# Protocol Integration and Implementation Problems

## Bringing PQC into practice

Stefan-Lukas Gazdag
<Stefan-Lukas_Gazdag@genua.eu>

5th ENISA/FORTH Summer School, 27th of September 2018

# Content

Introduction

The Challenges

Real-World Example: IKEv2

Real-World Example: Hash-Based Signatures

# Anecdote

# Anecdote

*How troublesome is it to integrate quantum-safe algorithms into protocols and applications and what time will it take to so?*

# Anecdote

*How troublesome is it to integrate quantum-safe algorithms into protocols and applications and what time will it take to so?*

*I'm not really into that topic but I guess it should be fairly easy. I assume that protocols are designed in a modular way so you simply got to exchange algorithms. Therefore it shouldn't take too much effort.*

# Protocols and Implementations

A little glossary for this presentation:

- *Protocols* describe how to communicate and how to handle data.
- *Implementations* are software instantiations of protocols.
- *Libraries* provide functionality for other software.

# How problems arise

Widespread communication (Internet) is a laboratory experiment gone wild.

# How problems arise

Widespread communication (Internet) is a laboratory experiment gone wild.

A lot of old technology mixed with new / updated protocols.

# How problems arise

Widespread communication (Internet) is a laboratory experiment gone wild.

A lot of old technology mixed with new / updated protocols.

A standard or documentation is more like guidelines rather than rules.

# How problems arise

Security-by-design remains a dream.

# How problems arise

Security-by-design remains a dream.

It's got to work first, then we can start thinking about security.

# How problems arise

Compatibility!!1!!eleven!!

# How problems arise

Compatibility!!1!!eleven!!

Optimization vs. modularity

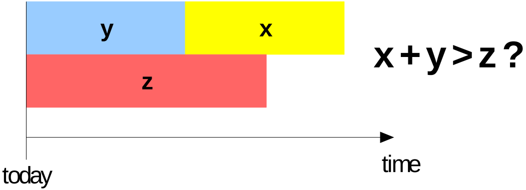
# Use Cases (not exhaustive)

- Digital signatures
  - Software updates / code signing
  - E-mail signatures
- Secure communication
  - Websites (online banking, ...)
  - Remote work
- Securing data
  - Passports / IDs
  - Health data
  - Payment data
- ...

# Tempus fugit

How soon do we need to worry?
(Michele Mosca, University of Waterloo)

- How long do you need encryption to be secure? ($x$ years)
- How much time will it take to re-tool the existing infrastructure with large-scale quantum-safe solution? ($y$ years)
- How long will it take for a large-scale quantum computer to be built (or for any other relevant advance)? ($z$ years)

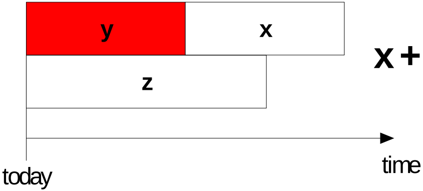


**x+y>z?**

# Tempus fugit

How soon do we need to worry?
(Michele Mosca, University of Waterloo)

- How long do you need encryption to be secure? ($x$ years)
- How much time will it take to re-tool the existing infrastructure with large-scale quantum-safe solution? ($y$ years)
- How long will it take for a large-scale quantum computer to be built (or for any other relevant advance)? ($z$ years)



$x + y > z$?

# The Challenges

# Check / To Do list (very high-level)

- Academic research
  - Building schemes
  - Optimization
  - Cryptanalysis, ...
- Standardization
- Practical Experience (secure usage, side channels, ...)
- Guidelines
- Integration in protocols
- Integration in software / libraries
- Widespread use

# Who wants to join the conversation?

- Academia
- Agencies
- Implementors / manufacturers
- Users / companies
- Standardization stakeholders
- Patent trolls

# What do we face?

Data size and timing demands

- Constraints in protocols
- Limitations in implementations
- Often depends on use case
- Sometimes depends on user requirements

# What do we face?

Brave New World

- Security Proofs / Quantum Setting
- New application
- Starting out with practical experience

# What do we face?

Complexity

- Quite often implementors and users are no experts in cryptography
- Correct use of cryptographic schemes isn't trivial and failures may not be obvious

# What do we face?

Debates on principles

- Are new crypto schemes necessary?

  (Meaning any new scheme. Why use SHA-3?)
- Crypto agility:
  - Who's gonna test it?
  - Again: Who needs it?
  - It's all just overhead...

