Vulnerability Disclosure (Lecture 11)

Prof. Mathy Vanhoef

DistriNet – KU Leuven – Belgium

KU LEUVEN DistriNet

Motivation: software is never 100% secure

- A **zero-day vulnerability**: vulnerability in hardware/software that the manufacturer of the product is not (yet) aware of
- A zero day in a widely used system is dangerous and powerful: can be used to break into and control that system
- Are valuable to various stakeholders:
  - Law enforcement / intelligence services to spy on users
  - Military for cyberweapons, criminals to build malware, …
- What if you/someone finds one?

Bug Bounties

Bug bounties

- Platform where users can report vulnerabilities
- Usually with cash reward for valid vulnerabilities

[Android and Google Devices Security Reward Program Rules:]

Code execution reward amounts

Description	Maximum Reward
Pixel Titan M with Persistence, Zero click	Up to $1,000,000
Pixel Titan M without Persistence, Zero click	Up to $500,000
Local App to Pixel Titan M without Persistence	Up to $300,000
Secure Element	Up to $250,000
Trusted Execution Environment	Up to $250,000
Kernel	Up to $250,000
Privileged Process	Up to $100,000

[KU Leuven / Online enrollment for students]

Low 0.1 - 3.9	Medium 4.0 - 6.9	High 7.0 - 8.9	Critical 9.0 - 9.4	Exceptional 9.5 - 10.0
€ 0	500	1,000	1,500	2,500

Bug bounty platforms

Each company/organization has a program that includes:

- **Rules of the program**, e.g., whether you are allowed to make your findings public once it has been fixed
- **Scope and limits**: which systems you are allowed to test
- **Safe harbor** language: protect those who disclosure vulnerabilities from legal action.

Bug bounty: best practices

- Do an internal audit first before starting a bug bounty program! You may otherwise get overwhelmed with reports.
- Internal employees are typically excluded.
- Reward amounts:
  - High enough to encourage submissions…
  - …but not too high? Internal employees might otherwise collude with bug hunters, experienced employees might quit and report bugs,…
- Have enough analysts to filter bug reports: public bug bounty platforms typically get a lot of invalid reports

Bug bounty variations I

Private bug bounty programs:

- Only selected researchers can report vulnerabilities
  - E.g., after finding bugs in public programs, after being invited,…
- Examples: zerocopter, Synack, Yogosha,…

HackerOne's Internet Bug Bounty

- For vulnerabilities in selected open-source software projects
- Sponsored by several companies to secure essential open-source software

Bug bounty variations II

Some companies provide their own bug bounties:

- Google VRP, Android, Chromium, etc.
- Facebook, Microsoft, …
- Apple Security Bounty (they have a reputation though)

Blockchain / crypto world:

- HackenProof: many crypto-related programs, but also others
- Immunefi: for a critical bug in Polygon, one researcher was awarded $2.2 million

Zero Day Initiative

Vendor-agnostic bug bounty program:

- Researcher provides info about vulnerability to ZDI. Including all details and a Proof-of-Concept (PoC)
- Info is analyzed, bug bounty offer is made
- If accepted, payment is made immediately before disclosure
- ZDI will then communicate with the affected vendor

Zero Day Initiative

Protection filters for ZDI customers are created & deployed while the bug is still being patched = zero-day protection

- To prevent leaks before disclosure, customers are given a generic description of the filter, not the vulnerability itself

List of disclosed vulnerabilities is available online:

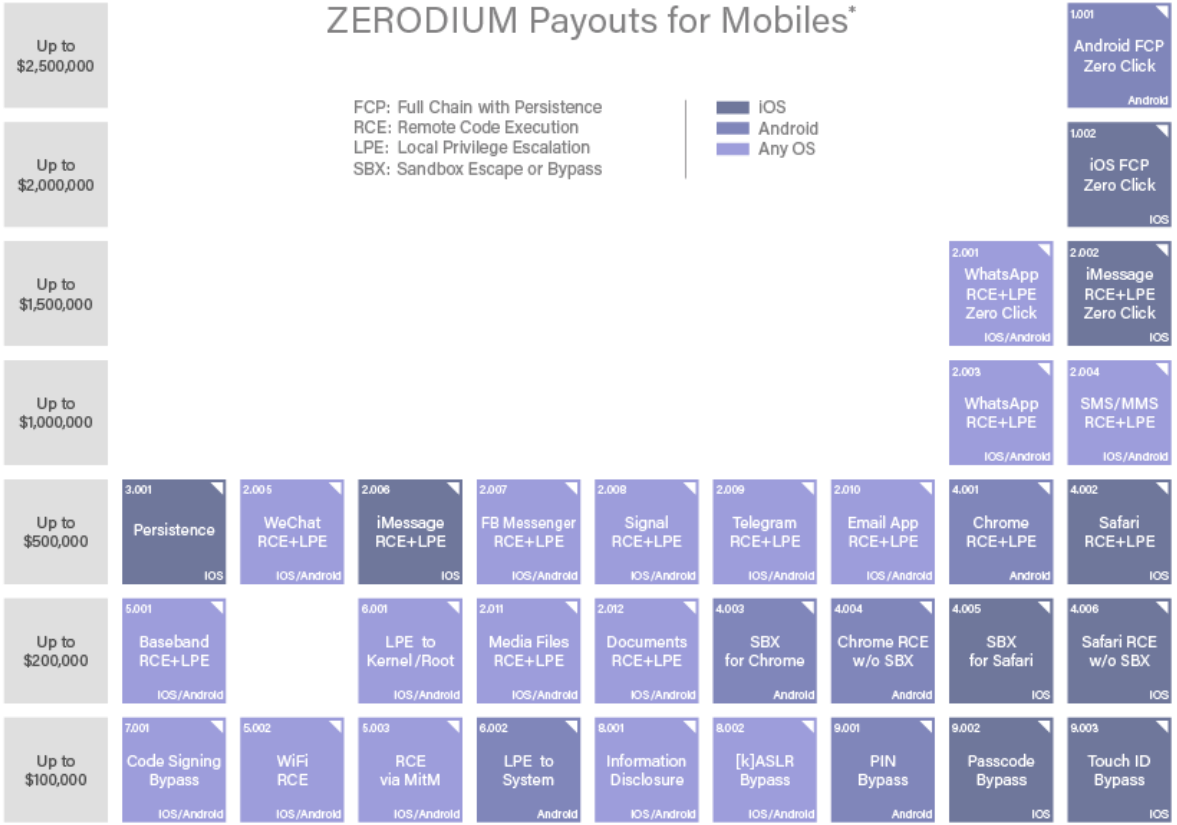
- https://www.zerodayinitiative.com/advisories/published/
- There's also a blog: https://www.zerodayinitiative.com/blog

Detectify

They focus on web technologies:

- Reported vulnerabilities are validated by them
- They create a module for Detectify so that customers will get a notification if they are vulnerable
- For as long as the module is live, you are rewarded for each unique hit in the Detectify customer base.
- For 0-days, they will frist contact the affected vendor, they do not scan for 0-days right away.

Or sell it as a weapon, e.g., Zerodium (2019)



Or sell it as a weapon, e.g., Zerodium

- Prices reflect how hard it is to compromise a device
- “Zerodium customers are government institutions (mainly from Europe and North America) in need of advanced zero-day exploits and cybersecurity capabilities.”
- Criticized by Reporters Without Borders for selling exploits to foreign governments that use them to spy on journalists

Not without risk

Maor Shwartz, worked as vulnerability broker:

- “I [..] got threatened to stop what I’m doing … I decided it wasn’t worth the risk and closed the company”
- “Effectively threatened in broad daylight at a café.”

