00-14:00, Wednesday

Jie Zhang
Department of Computer Science and Software Engineering
Email: jie.zhang01@xjtlu.edu.cn
Office: SD561

Outline

- Introduction
- IEEE 802.15.6
- Bluetooth 5.0
- TLS 1.3

Tasks

- Get familiar with experimental codes and environment of AKE protocols in IEEE 802.15.6, Bluetooth 5.0, and TLS 1.3
- 3 weeks

1. Introduction

Communication

- Communication takes place at every moment.
- Many of them involve sensitive information.





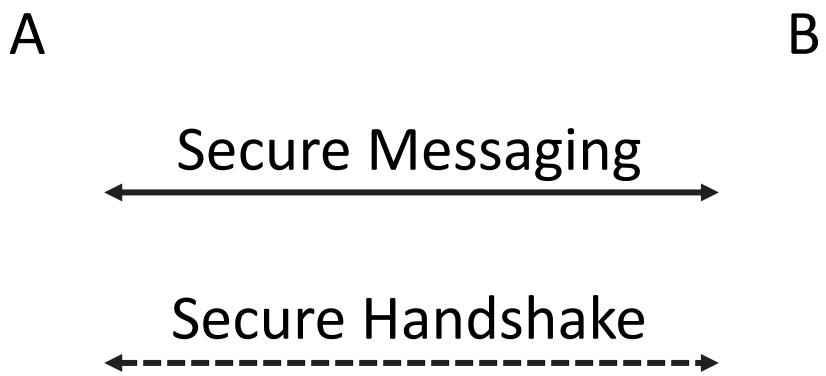
How to protect communication

- Confidentiality
  - Encryption
    - E.g., AES
- Integrity / Authenticity
  - Message authentication code
    - E.g., HMAC,CBC-MAC
  - Digital signature

Shared key

Protecting communication

- 2 phases



Protecting communication

- Phase 1, Secure handshake
  - Establish authenticated shared keys
  - Also known as
    - Authenticated key establishment.
    - Authenticated association (IEEE 802.15.6)
    - Secure Pairing (Bluetooth)
- Phase 2, Secure messaging
  - Use Encryption, Message Authentication Code, etc., to protect messages under shared keys.

Authenticated key exchange (AKE)

- Definition
  - Authenticated 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.

AKE in international standards

- Communication standards
  - IEEE Standard 802.15.6 (WBAN)
  - Bluetooth Specification
- Security standards
  - Transport Layer Security (TLS)

IEEE standard 802.15.6 (WBAN)

- WBAN – Wireless Body Area Networks
  - Short-range, wireless communications in the vicinity of, or inside, a human body (but not limited to humans)
- Security Services
  - Based on ECDH
- Security Association
  - Password Authenticated Association
  - Display Authenticated Association

Bluetooth specification v 5.0

- Bluetooth wireless technology
  - a short-range communications system intended to replace the cable(s) connecting portable and/or fixed electronic devices.
  - key features: robustness, low power consumption, and low cost.
- Security Overview
  - v 2.1+ introduced secure simple pairing (utilizes ECDH)
- Secure Simple Pairing
  - Numeric Comparison
  - Just Works
  - Out Of Band
  - Passkey Entry

Transport layer security (TLS)

- Primary goal
  - provide a secure channel between two communicating peers
- Components:
  - A handshake protocol
    - authenticates the communicating parties, negotiates cryptographic modes and parameters, and establishes shared keying material
  - A record protocol
    - uses the parameters established by the handshake protocol to protect traffic between the communicating peers

2. ECDH

Diffie-Hellman key agreement

G: cyclic group

q: prime, order of G

g: generator of G

Decisional Diffie-Hellman (DDH) problem:
Given g^x, g^y, distinguish g^{xy} from a uniform group element

Hardness

G, q, g

h_1

h_2

$k_1 = (h_2)^x$

$x \leftarrow Z_q$

$h_1 = g^x$

$k_2 = (h_1)^y$

$y \leftarrow Z_q$

$h_2 = g^y$

Elliptic curve Diffie-Hellman key agreement

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

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

Hardness

E, G, q

h_1

h_2

$k_1 = xh_2$

$x \leftarrow Z_q$

$h_1 = xG$

$k_2 = yh_1$

$y \leftarrow Z_q$

$h_2 = yG$

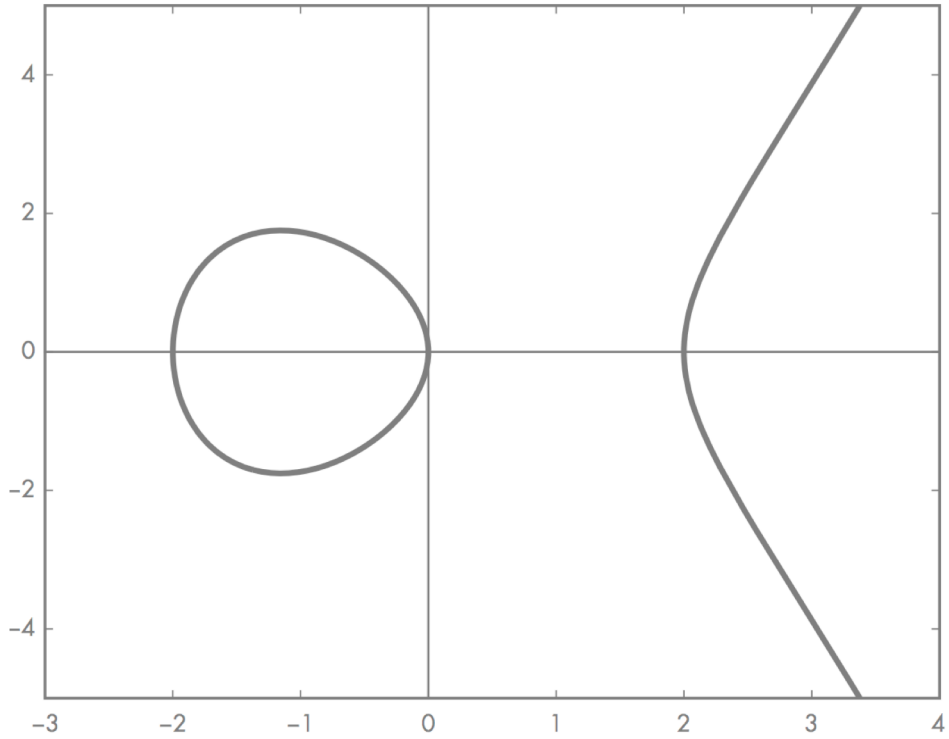
Elliptic curve used in cryptography

- An elliptic curve as used in cryptography is typically a curve whose equation is of the form

$$y^2 = x^3 + ax + b$$

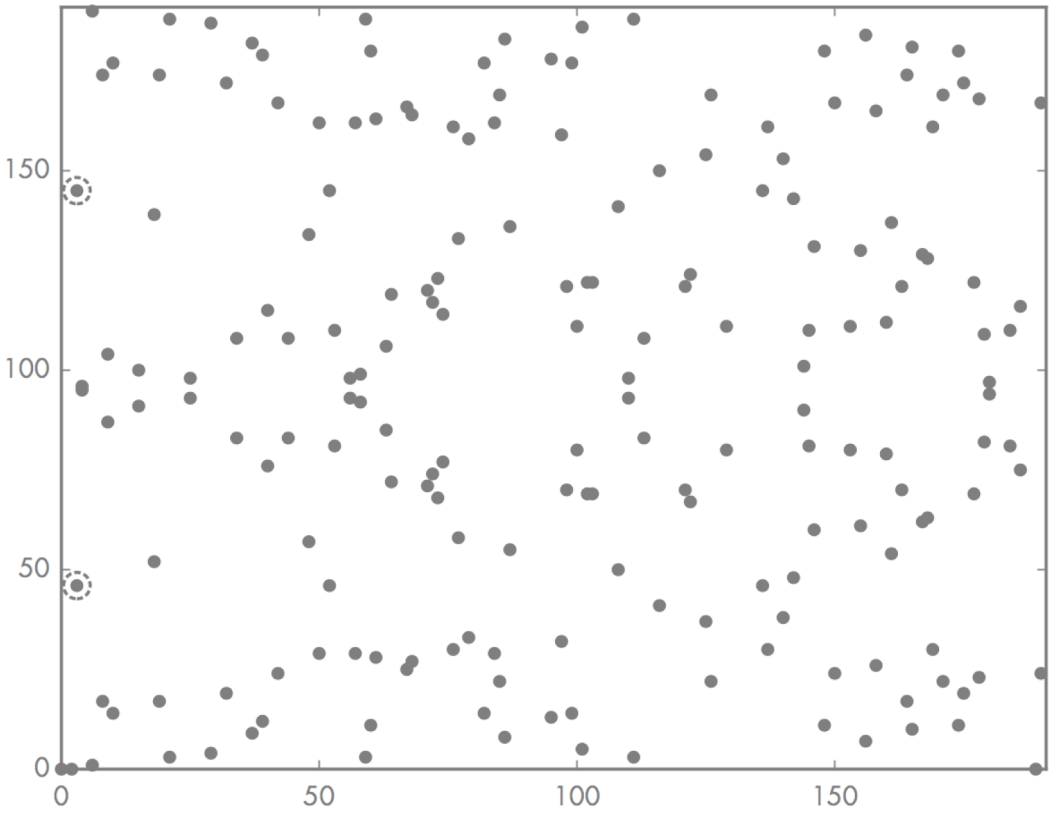
- a, b: constants that define the shape of the curve
- Weierstrass form