# Real-World Example: IKEv2

# Virtual Private Network: IPsec using IKEv2

Internet Protocol Security (IPsec)

- Suite for secure communication
- A secure *tunnel* to send data through
- Symmetric crypto only

# Virtual Private Network: IPsec using IKEv2

Internet Protocol Security (IPsec)

- Suite for secure communication
- A secure *tunnel* to send data through
- Symmetric crypto only

Internet Key Exchange Version 2 (IKEv2)

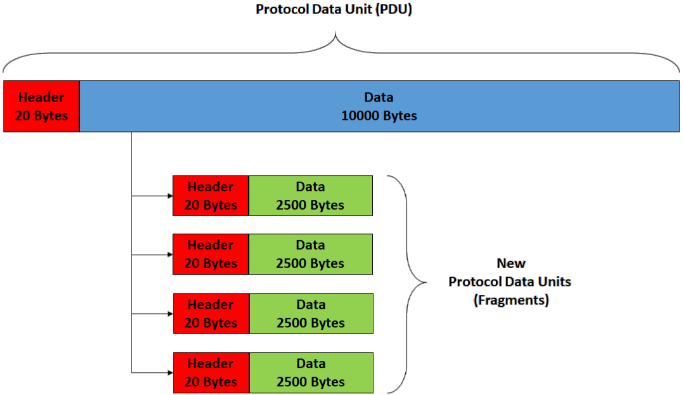
- Key Exchange Protocol
- Diffie-Hellman key exchange
- Authentication

# Maximum Transmission Unit

Maximum data size (frame) to send data from machine to machine until you reach the actual recevier.

Imagine a machine on the way to your destination that is so old, it can't handle forwarding a single packet that is too big with no alternative route.

# Fragmentation



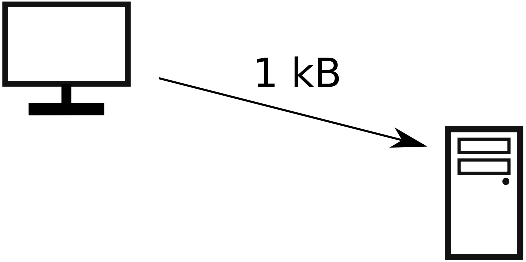
# Fragmentation

Important feature to split big data in suitable smaller chunks

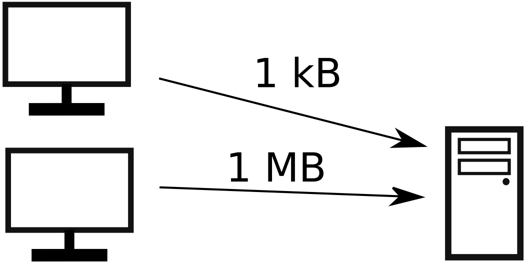
- IP fragmentation avoided in practice
- Some boxes drop fragmented packets

  ⇒ Fragmentation handled by higher protocols
- (Initial) Packets need to fit the MTU

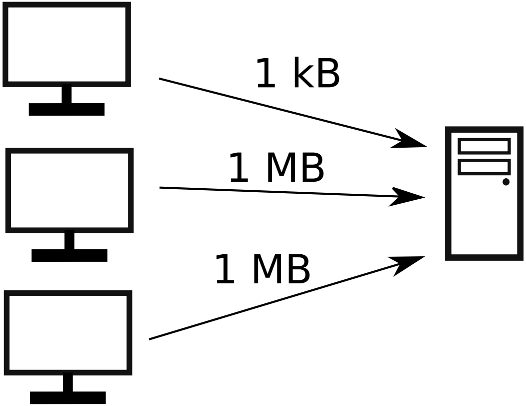
# Denial of Service



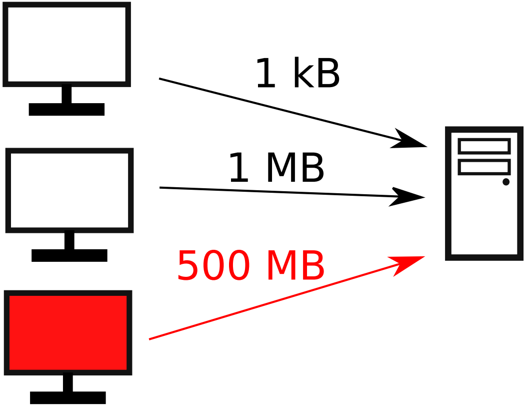
# Denial of Service



# Denial of Service

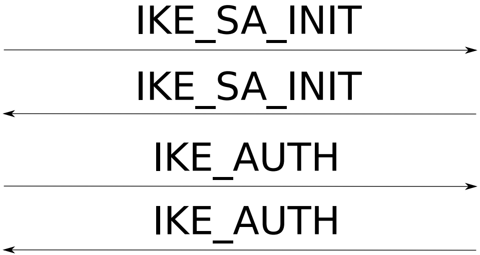


# Denial of Service



# Approaches

Classic IKEv2



IKE_SA_INIT

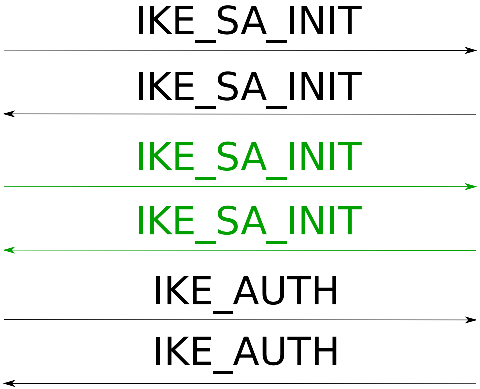
IKE_SA_INIT

IKE_AUTH

IKE_AUTH

# Approaches

Hybrid Key Exchange (draft-tjhai-ipsecme-hybrid-qske-ikev2), e.g.:



