Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2

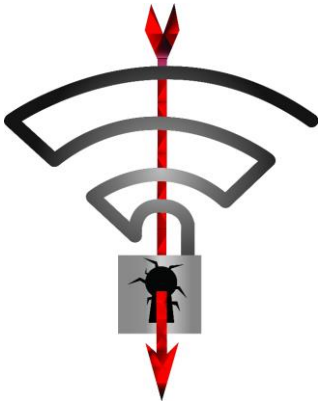
Mathy Vanhoef, PhD

Wi-Fi Alliance meeting Bucharest, 24 October 2017

KU LEUVEN DistriNet

Overview

1. **Key reinstallation in 4-way handshake**



2. Misconceptions and remarks

3. Steps to improve Wi-Fi security?

The 4-way handshake

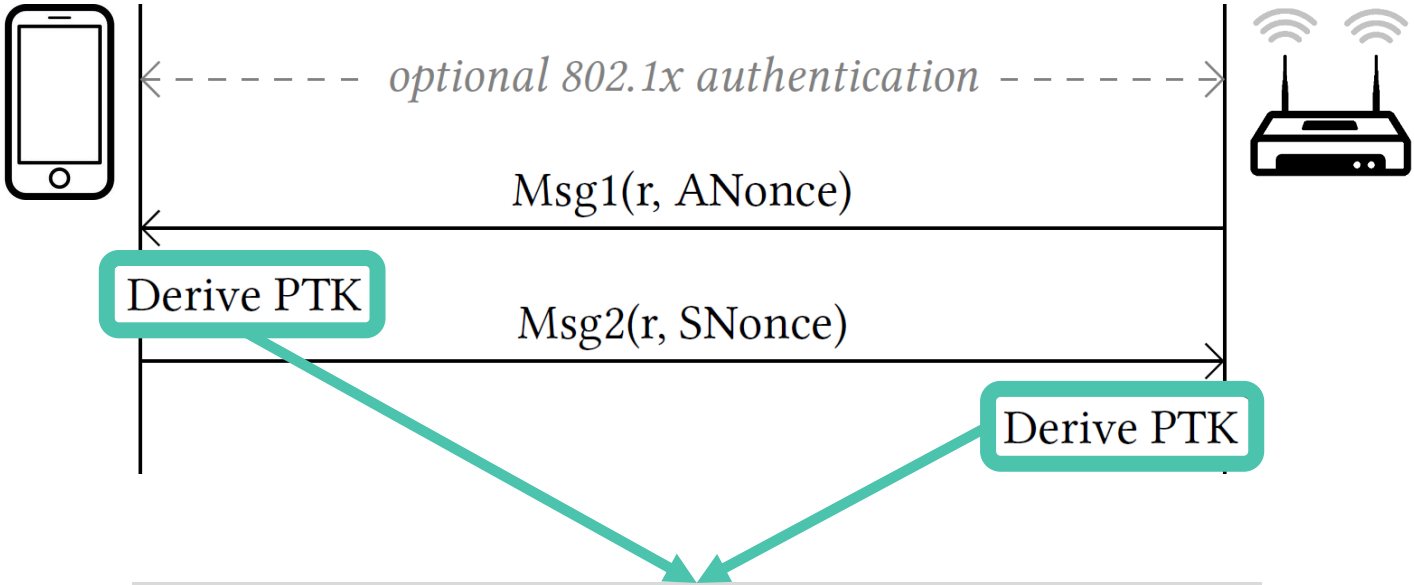
Two main purposes:

- Mutual authentication
- Negotiate fresh PTK: pairwise temporal key

Appeared to be secure:

- No attacks in more than a decade
- Proven as secure in 2005[1]
- That is: negotiated key (PTK) is secret

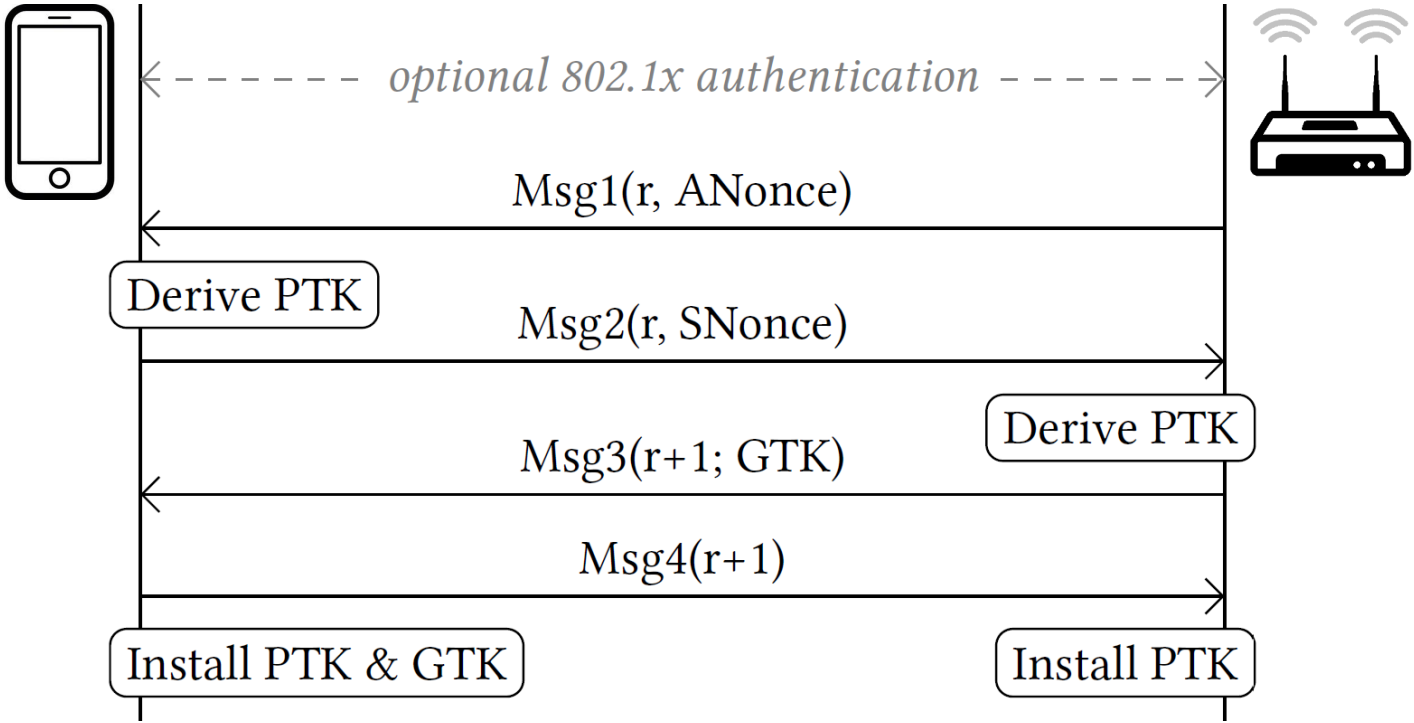
Wi-Fi handshake (simplified)