Some companies may sue you when reporting vulnerabilities

- Volkswagen sued researchers at Radboud University to prevent them from disclosing vulnerability details
- Eventually made public, had to delete 1 sentence from paper

[1] “Selling 0-Days to Governments and Offensive Security Companies” by Maor Shwartz at Black Hat US, 2019

Other aspects

Competitions: Pwn2own, PWNoRAMA, Hack2Win, Zer0Fest

- Cash rewards if you can successfully exploit a product

Wassenaar arrangement:

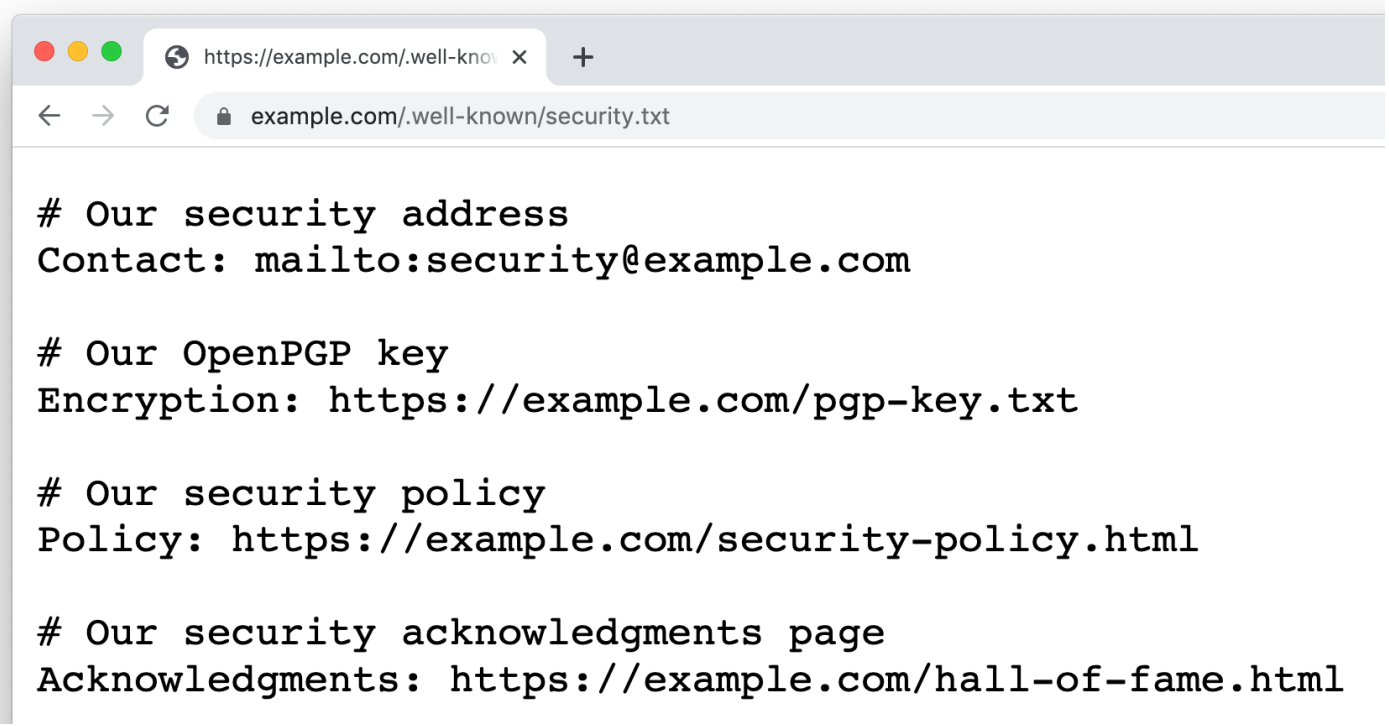
- Export controls for conventional arms and dual-use goods
- 0-day vulnerabilities themselves are *not* controlled
- But “the software to deliver 0-day exploits” *is* controlled
  - E.g., the command-and-control software
  - Exception for research tools, e.g., tools like Metasploit

This is not legal advice…

(Multi-Party) Coordinated Disclosure

1. Who to contact: single vendor

› See if they have a security.txt file on their website:



1. Who to contact: single vendor

- See if they have a security.txt file on their website:
- Look if their website has a **security contact** somewhere
- If they have a **bug tracker**: can you create private reports?
- See if they have a bug **bounty program**
- If all that fails, try to contact a developer directly…
- If all that fails, try to get help from a country's Computer Emergency Response Team (CERT)…
- … if everything fails, publicly disclose it.

1. Who to contact: multi-party disclosure

What if the vulnerability affects multiple vendors?

- For instance, you found a vulnerability in WPA2 ☺
- Contact companies individually? Gets very hard to manage..

Get help from dedicated organizations:

- CERT/CC can help with coordinated disclosures
- FIRST.org: companies can join FIRST to more easily share information about vulnerabilities (e.g., under NDA)

1. Who to contact: multi-party disclosure

How many organizations to include in the disclosure?

1. Contact all affected vendors
   - Higher chance of (un)intentional leaks
   - Higher chance that malicious actors learn about the vulnerability
2. Contact a selected set of companies
   - Who to select? Depends on specific case. Common criteria are big companies that have more experience and/or have most users.
   - Might be influenced by political landscape and international sanctions, e.g., FIRST.org suspended Russian members on March 25th, 2022

2. How to secure communicate?

With a single vendor:

- PGP is often suggested, but has bad usability in practice.
- Try to look for submission forms that support HTTPS.

If CERT/CC supports your multi-party disclosure:

- Use VINCE (Vulnerability Information and Coordination Environment) ≈ a secure forum where multiple organizations can post messages and share code

3. Referring to a vulnerability: CVE identifiers

MITRE maintains database of identifiers for vulnerabilities

- E.g., Heartbleed is assigned CVE-2014-0160

> *"The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle **Heartbeat Extension packets**, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a **buffer over-read, as demonstrated by reading private keys**, related to d1_both.c and t1_lib.c, aka the Heartbleed bug."*

3. Referring to a vulnerability: CVE identifiers

MITRE maintains database of identifiers for vulnerabilities

- E.g., Heartbleed is assigned CVE-2014-0160

How to request a CVE identifier?

- MITRE has an online form to request CVE identifier(s)
- Approved vendors can assign CVEs for their own software

A CVE gives no indication about a vulnerability's severity

- In practice, people (wrongly) associate a certain importance to getting CVEs. But this is, at least in general, not the case.

4. How long should the embargo be?

- CERT/CC: default is 45 days, but they say 90 is common too
- CERT Europe: 90 days
- Google Project Zero: 90 days, with an extra 30 days if a patch is made within the initial 90 days.

The above is tailored towards software vulnerabilities

- Hardware vulnerabilities can take longer to patch
- Longer embargo for complex supply chains, e.g., firmware by 3rd party that is then incorporated by downstream vendor(s)

5. Disclose limited info before full disclosure?

Can consider an early public warning

- E.g., OpenSSL notifies about upcoming disclosures:

> *"The OpenSSL project team would like to announce the forthcoming release of OpenSSL version 3.0.7.*
>
> *This release will be made available on Tuesday 1st November 2022 between 1300-1700 UTC.*
>
> *OpenSSL 3.0.7* ***is a security-fix release****. The highest severity issue fixed in this release is* ***CRITICAL****"*

Source: https://mta.openssl.org/pipermail/openssl-announce/2022-October/000238.html

Example:
WPA2 KRACK Attack

The 4-way handshake

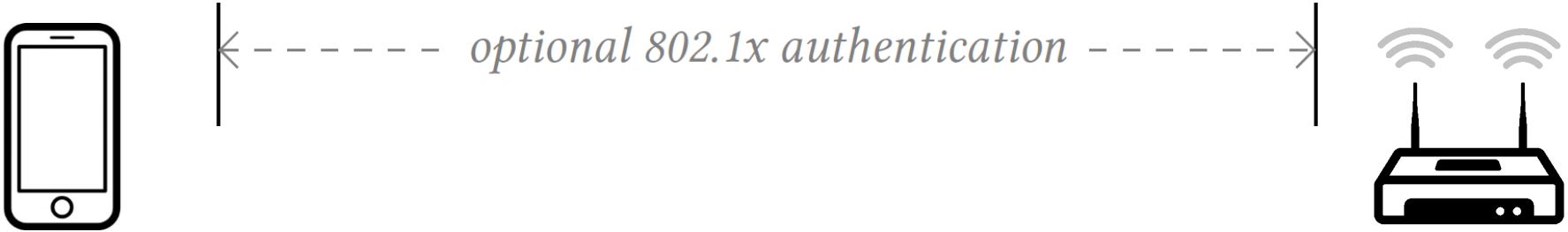
Used to connect to any protected Wi-Fi network

- Provides mutual authentication & negotiates fresh session key called the PTK (Pairwise Temporal Key)

