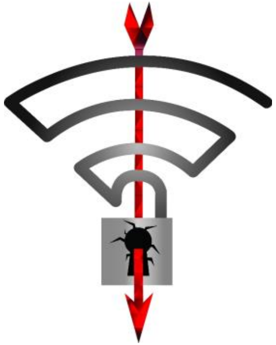
Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2

Mathy Vanhoef — @vanhoefm

CCS 2017, 1 October 2017

KU LEUVEN DistriNet

Overview



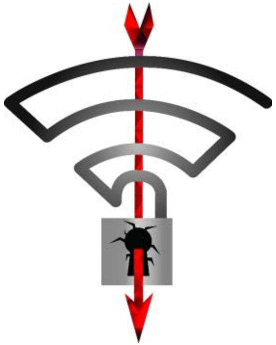
Key reinstalls in
4-way handshake

Misconceptions

Practical impact

Lessons learned

Overview



Key reinstalls in 4-way handshake

Misconceptions

Practical impact

Lessons learned

The 4-way handshake

Used to connect to any protected Wi-Fi network

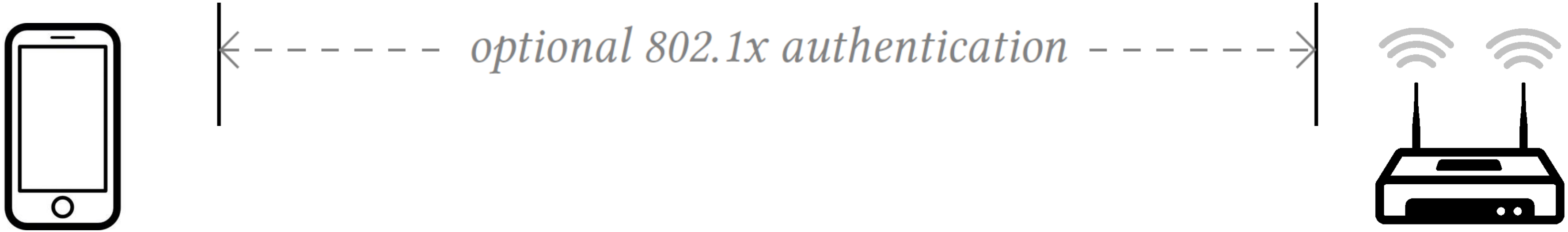
Two main purposes:

- Mutual authentication
- Negotiate fresh PTK: pairwise temporal key

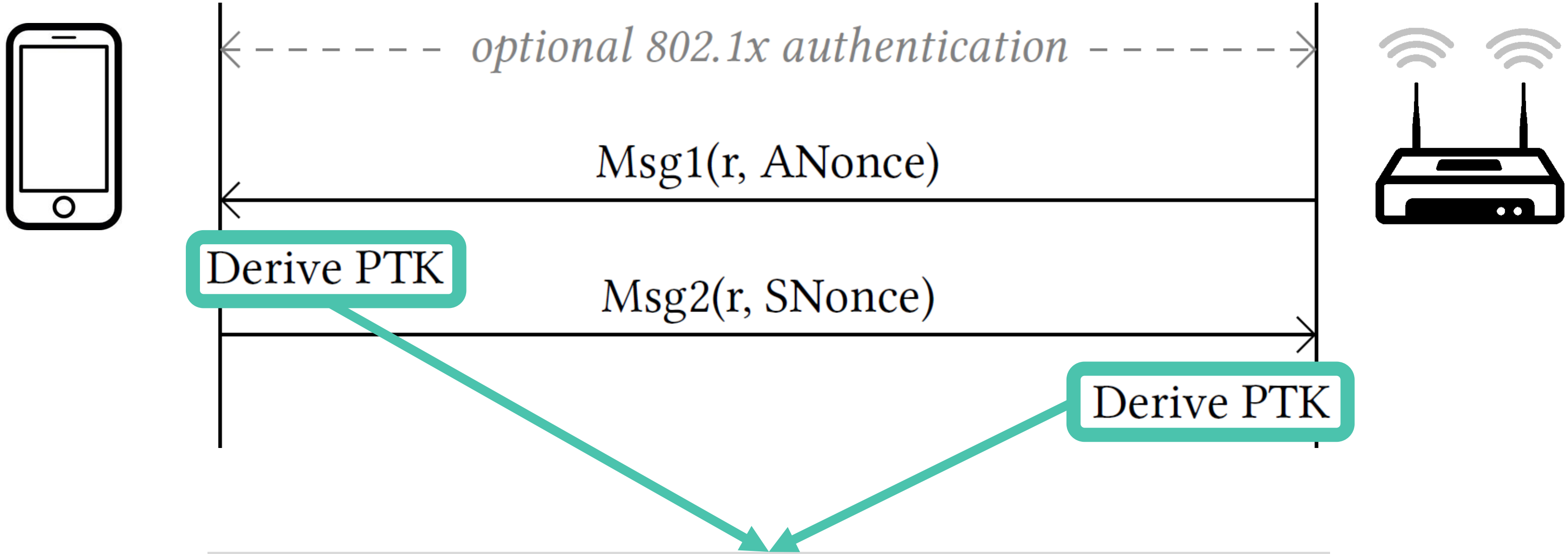
Appeared to be secure:

- No attacks in over a decade (apart from password guessing)
- Proven that negotiated key (PTK) is secret[1]
- And encryption protocol proven secure[7]

4-way handshake (simplified)



4-way handshake (simplified)



optional 802.1x authentication

Msg1(r, ANonce)

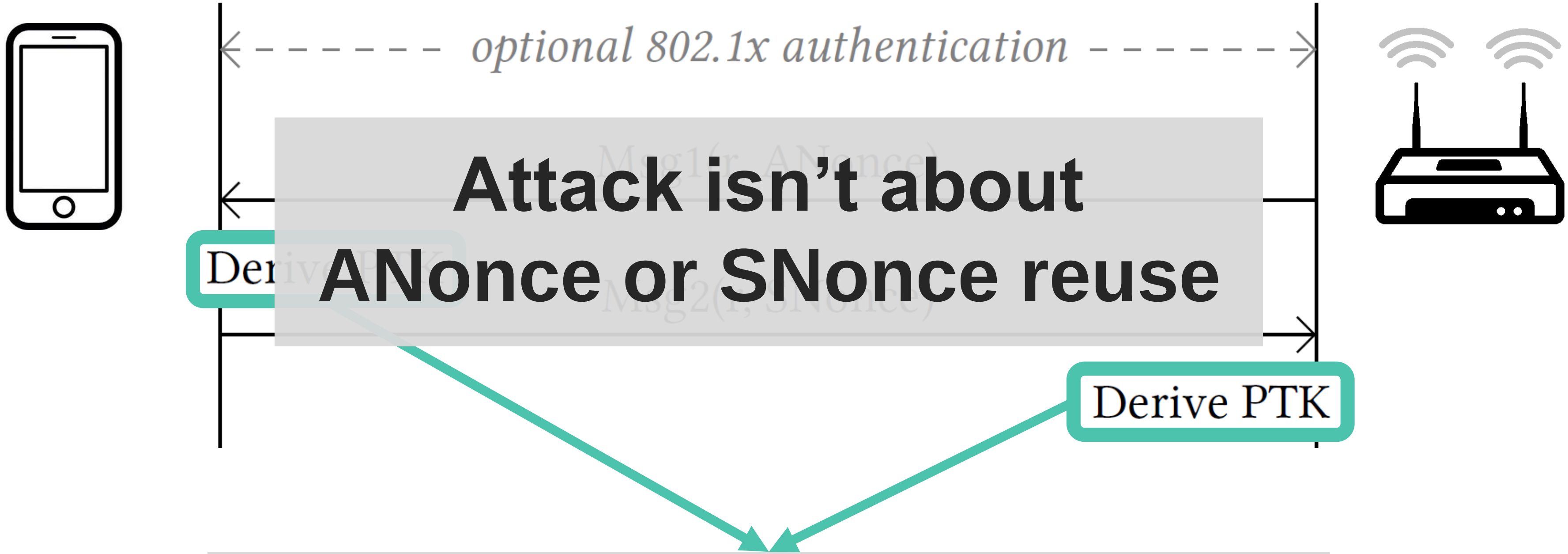
Derive PTK

Msg2(r, SNonce)

Derive PTK

PTK = Combine(shared secret, ANonce, SNonce)