- Example: $y^2 = x^3 - 4x$
  - over real numbers



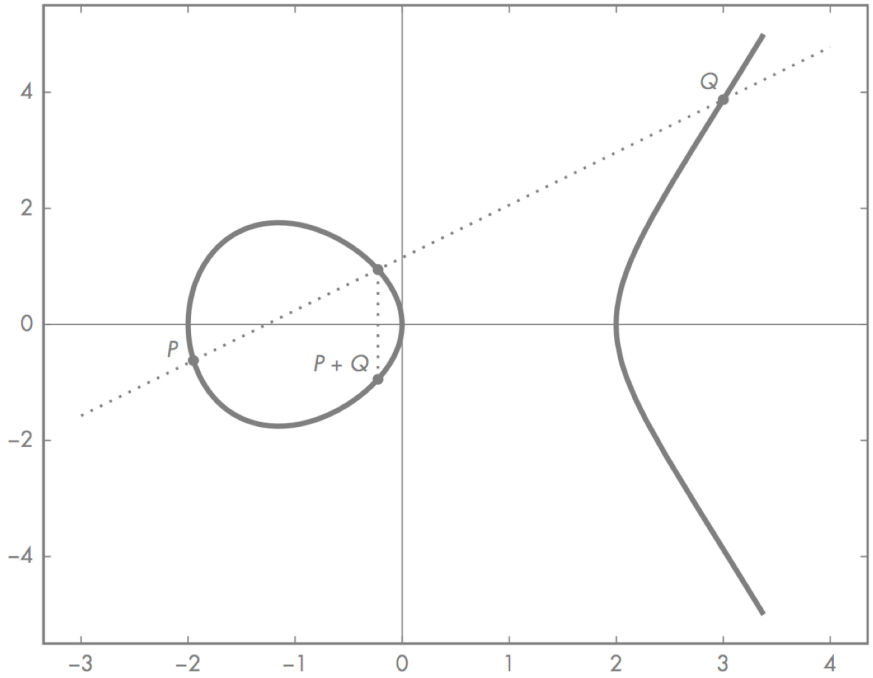
Elliptic curve over Integers

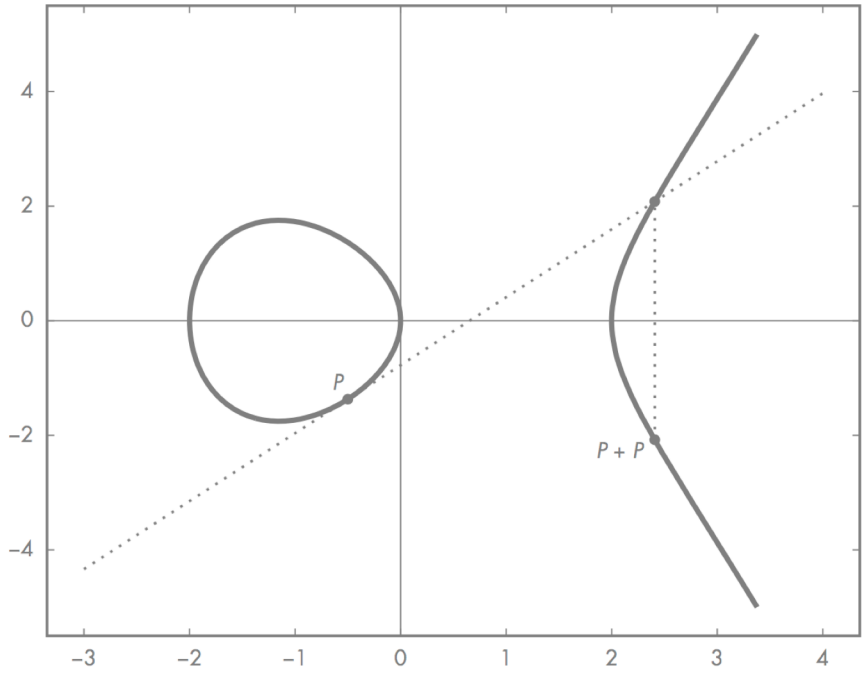
- Example: $y^2 = x^3 - 4x$ over Z_{191}
  - Z_{191}: the set of integers modulo 191



Operations

- Addition
  - $R = P + Q$
- Scalar multiplication
  - $Q = tP = P + \cdots + P$





Difficult problems

- ECDLP
  - Given a base point P and a point Q in the group E where $Q = kP$, finding the number k
- ECCDH
  - Given a base point P and two points Q_1 and Q_2 in the group E where $Q_1 = k_1P$ and $Q_2 = k_2P$, finding the point $\mathrm{Q} = k_1k_2P$
- ECDDH
  - Given a base point P and two points Q_1 and Q_2 in the group E where $Q_1 = k_1P$ and $Q_2 = k_2P$, distingish the point $\mathrm{Q} = k_1k_2P$ from a uniform element of E.

Man-in-the-middle to EC(DH)

$$E, G, q$$

$k_1 = xh$

$x \leftarrow Z_q$

$h_1 = xG$

$k_3 = ch_1, k_4 = ch_2$

$c \leftarrow Z_q$

$h = cG$

$k_2 = yh$

$y \leftarrow Z_q$

$h_2 = yG$

- Solution:
  - Introduce authentication in the protocol.

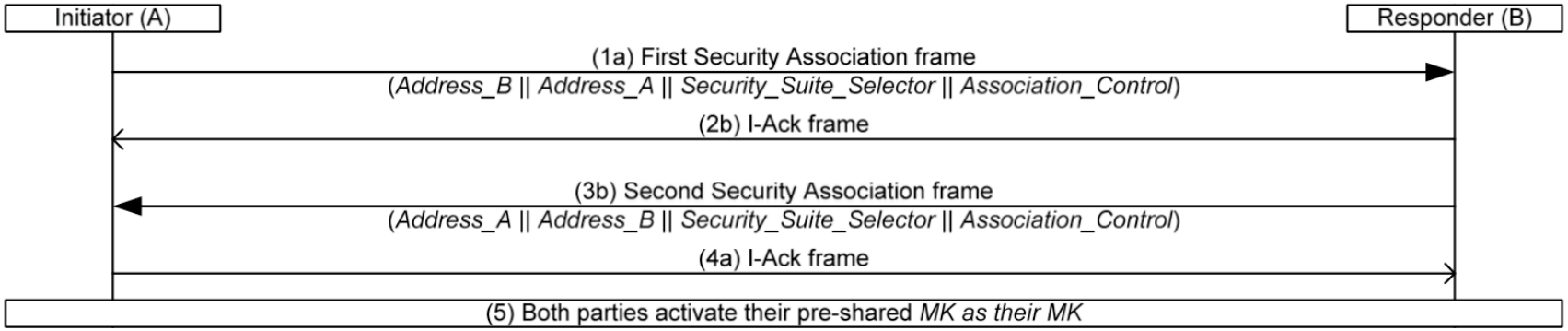
3. ECDH-based AKE in Standards

ECDH-based AKE in IEEE 802.15.6

- Secure association:
  - 1 Activating a pre-shared or generating a new shared master key (MK)
    - Master key pre-shared association
    - Unauthenticated association
    - Public key hidden association
    - Password authenticated association
    - Display authenticated association
  - 2 Pairwise temporal key (PTK) creation
- Group temporal key (GTK) distribution