IKE_SA_INIT

IKE_SA_INIT

IKE_SA_INIT

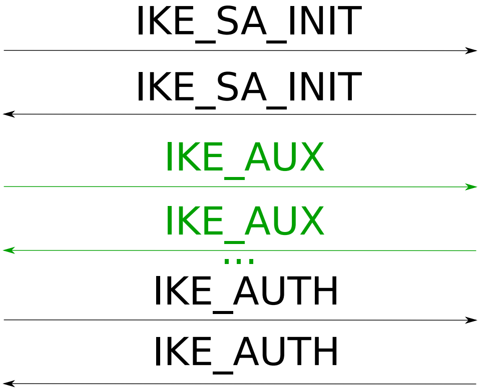
IKE_SA_INIT

IKE_AUTH

IKE_AUTH

# Approaches

Auxiliary Exchange (draft-smyslov-ipsecme-ikev2-aux)



IKE_SA_INIT

IKE_SA_INIT

IKE_AUX

IKE_AUX

...

IKE_AUTH

IKE_AUTH

# IKEv3

I like IKEv3

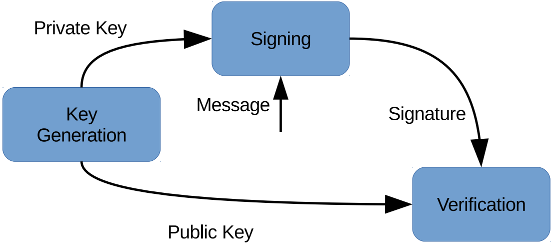
# Real-World example: Hash-Based Signatures

# Stateful Hash-Based Signatures

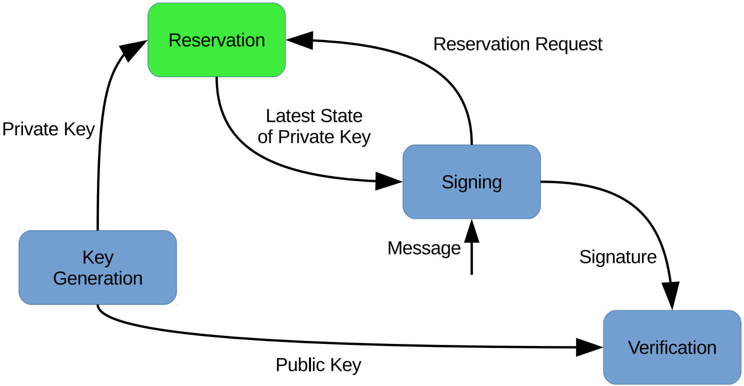
Some hash-based schemes have a state.

- Secret key becomes critical resource!
- Copies of the key may leak old state!

# Classical Signatures



# Reservation Approach



Reservation

Reservation Request

Private Key

Latest State
of Private Key

Signing

Key
Generation

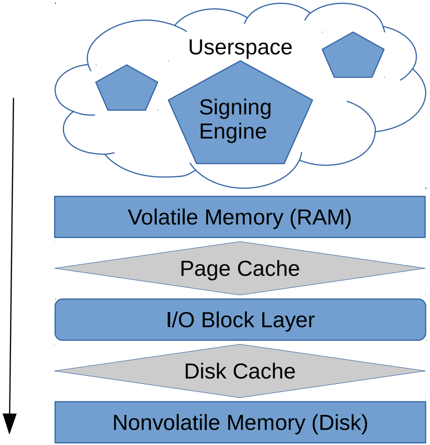
Message

Signature

Verification

Public Key

# Writing the key



# Consequences

- Software was never expected to support this
  ⇒ Software has to be altered / updated
- Key management required
  ⇒ totally different approach and security anchor necessary
- Each different system / architecture may have different requirements
- Different approaches for different scenarios

# Conclusion

- Integration of PQC in protocols and software is not trivial

# Conclusion

- Integration of PQC in protocols and software is not trivial
- We can handle that!

# Conclusion

- Integration of PQC in protocols and software is not trivial
- We can handle that!
- But we need:
  - More experience
  - The will to change the status-quo
  - Time (also meaning effort and money)

# Questions?

Stefan-Lukas_Gazdag@genua.eu