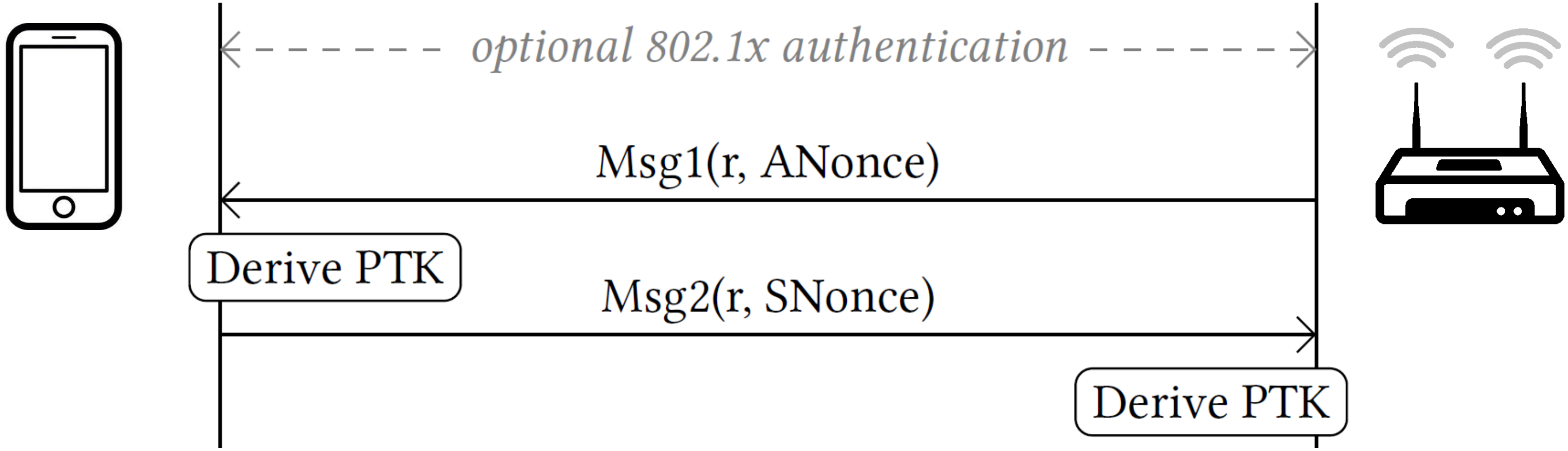
4-way handshake (simplified)



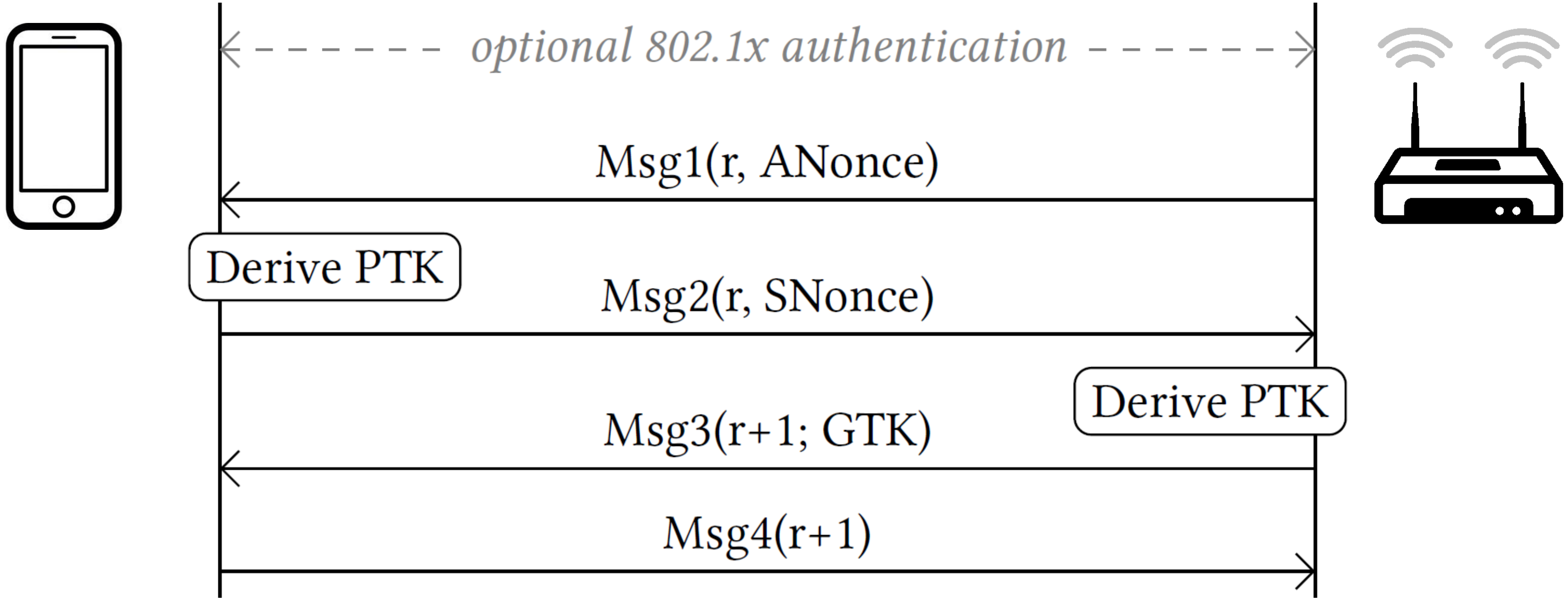
Attack isn’t about ANonce or SNonce reuse

PTK = Combine(shared secret, ANonce, SNonce)

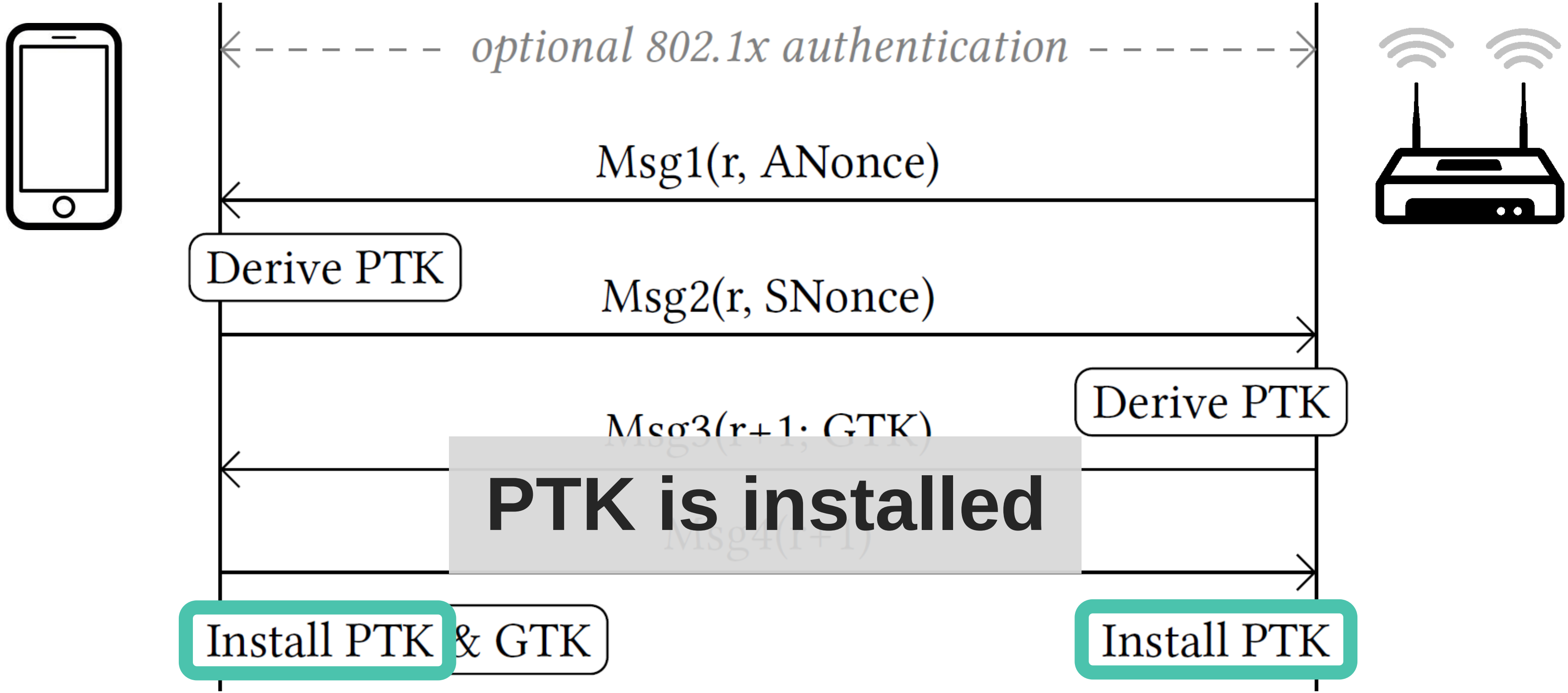
4-way handshake (simplified)



4-way handshake (simplified)



4-way handshake (simplified)



optional 802.1x authentication

Msg1(r, ANonce)

Derive PTK

Msg2(r, SNonce)

Derive PTK

Msg3(r+1; GTK)