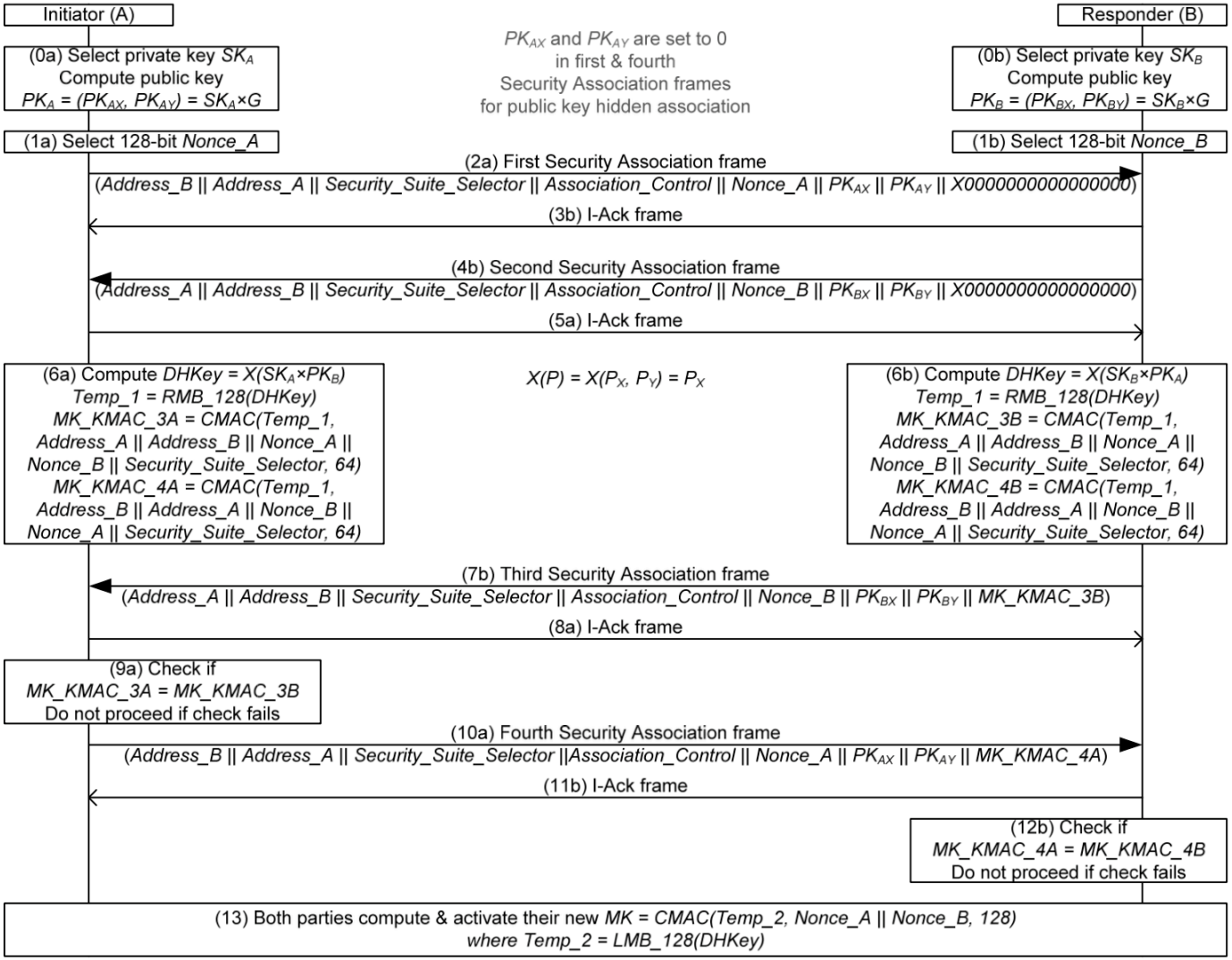
ECDH-based AKE in IEEE 802.15.6

- Master key pre-shared association



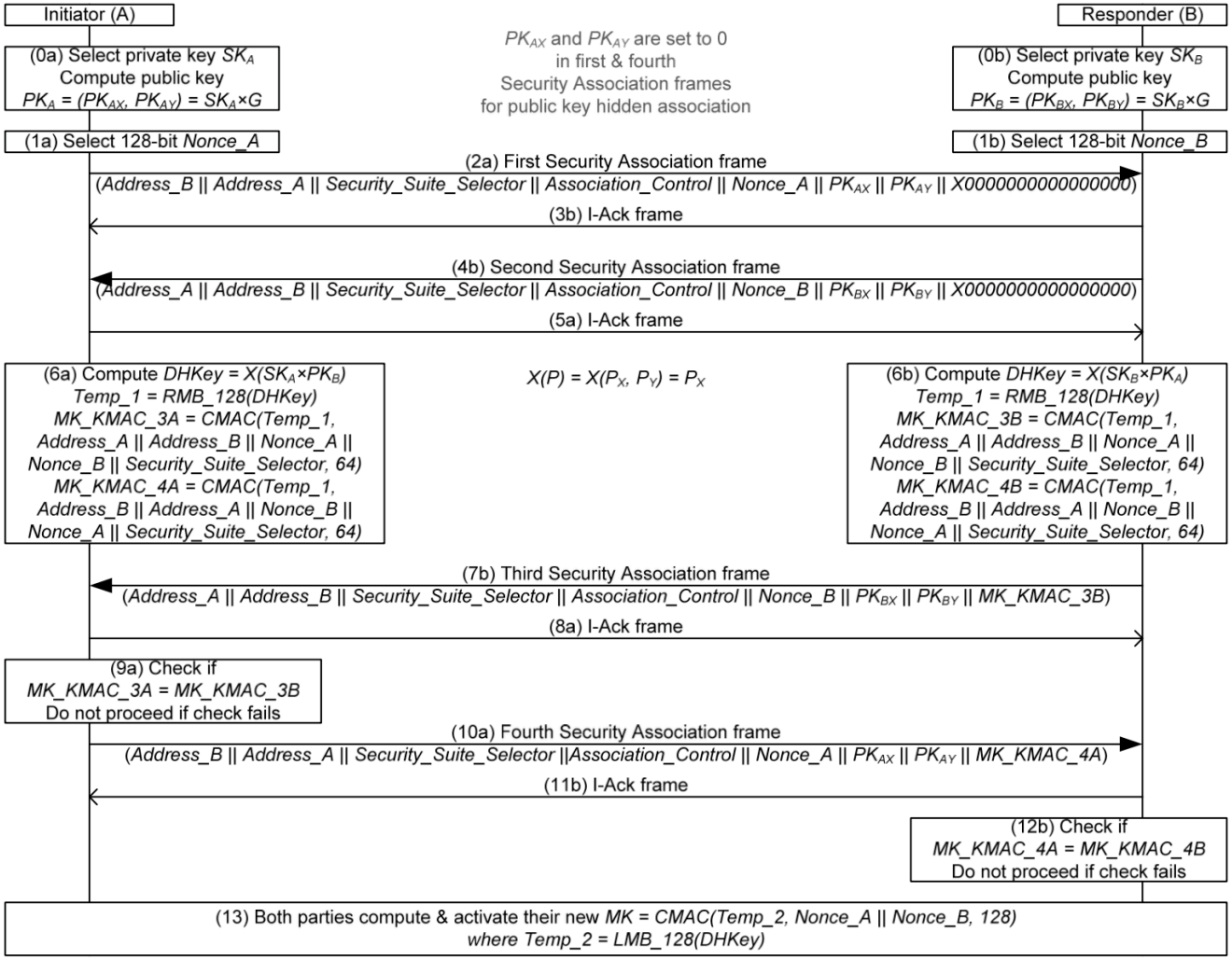
ECDH-based AKE in IEEE 802.15.6

- Unauthenticated association



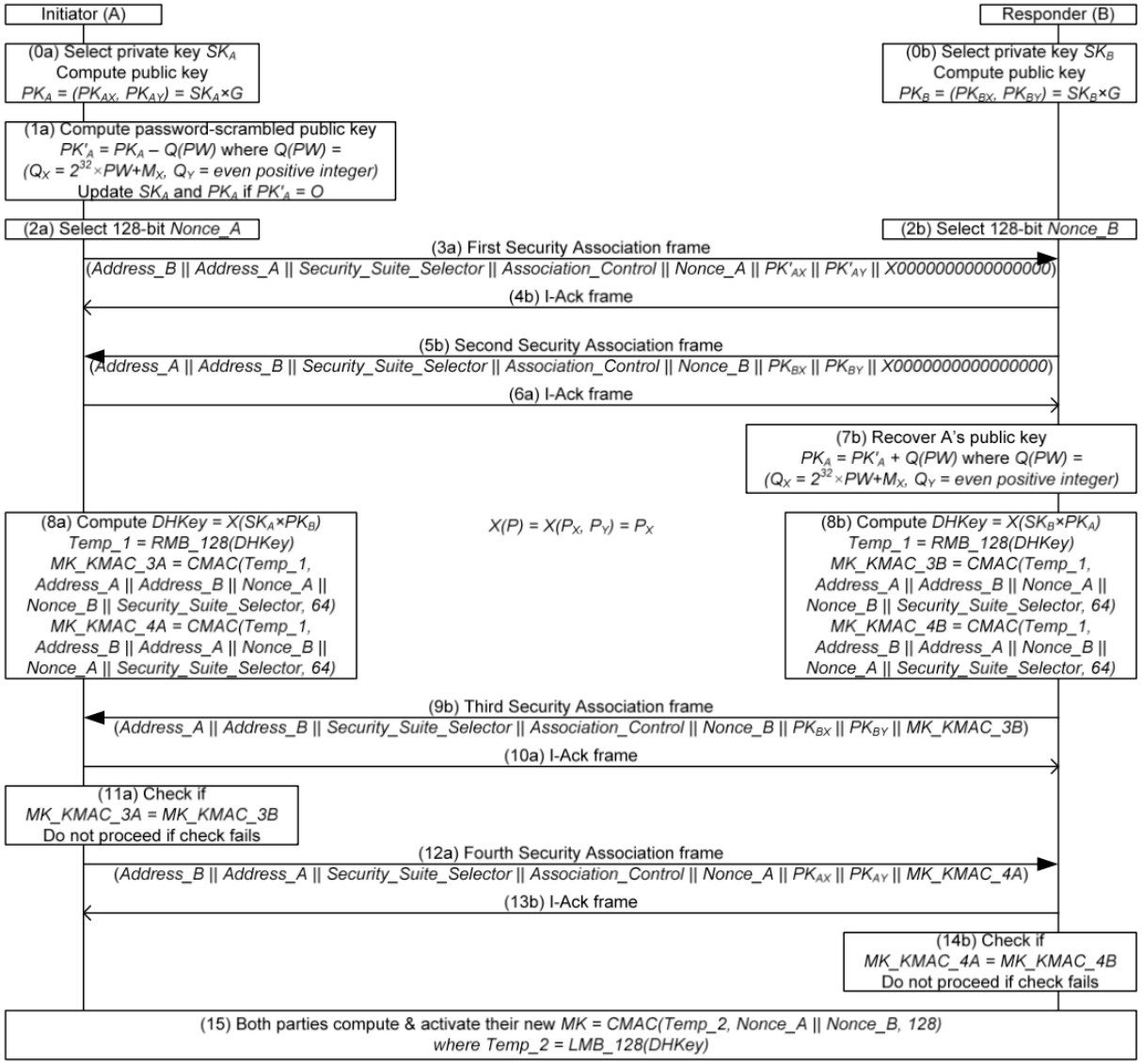
ECDH-based AKE in IEEE 802.15.6

- Public key hidden association
  - A's public key is secured transferred to the hub



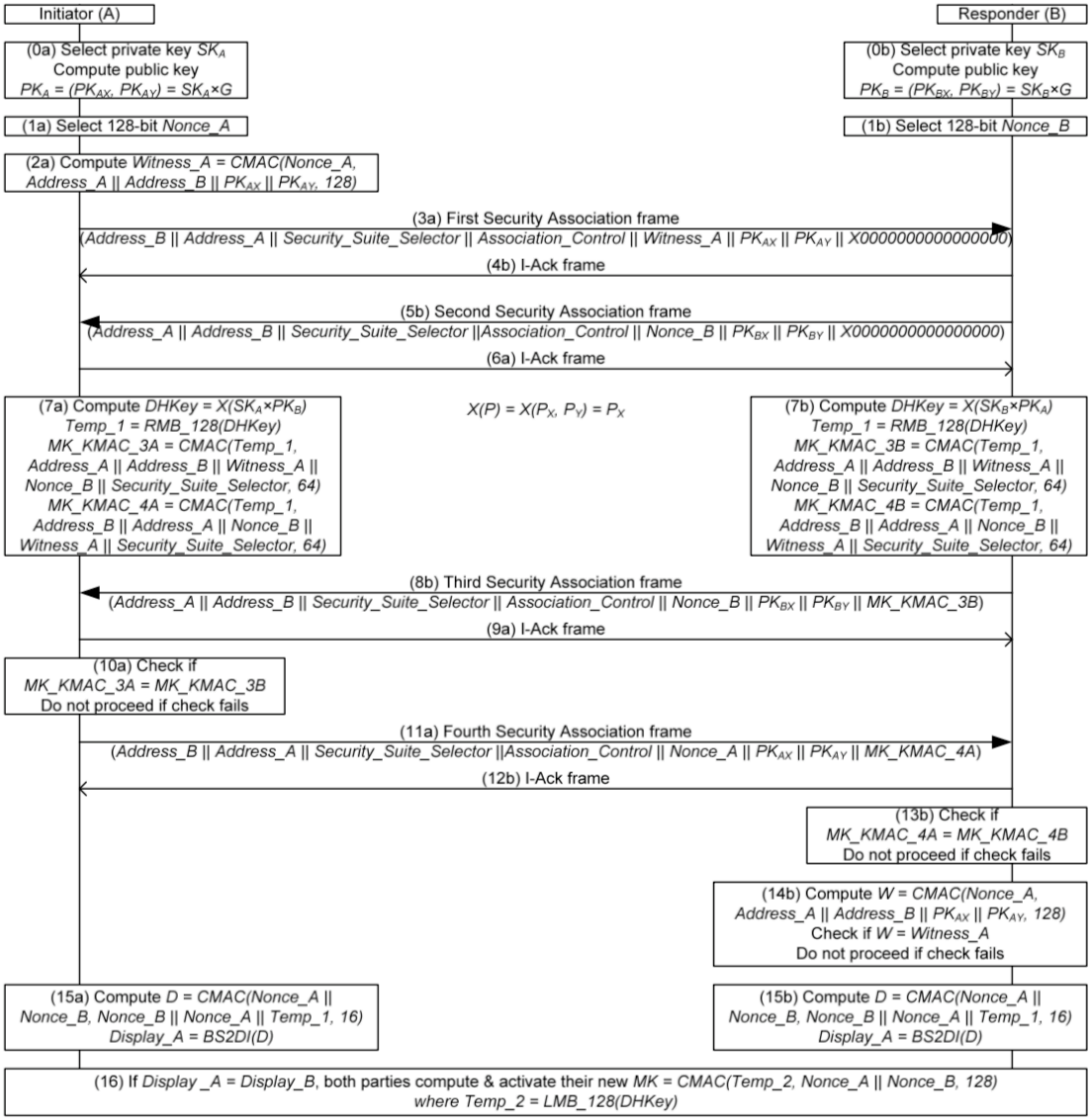