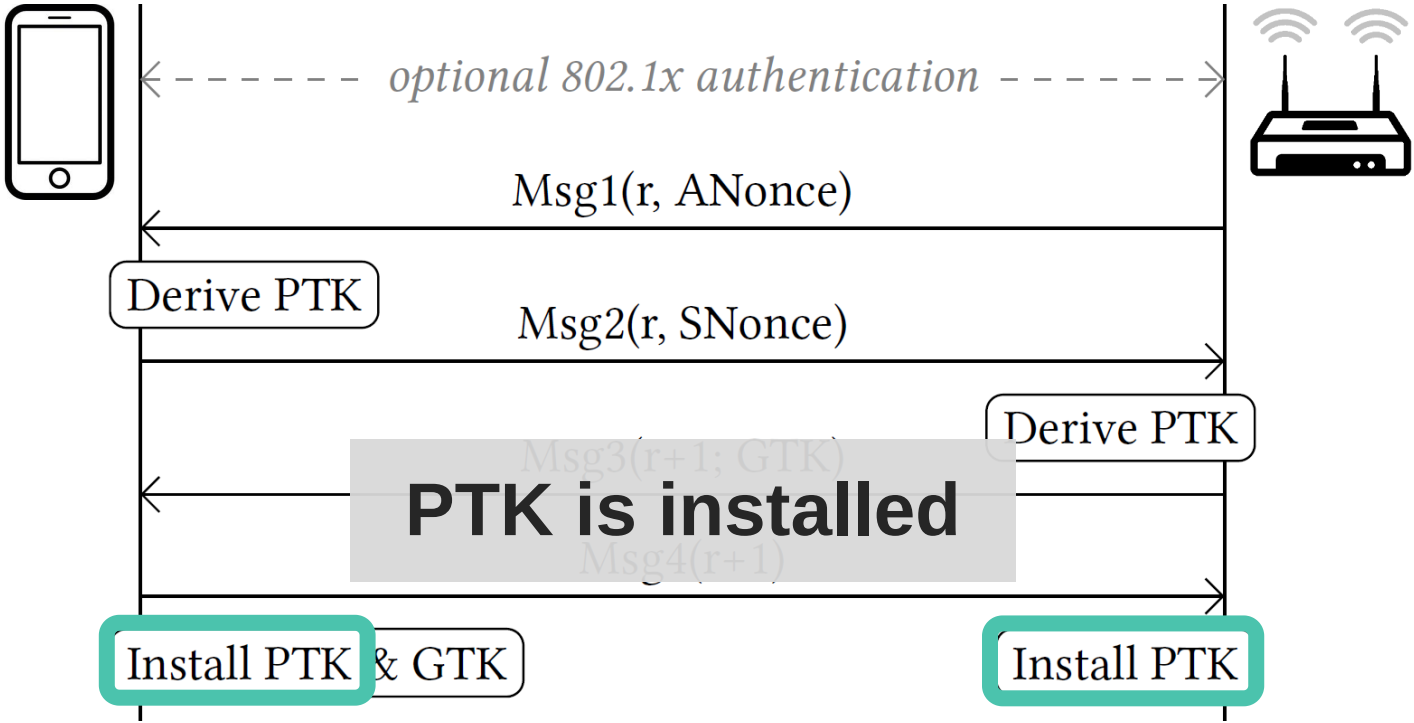
PTK = Combine(shared secret, ANonce, SNonce)

Wi-Fi handshake (simplified)

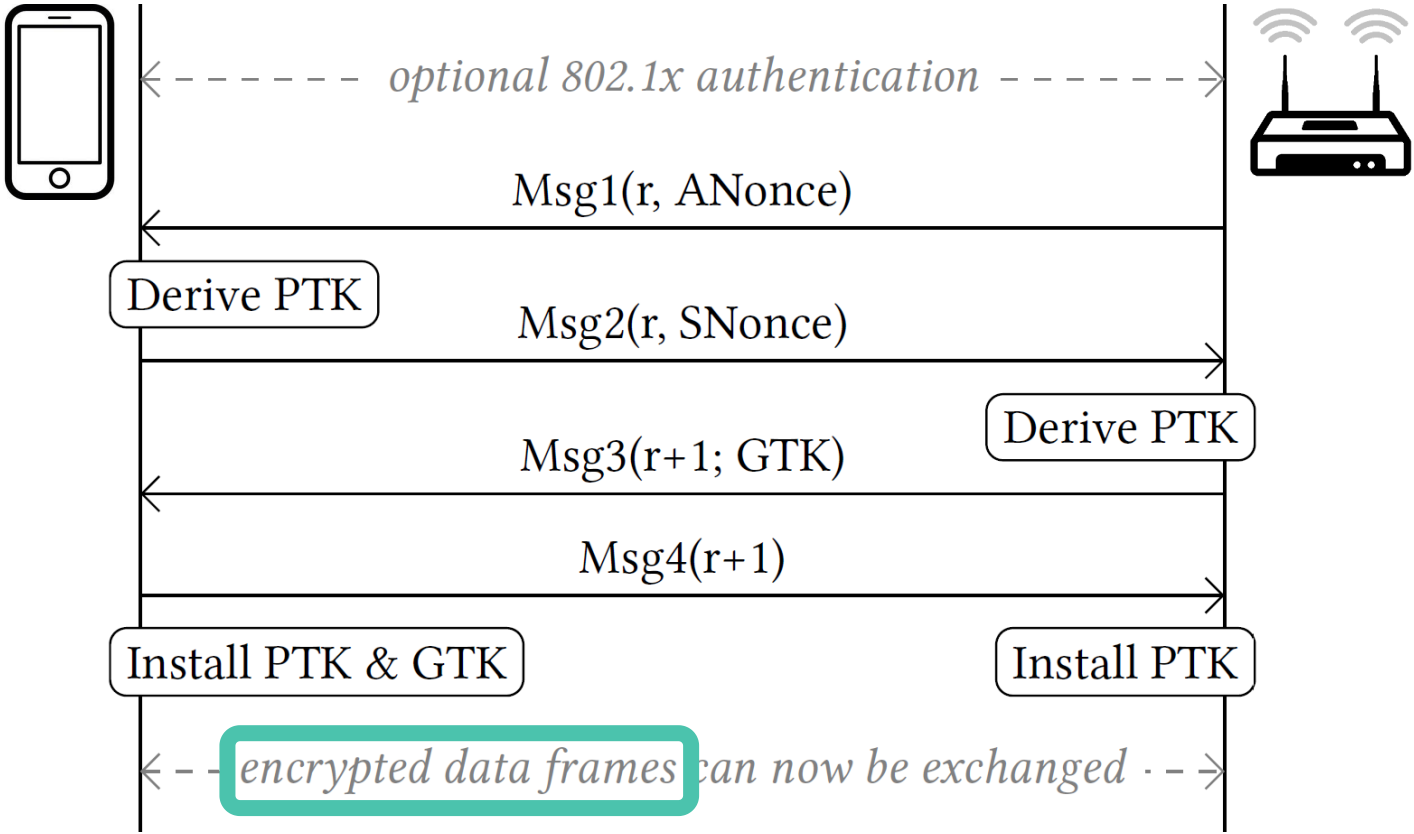
Wi-Fi handshake (simplified)



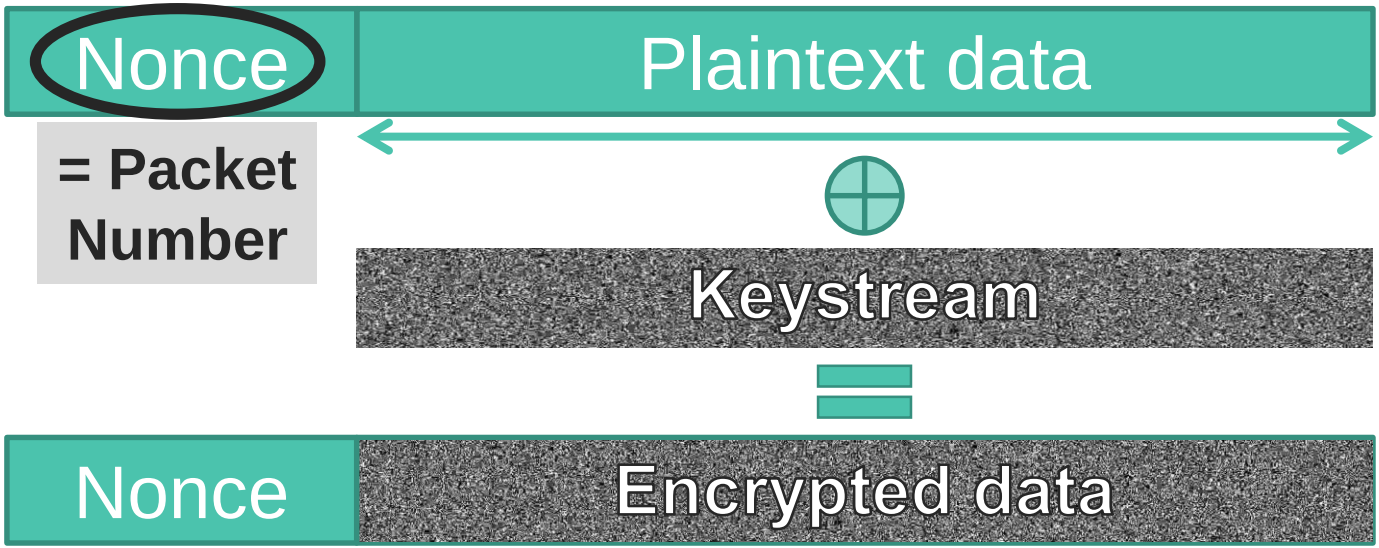
Wi-Fi handshake (simplified)