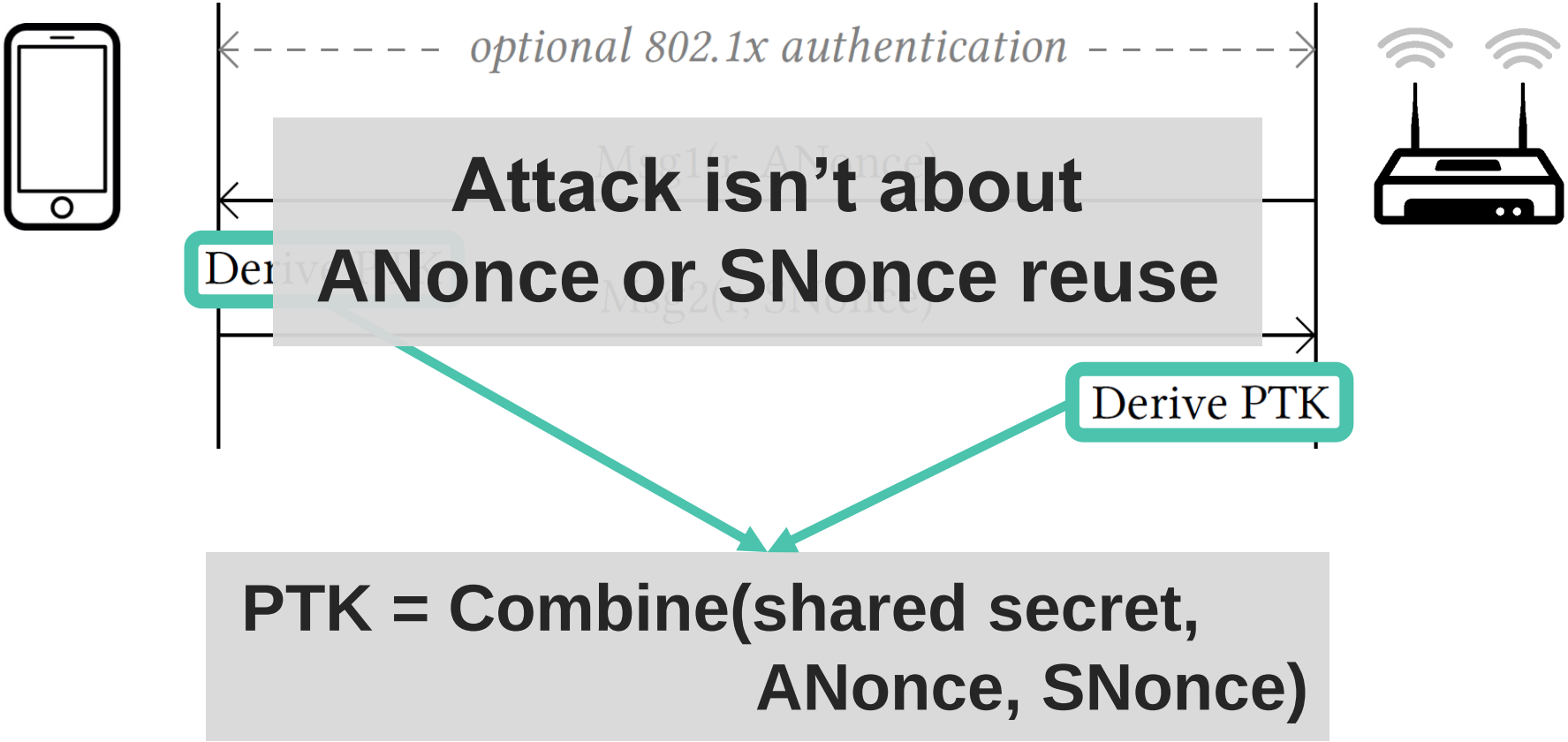
Appeared to be secure:

- No attacks in over a decade (apart from password guessing)
- Proven that negotiated key (PTK) is secret[1]
- And encryption protocol proven secure[2]

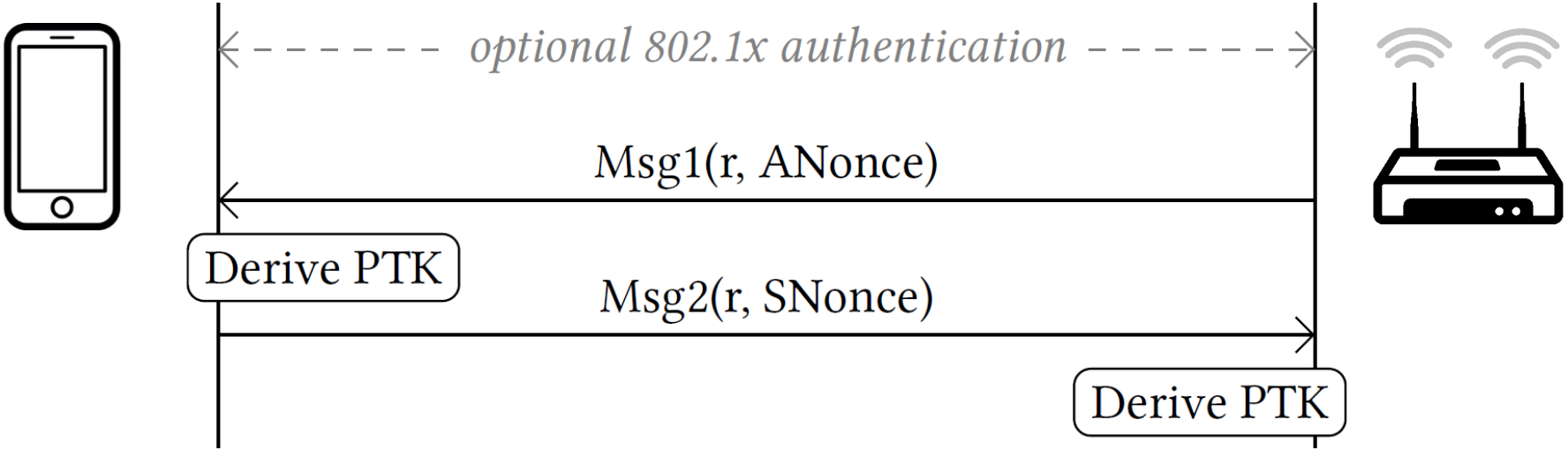
4-way handshake (simplified)



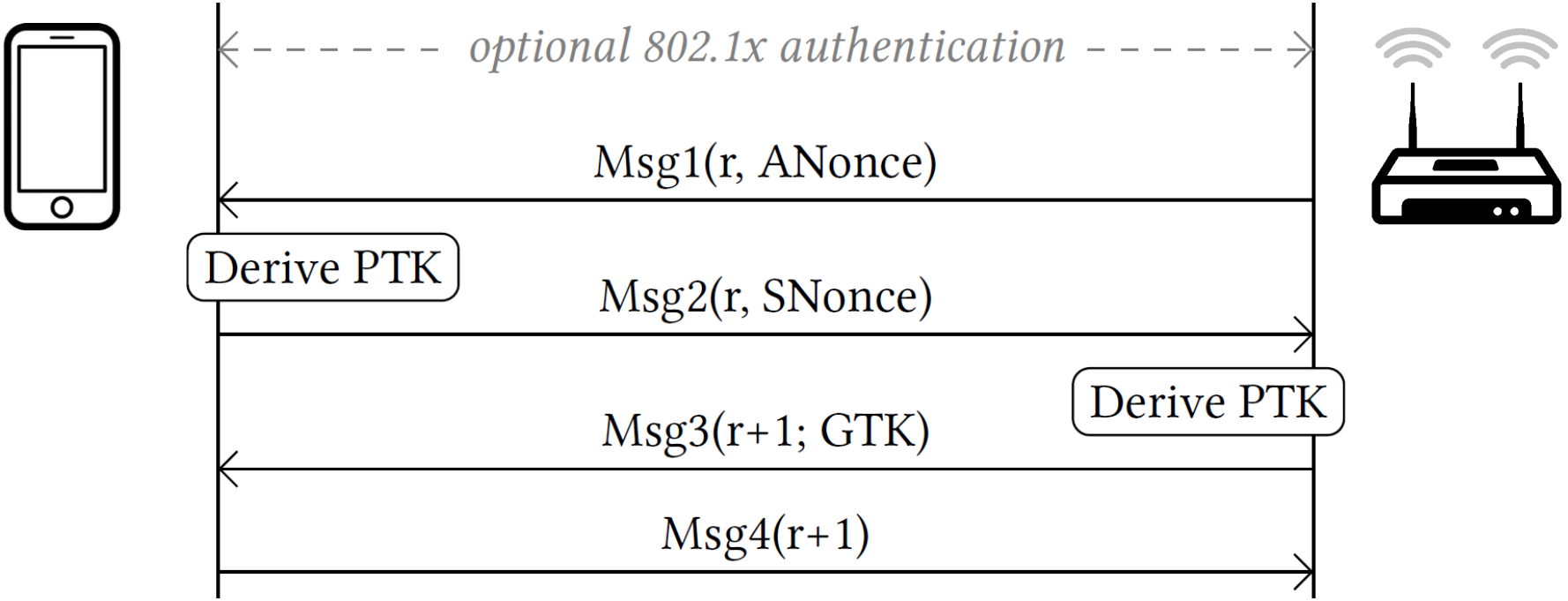
4-way handshake (simplified)