ECDH-based AKE in IEEE 802.15.6

- Password authenticated association



ECDH-based AKE in IEEE 802.15.6

- Display authenticated association

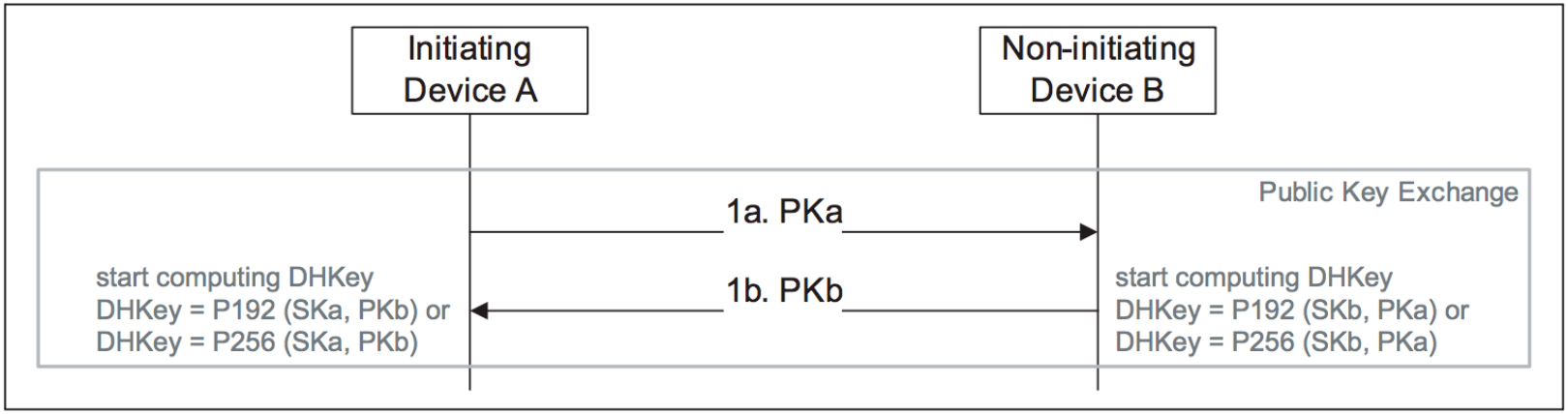


ECDH-based AKE in Bluetooth 5.0

- Bluetooth secure simple pairing
  - 1 Public key exchange
  - 2 Authentication stage 1
    - Numeric comparison
    - Out of band
    - Passkey entry
  - 3 Authentication stage 2
  - 4 Link key calculation