Wi-Fi handshake (simplified)



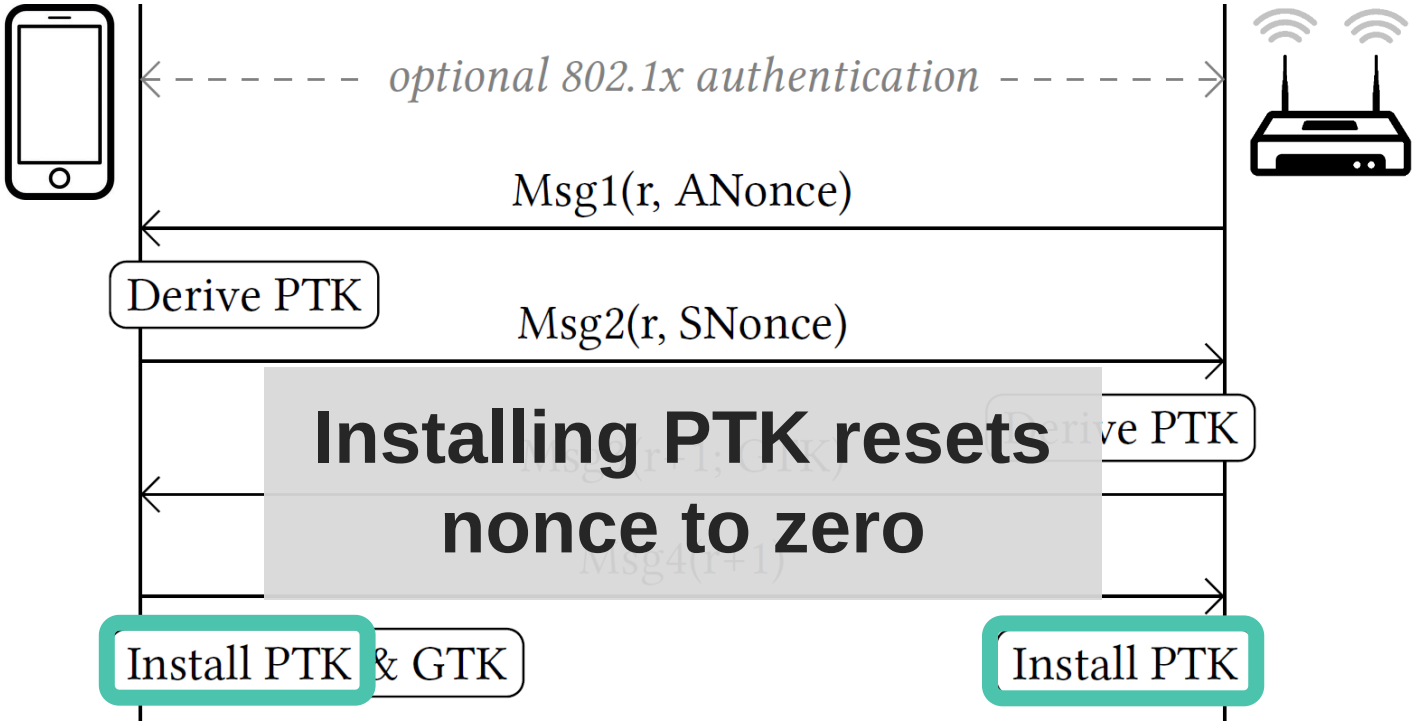
Encrypting data frames (simplified)



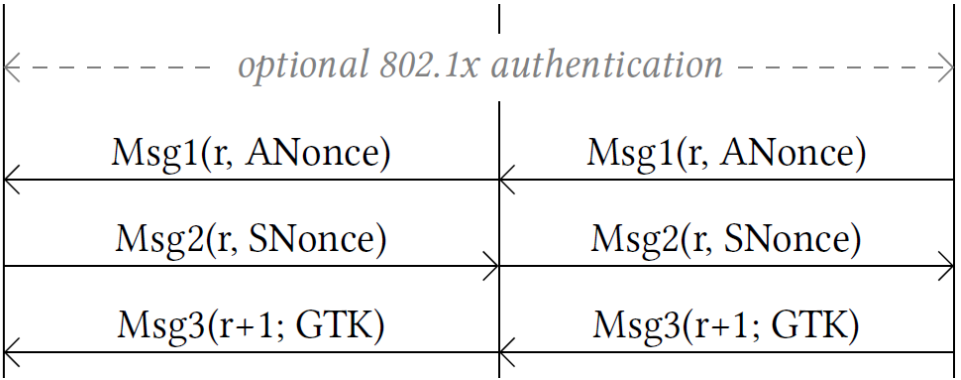
Keystream should never be reused

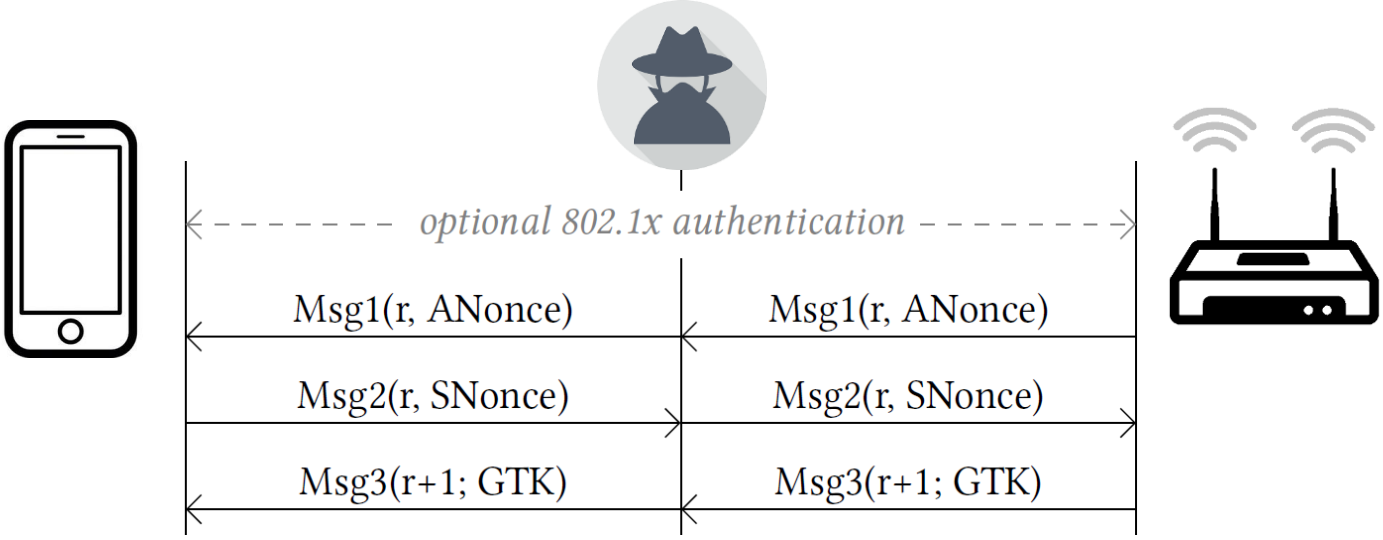
- Each nonce results in a unique keystream