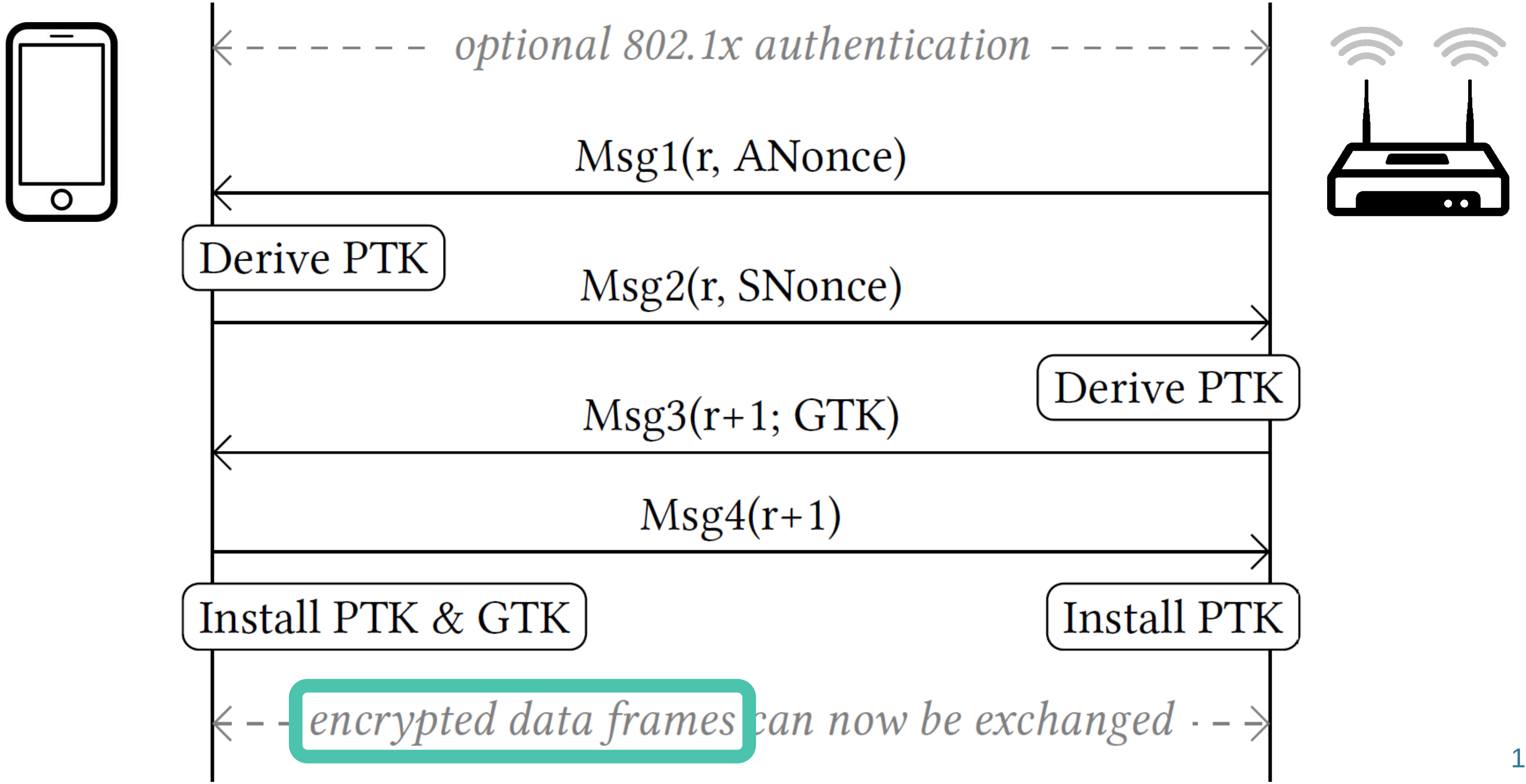
PTK is installed

Msg4(r+1)

Install PTK & GTK

Install PTK

4-way handshake (simplified)



optional 802.1x authentication

Msg1(r, ANonce)

Derive PTK

Msg2(r, SNonce)

Derive PTK

Msg3(r+1; GTK)

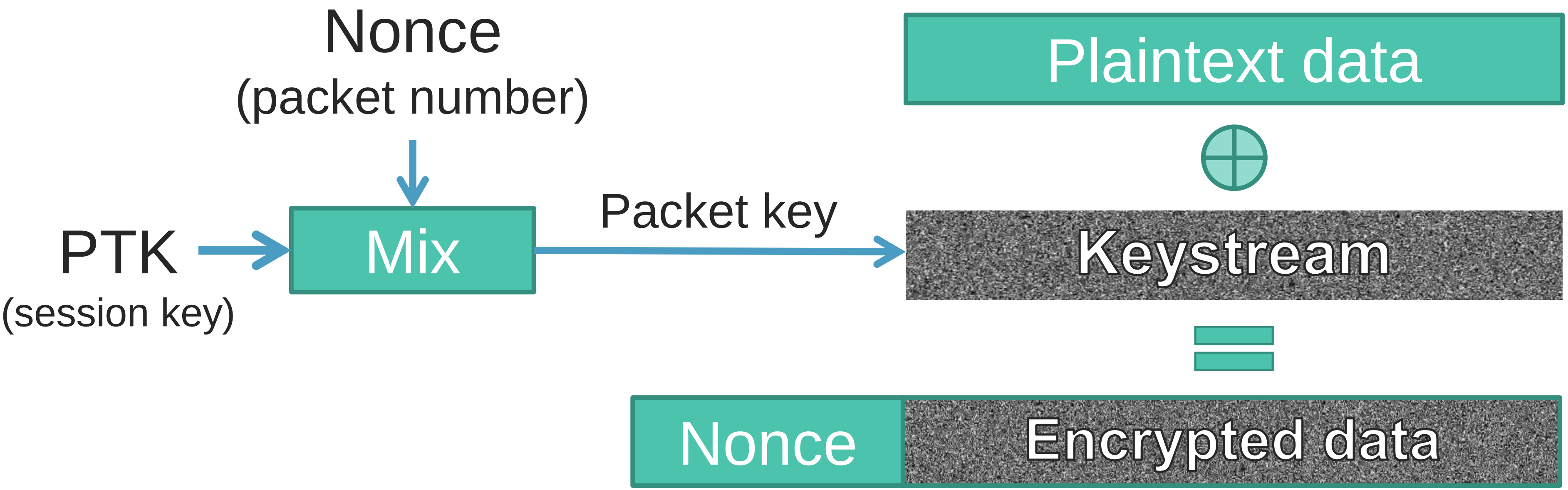
Msg4(r+1)

Install PTK & GTK

Install PTK

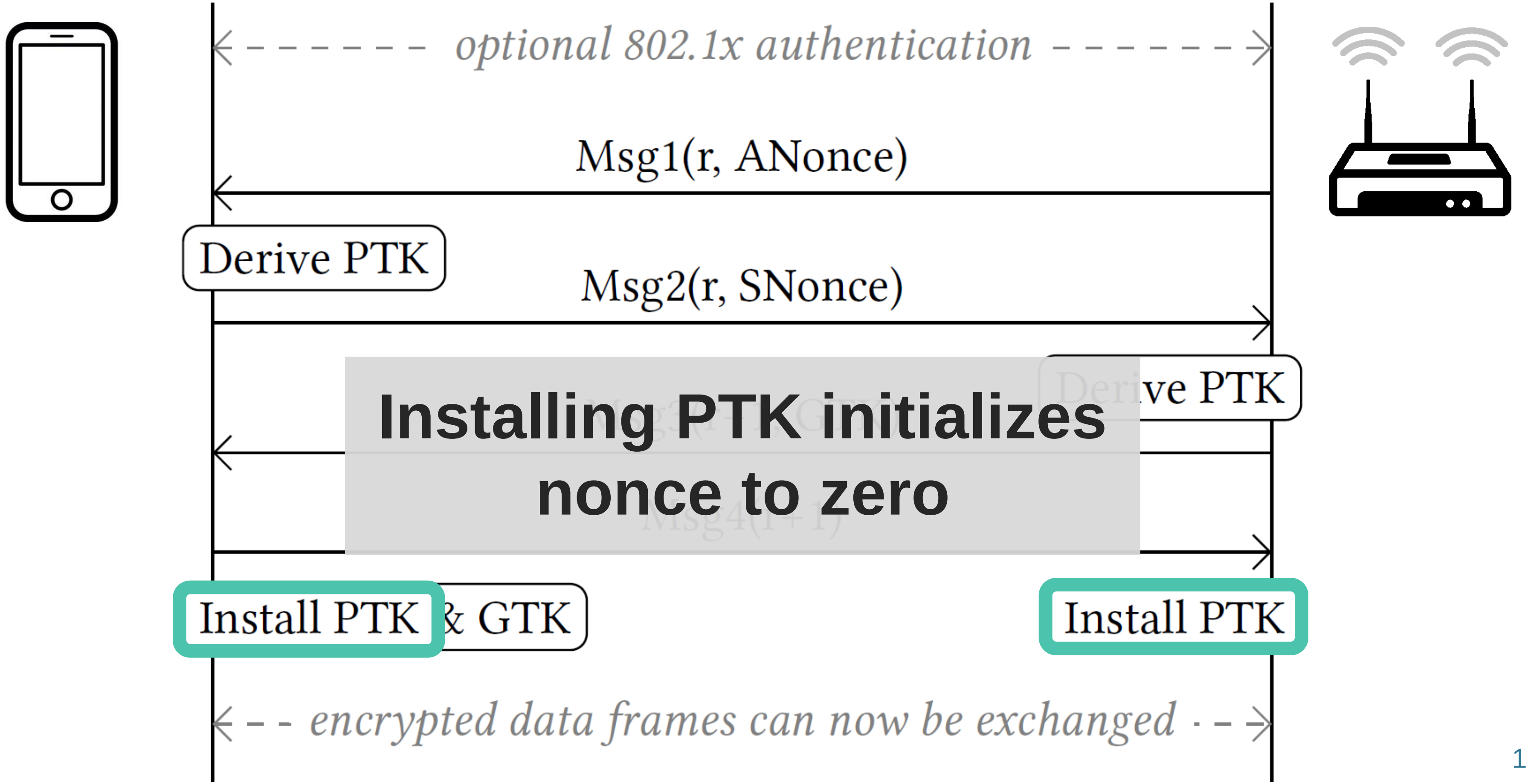
encrypted data frames can now be exchanged