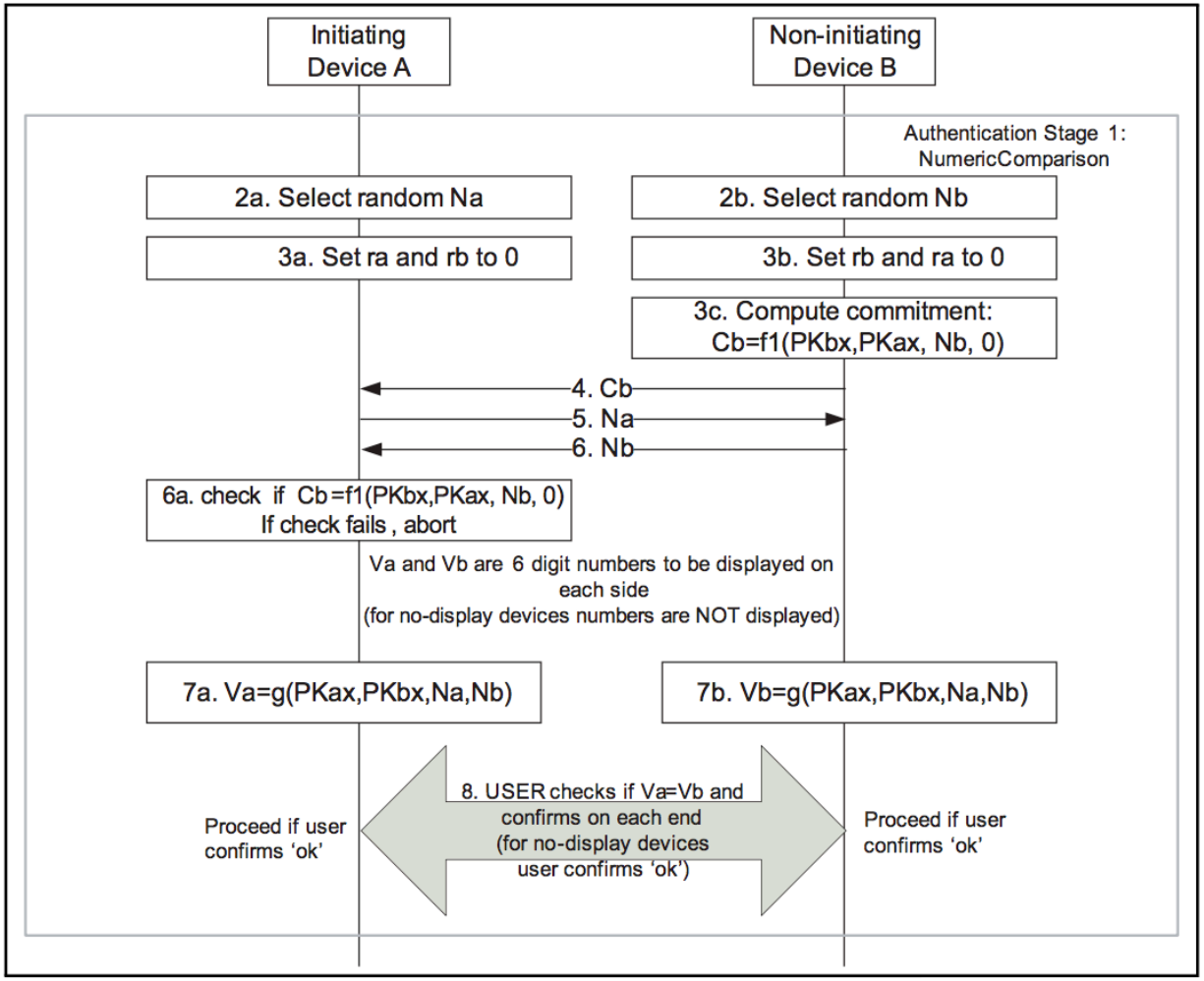
ECDH-based AKE in Bluetooth 5.0

- 1 Public key exchange



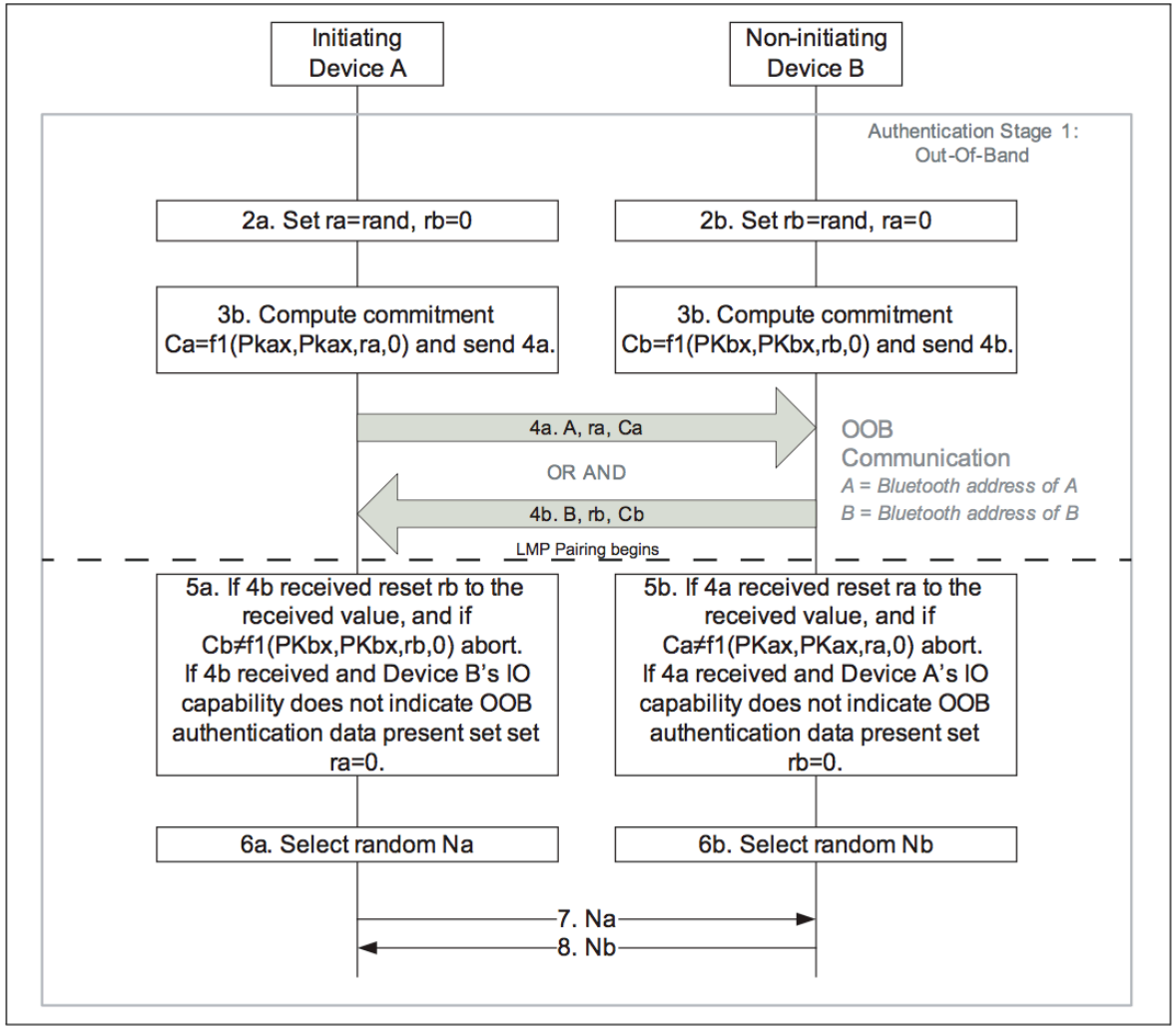
ECDH-based AKE in Bluetooth 5.0

- 2 Authentication stage 1 with numeric comparison



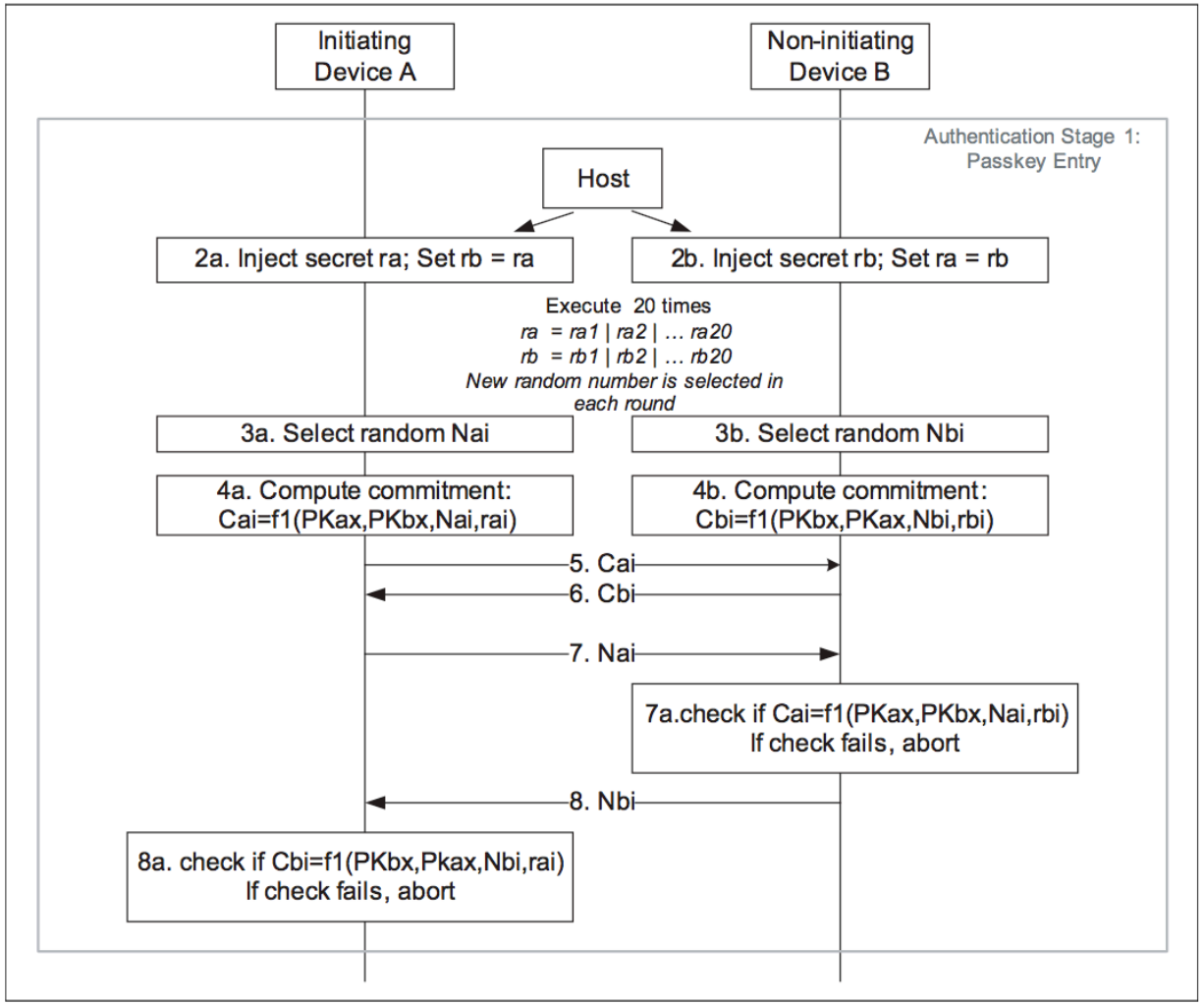
ECDH-based AKE in Bluetooth 5.0

- 2 Authentication stage 1 with out of band



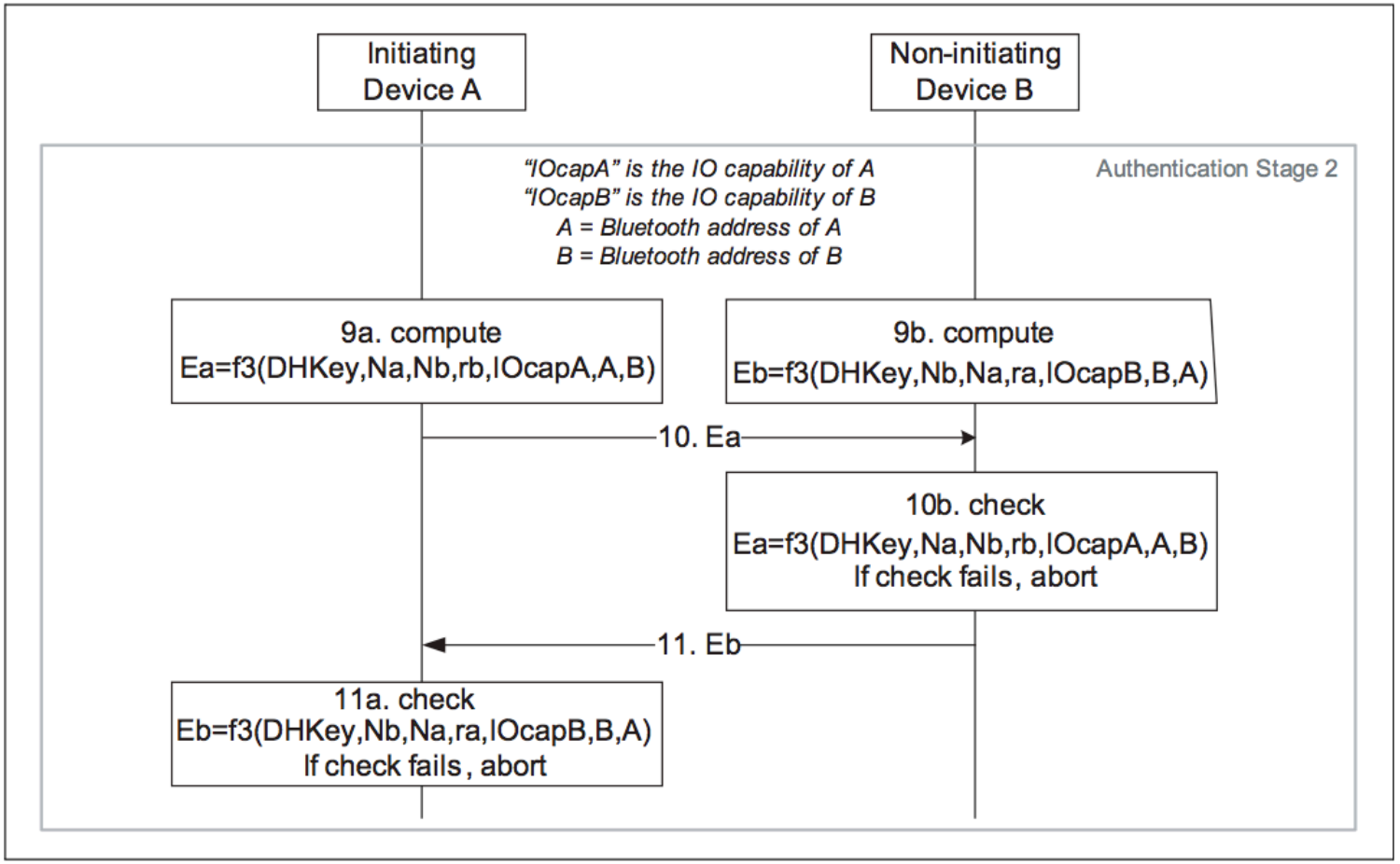
ECDH-based AKE in Bluetooth 5.0

- 2 Authentication stage 1 with passkey entry



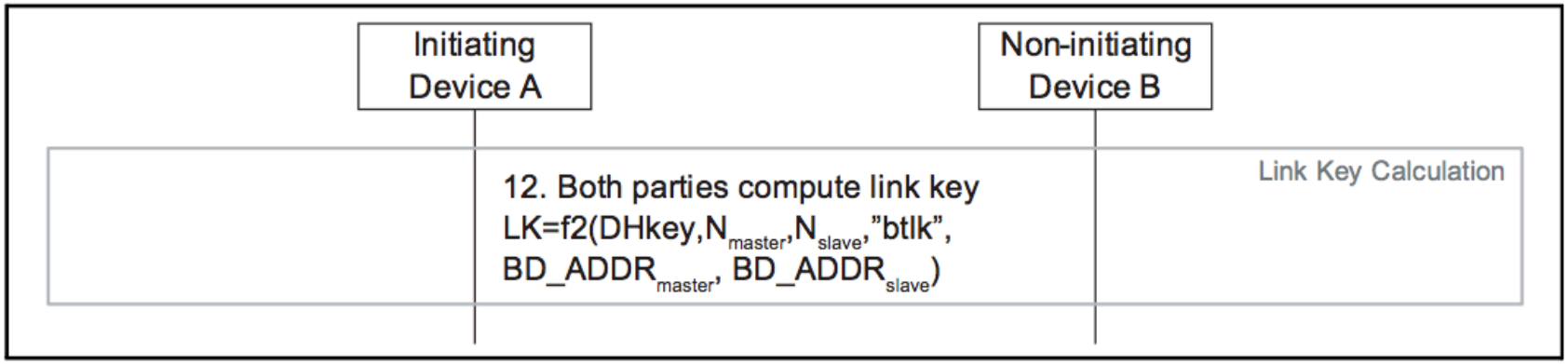
ECDH-based AKE in Bluetooth 5.0

- 3 Authentication stage 2