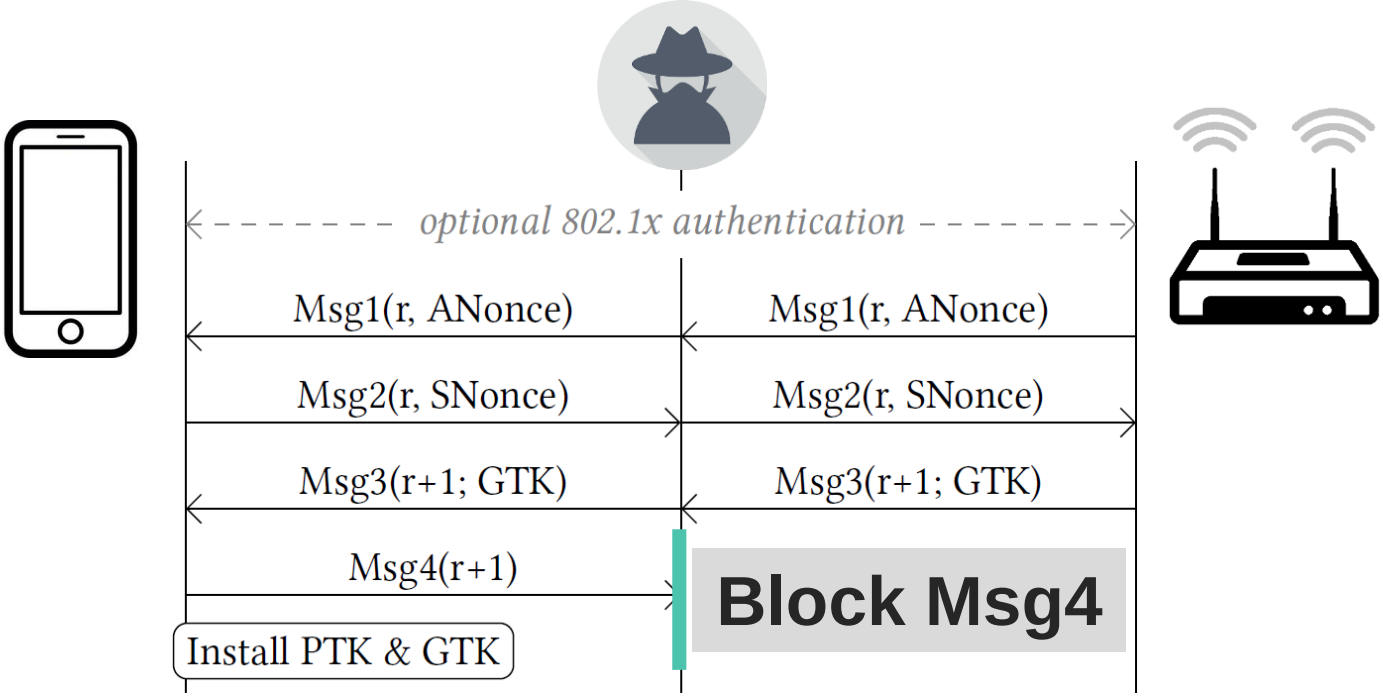
Wi-Fi handshake (simplified)