Frame encryption (simplified)



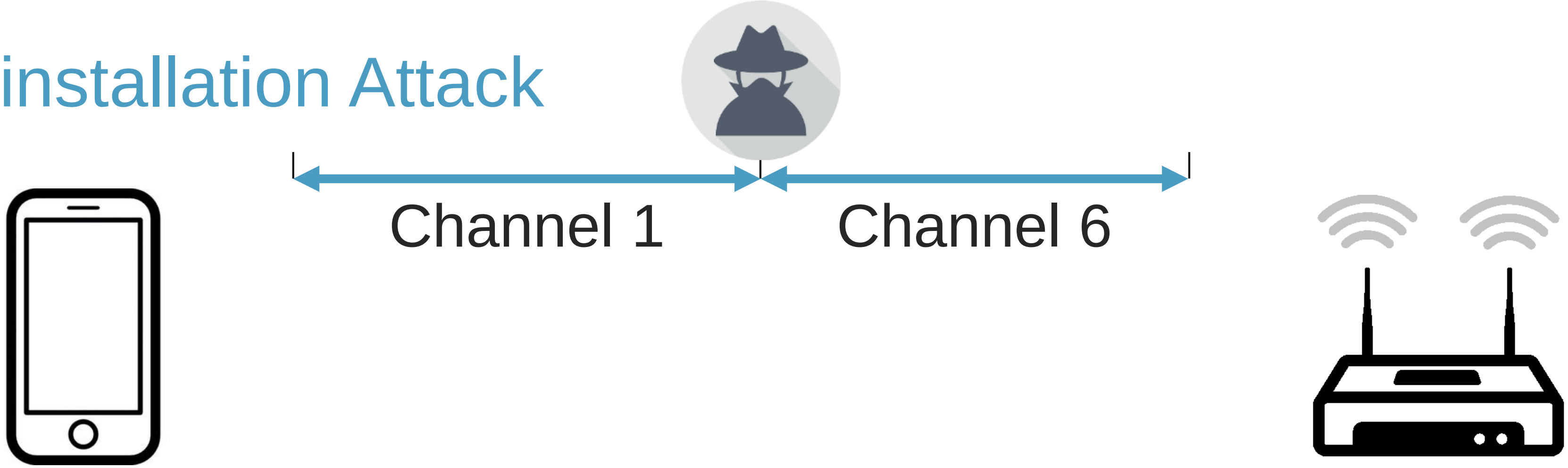
→ Nonce reuse implies keystream reuse (in all WPA2 ciphers)

4-way handshake (simplified)

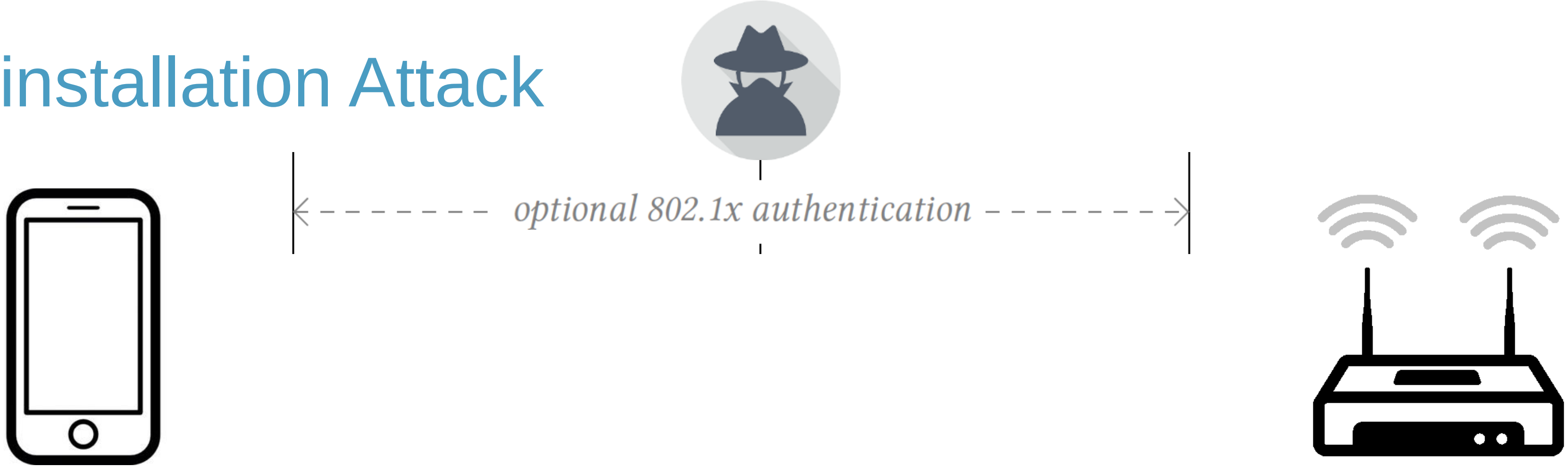


Installing PTK initializes nonce to zero

Reinstallation Attack

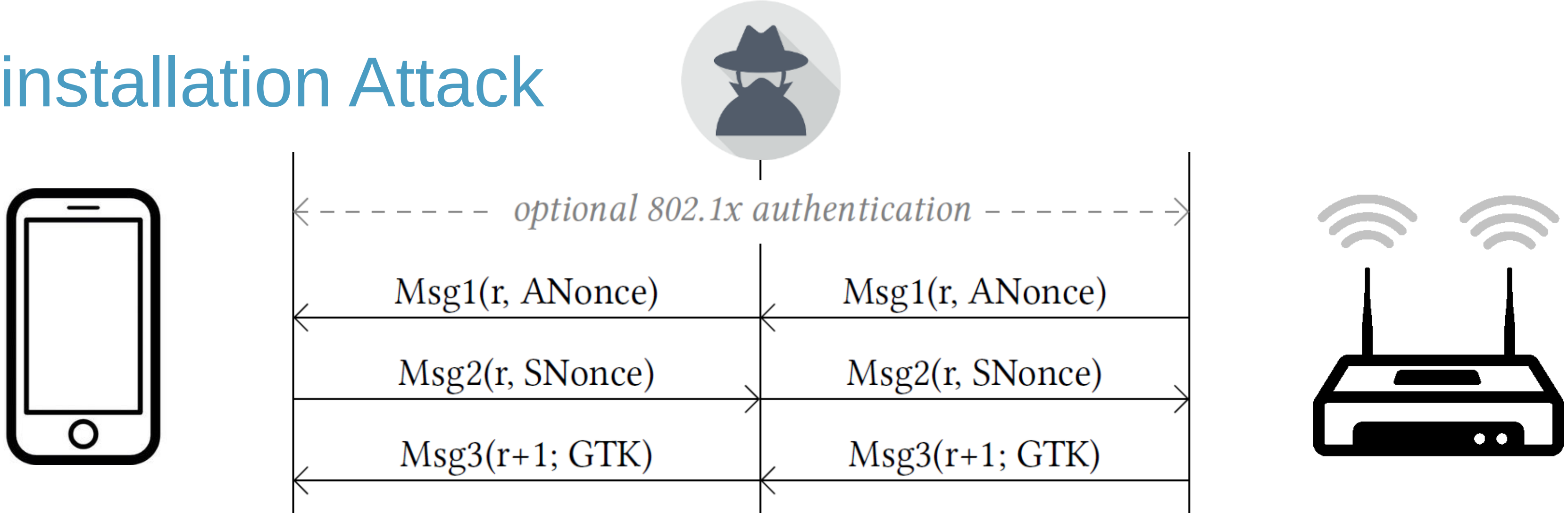


Reinstallation Attack



optional 802.1x authentication

Reinstallation Attack



optional 802.1x authentication

Msg1(r, ANonce)