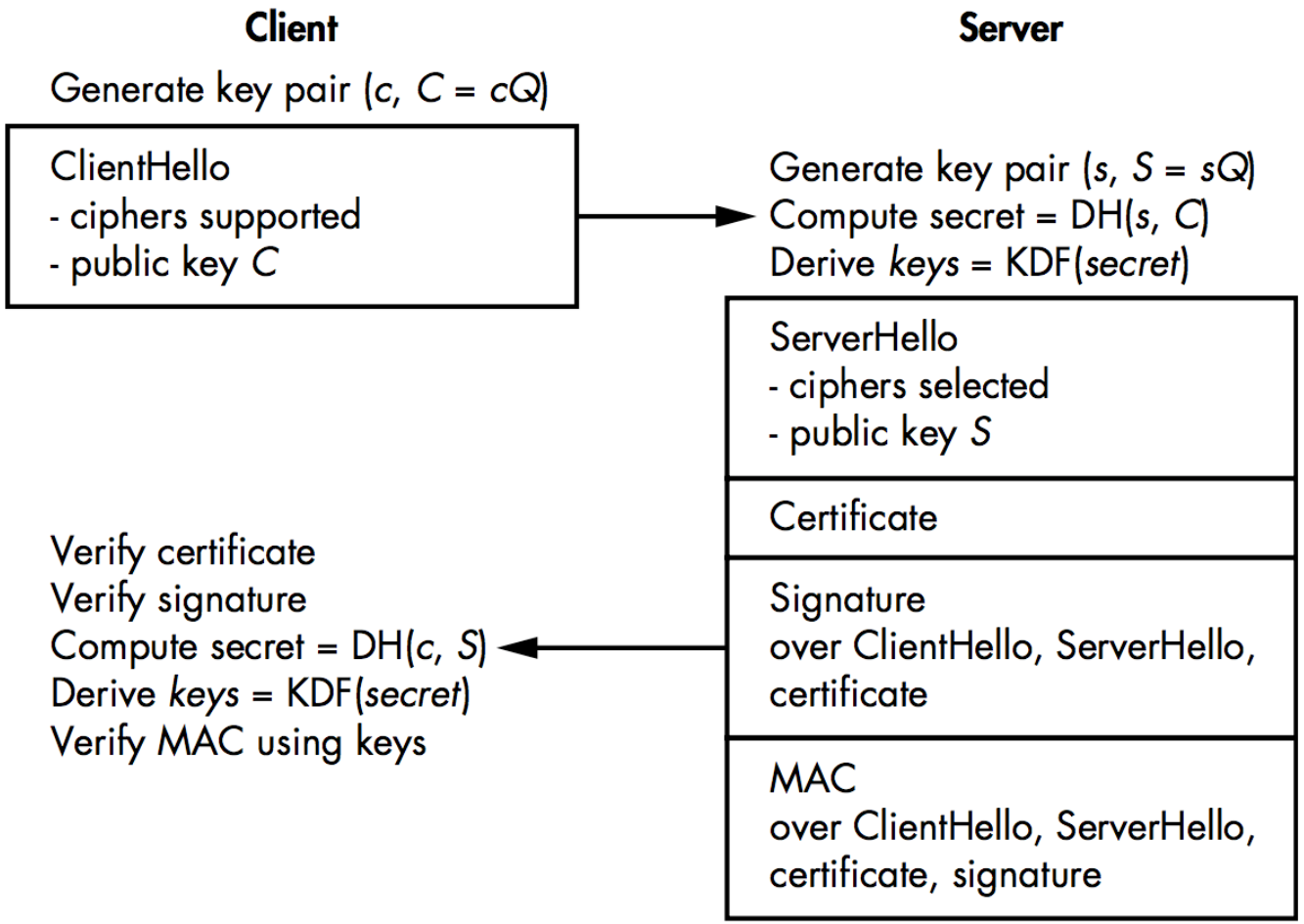
ECDH-based AKE in Bluetooth 5.0

- 4 Link key calculation



ECDH-based AKE in TLS 1.3

- TLS handshake with certificate



Experiments

- In the package “ieee802.15.6”

- wban1coordinator.py
- wban1sensor.py

Public key hidden association

- wban2coordinator.py
- wban2sensor.py

Password authenticated association

- wban3coordinator.py
- wban3sensor.py

Display authenticated association

Experiments

- In the package “bluetooth5.0”

- bluetooth1A.py
- bluetooth1B.py

out of band-based

- bluetooth2A.py
- bluetooth2B.py

numeric comparison-based

Experiments