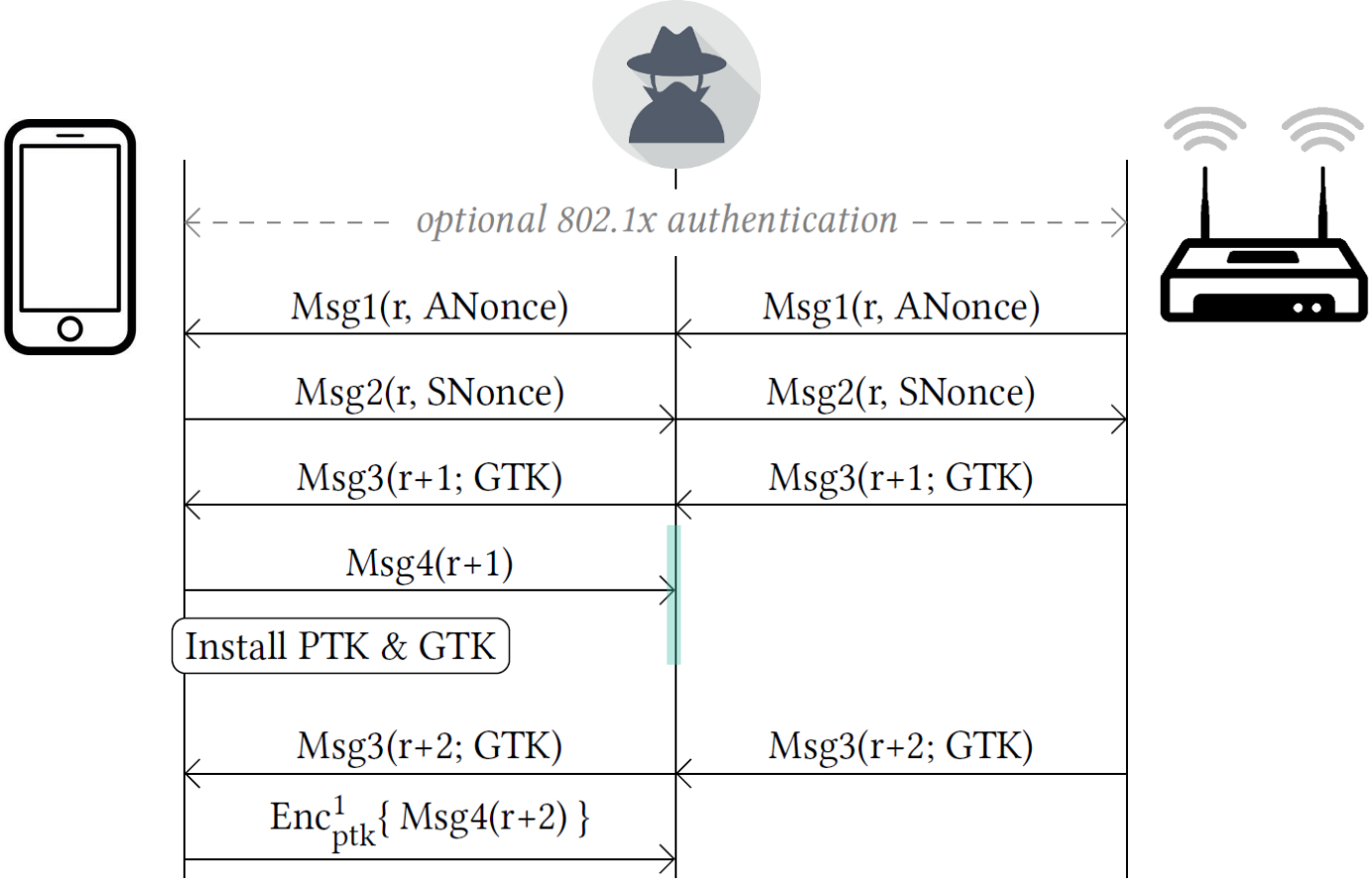


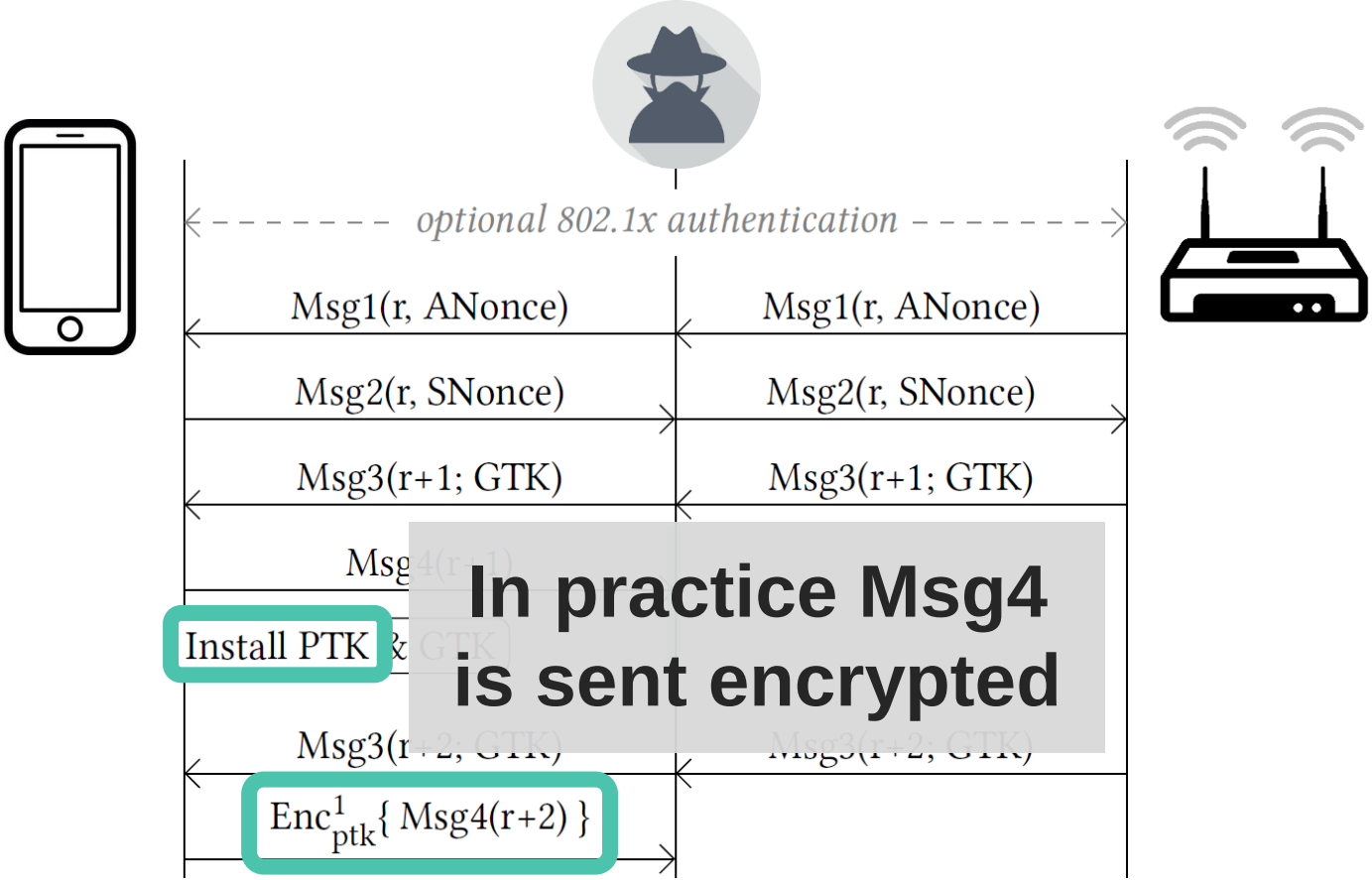
Key Reinstallation Attack



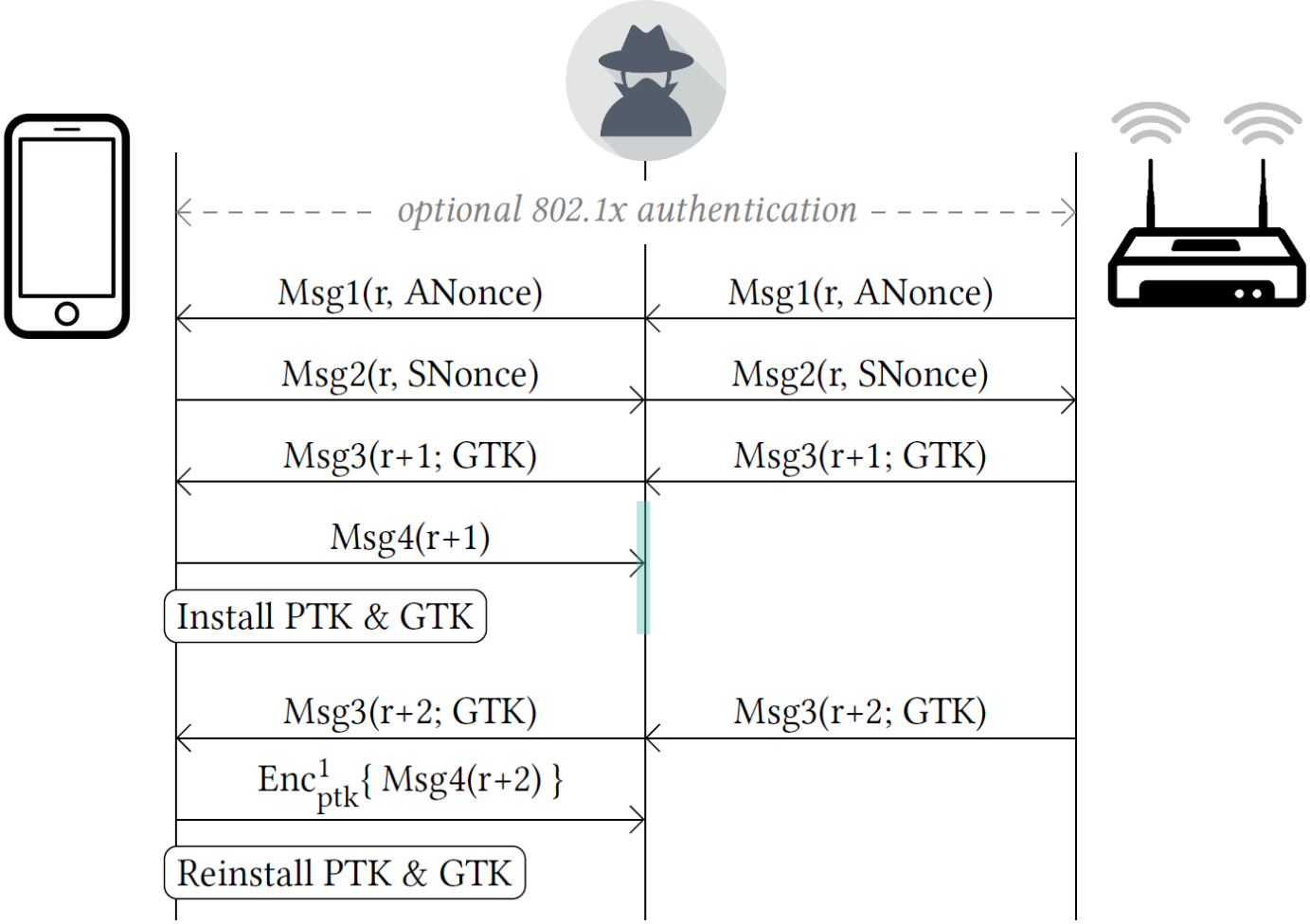

optional 802.1x authentication
Msg1(r, ANonce)
Msg1(r, ANonce)
Msg2(r, SNonce)
Msg2(r, SNonce)
Msg3(r+1; GTK)
Msg3(r+1; GTK)