Msg1(r, ANonce)

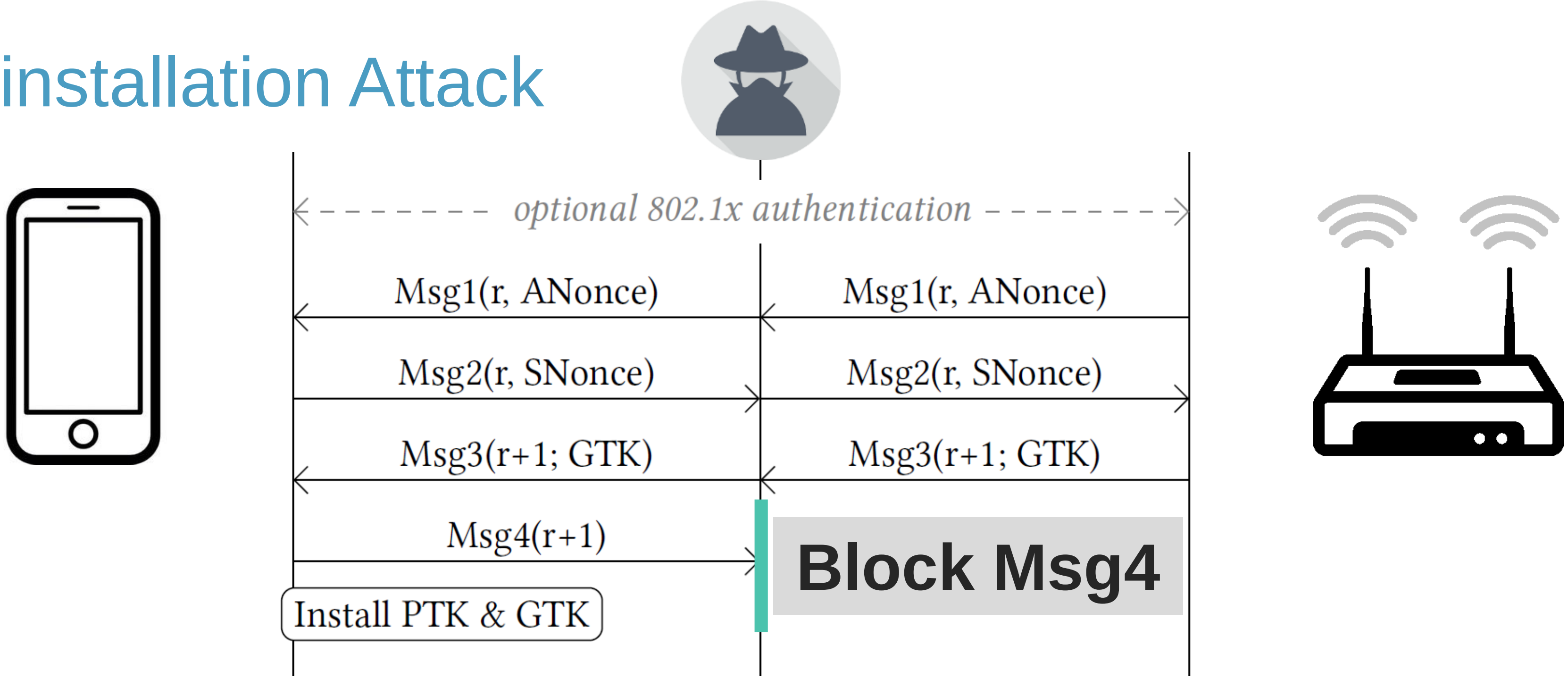
Msg2(r, SNonce)

Msg2(r, SNonce)

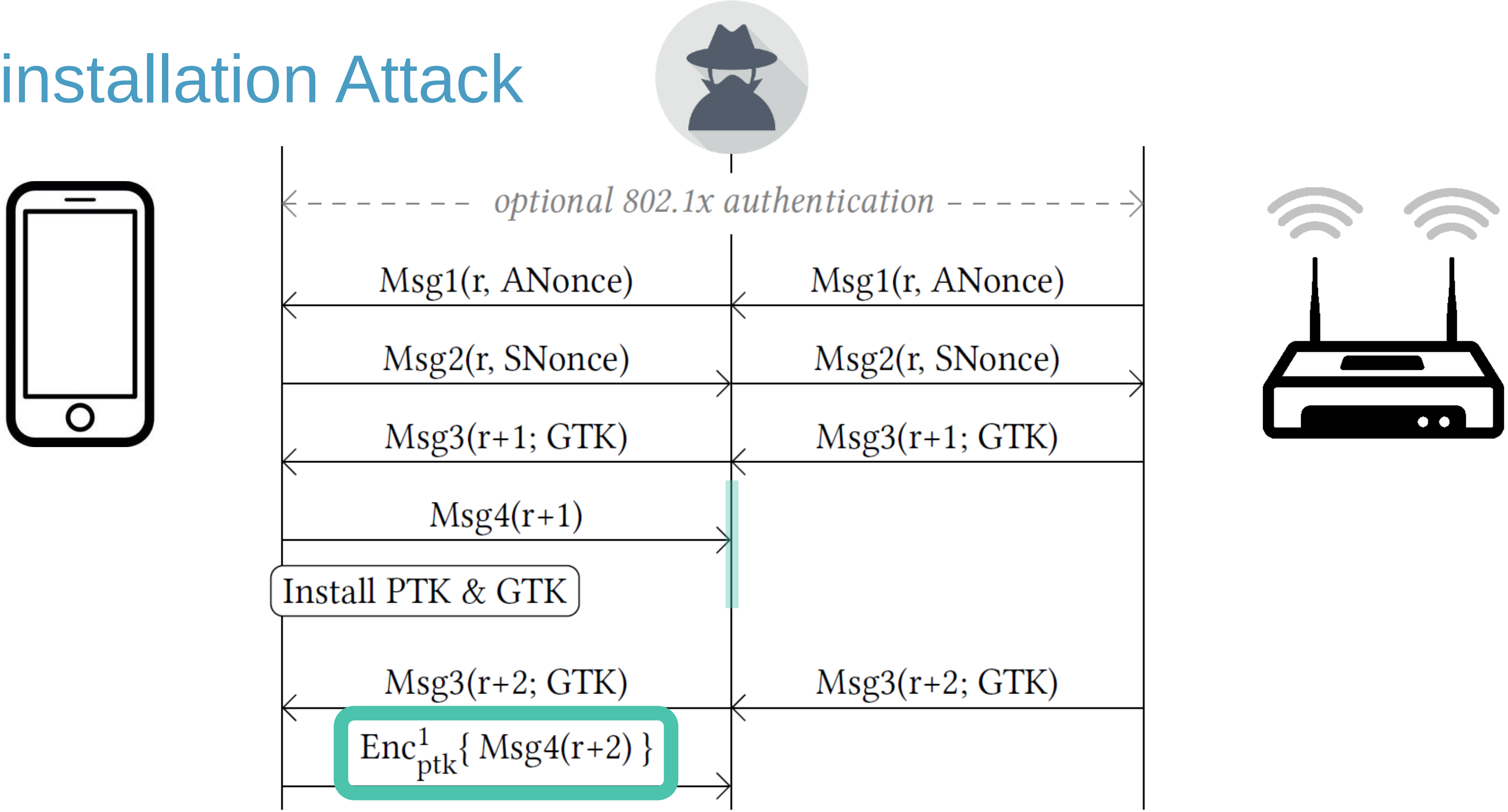
Msg3(r+1; GTK)

Msg3(r+1; GTK)