- In the package “tls1.3”

- tlsclient.py
- tlsserver.py

To run the protocols

- Python2.7
- ecc package

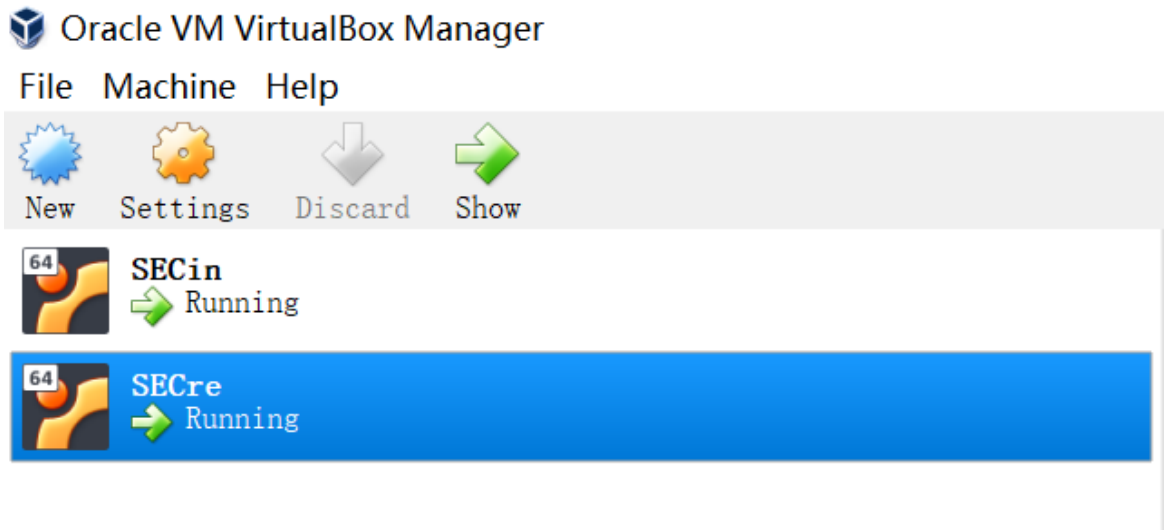
- Run coordinator.py and sensor.py respectively in
  - two virtual machines
  - two laptops
  - a laptop and a virtual machine

```
HOST = '10.8.106.95'
PORT = 6633
```

- Change the address into the ip of your coordinator machine
- Run coordinator.py before sensor.py

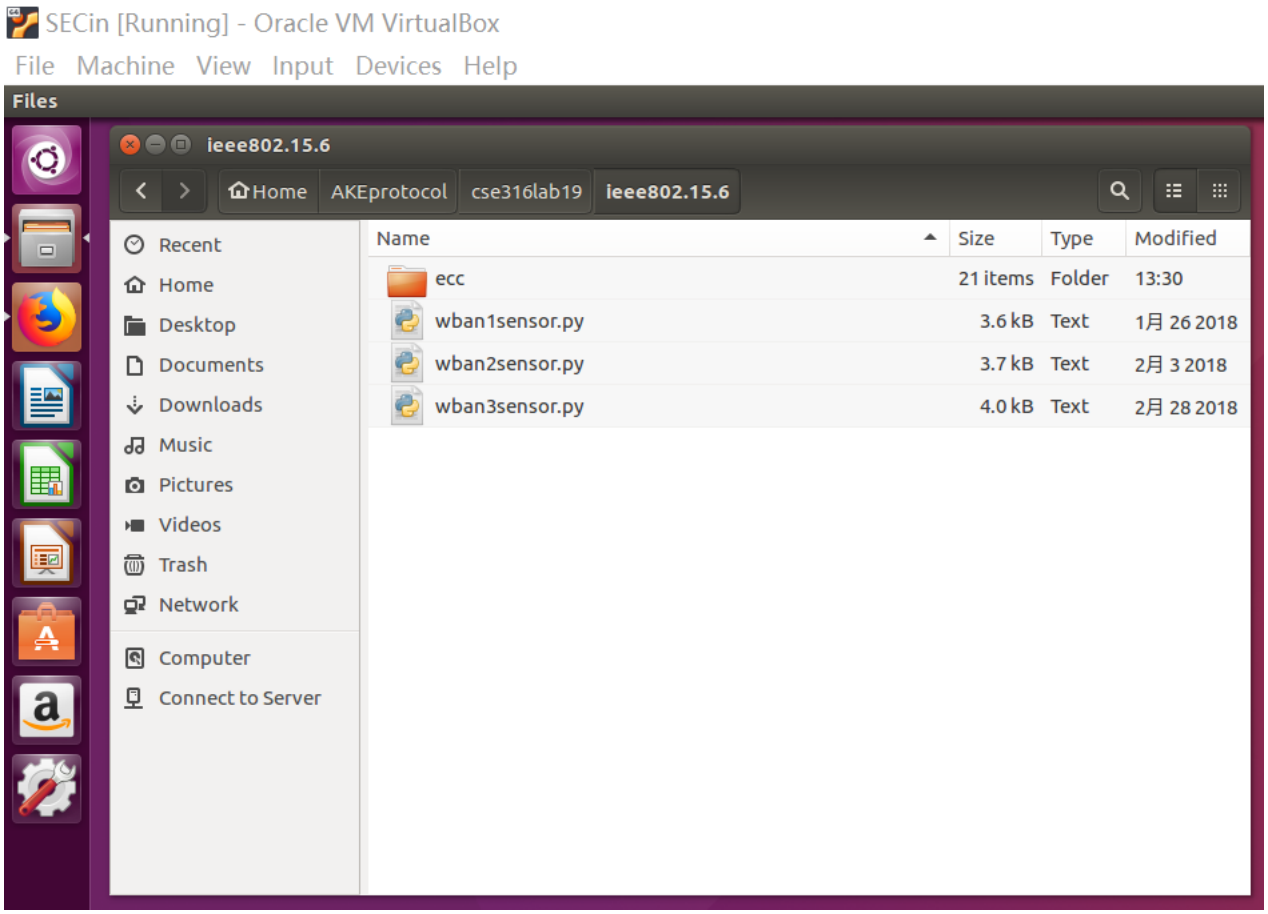
Example: WBAN1

- Tow virtual machines with the same configuration
  - SECin: initiator, client, sensor
  - SECre: responder, server, coordinator