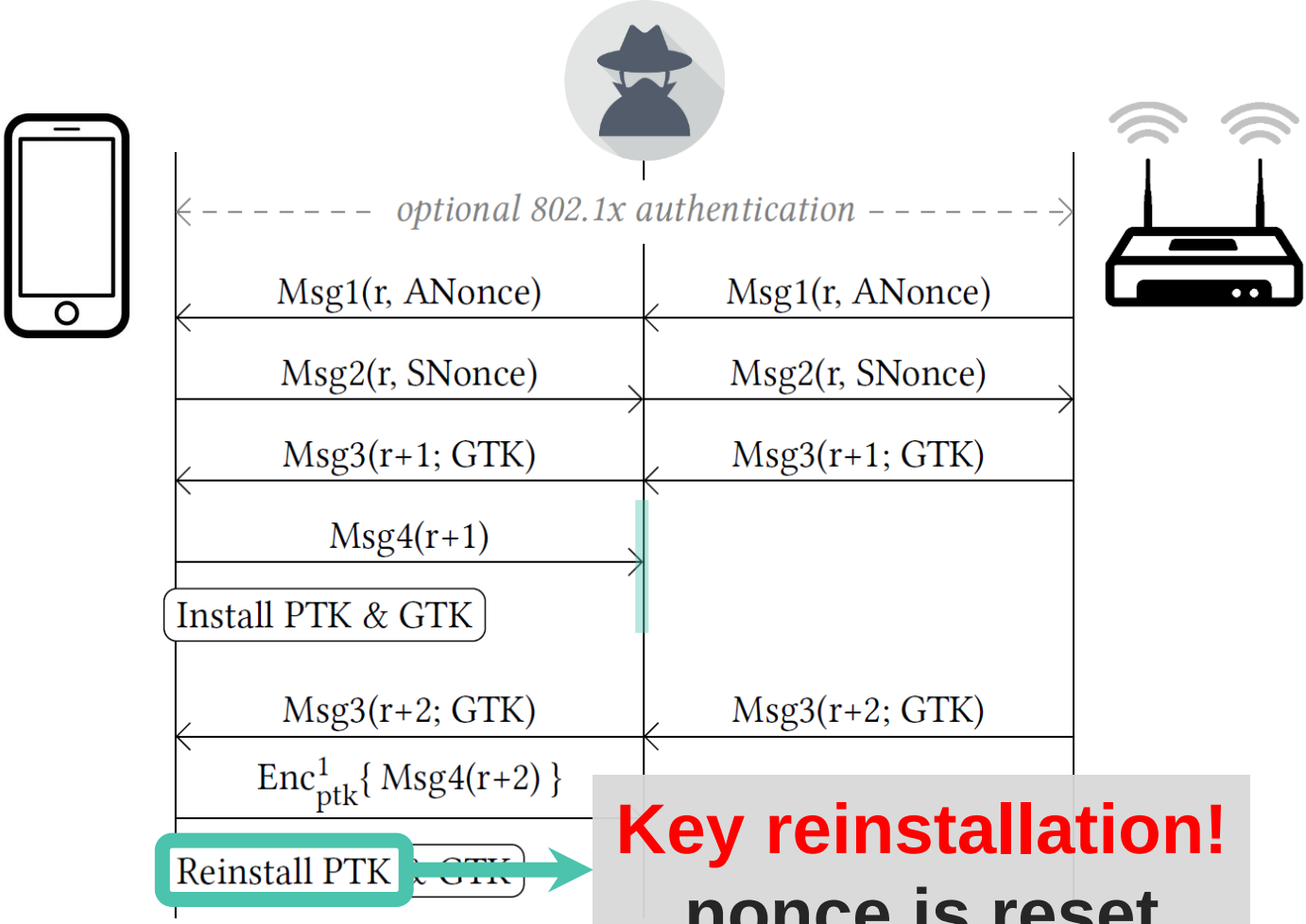

optional 802.1x authentication
Msg1(r, ANonce)
Msg1(r, ANonce)
Msg2(r, SNonce)
Msg2(r, SNonce)
Msg3(r+1; GTK)
Msg3(r+1; GTK)
Msg4(r+1)
Block Msg4
Install PTK & GTK

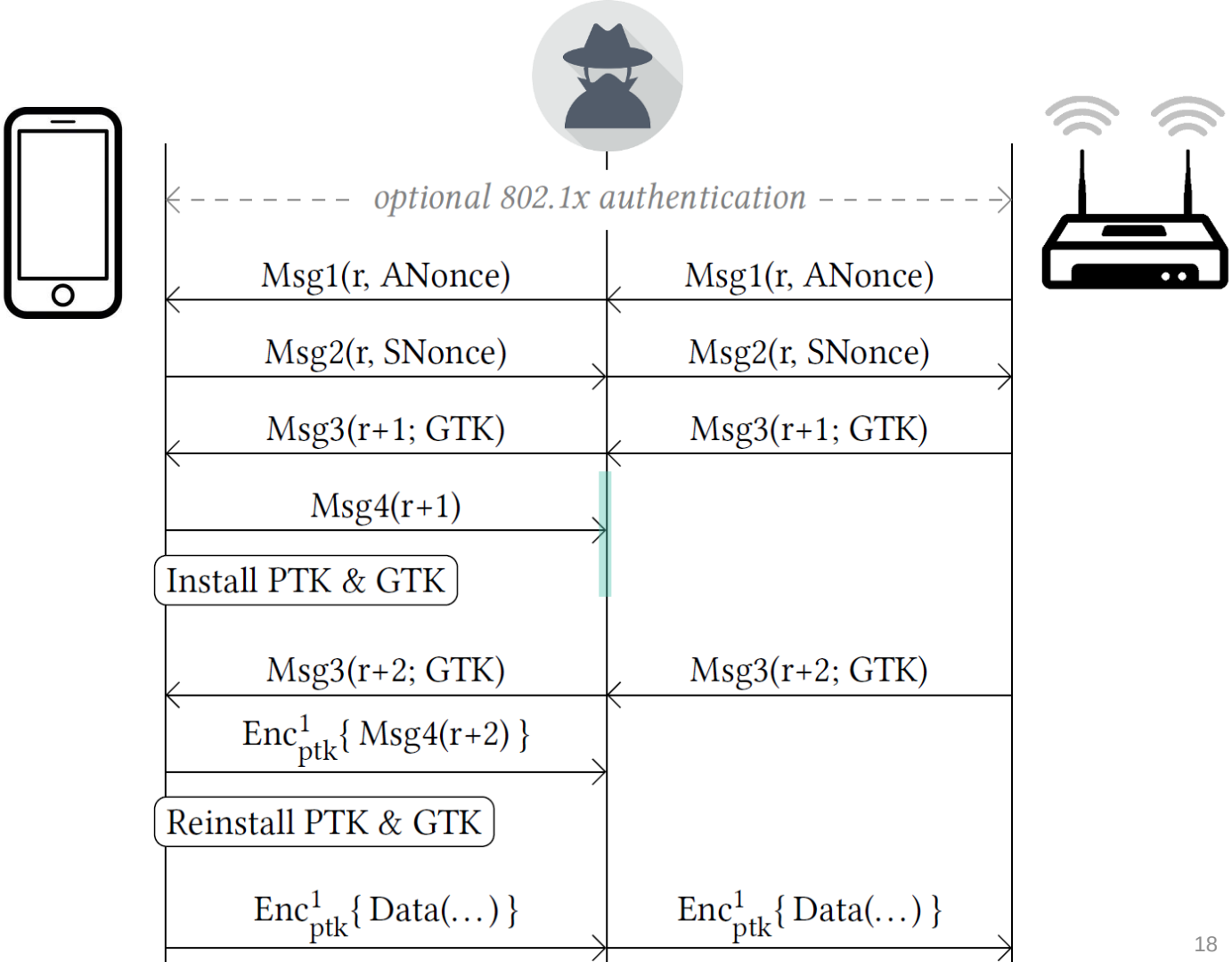

optional 802.1x authentication
Msg1(r, ANonce)
Msg1(r, ANonce)
Msg2(r, SNonce)
Msg2(r, SNonce)
Msg3(r+1; GTK)
Msg3(r+1; GTK)
Msg4(r+1)
Install PTK & GTK
Msg3(r+2; GTK)
Msg3(r+2; GTK)
Enc1ptk{ Msg4(r+2) }