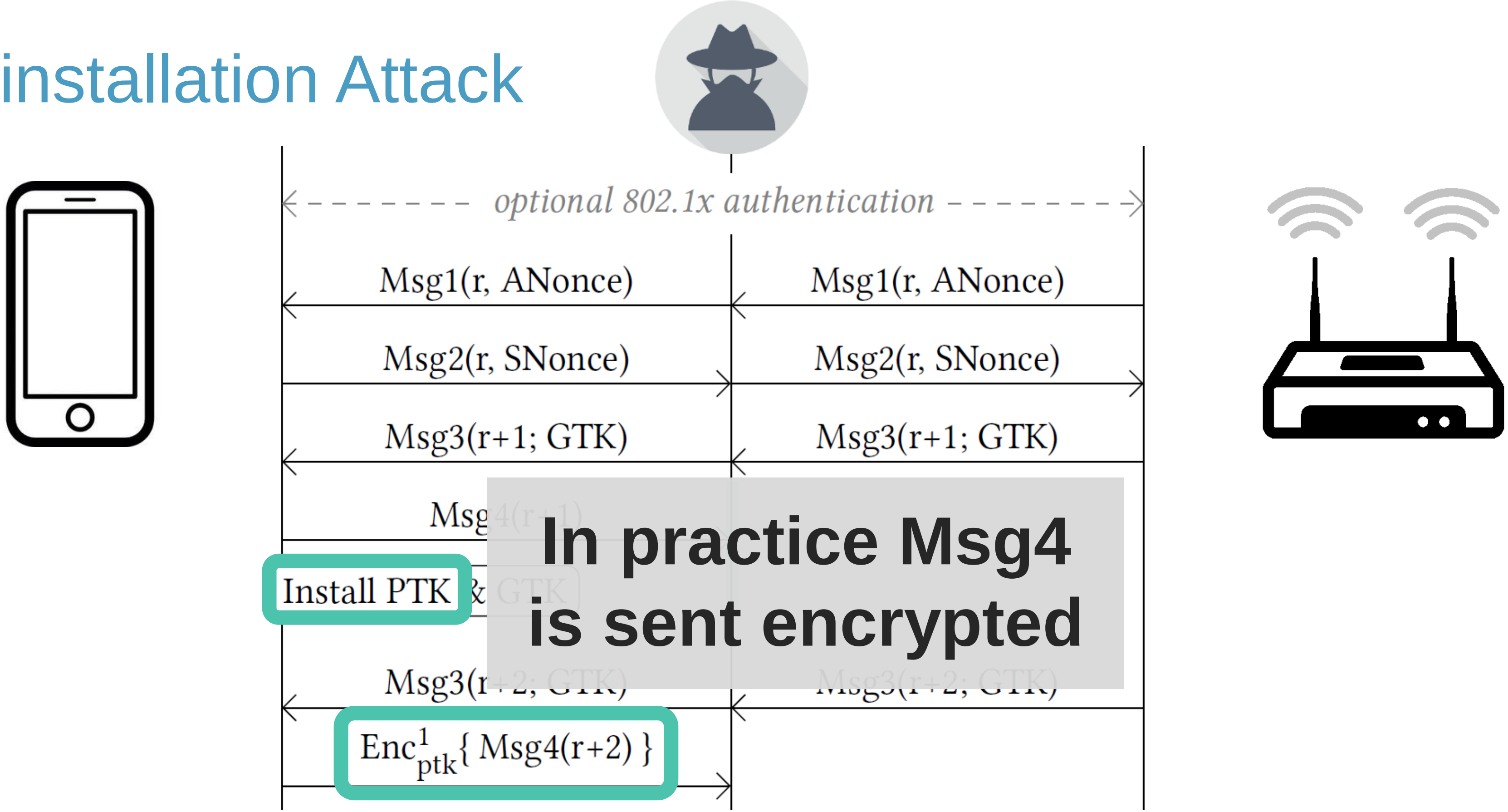
Reinstallation Attack



Reinstallation Attack

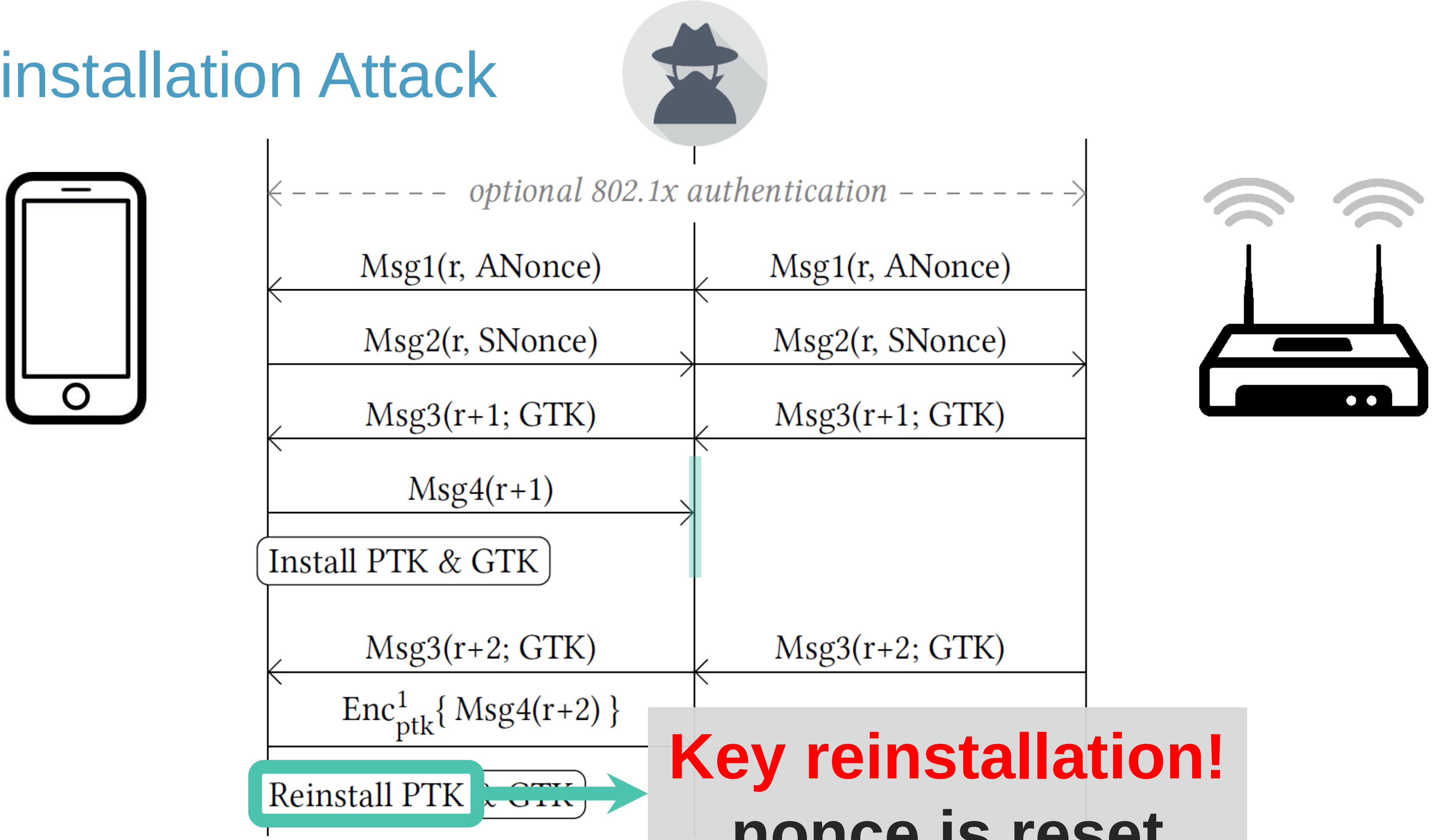


Reinstallation Attack



In practice Msg4 is sent encrypted

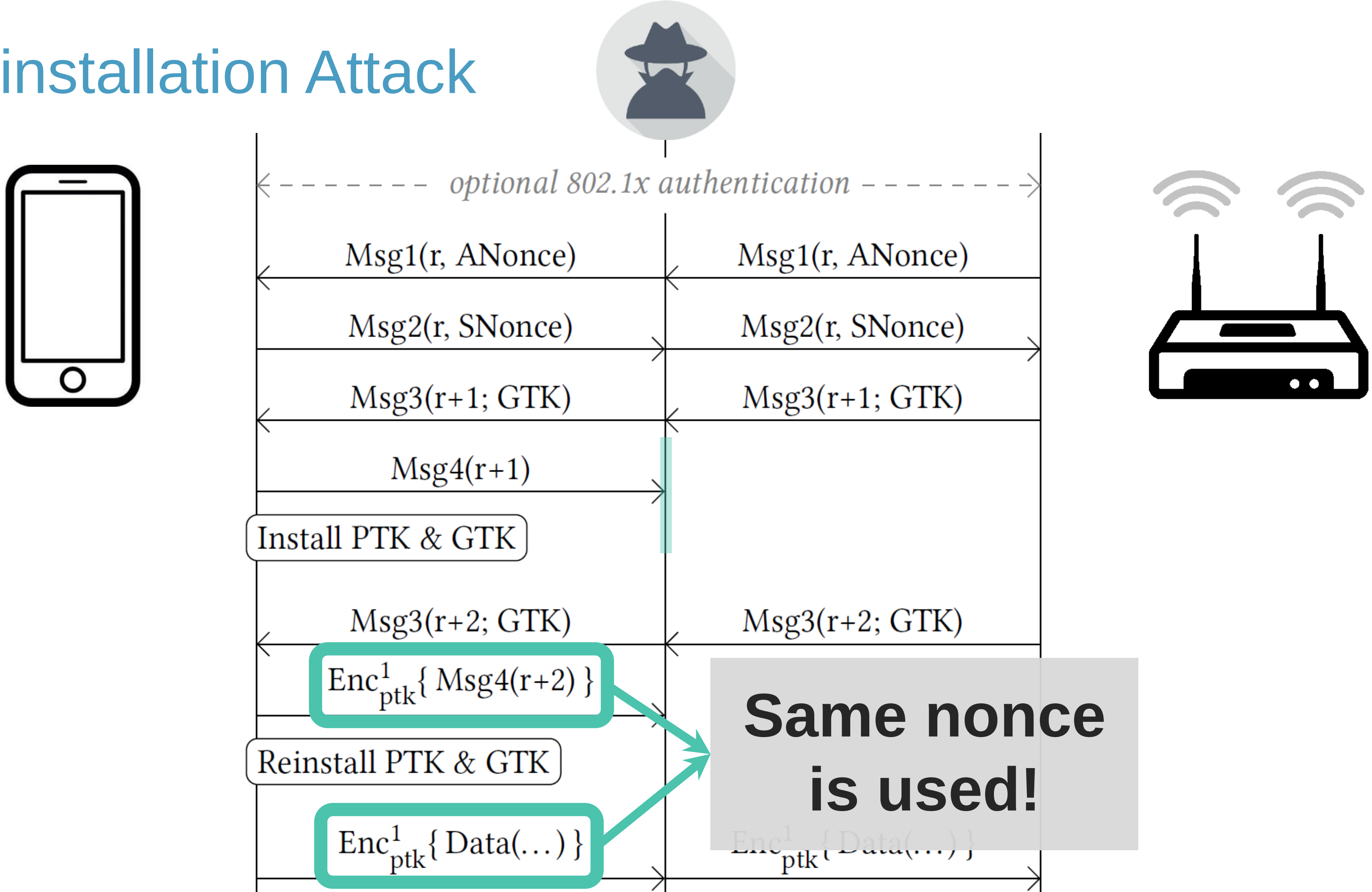
Reinstallation Attack



Key reinstallation!

nonce is reset

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)

Msg4(r+1)

Install PTK & GTK

Msg3(r+2; GTK) Msg3(r+2; GTK)

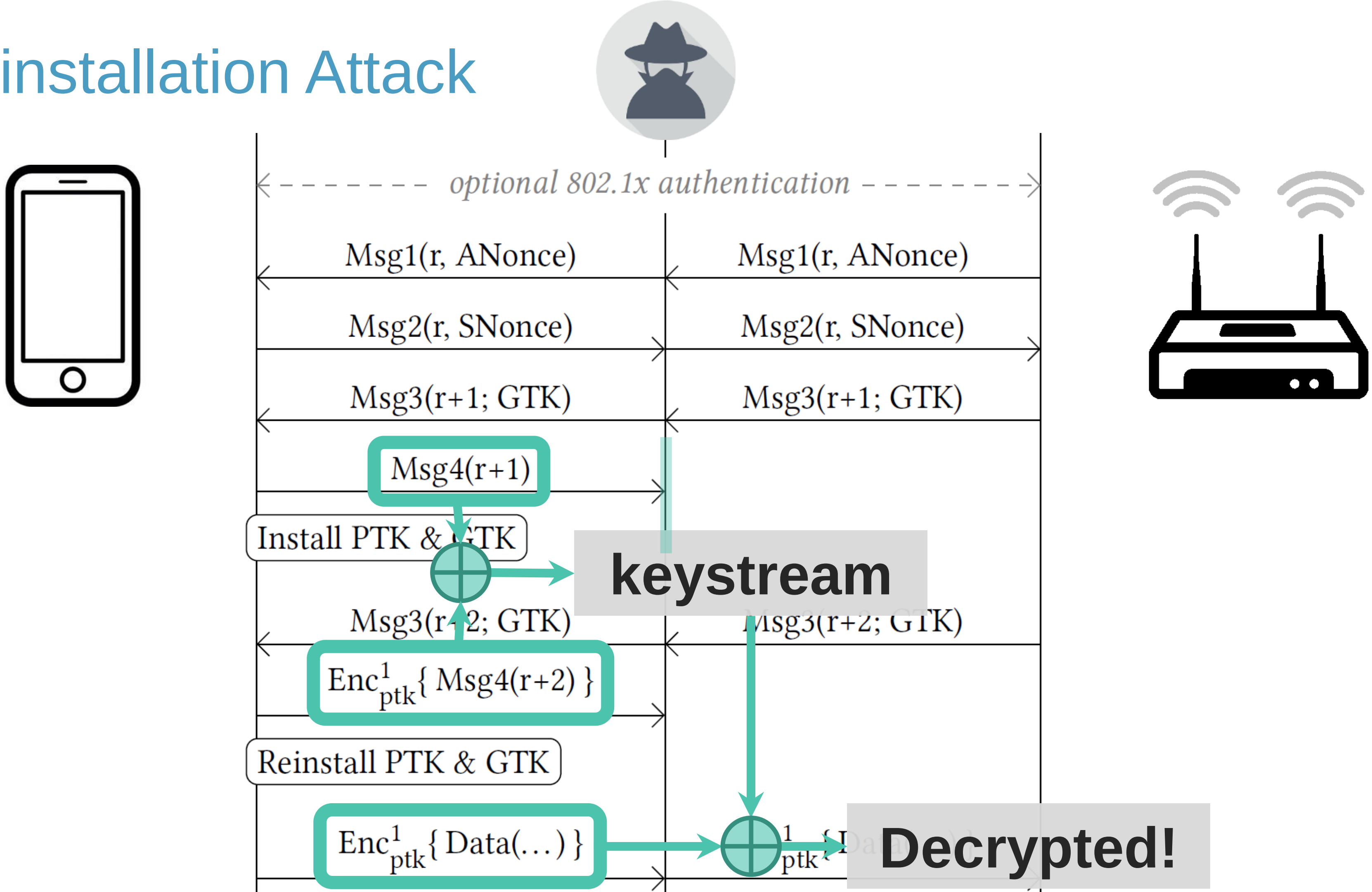
$\text{Enc}^{1}_{\text{ptk}}\{\ \text{Msg4}(r+2)\ \}$

Reinstall PTK & GTK

$\text{Enc}^{1}_{\text{ptk}}\{\ \text{Data}(\ldots)\ \}$ $\text{Enc}^{1}_{\text{ptk}}\{\ \text{Data}(\ldots)\ \}$

Same nonce is used!

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)

Msg4(r+1)

Install PTK & GTK

keystream

Msg3(r+2; GTK)

Msg3(r+2; GTK)

$\text{Enc}^{1}_{\text{ptk}}\{\text{ Msg4(r+2) }\}$

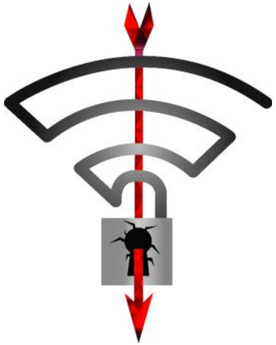
Reinstall PTK & GTK

$\text{Enc}^{1}_{\text{ptk}}\{\text{ Data(...) }\}$

$\text{Enc}^{1}_{\text{ptk}}\{\text{ Data(...) }\}$

Decrypted!