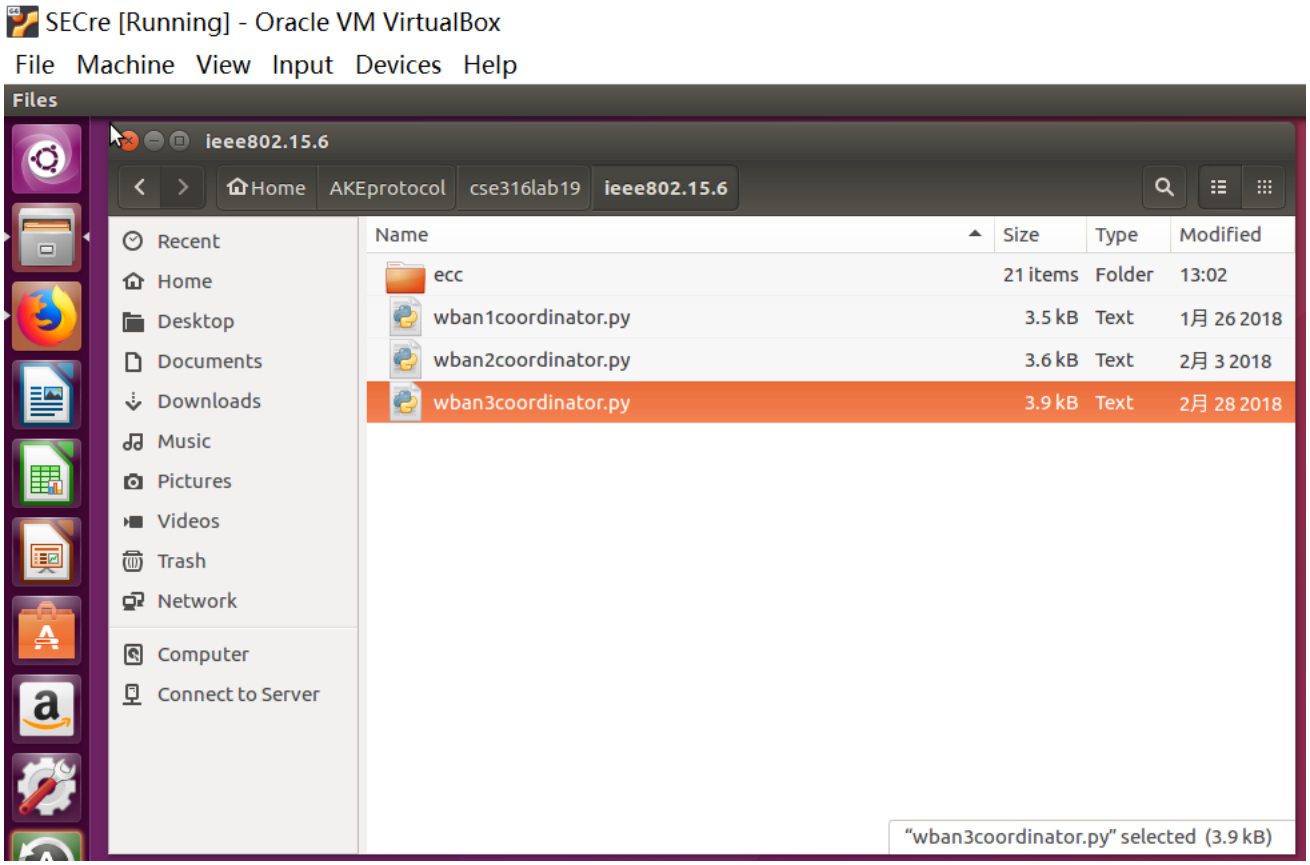
Example: WBAN1

- SECin: initiator, client, sensor



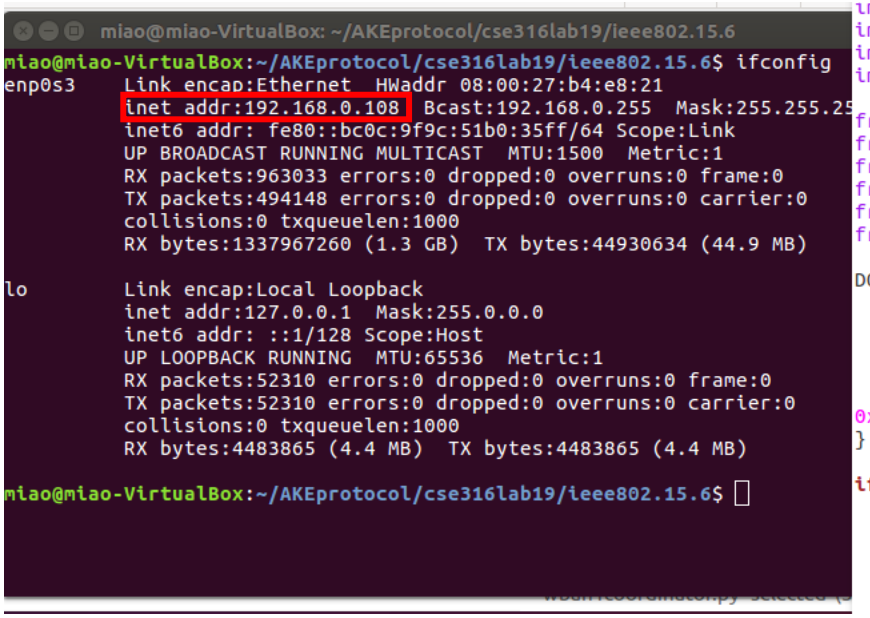
Example: WBAN1

- SECre: responder, server, coordinator



Example: WBAN1

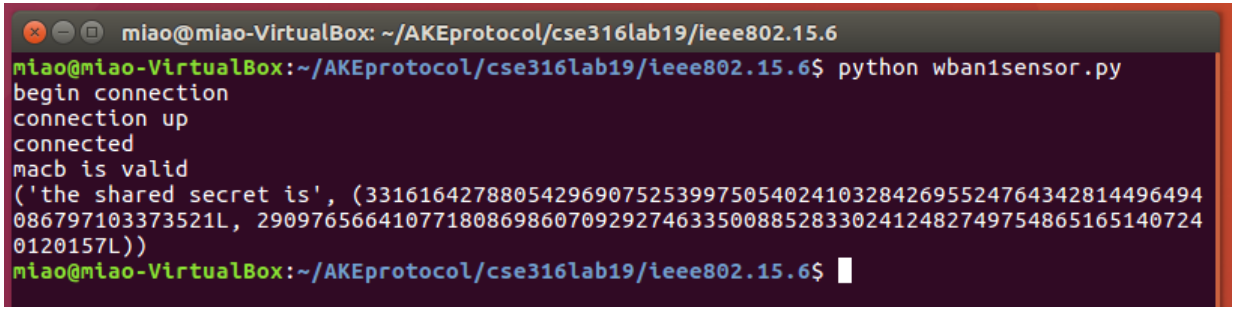
- IP address in SECre is 192.168.0.108
- Change the value in both wban1coordinator.py and wban1sensor.py into 192.168.0.108




```
import socket
import time
import random
import hmac

from collections import OrderedDict
from ecc.Key import Key
from hashlib import sha256
from ecc.elliptic import mul,add,neg
from ecc.curves import get_curve
from time import clock

DOMAINS = {
    # Bits : (p, order of E(GF(P)), parameter b, base point x, base point y)
    256: (0xffffffff00000001000000000000000000000000ffffffffffffffffffffffffL,
          0xffffffff00000000ffffffffffffffffbce6faada7179e84f3b9cac2fc632551L,
          0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604bL,
          0x6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296L,
0x4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5L)
}

if __name__ == '__main__':

    global Ta,Rb,p,n,b,x,y,c_p,c_q,c_n,M1,M2,M3,Kb

    HOST = '192.168.0.108'
    PORT = 6633
```

Example: WBAN1

- Run wban1coordinator.py in SECre first, then run wban1server.py in SECin.
- Your result should be like this



SECin

SECre