SHANGHAITECH UNIVERSITY

Blockchain Wi-Fi Authentication

Huang Zezhe **Gao Yining** **Wang Zhuoli**

CS222 Mobile & wireless system

Abstract

Our project WiBlock is the fusion of blockchain technique and Wi-Fi authentication. There are some existed weakness in WPA(2)-Enterprise or WPA(2)-PSK: mandatory key reset, extracting key from multiple handshake packages or even brute forcing. Besides, we know the advantages of blockchain: irreversibility, decentralization as well as some cryptographer and consensus algorithms, which may construct a new system with special features, i.e., WiBlock.

Introduction of the Main Idea

We construct a blockchain using in the system with mix coins: AuthCoin is used for transactions between STA and AP in a connection. RewardCoin is used as a authoritative reward when blockchain nodes mine successfully to create a new block.

For STA, we don't need a different account password any more when we connect different AP terminals. We don't need to worry about the login secret keys stolen. For AP, we don't need to make our own account password, store user information in our own database, and it will be easy to maintain.

Introduction to the Protocal

Maintain block chain and current transaction

$$\mathrm{Tx} = \begin{cases} \text{Coin\_type} \begin{cases} \text{Auth Coin} \\ \text{Reward Coin} \end{cases} \\ \text{Trans\_type} \begin{cases} \text{Issue} \\ \text{Connect} \\ \text{Disconnect} \end{cases} \\ \text{Sender} \\ \text{Recipinent} \\ \text{Amount} \end{cases}$$

Protocal #1

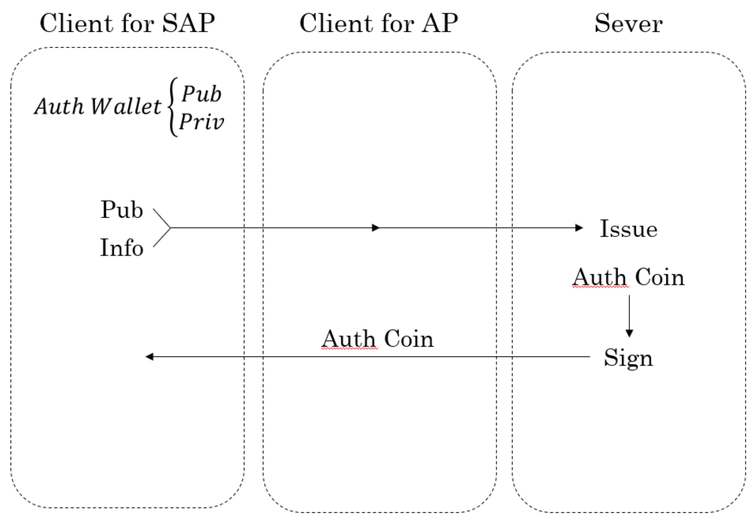

Figure 1: Protocal #1

Protocal #2

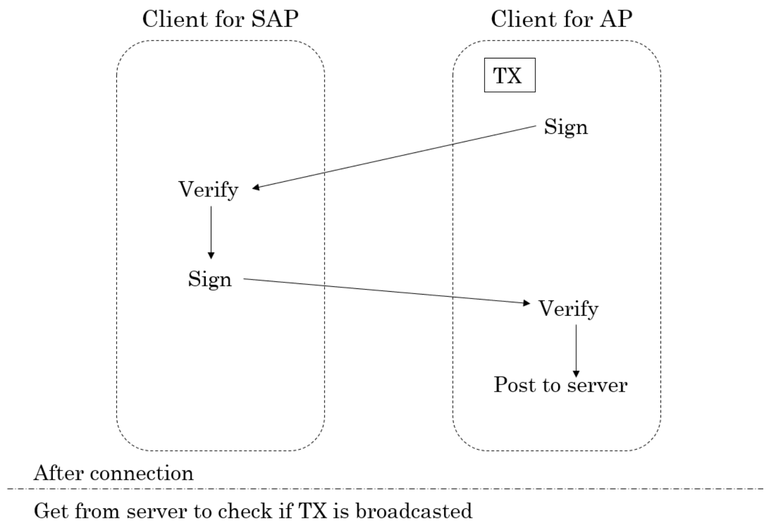

Figure 2: Protocal #2

Details of Clients

Client for AP

1: **def** register
2: Register when AP client is used first time
3: **def** handle_data
4: Recognize type of data received from socket
5: 1. connection TX
6: 2. disconnection TX
7: 3. STA registration
8: **def** verify_signature
9: Verify authority of Auth Coin
10: **def** generate_transaction
11: Generate raw transaction data for sending to STA

Client for STA

1: **def** register
2: Register when STA client is used first time
3: **def** validate_tx
4: Verify if TX is broadcasted
5: **def** handle_recv
6: Recognize type of data received from socket and handle them
7: 1. connection TX
8: 2. AuthCoin
9: **def** exit_handler
10: Disconnect and handle disconnection TX

APIs

POST /register
form data:
public_key
info
response:
AuthCoin

POST /register4AP
form data:
public_key
responseL
registration status

GET /chain
response:
blockchain

POST /transactions/generate
form data:
raw: TX data
signature: sender's signature

GET /coin_owner
request data:
asset_id
response data:
public_key

GET /transactions/current
response data:
TXs not be stored in chain

GET /mine
response data:
index of new block created

POST /nodes/register
form data:
nodes' IDs
response data:
registration status

GET /nodes/resolve
response data:
result of resolving (Replaced / Authoritative)

Simulation — Stolen AuthCoin

We program a little impersonation attack to try to connect a AP with the stolen authcoin.

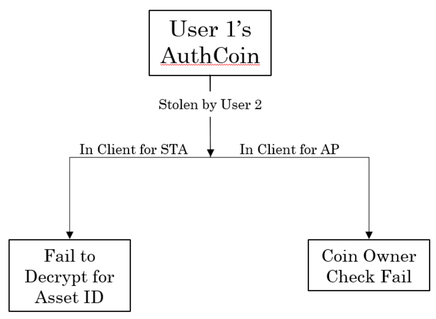

Figure 3: Stolen AuthCoin

The client for AP will check if the owner of authcoin received is the target user: request GET coin owner API served by blockchain node.

The STA client will encounter failure while it tries to decrypt authcoin binary code to extract asset ID with a wrong private key.

Features

- AuthWallet for generating asymmetric keys and maintaining identity.
- Verify a transaction with signature.
- Other APIs for blockchain and transaction.

References

[1] Source Code https://github.com/kiki0805/WiBlock/

[2] http://ieeexplore.ieee.org/document/7800479/

[3] Sanda, Tomoyuki, and H. Inaba. "Proposal of new authentication method in Wi-Fi access using Bitcoin 2.0." Consumer Electronics, 2016 IEEE, Global Conference on IEEE, 2016:1-5.

[4] Third Part Tools https://github.com/oblique/create_ap