Overview



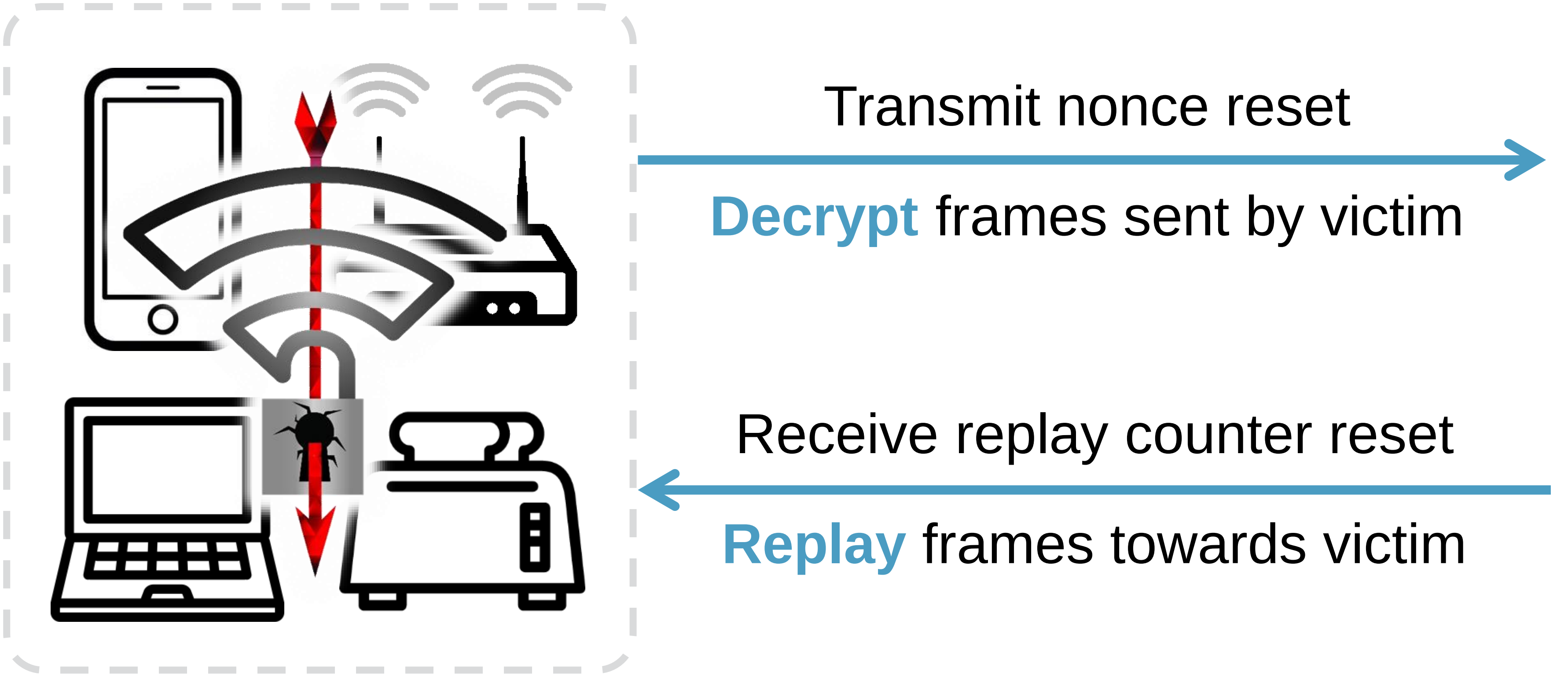
Key reinstalls in 4-way handshake

Misconceptions

Practical impact

Lessons learned

General impact



Transmit nonce reset

Decrypt frames sent by victim

Receive replay counter reset

Replay frames towards victim

Cipher suite specific

AES-CCMP: No practical frame forging attacks

WPA-TKIP:

- Recover Message Integrity Check key from plaintext[4,5]
- **Forge/inject** frames sent by the device under attack

GCMP (WiGig):

- Recover GHASH authentication key from nonce reuse[6]
- **Forge/inject** frames in **both directions**

Handshake specific

Group key handshake:

- Client is attacked, but only AP sends real broadcast frames
- Can only replay broadcast frames to client

4-way handshake:

- Client is attacked → replay/decrypt/forge

FT handshake (fast roaming = 802.11r):

- Access Point is attacked → replay/decrypt/forge
- **No MitM required, can keep causing nonce resets**

Implementation specific

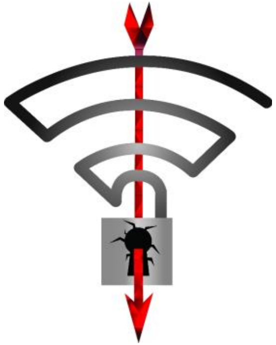
Windows and iOS: 4-way handshake not affected

- **Cannot decrypt unicast traffic** (nor replay/decrypt)
- But group key handshake is affected (replay broadcast)

wpa_supplicant 2.4+

- Client used on Linux and Android 6.0+
- On retransmitted msg3 will **install all-zero key**

Overview



Key reinstalls in 4-way handshake

Misconceptions

Practical impact

Lessons learned

Misconceptions I

Updating only the client or AP is sufficient

- Both vulnerable clients & vulnerable APs must apply patches

Need to be close to network and victim

- Can use special antenna from afar

No useful data is transmitted after handshake

- Trigger new handshakes during TCP connection

Misconceptions II

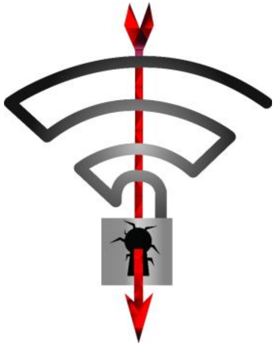
Obtaining channel-based MitM is hard

- Nope, can use channel switch announcements

Attack complexity is hard

- Script only needs to be written once …
- … and some are already doing this!

Overview



Key reinstalls in 4-way handshake

Misconceptions

Practical impact

Lessons learned

Limitations of formal proofs

- 4-way handshake proven secure
- Encryption protocol proven secure

The combination was not proven secure!

Model vs. implementation

Abstract model ≠ real code

- Must **assure code matches specification**

The wpa_supplicant 2.6 case

- Complex state machine & turned out to still be vulnerable
- Need **formal verification of implementations**

On a related note…

Workshop on:

Security Protocol Implementations: Development and Analysis (SPIDA)

Co-located with EuroS&P 2018

“focuses on improving development & analysis of security protocols implementations”

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/
4. E. and M. Beck. Practical attacks against WEP and WPA. In WiSec, 2009.
5. M. Vanhoef and F. Piessens. Practical verification of WPA-TKIP vulnerabilities. In ASIA CCS, 2013.
6. A. Joux. Authentication failures in NIST version of GCM. 2016.
7. J. Jonsson. On the security of CTR+ CBC-MAC. In SAC, 2002.

Countermeasures

Problem: many clients won’t get updates

Solution: AP can prevent (most) attacks on clients!

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

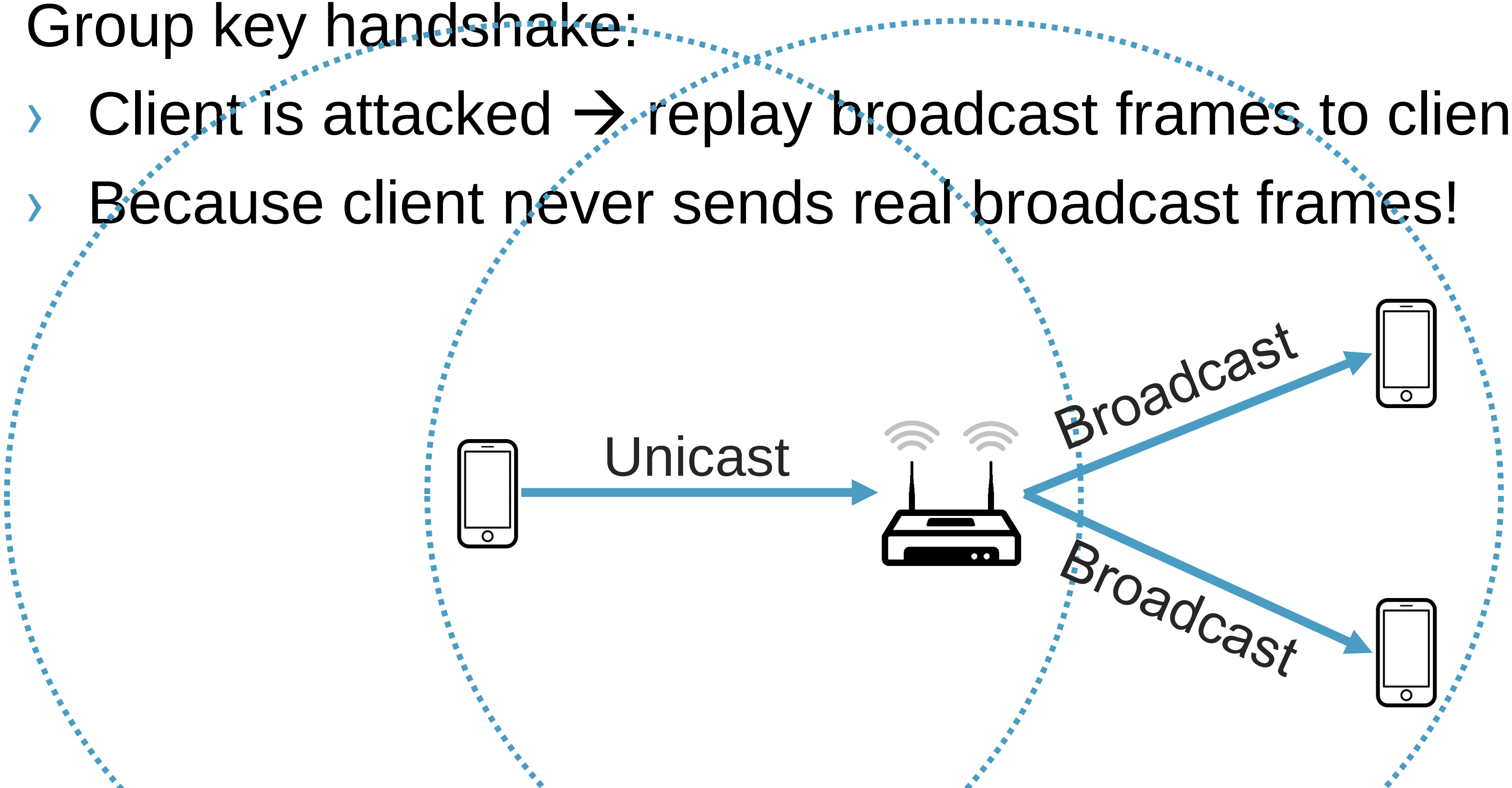
However:

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

Handshake specific

Group key handshake:

- Client is attacked → replay broadcast frames to client
- Because client never sends real broadcast frames!


Unicast
Broadcast
Broadcast