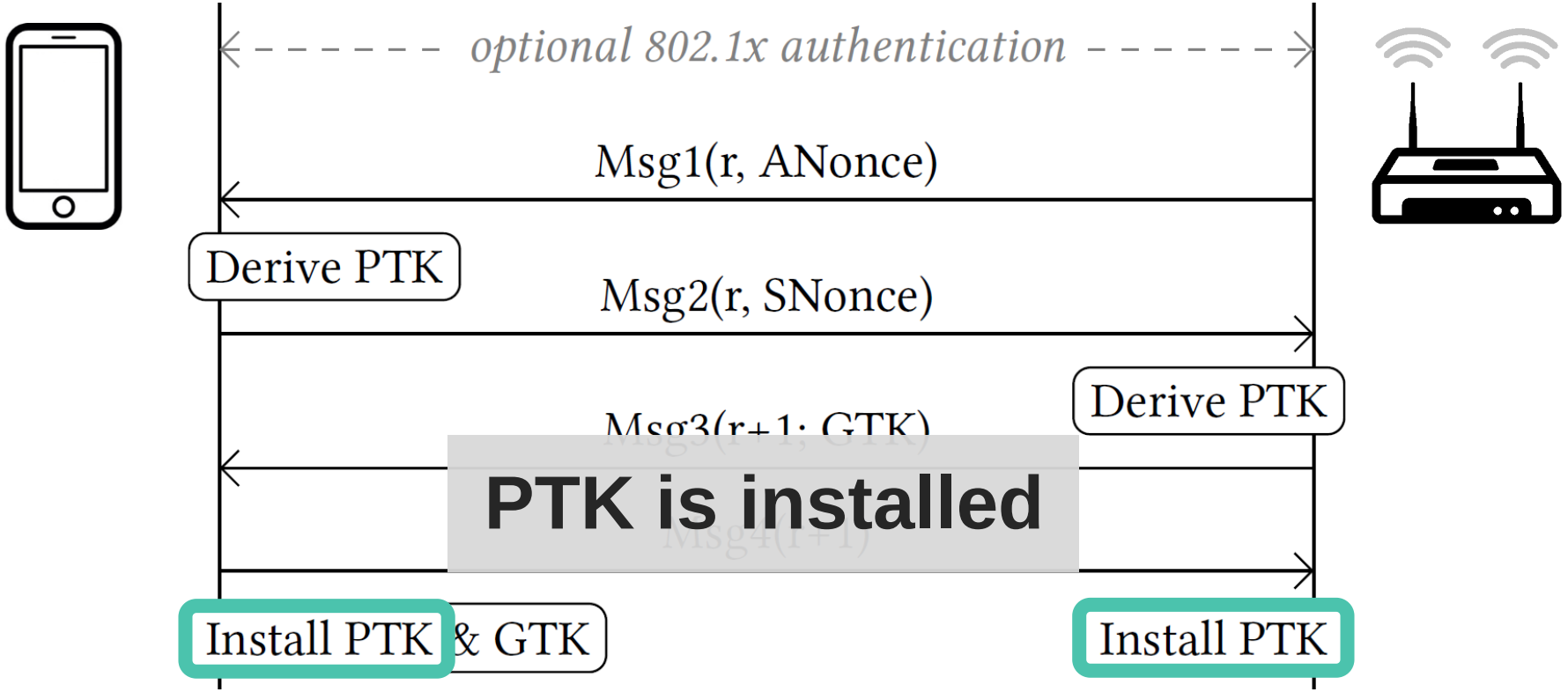
4-way handshake (simplified)



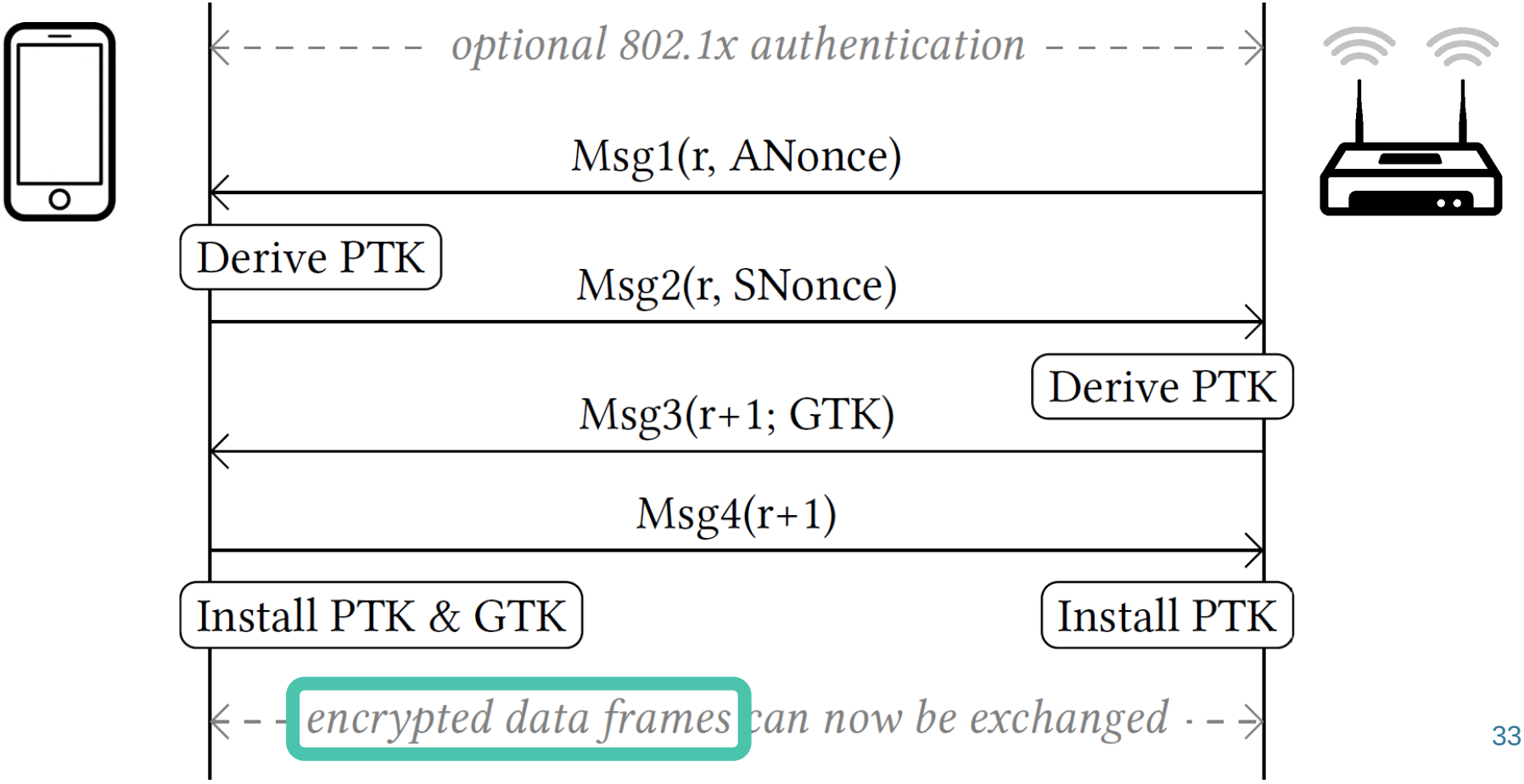
4-way handshake (simplified)