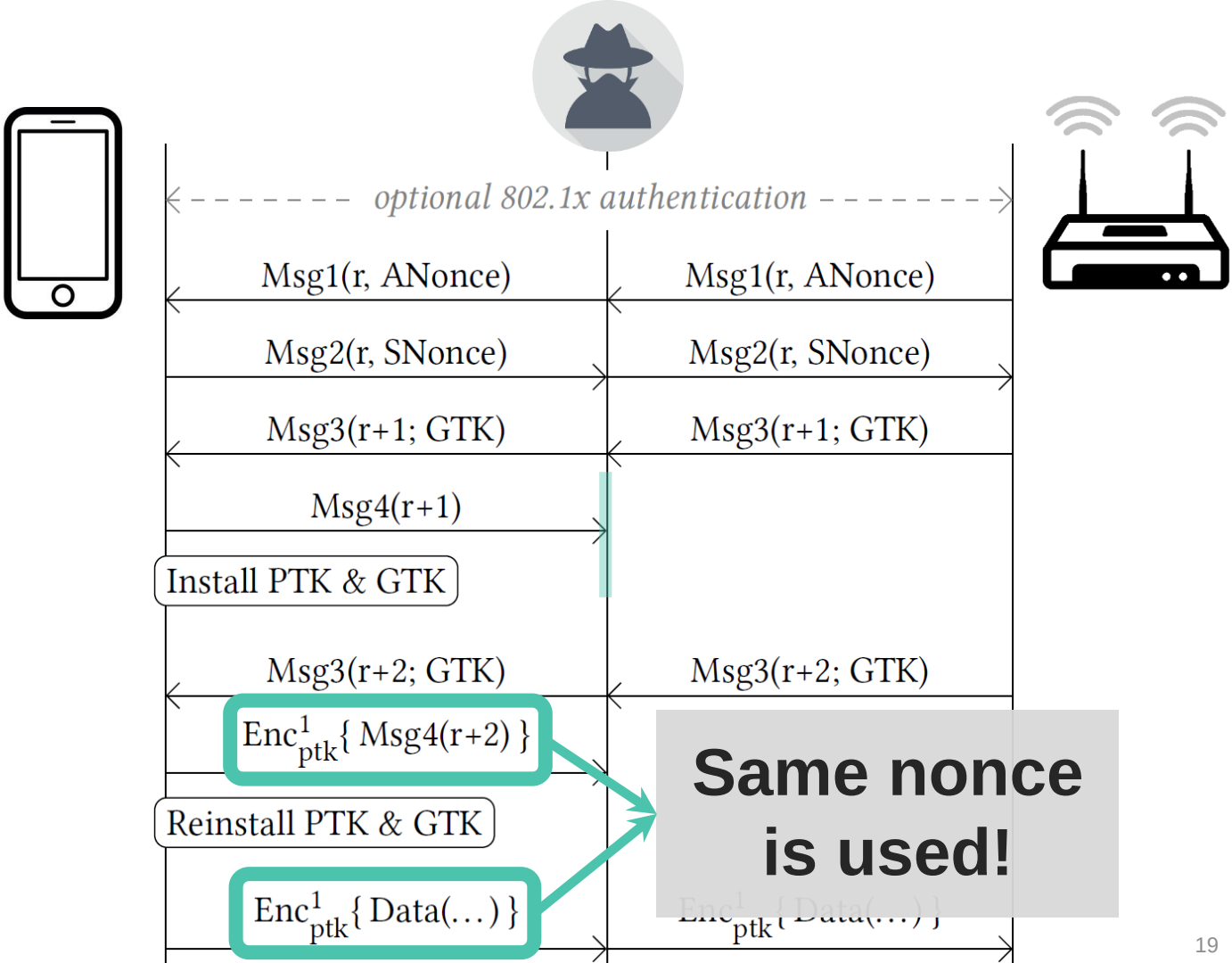


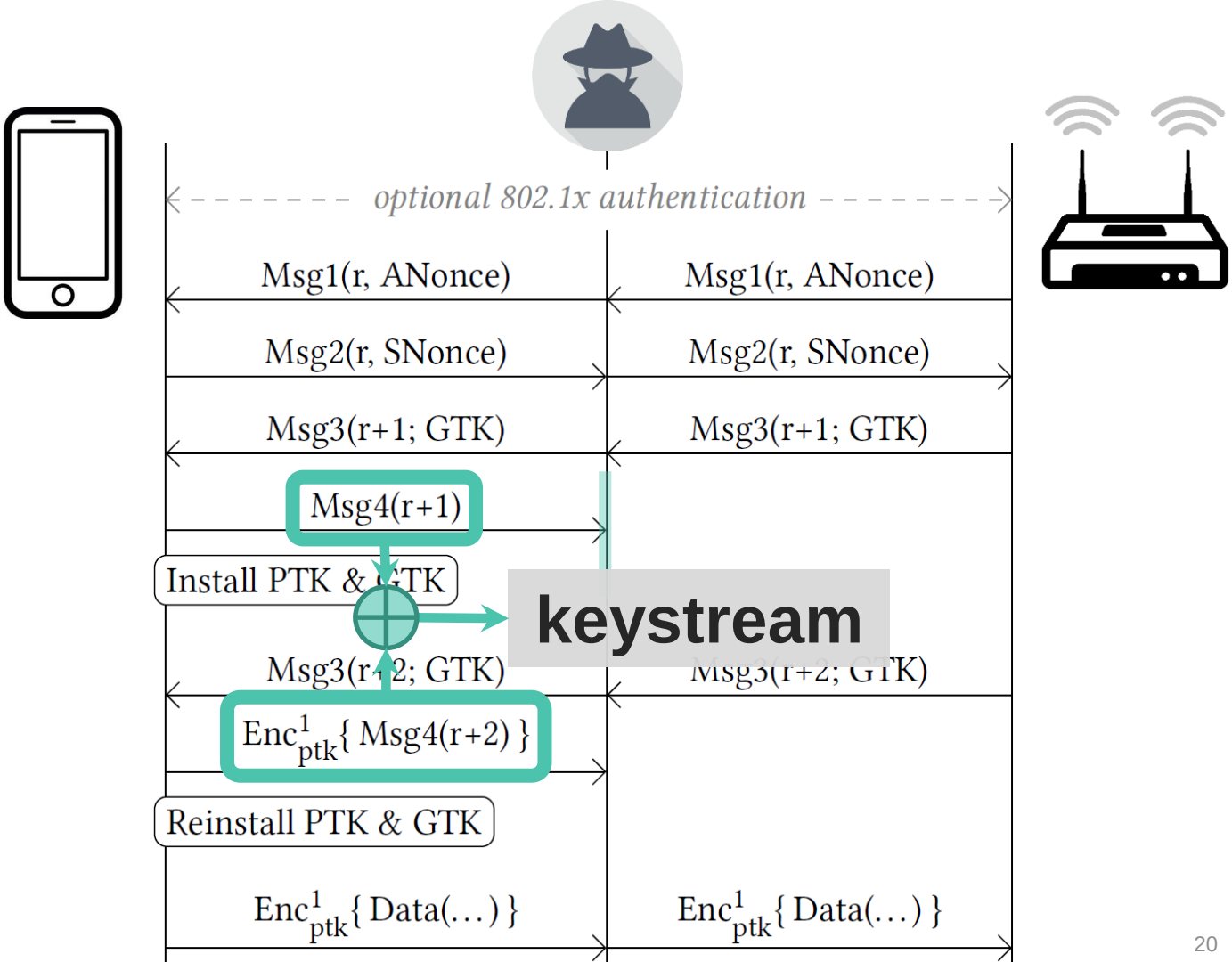
In practice Msg4 is sent encrypted

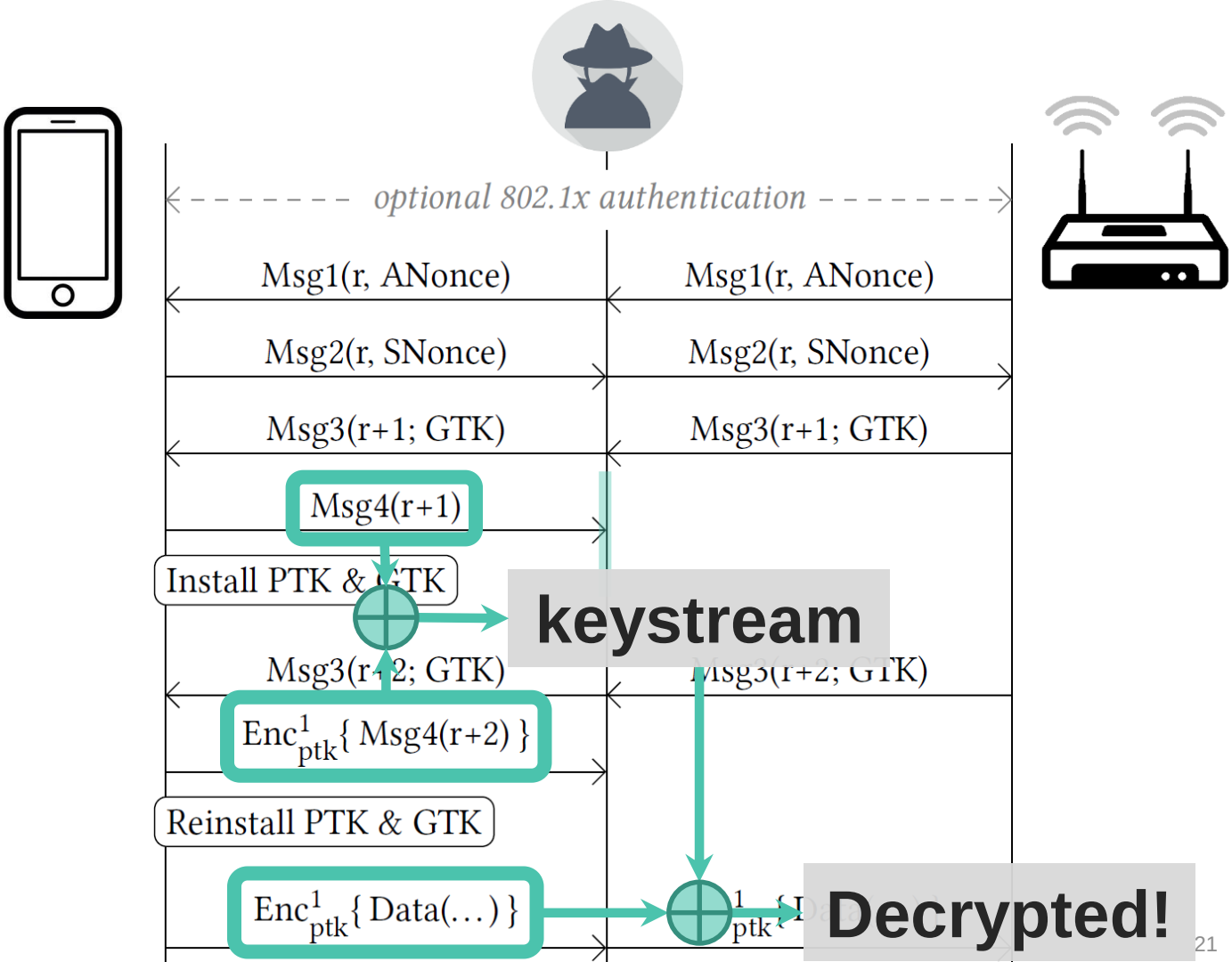

optional 802.1x authentication
Msg1(r, ANonce)
Msg1(r, ANonce)
Msg2(r, SNonce)
Msg2(r, SNonce)
Msg3(r+1; GTK)
Msg3(r+1; GTK)
Msg4(r+1)
Install PTK & GTK
Msg3(r+2; GTK)
Msg3(r+2; GTK)
Enc1ptk{ Msg4(r+2) }
Reinstall PTK & GTK


optional 802.1x authentication
Msg1(r, ANonce)
Msg1(r, ANonce)
Msg2(r, SNonce)
Msg2(r, SNonce)
Msg3(r+1; GTK)
Msg3(r+1; GTK)
Msg4(r+1)
Install PTK & GTK
Msg3(r+2; GTK)
Msg3(r+2; GTK)
Enc1ptk{ Msg4(r+2) }
Reinstall PTK
Key reinstallation!
nonce is reset


optional 802.1x authentication
Msg1(r, ANonce)
Msg1(r, ANonce)
Msg2(r, SNonce)
Msg2(r, SNonce)
Msg3(r+1; GTK)
Msg3(r+1; GTK)
Msg4(r+1)
Install PTK & GTK
Msg3(r+2; GTK)
Msg3(r+2; GTK)
Enc1ptk{ Msg4(r+2) }
Reinstall PTK & GTK
Enc1ptk{ Data(...) }
Enc1ptk{ Data(...) }

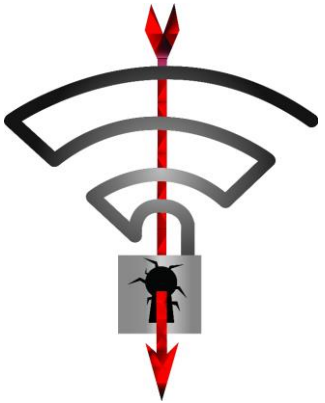

optional 802.1x authentication
Msg1(r, ANonce)
Msg1(r, ANonce)
Msg2(r, SNonce)
Msg2(r, SNonce)
Msg3(r+1; GTK)
Msg3(r+1; GTK)
Msg4(r+1)
Install PTK & GTK
Msg3(r+2; GTK)
Msg3(r+2; GTK)
Enc1ptk{ Msg4(r+2) }
Reinstall PTK & GTK
Enc1ptk{ Data(…) }
Same nonce is used!


optional 802.1x authentication
Msg1(r, ANonce)
Msg1(r, ANonce)
Msg2(r, SNonce)
Msg2(r, SNonce)
Msg3(r+1; GTK)
Msg3(r+1; GTK)
Msg4(r+1)
Install PTK & GTK
keystream
Msg3(r+2; GTK)
Msg3(r+2; GTK)
Enc1ptk{ Msg4(r+2) }
Reinstall PTK & GTK
Enc1ptk{ Data(...) }
Enc1ptk{ Data(...) }


optional 802.1x authentication
Msg1(r, ANonce)
Msg1(r, ANonce)
Msg2(r, SNonce)
Msg2(r, SNonce)
Msg3(r+1; GTK)
Msg3(r+1; GTK)
Msg4(r+1)
Install PTK & GTK
keystream
Msg3(r+2; GTK)
Enc1ptk{ Msg4(r+2) }
Reinstall PTK & GTK
Enc1ptk{ Data(…) }
Decrypted!

Overview

1. Key reinstallation in 4-way handshake



2. **Misconceptions and remarks**

3. Steps to improve Wi-Fi security?

Misconceptions I

No useful data is transmitted after handshake

- Trigger handshakes during TCP connection

Difficult to derive keystream

- Already have 82 bytes from encrypted Msg4

Need high signal strength to get MitM

- Use channel switch announcements, BSS Transition Requests, jammers, …

Misconceptions II

Need to be close to network

- Can use special antenna[2,3]

Using (AES-)CCMP mitigates the attack

- No, still allows decryption & replay of frames

Enterprise networks (802.1x) are not vulnerable

