A Security Analysis of WPA3-PK:

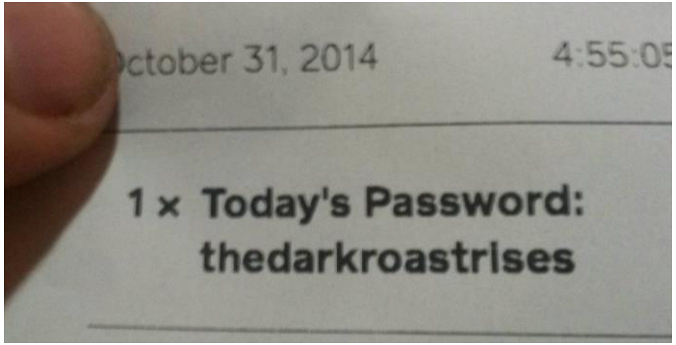
Implementation & Precomputation Attacks

Mathy Vanhoef and Jeroen Robben

Applied Cryptography and Network Security (ACNS), 5-8 March 2024

KU LEUVEN DistriNet

Protected hotspots: publicly sharing the password



- Can passively decrypt traffic
- No forward secrecy: decrypt old traffic

} Solved by WPA3

- Create rogue clone using shared password

WPA3 Public Key

Goal of WPA3 Public Key, also called SAE-PK:

- Authenticate a Wi-Fi hotspot using a password…
- …but prevent an adversary from cloning the network

→ Accomplished by using asymmetric crypto

High-level overview of SAE-PK

1. Access Point (AP) generates public/private key
2. Wi-Fi password is derived from public key
3. Public key is sent to the client when connecting
4. Client uses password to verify this public key

→ The **password forms a signature** of the public key

The SAE-PK password

Password is the truncated output of:

Hash(SSID || Modifier M || public key)

- SSID: name of the Wi-Fi network
- Modifier M: chosen so output has 3 or 5 leading zero bytes
  - Number of leading zero bytes is a security parameter

The SAE-PK password

Password is the truncated output of:

Hash(SSID || Modifier M || public key)

Output is converted into a human-readable form

- Example password: `2udb-slxf-3ijn-dbu3-...`
- Password length is decided by administrator…
- …must encode at least 52 bits, excluding leading zero bytes

Attack: creating a clone of the network?

Find a modifier M & public key that result in the same password

- What is the complexity of this in the best case?

Hash(SSID || Modifier M || public key)

- Hash output must start with at least 3 zero bytes → 2^{24}
- Remaining output must equal the password → 2^{52}

Total time complexity of 2^{76} to perform a naïve attack

Observation & better attack

The **same SSID is often attacked multiple times**

- Common names such as `xfinitewifi` or `linksys`
- Attacking same network after they update keys

Time-memory trade-off attacks:

- Naïve: table to maps SAE-PK passwords to a private key
- Can construct **rainbow tables** to optimize the attack
  - Estimate: ~6TB table inverts password in ~2 weeks on AWS
  - Defense: longer password or using 5 leading zero bytes

Downside of computed table

Table converts password into hash input:

Hash(SSID || Modifier M || public key)

Downside of computed table

Table converts password into hash input:

ssid | modifier | public key

Hash(69 64 5F 32 || 00 … B5 30 8A || 30 88 00 … 18)

Downside of computed table

Table converts password into hash input:

ssid | modifier | public key

Hash(69 64 5F 32 || 00 … B5 30 8A || 30 88 00 … 18)

= Hash(69 64 || 5F 32 00 … B5 || 30 8A 30 88 00 … 18)

ssid' | modifier' | public key'

→ Same table output now targets a different **ssid'**!

Suggested defense

Start with single input byte encoding **length of SSID**:

ssid | modifier | public key

Hash(**04** 69 64 5F 32 || 00 … B5 30 8A || 30 88 00 … 18)

→ Hash input now has a single interpretation

Intercepting traffic at network layer

1. Can get MitM using ARP poisoning
2. Can abuse symmetric group key to spoof broadcast traffic

Can even put unicast IP packet in a broadcast Wi-Fi frame:

- Vulnerable: Windows 10, Huawei Y6', iPad, Android 5X, Linux
- Not vulnerable: Android Pixel 4XL

Intercepting traffic at network layer

1. Can get MitM using ARP poisoning
2. Can abuse symmetric group key to spoof broadcast traffic

Defenses:

1. Block client-to-client traffic
2. Disable broadcast traffic (see Passpoint standard)

Implementation Attacks

In a private home network, password must remain secret:

Password = Hash(SSID || **Modifier M** || public key)

SSID is known & public key sent in plaintext when connecting

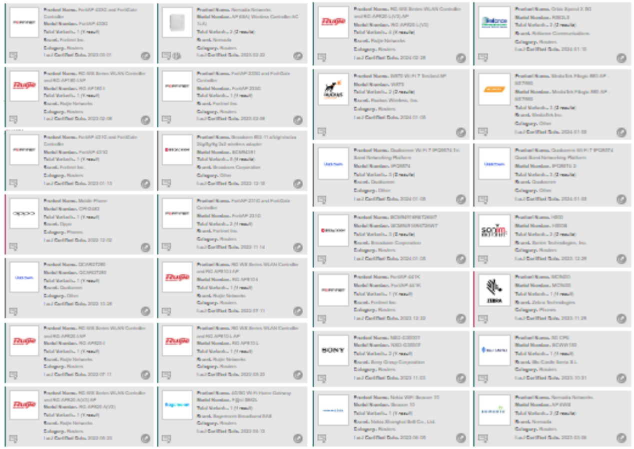
- **Modifier must be unpredictable** to keep password private
- In 2 of 3 tested implementations, modifier was predictable
  - Fortunately, those two implementations weren't used widely

Conclusion

Overall, SAE-PK looks decent

Prevent network attacks by both:

1. Disabling client-to-client traffic
2. Disabling broadcast traffic



Growing number of devices support SAE-PK!

Prevent rogue networks using either:

- 16+ long passwords
- Using passwords with “5 leading zero bytes”