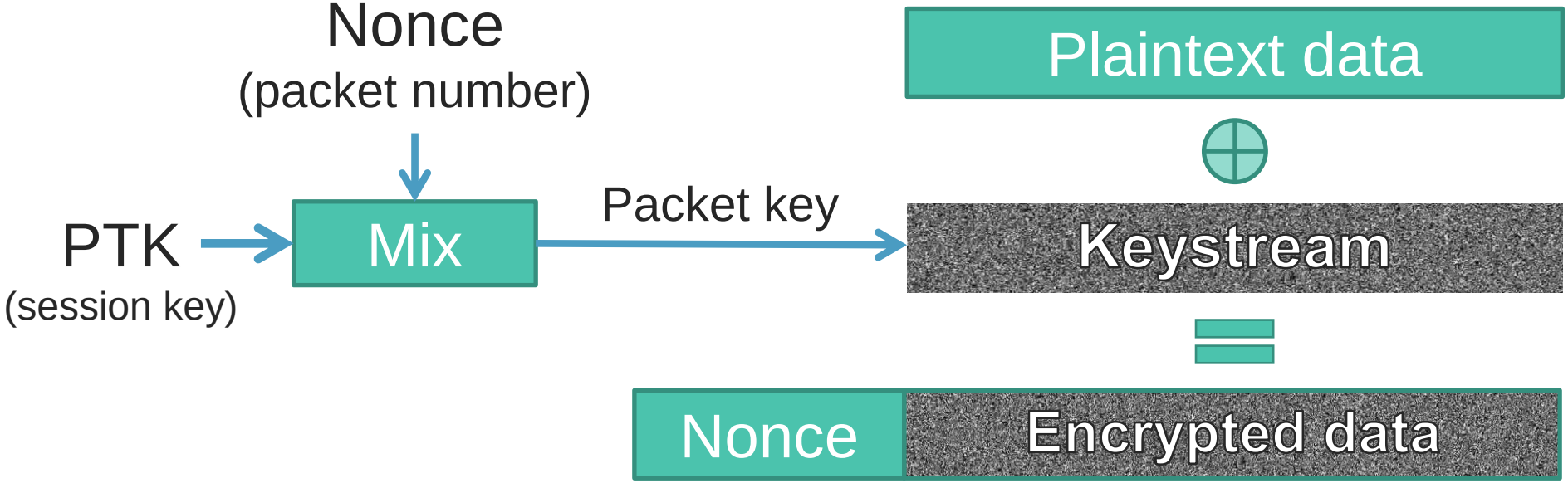
4-way handshake (simplified)



4-way handshake (simplified)

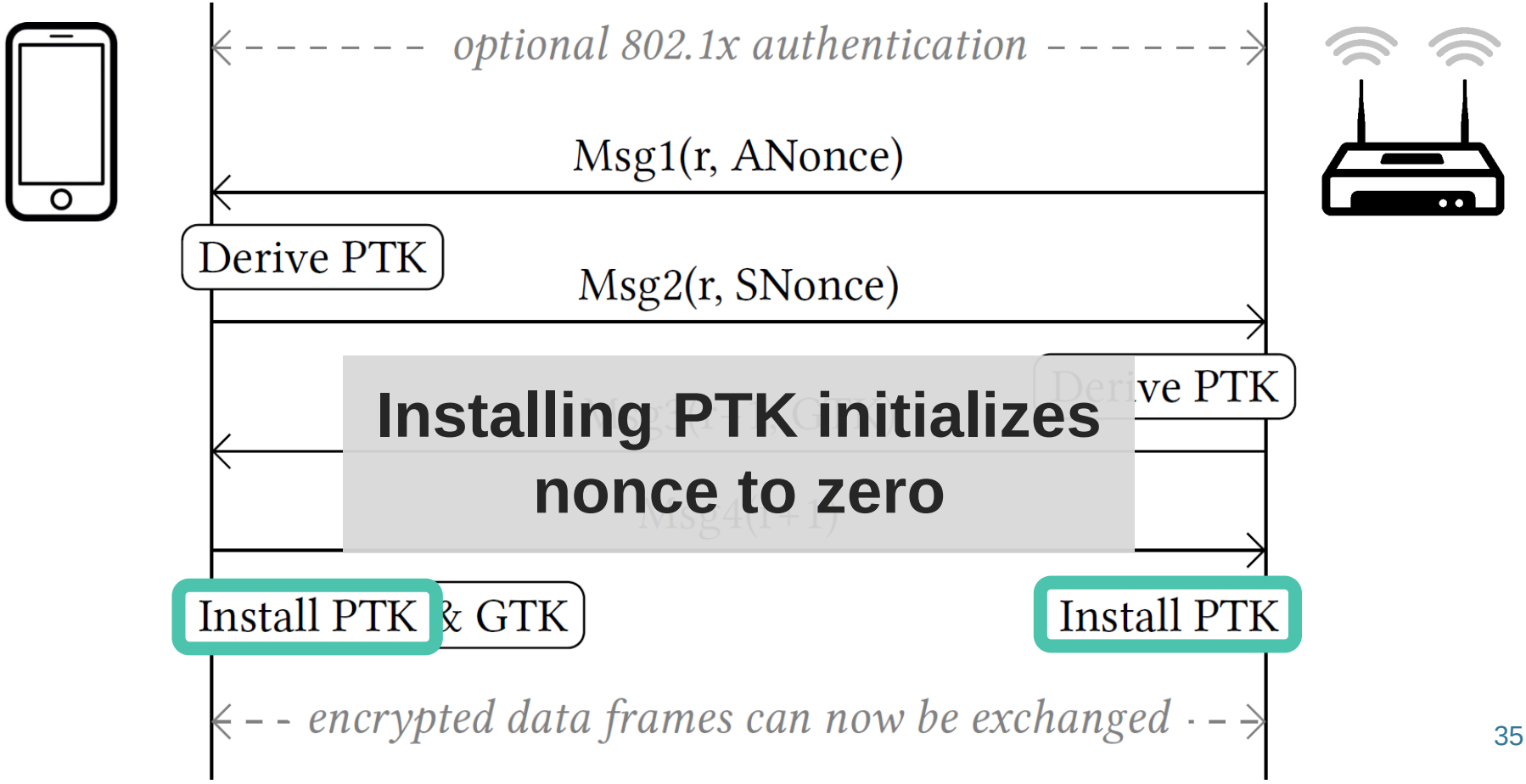


Frame encryption (simplified)



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

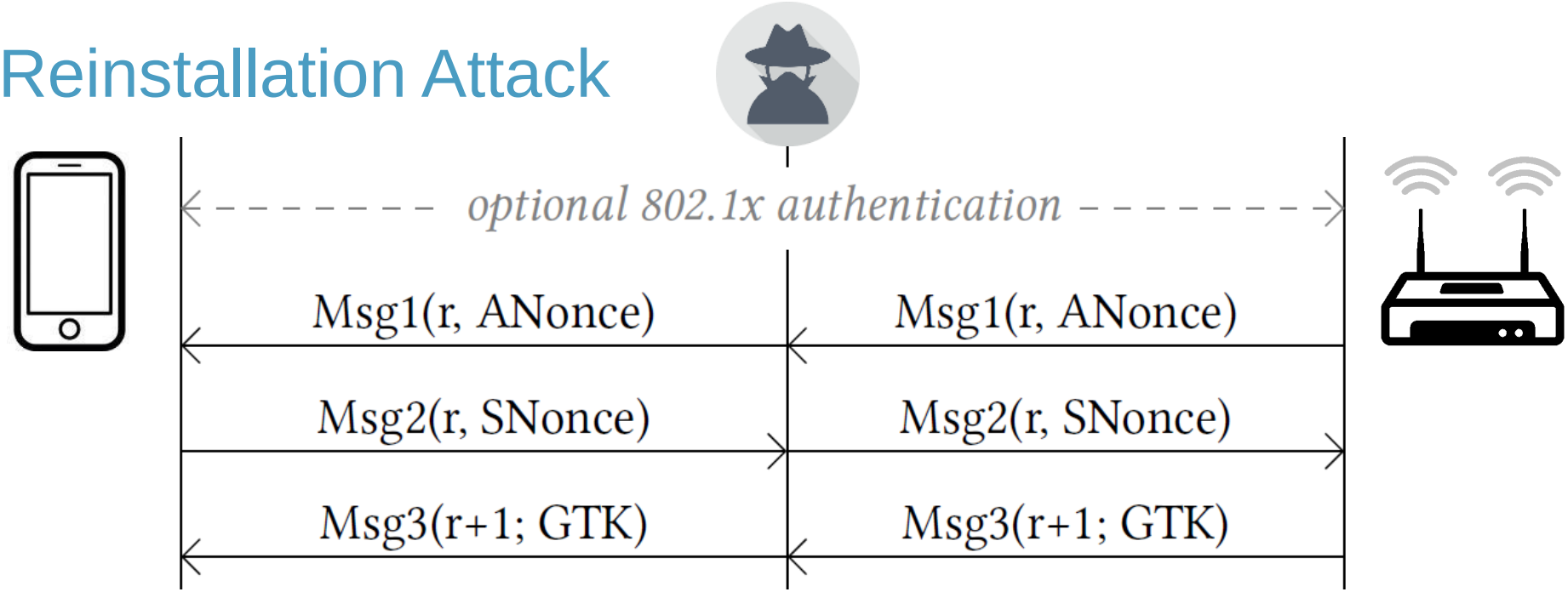
4-way handshake (simplified)