- Also use 4-way handshake and are affected

Misconceptions III

You need the password to perform attacks

- Nope. Then you could decrypt all already …

Updating only client or AP is sufficient

- Both vulnerable clients and vulnerable APs need to apply patches

Attack complexity is hard

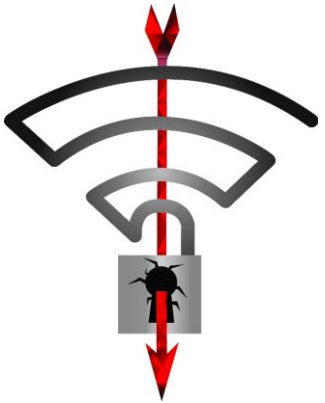
- Script only needs to be written once

“Attacks only get better, they never get worse.”

— Bruce Schneier

Overview

1. Key reinstallation in 4-way handshake



2. Misconceptions and remarks

3. Steps to improve Wi-Fi security?

Countermeasures

Problem: many clients will not get updated

Solution: AP can prevent attacks on clients!

- Don’t retransmit message 3/4
- Don’t retransmit group message 1/2

However:

- Impact on reliability currently unclear
- Clients still vulnerable when connected to other unmodified APs

Fuzzing

Basic fuzzing as part of device certification

- Test against key reinstallations
- Fuzzing length fields: avoid well-known bugs
- Plaintext frames rejected if encryption enabled?
- …

Advanced fuzzing of widely used tools:

- Can do more costly fuzzing on specific tools
- Make these fuzzing tools open source

“Millions of dollars saved (for Microsoft and the world).”

Patrice Godefroid, Microsoft Research

Other recommendations

Not Wi-Fi Alliance task, but …

- Make standards easier to access. Just a download link, nothing on top.

- Anyone should be able to easily follow discussions. Mailing list?

Need open source firmware

Code is getting more closed:

- Functionality is offloaded to closed firmware
- E.g. 4-way handshake is being offloaded
- We cannot trust this code!

At least open source security critical parts?

- Catch problems earlier & get help

Long-term: formal verification

Programming is hard. Are patches correct?

- Missed attack against wpa_supplicant 2.6

Collaboration with academia:

- Create formal and precise state machines
- Formal verification of core code
- E.g. prove correctness of open source tools

Thank you!

Questions?

krackattacks.com

References

1. C. He, M. Sundararajan, A. Datta, A. Derek, and J. Mitchell. A Modular Correctness Proof of IEEE 802.11i and TLS. In CCS, 2005.
2. S. Antakis, M. van Cuijk, and J. Stemmer. Wardriving - Building A Yagi Pringles Antenna. 2008.
3. M. Parkinson. Designer Cantenna. 2012. Retrieved 23 October 2017 from https://www.mattparkinson.eu/designer-cantenna/