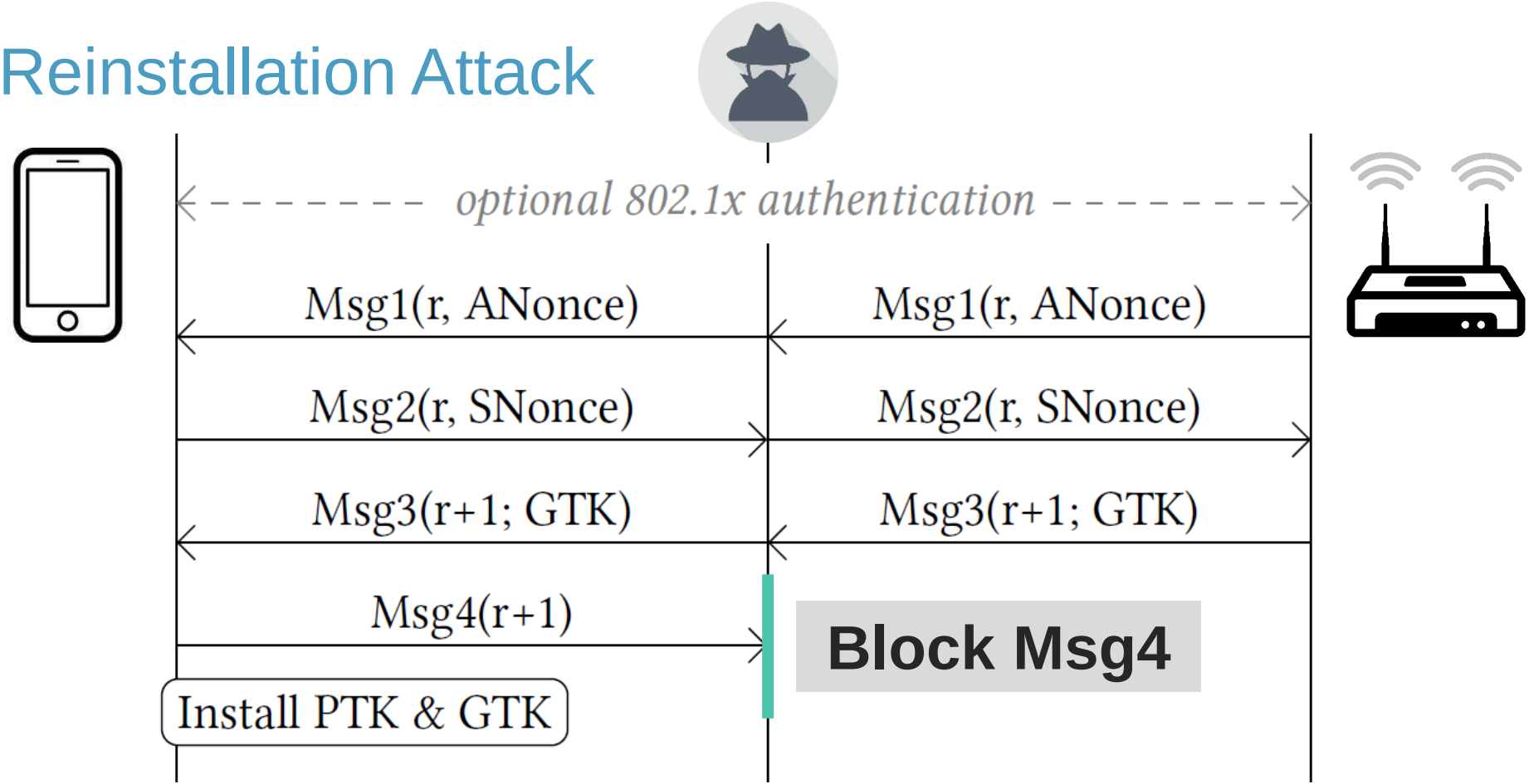
Reinstallation Attack

Reinstallation Attack

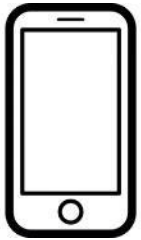
Reinstallation Attack

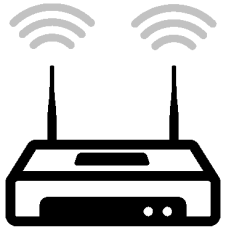


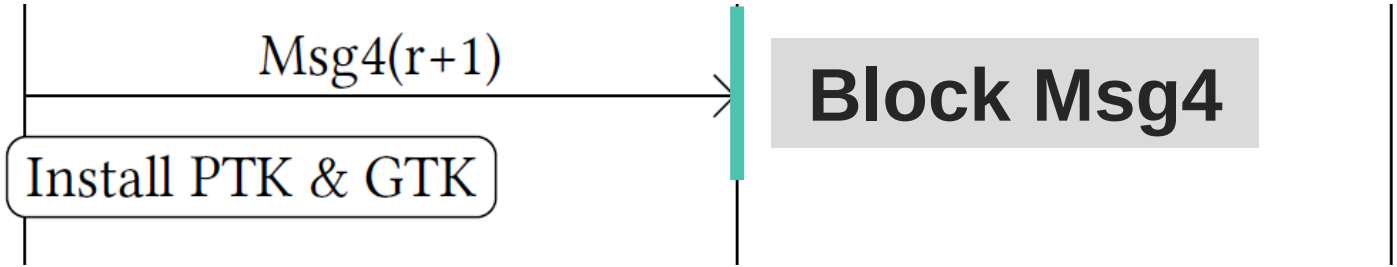
Reinstallation Attack



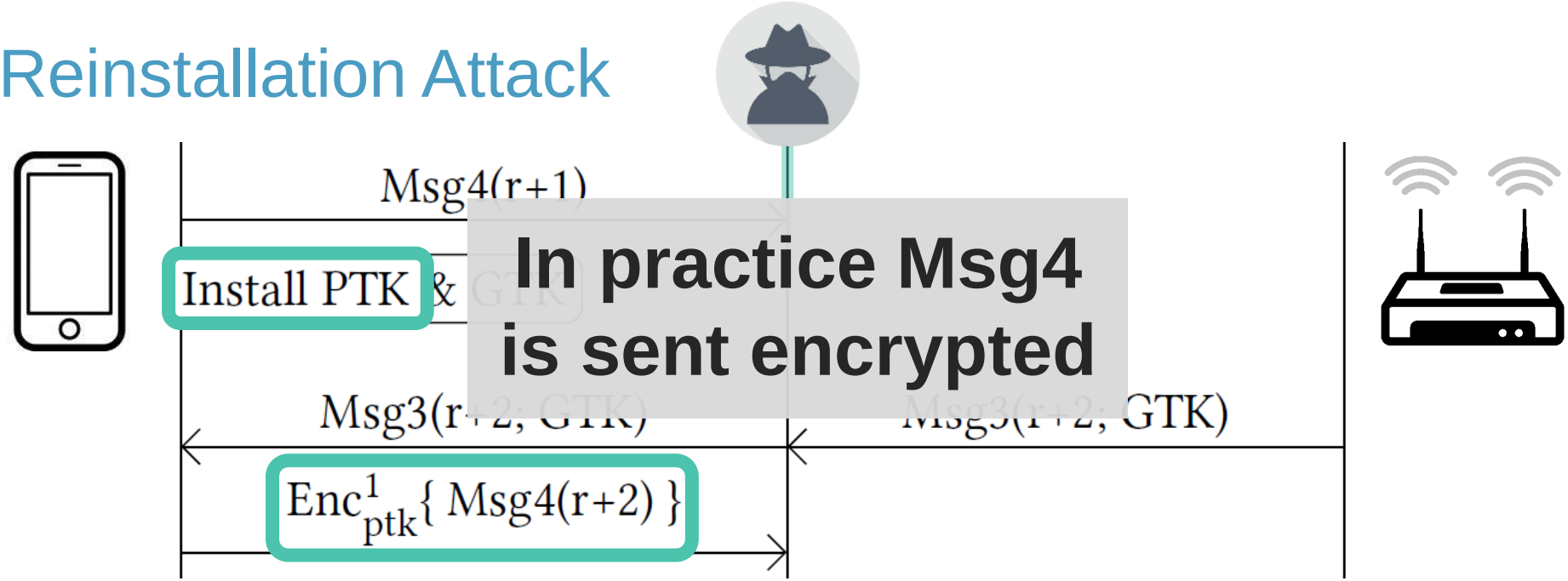
Reinstallation Attack





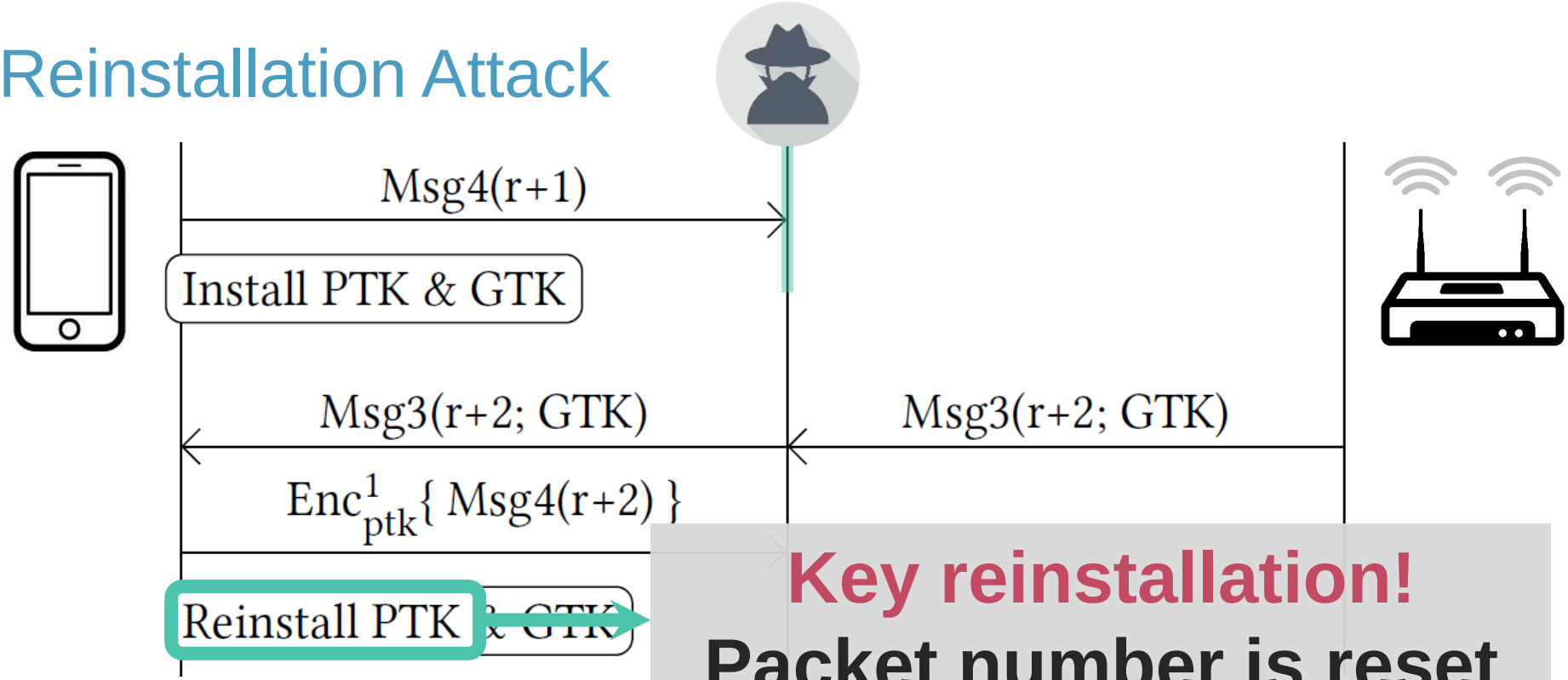


Reinstallation Attack



In practice Msg4 is sent encrypted

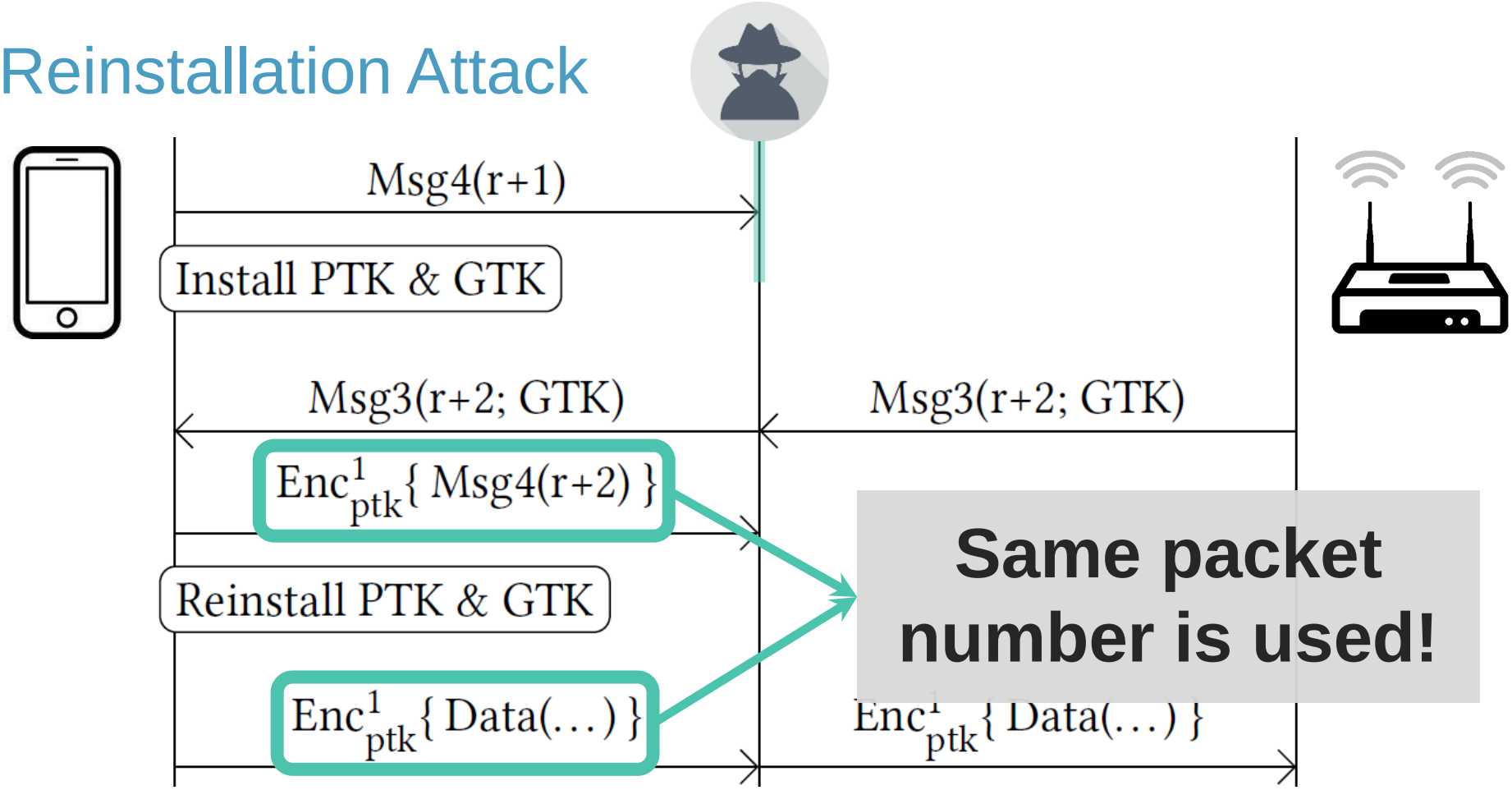
Reinstallation Attack



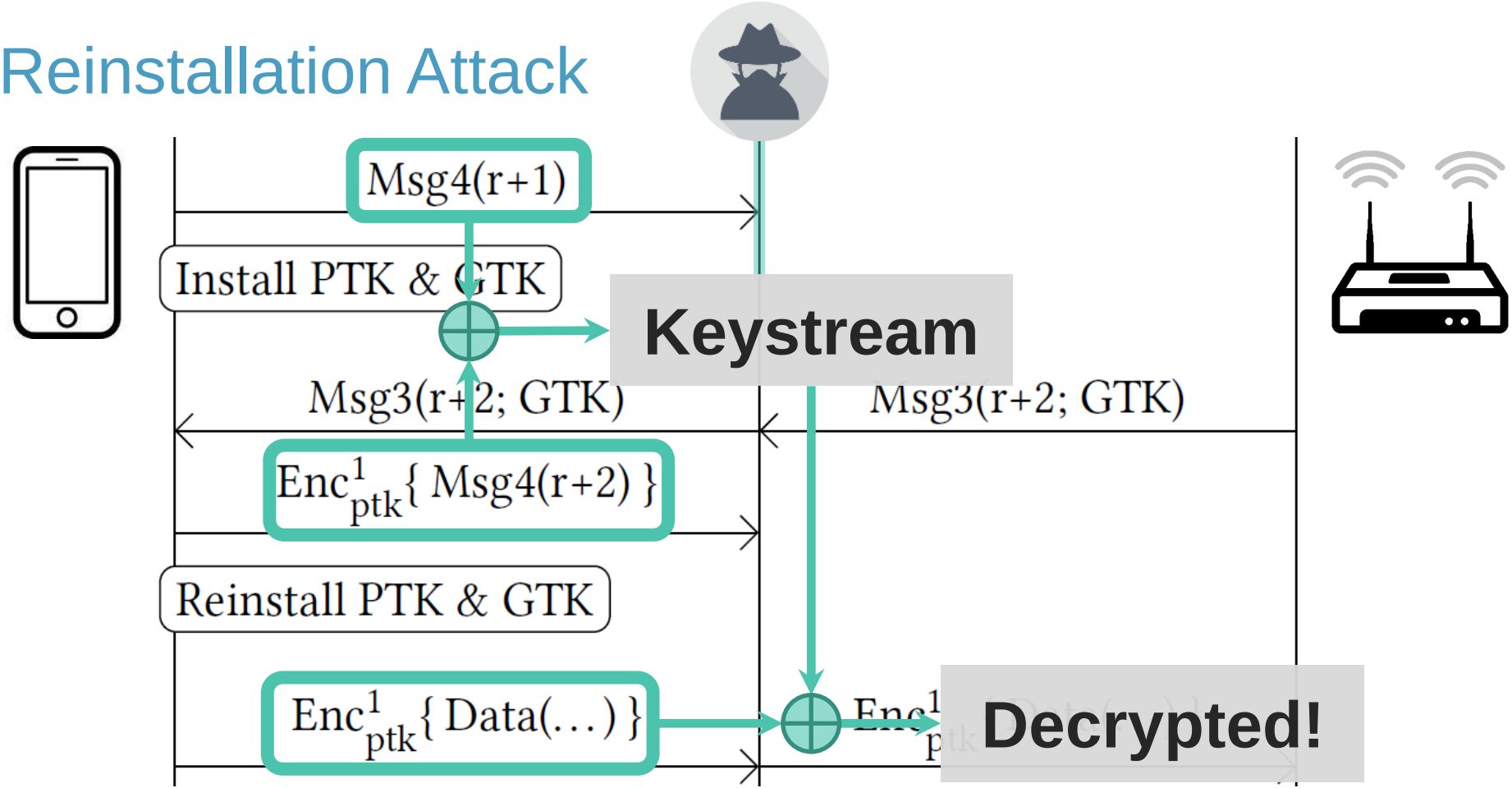
Key reinstallation!

Packet number is reset

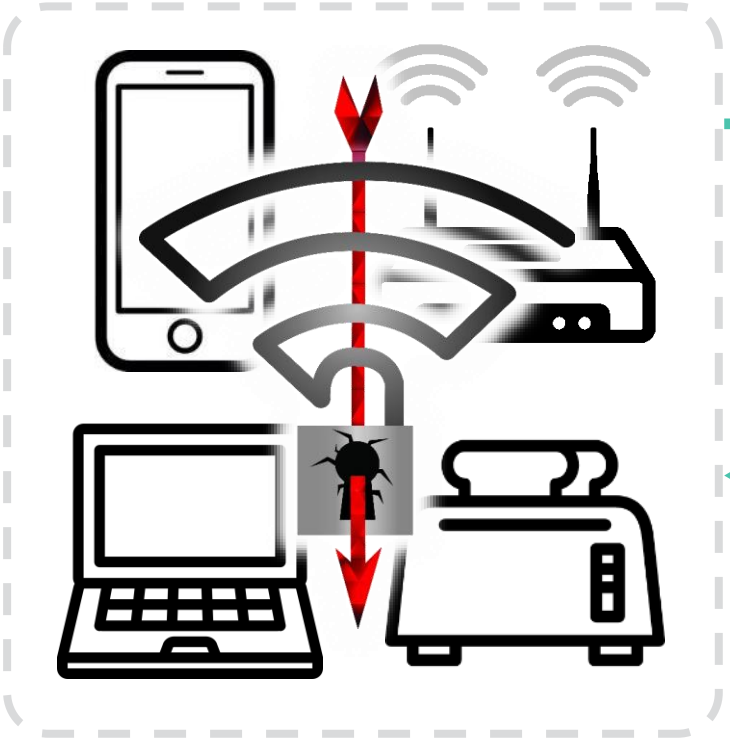
Reinstallation Attack



Reinstallation Attack



General impact



Transmit packet number reset

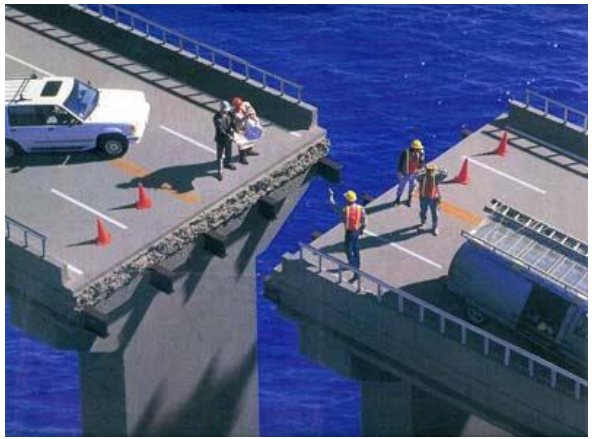
Decrypt frames sent by victim

Receive replay counter reset

Replay frames towards victim

Limitations of formal proofs

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



The combination was not proven secure!

Disclosure process of KRACK

- Disclosed with help of CERT/CC, ICASI, and Wi-Fi Alliance
  - ICASI is now part of FIRST.org
- Embargo of 49 days. Non-trivial, affected all Wi-Fi devices. Often required firmware patches (complex supply chain).

Current status:

- Certified Wi-Fi devices are tested against the KRACK attack
- The 802.11 standard that underpins Wi-Fi was updated
- The Wi-Fi Alliance now has a form to report vulnerabilities ☺

The end ☺

Optional reading / more information:

- How to request CVEs
- Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure