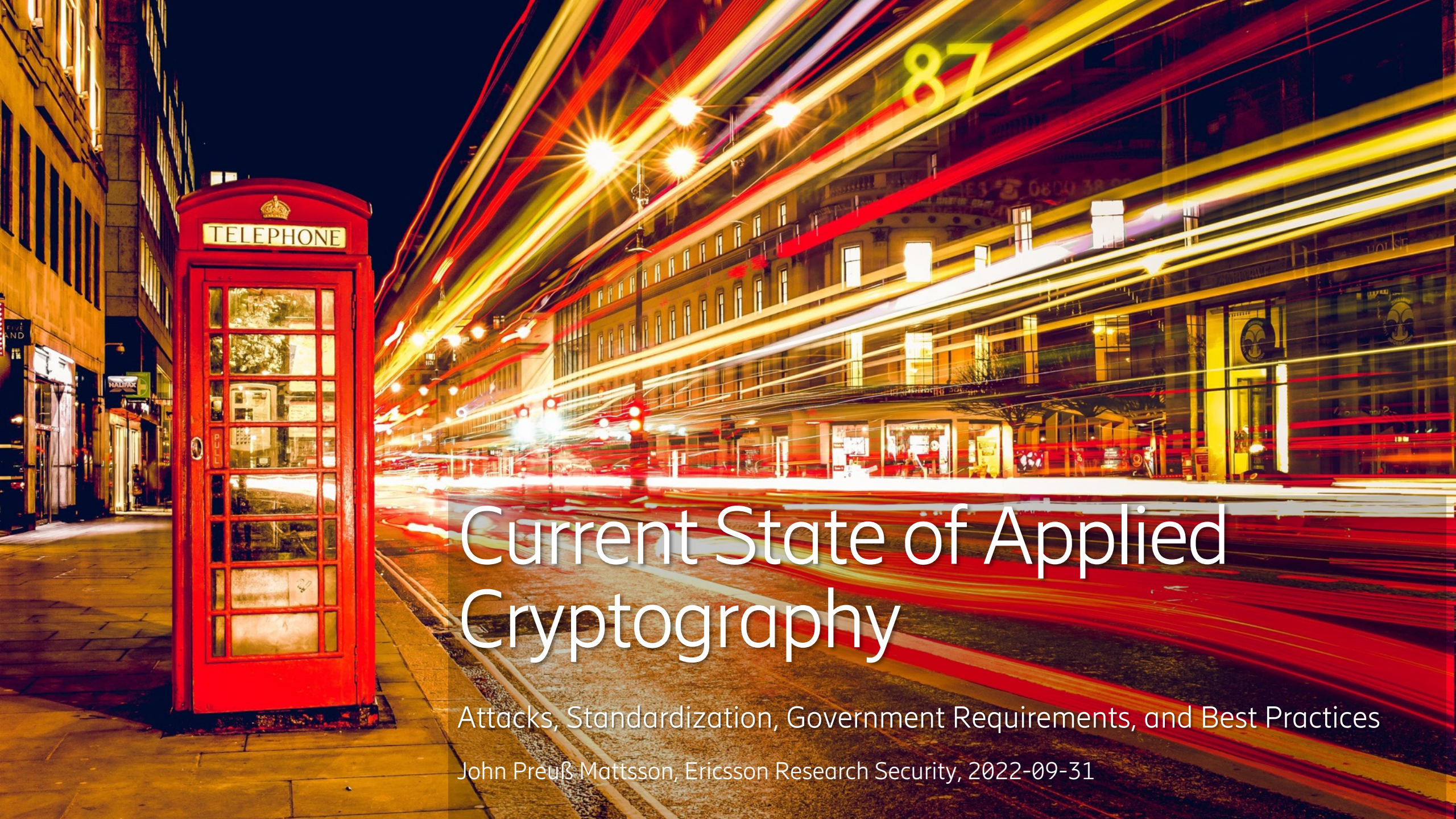
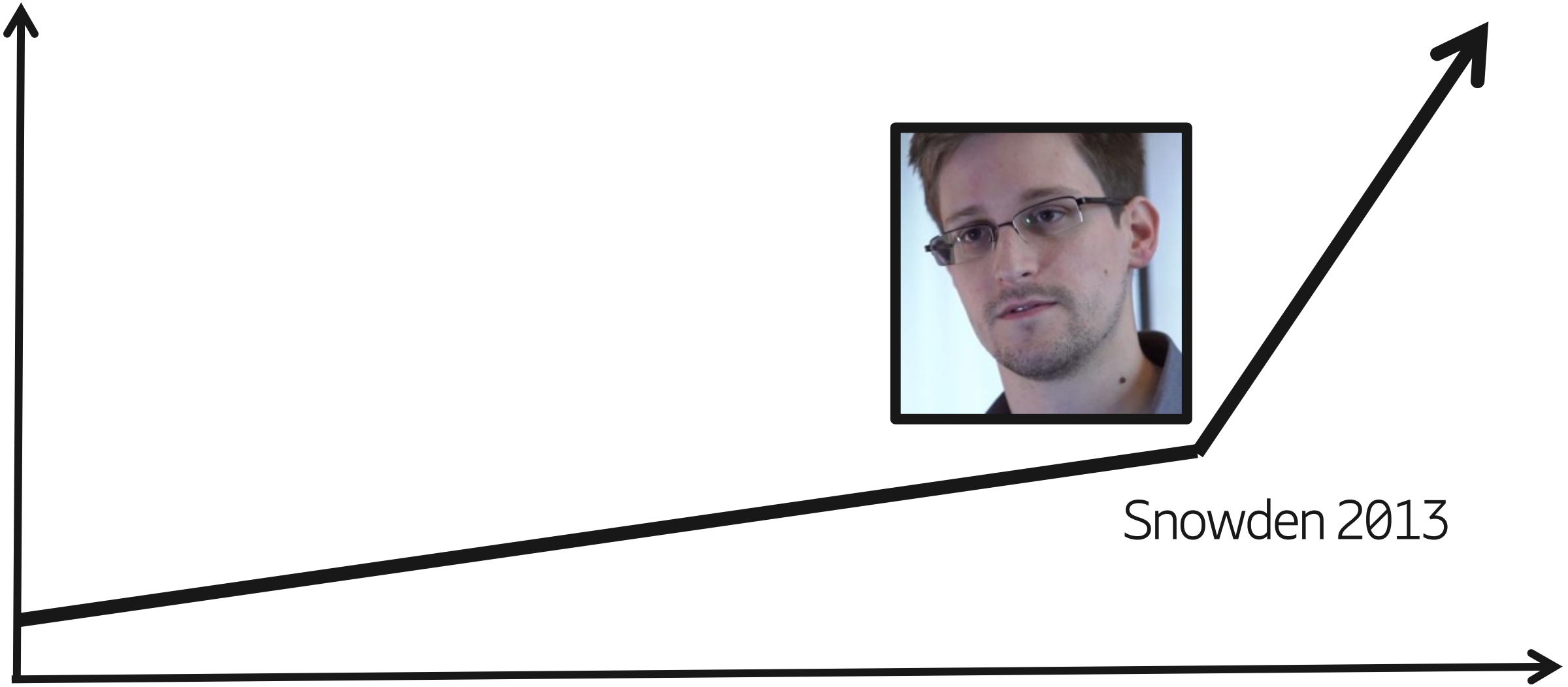


Current State of Applied Cryptography

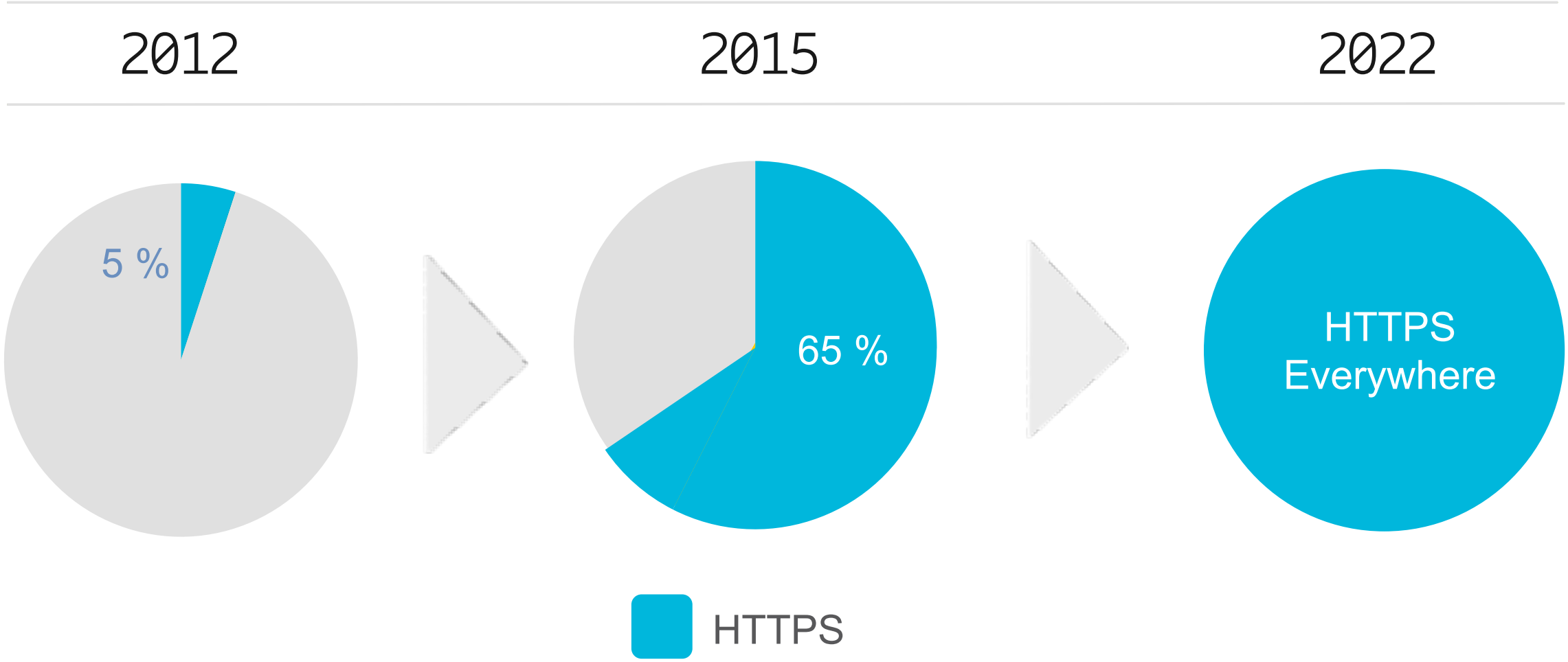
Attacks, Standardization, Government Requirements, and Best Practices

John Preuß Mattsson, Ericsson Research Security, 2022-09-31

Interest in Security and Privacy

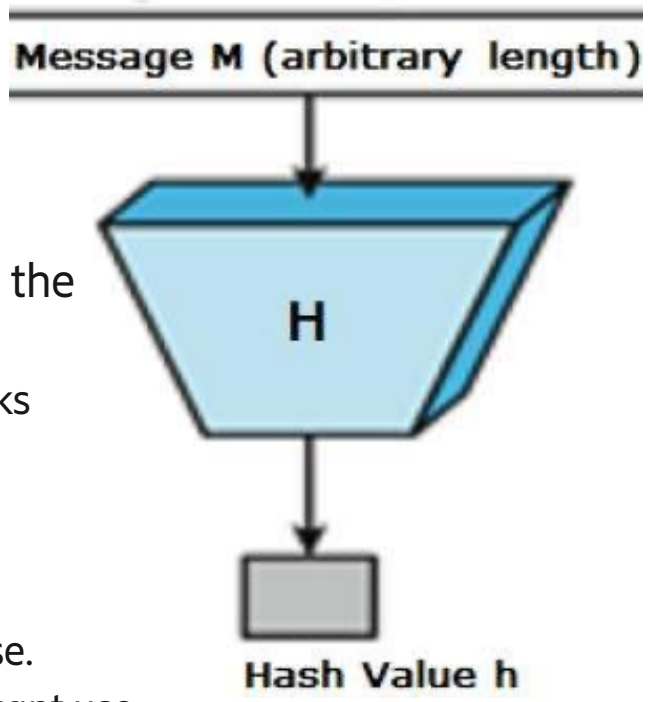


Security and Privacy Everywhere



Hash Functions

Hash functions (including keyed and variable length)



- Researchers have computed actual collisions for SHA-1.
  - No practical application on HMAC-SHA1. NIST is planning to disallow all uses of SHA-1.
  - MD5 and SHA-1 is being removed from standards and deployments (3GPP, TLS).
- SHA-2 (SHA-256, SHA-512/256, SHA-384, SHA-512) remains strong and is currently the default in most deployments.
  - SHA-256, SHA-384, SHA-512, but not SHA-512/256 is vulnerable to length extension attacks
    - Given Hash(m1) an attacker can calculate Hash(m1 || m2) for a any m2 …
- SHA-3 comes in a lot of variants:
  - SHA3-256, SHA3-384, SHA3-512, the drop-in replacements for SHA-2 are not seeing any use.
  - The more efficient SHAKE128 and SHAKE256 with variable length output are seeing significant use.
  - As SHA-3 is not vulnerable to length extension attacks, HMAC is overkill, and the more efficient KMAC128 and KMAC256 can be used instead. Simplified description below:
    - HMAC(K, m): H(K || H(K || m)) -> output width bytes (32 for SHA-256)
    - KMAC(K, m, L): H(K || m, L) -> L bytes

Authenticated encryption and AEAD

Authenticated Encryption and AEAD

- Authenticated encryption with Associated Data (AEAD) such as AES-GCM, AES-CCM, and ChaCha20-Poly1305 is taking over in standards and deployments. Several different security reasons + Performance.

- Before AEAD, the most used algorithms where 3DES or AES in CBC mode (for confidentiality) and MD5 or SHA-1 in HMAC mode (for integrity protection) in one of three compositions:
  - MtE ENC(P || MAC(P)) used in TLS and 3GPP PDCP
  - E&M ENC(P) || MAC(P) used in SSH
  - EtM ENC(P) || MAC(ENC(P)) used in IPsec, SRTP, and 3GPP NAS

- Turns out that CBC mode (irrespectively of padding) have significant weaknesses when used alone or in MtE and E&M compositions.

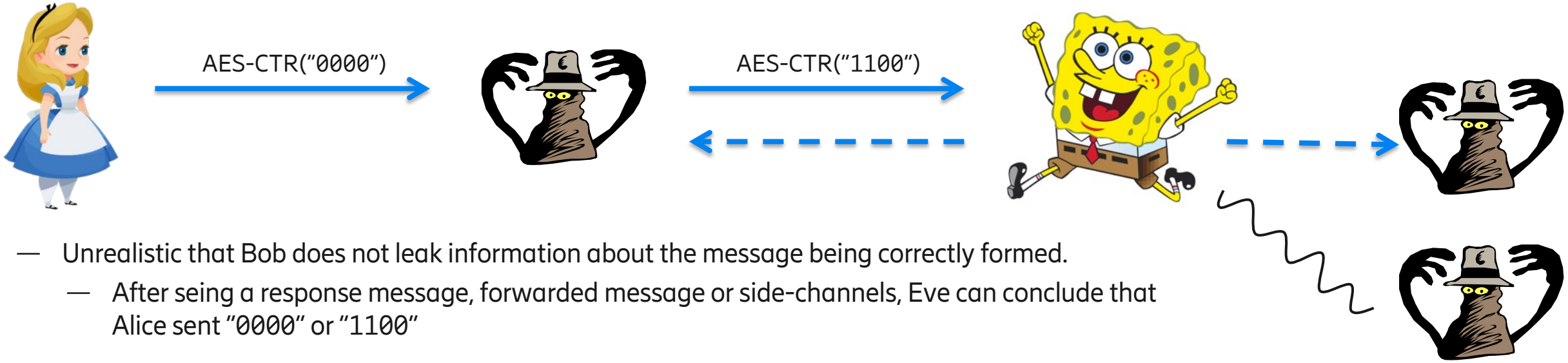
- No integrity protection opens up for “suprising” attacks. Mandating both encryption and integrity is current best practice. Optionality in choosing encryption and/or integrity opens up for attacks. Makes protocols and systems hard to analyse. Having a NULL algorithm is not seen as acceptable anymore.

- AEAD provides a simplified interface for IND-CCA encryption. The interface C = AEAD(K, P, A, N) is harder to misuse.

IND-CPA encryption

— IND-CPA encryption like AES-CTR only provides confidentiality against passive attackers.

Assume tha Alice sends Bob a message in the set:
{"0000", "0011", "0110", "1001", "1100"}



— Unrealistic that Bob does not leak information about the message being correctly formed.
  — After seing a response message, forwarded message or side-channels, Eve can conclude that Alice sent "0000" or "1100"

— Similar attacks on some MAC algorithms with short tags
  — e.g. GCM with 64 bit tags only offers 64 bit security (2^64) forged messages to recover the authentication key.

— Confidentiality against active attackers (current best practice) requires integrity protection.

Symmetic Algorithm Performance

- Old algorithms like 3DES, AES-CBC, and SNOW 3G has quite low performance. AES-GCM is often 10 times faster AES-CBC + HMAC-SHA-1 on modern x64 processors. Integrity is almost free.
- The new 256-bit 5G algorithms will have much faster performance than the old 128-bit algorithms.

Table 3: Performance comparison of SNOW-V-(GCM) and best OpenSSL's algorithms. Performance values are given in Gbps.

Encryption only	Size of input plaintext (bytes)						
	16384	8192	4096	2048	1024	256	64
SNOW-3G-128 (C++)	9.22	9.07	8.89	8.50	7.81	5.38	2.37
AES-256-CBC (asm)	8.50	8.50	8.49	8.48	8.42	8.11	7.07
ChaCha20 (asm)	26.53	26.41	26.29	25.86	24.99	11.80	5.61
AES-256-CTR (asm)	35.06	34.82	34.16	32.94	30.95	22.67	11.32
SNOW-V (C++)	**58.25**	56.98	54.60	50.70	45.28	26.37	9.85
AEAD mode							
ChaCha20-Poly1305 (asm)	18.46	18.24	18.16	17.54	16.99	8.98	4.29
AES-256-GCM (asm)	34.42	33.86	32.74	30.49	27.22	17.32	8.54
SNOW-V-GCM (C++)	**38.91**	37.66	34.86	30.71	26.16	13.93	5.16

Elliptic Curves

Elliptic Curves

— Elliptic curve cryptography (EDCH, ECDSA, EdDSA, ECIES,) have quickly replaced cryptography over finite fields (MODP) and RSA encryption, and is slowly replacing RSA signatures. Much smaller key and signature sizes, and higher performance.

— There are 5 relevant specifications (NIST, ANSI, SECG, CFRG, Brainpool) of Elliptic curves. Significant overlap.

— NIST has deprecated all curves over binary fields. Only curves over prime fields are used today.

SECG	ANSI X9.62	NIST
sect163k1		NIST K-163
sect163r1		
sect163r2		NIST B-163
sect193r1		
sect193r2		
sect233k1		NIST K-233
sect233r1		NIST B-233
sect239k1		
sect283k1		NIST K-283
sect283r1		NIST B-283
sect409k1		NIST K-409
sect409r1		NIST B-409
sect571k1		NIST K-571
sect571r1		NIST B-571
secp160k1		
secp160r1		
secp160r2		
secp192k1		
secp192r1	prime192v1	NIST P-192
secp224k1		
secp224r1		NIST P-224
secp256k1		
secp256r1	prime256v1	NIST P-256
secp384r1		NIST P-384
secp521r1		NIST P-521

Elliptic Curves

- All curves can be written in Weiestraß form. Some curves can be written in other special forms.
- Weiestraß form $y^2 = x^3 + ax + b$
- Montgomery form $M_{A,B} : By^2 = x^3 + Ax^2 + x$
- (Twisted) Edward form $E_{E_{a,d}} : ax^2 + y^2 = 1 + dx^2y^2$
- The Montgomery and Edwards forms support faster formulas for elliptic-curve group operations.
- IRTF CFRG has standardized Curve25519/X25519 and curve448 in Montgomery form and ed25519 and ed448 in (Twisted) Edwards form. NIST will standardize the CRFG curves.
- Montgomery and Edwards curves have a cofactor. Curve25519 is defined over the field GF(2^255 - 19) but the largest subgroup is only ≈ 2^252.
- Curve25519 is the new default standard in HTTPS and is also used in and 3GPP SUCI. In addition to higher performance it is more secure against implementation vulnerabilities.

Signature Algorithms

Signature Algorithms

- RSA signatures with PKCS#1 1.5 padding is now non-recommended due do side-channel attacks and lack of security proofs.

- RSA signatures with PSS padding is used in TLS 1.3 but not in signatures. Criticism that the algorithm is complex. Still large keys, signatures. ECDSA is slowly replacing RSA.

- IETF now recommends deterministic ECDSA for many things. NIST will standardize use of deterministic ECDSA. Deterministic signatures are quite vulnerable to side-channel and fault attacks. CFRG and NIST are now looking at deterministic signatures with readded randomness.

- EdDSA offers better security and performance than deterministic ECDSA. Uses Schnorr signatures instead of the weird ECDSA scheme (was created to avoid Schnorr IPR). NIST will standardize EdDSA.

- For security and performance. EdDSA only comes in two algorithms that fixes the hash function and elliptic curve.
  - Ed25519: ed25519 + SHA-512
  - Ed448: ed448 + SHAKE256

Random and Deterministic signature generation

Both ECDSA and EdDSA algorithms uses a per-message message ‘k’ which is very important for the security.

- **Randomized ECDSA**
  - generate a random k and calculate g^k ….
  - Vulnerable to weak PRNGs
  - Sony used a fixed k for signing games in PS3

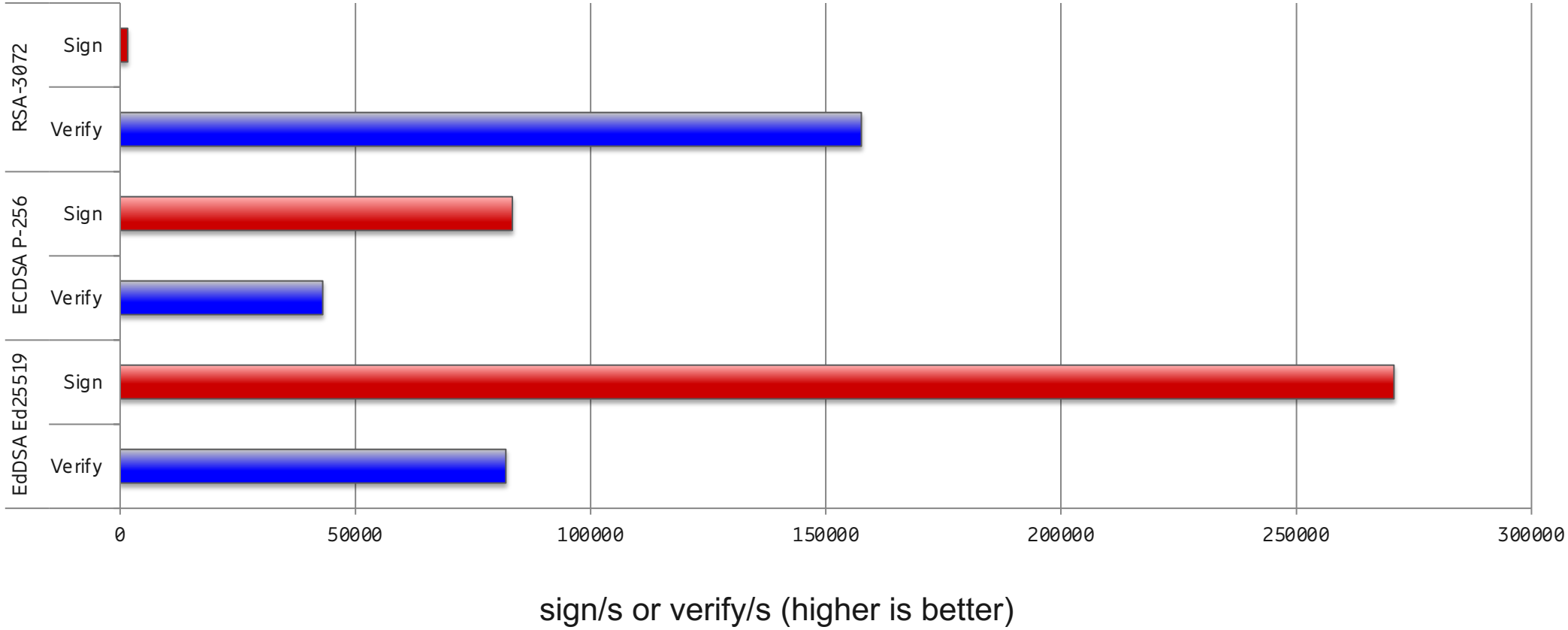
- **Deterministic ECDSA**
  - generate k as HMAC(private key, message)
  - This makes the algorithm vulnerable to side-channel attacks
  - Easy to test implementation

- **Hedged ECDSA**
  - generate k as HMAC(Z, private key, 000….., message)
  - Security wise the best solution

- EdDSA has no purely randomized mode specified. Similar side-channel and fault attacks on EdDSA.

Asymmetric Algorithm Speed

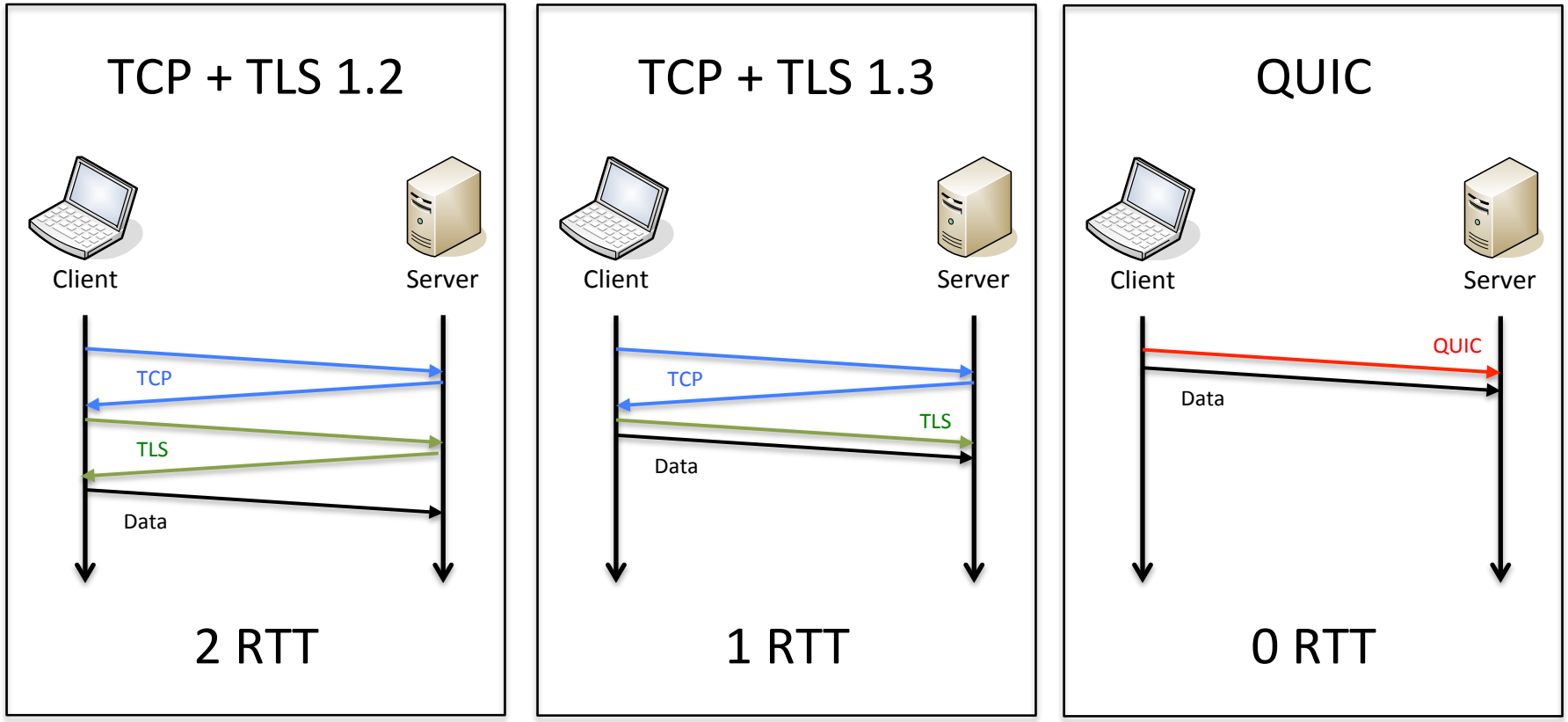
— ECDSA has much better signing performance than RSA, and EdDSA is even faster.



Latency and message overhead

Security Protocol Connection setup latency

- New protocols such as (D)TLS 1.3 and QUIC significantly reduce connection setup latency
- HTTP/3 is HTTP over QUIC



Repeated connection establishment using TLS 1.2, TLS 1.3, and QUIC

Security Protocol "Record layer" overhead

- In many IoT systems, the radio is more constained than the devices. The Low Power Wide Area Networks (LPWANs) market is expected to reach over 1000 billion USD globally by 2027.
- The IoT protocol OSCORE developed by IETF CORE WG has inspired (D)TLS to focus more on less overhead. AEAD enables less overhead.

Protocol	Overhead
DTLS 1.2	21
DTLS 1.3	3
TLS 1.2	13
TLS 1.3	6
OSCORE request	5
OSCORE response	3

Fig: Overhead (excluding MAC / tag) in bytes

Security Protocol "Handshake" overhead

- Message sizes in bytes for the different flights #1, #2, #3
- The EDHOC AKA developed by the IETF LAKE WG takes 1/6 as many bytes for RPK+ECDHE
  - Possible by using CBOR encoding, Certificates by reference, Ephemeral keys as Random, Noise XX,
  - TLS WG is now working on a compressed TLS handshake (cTLS)

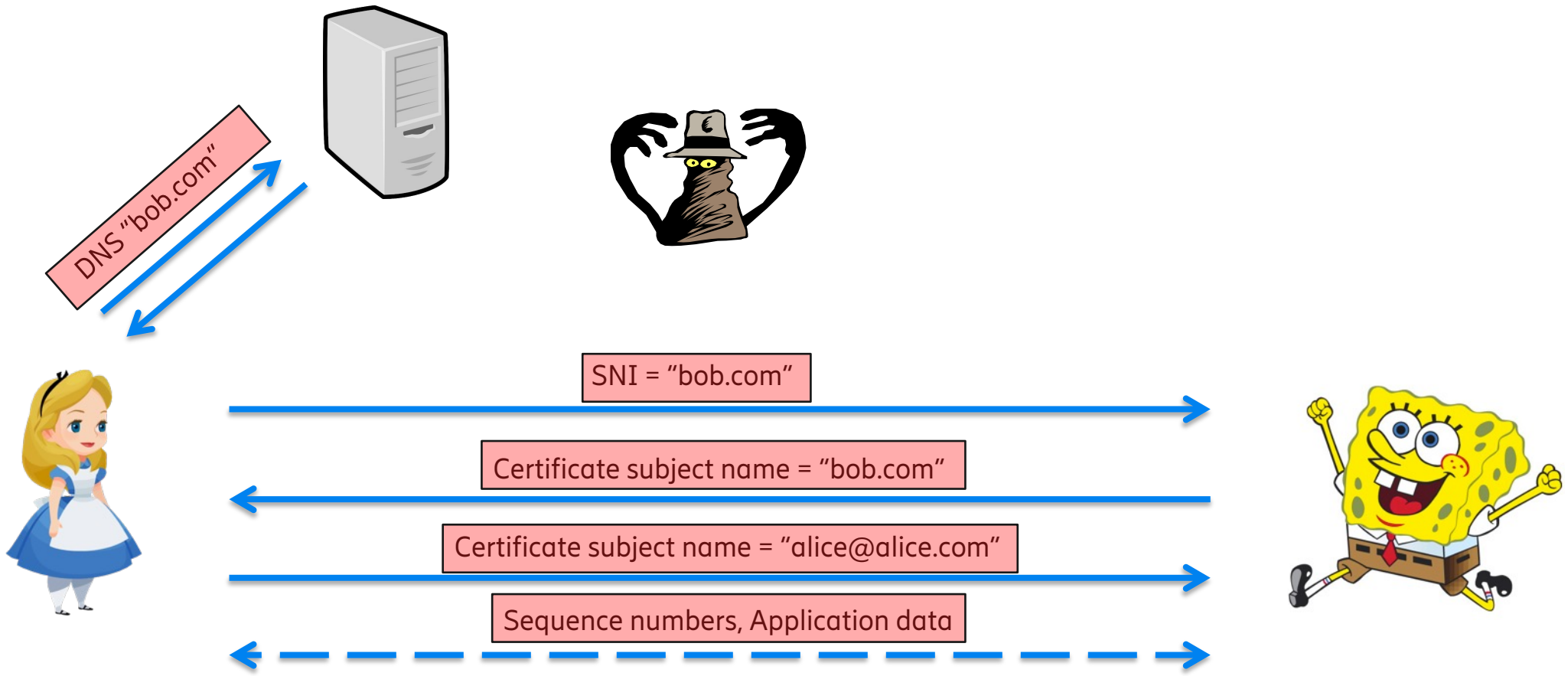
Flight	#1	#2	#3	Total
DTLS 1.3 RPK + ECDHE	150	373	213	736
DTLS 1.3 PSK + ECDHE	184	190	57	431
DTLS 1.3 PSK	134	150	57	341
EDHOC RPK + ECDHE	37	46	20	103
EDHOC PSK + ECDHE	38	44	10	92

Figure 1: Comparison of message sizes in bytes with Connection ID

Privacy

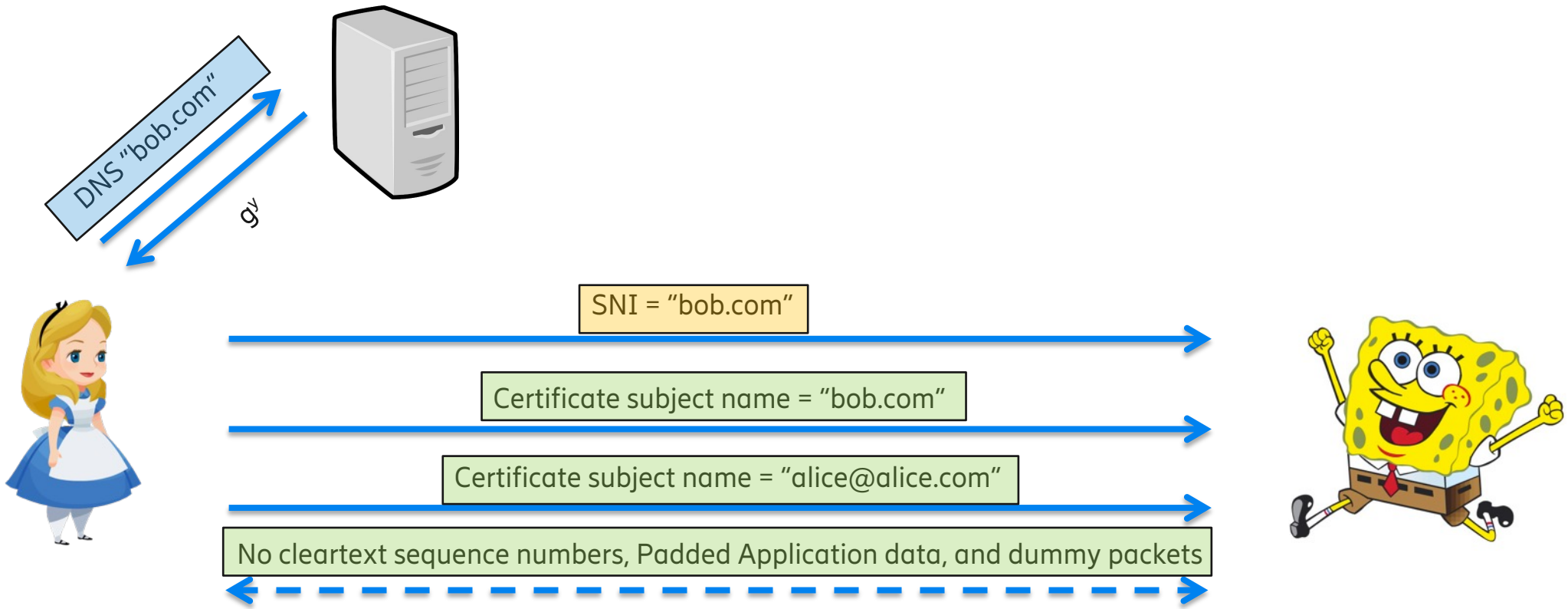
Privacy

— A huge focus on concealing information to make harder in new protocols. In TLS 1.2 most of the handshake is not encrypted. Futhermore DNS is typically not encrypted and traffic flows and packet sizes can be analysed.



Privacy

— In TLS 1.3 as much information as possible is encrypted, especially when used with DoH (DNS over HTTPS) and ECHO (Encrypted Client Hello). Encrypting as much as possible is now best practice.



— IPsec have identity confidentiality, padding, and dummy packets since many years.

— DTLS 1.3 encrypts sequence numbers by using the payload ciphertext as nonce. TLS 1.3 use implicit sequence numbers.

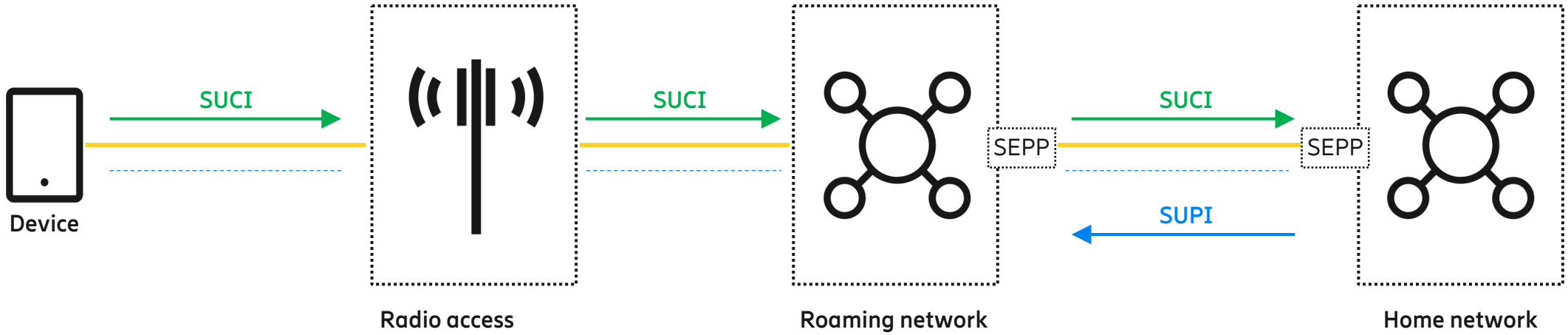
5G: An End to the Battle Against False Base Stations?

Many mitigations against false base stations are part of the 5G standard.

- Encryption of permanent identifier
- Strict refreshment of temporary identifier
- Decoupling of permanent identifier from the paging mechanism
- Integrity protection of user plane traffic
- Secure radio redirections
- False base station detection

Most important is encryption of permanent identifiers. With this enabled, the permanent identifier is never sent in clear over the air.

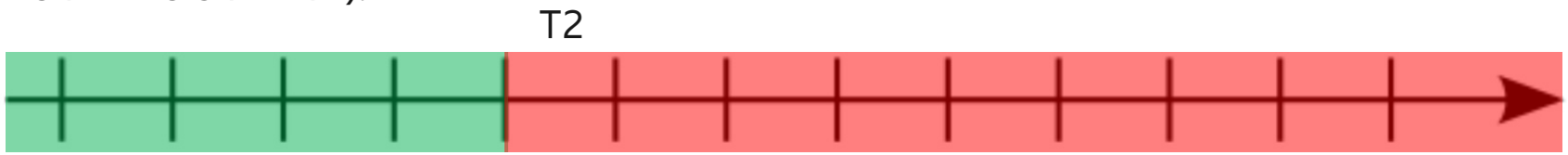
- Encryption is done using a public key stored on the device. Uses state-of-the-art cryptography like Curve25519.
- In 5G, the long-term identity is called SUPI. SUPI is often an IMSI. An encrypted SUPI is called SUCI.



Perfect Forward Secrecy PFS

Current Best Practice for Diffie-Hellman

- **Forward secrecy only limits the effect of key leakage in one direction** (compromise of a key at time T2 does not compromise some key at time T1 where T1 < T2).



- Protection (from passive attackers) in the other direction (compromise at time T2 does not compromise keys at time TX) can be achieved by rerunning EC(DHE).

- Using the terms in RFC 7624 *"Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement"*, forward secrecy without rerunning EC(DHE) does not stop an attacker from doing **static key exfiltration**. Frequently rerunning EC(DHE) forces an attacker to do **dynamic key exfiltration (or content exfiltration)**.
  - Static key exfiltration (transfer of keys happens once or rarely)
  - Dynamic key exfiltration (transfers of keys happens frequently)
  - Content exfiltration (transfer of content instead of keys)

- **Forcing attacker to do dynamic key exfiltration (or content exfiltration) should be considered best practice.** This significantly increases the risk of discovery for the attacker. **RFC 7624 (2015) is great and should be referenced and followed much more.**

Current Best Practice for Diffie-Hellman

- **Assuming breach (e.g. key exfiltration) and minimizing the impact from breach is an essential part of zero-trust.**
  - *"The Zero Trust security model assumes that a breach is inevitable or has likely already occurred"* https://media.defense.gov/2021/Feb/25/2002588479/-1/-1/0/CSI_EMBRACING_ZT_SECURITY_MODEL_UOO115131-21.PDF
  - *"If a breach does occur, minimizing the impact of the breach is critical."* https://www.google.com/search?channel=crow5&client=firefox-b-d&q=limit+impact+of+breach+zero+trust
- ANSSI (France) (2015) recommendation for IPsec is to rerun ephemeral Diffie-Hellman at least every hour or every 100 GB. Nothing special about IPsec. **This should be considered best practice for all non-constrained protocols and systems.**

R12

If available, one shall activate the PFS property in IKEv2 second phase (a.k.a "quick mode") using a Diffie-Hellman key exchange or its elliptic curve variant.

R13

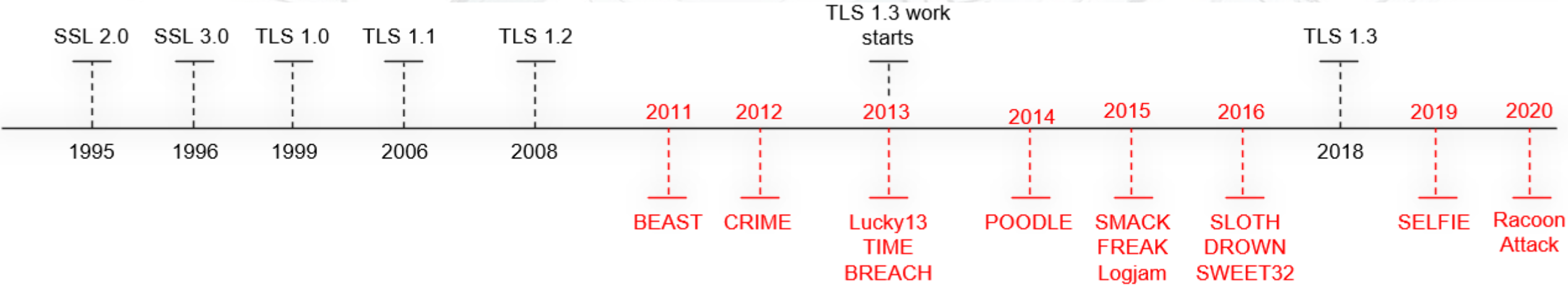
It is recommended to force the periodic renewal of the keys, e.g. every hour and every 100 GB of data, in order to limit the impact of a key compromise.

- The Double Ratchet Algorithm enables frequent Diffie-Hellman in the Signal protocol. In TLS 1.3 new connections are needed.

SIGMA-I and Noise Protocol

The Dark Ages: SSL 1.0 - TLS 1.2

- SSL 1.0 was an ad hoc design without any theoretical foundation.
- SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, and TLS 1.2 (as specified in RFC 5246) are all horribly weak.
  - Each version were broken due to a combination of bad design and bad cryptographic algorithms.
  - Each version tried to patch the fundamental weaknesses found in the previous version but failed.
- TLS 1.0 cannot be made secure at all. US government assessment is that it so weak that it gives a false sense of protection. Do not assume it gives any protection.
- TLS did not become popular because of its good security. TLS has a terrible security history.

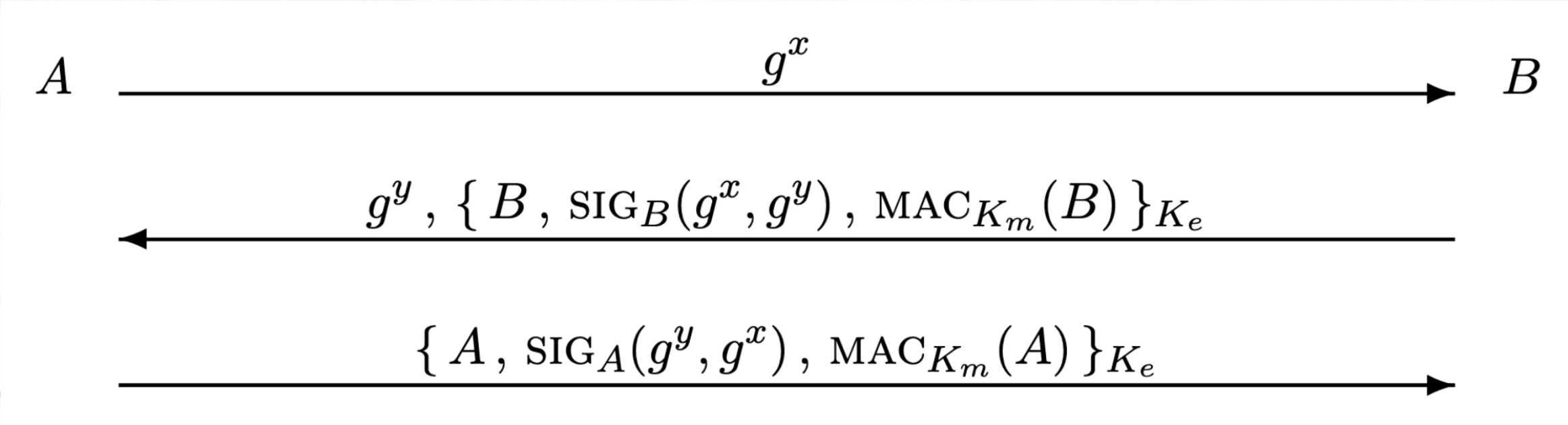


Age of Enlightenment: SIGMA-I with identity protection

— **Theoretical security protocol** with proven security.

— "Secure" without encryption. Encryption is a separate layer (record layer in TLS).

— SIGMA-I is the basis for IKE, IKEv2, TLS 1.3, and EDHOC method 0.



Age of Enlightenment: Noise Protocol Framework

- Set of *formally verified* **theoretical security protocols** using only ephemeral (e) and static (s) ECDH.
- No signatures. Ephemeral-Static ECDH gives implicit authentication by the ability to compute the session key (like Static RSA in TLS).
- A single ephemeral key used reused in different ECDH functions and static-static ECDH is used.
- EDHOC method 3 use the Noise XX pattern and WireGuard use the Noise IK pattern.

```
XX:
  -> e
  <- e, ee, s, es
  -> s, se
  <-
  ->
  ...
```

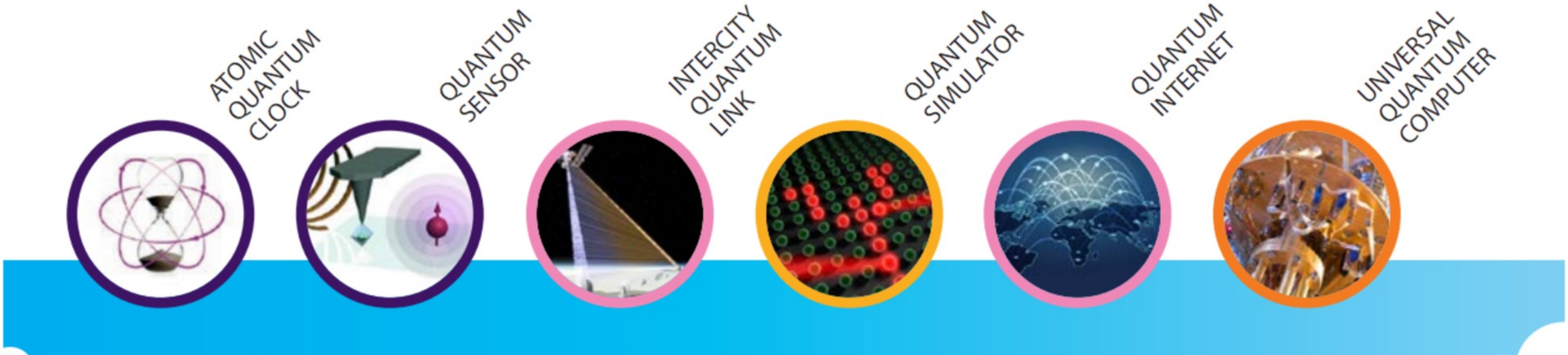
```
IK:
 <- s
 ...
 -> e, es, s, ss
 <- e, ee, se
 ->
 <-
 ...
```

Quantum-Resistant Cryptography

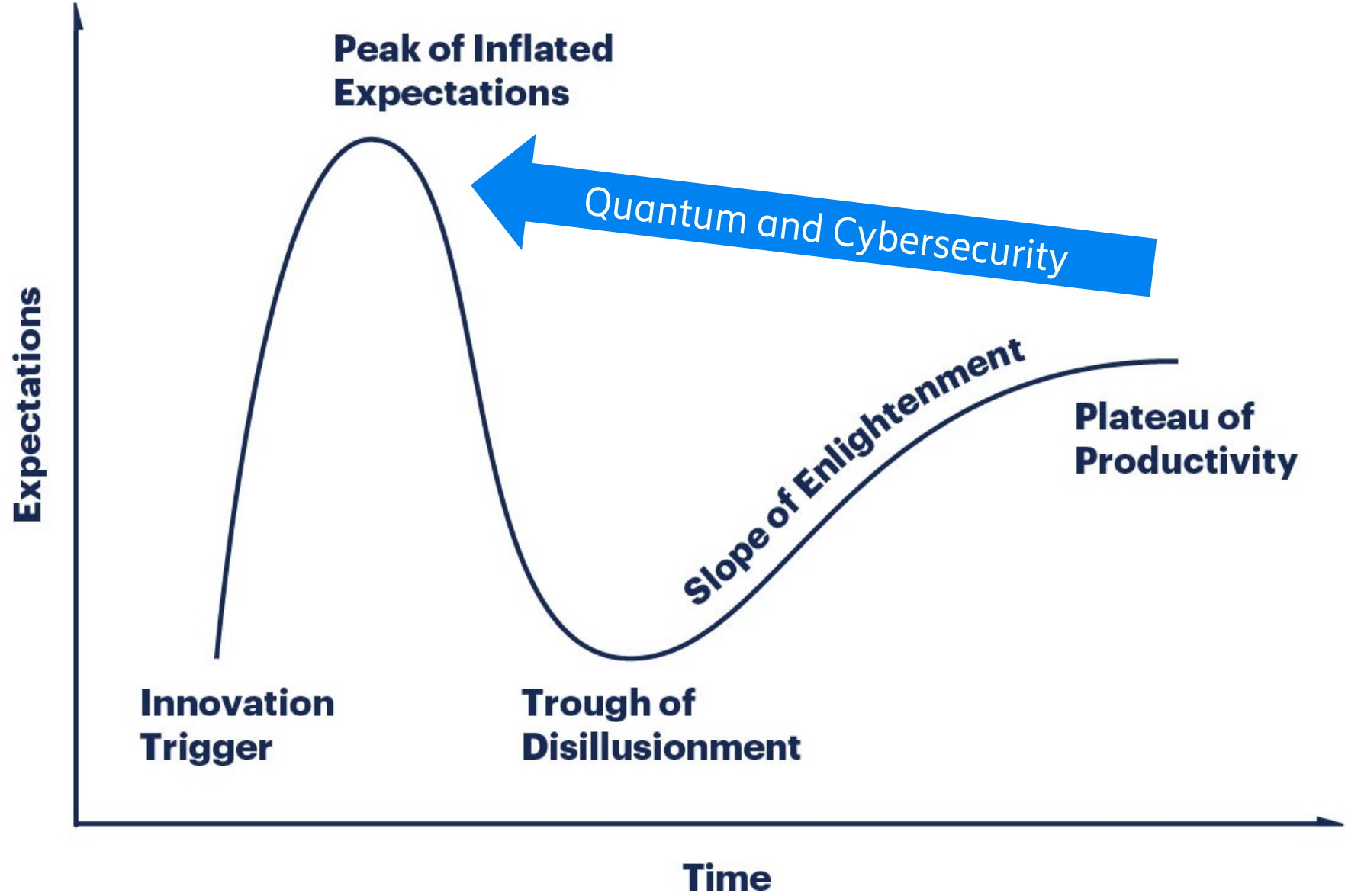
Second quantum revolution

— The first quantum revolution enabled inventions such as lasers, transistors, atomic clocks, and magnetic resonance imaging (MRI). This gave us us computers, optical fiber communication, and the global positioning system and led to the 20th-century technological revolution.

— The second quantum revolution is about **controlling individual quantum systems**, which enables even more powerful applications of quantum information.

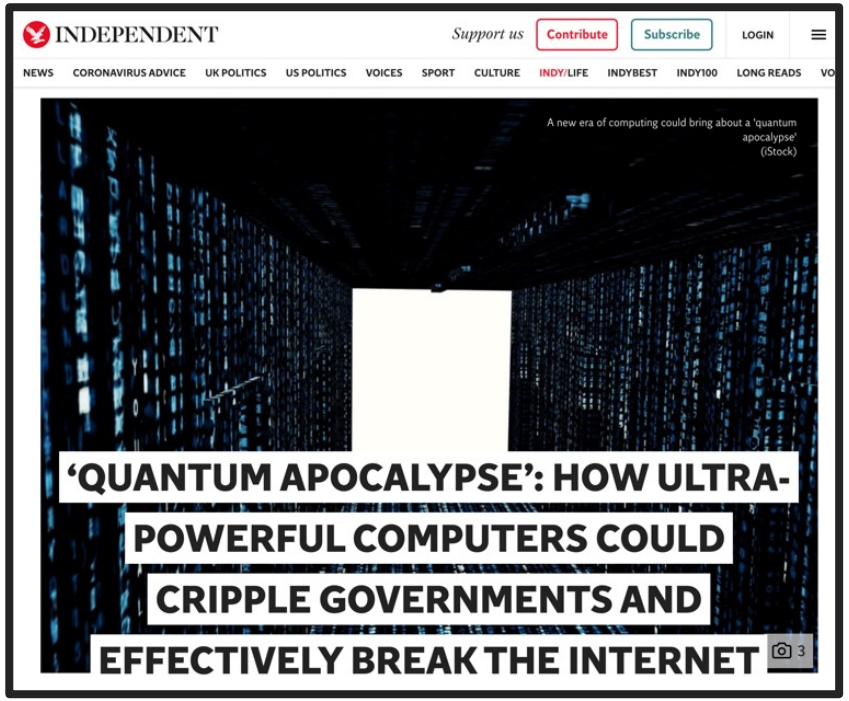


Quantum and security is on top of the hype curve



Quantum and cybersecurity are at the top of the hype curve

— (too) much talk about quantum technology and security.



— A lot of snakeoil like "Quantum Blockchain with AI deep learning"....

— Risks taking focus from other more acute and relevant security aspects.

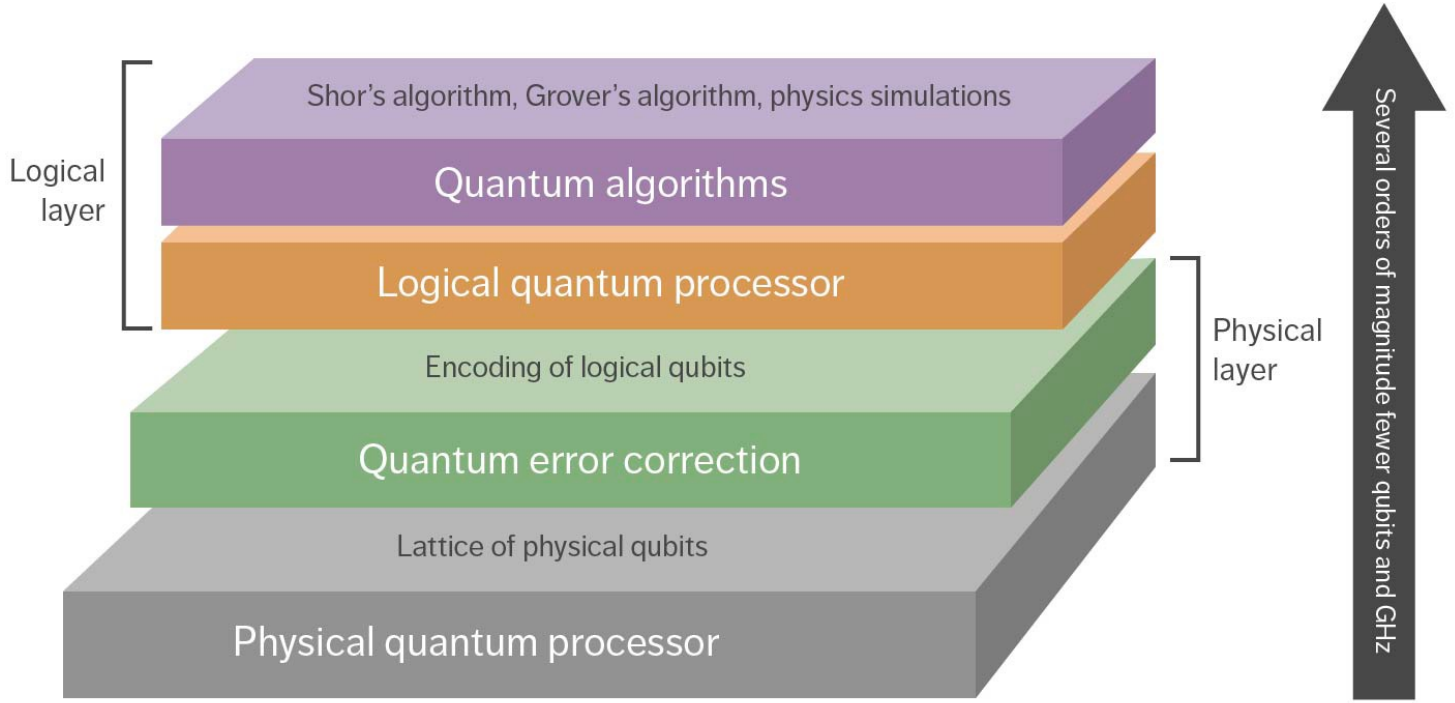
System view of a quantum processor

— Just like 5G and WiFi, a quantum processor consists of a physical and a logical layer.

— All press releases about quantum computers talk about physical qubits.

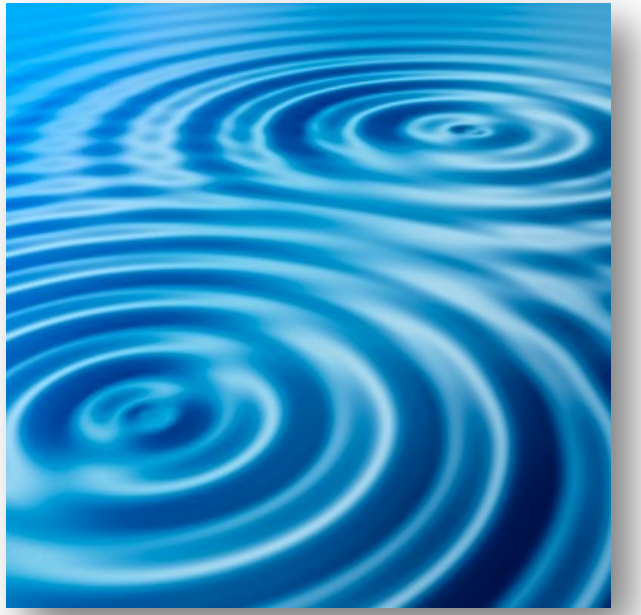
— All quantum attack papers talk about logical qubits.

— Due to quantum error correction, the logical layer typically has 10^3 fewer qubits and 10^3 lower clock speed.



Quantum computers and algorithms

- Large, robust quantum computers have proven to be notoriously difficult to build.
- There are only a few quantum algorithms. They do not look like classic algorithms and work by manipulating probability waves.
- Quantum computers are only useful for the few problems for which quantum algorithms exist.



Attacks from quantum computers – Shor's algorithm

- Shors quantum algorithm on a sufficiently large and robust quantum computer would crack the two asymmetric crypto-algorithms (RSA, ECC) used today. Such an attacker could decrypt communications and forge signatures. Communication today can be saved and decrypted in the future.

- A cryptographically relevant quantum computer (CRCQ) requires millions of robust physical qubits and trillions of quantum gates. The largest quantum computer currently has about 100 unstable physical qubits.

- It is unclear when or if CRQC will be built. If the number of qubits follows Moore's law (doubled every two years), it takes 25-30 years. Some researchers believe that trillions of "quantum gates" are an even more difficult problem.

- An expert committee said in 2019 that the emergence of a CRQC in the coming decade would be very unexpected. The Committee also pointed out that there are no known applications for medium-sized quantum computers. If no applications are found, the investments will probably pay off.

Attacks from quantum computers – Shor's algorithm

- The NSA said in September 2021 that if you do not need more than 10 years of protection, you should worry more about things like malware and security in the supply chain in the immediate future.

- In December 2021, the ASD (Australian Signals Directorate) told the Australian Senate that they believe that quantum computers do not pose an immediate threat. Current encrypted data will remain secure from decryption for as long as the information is confidential.

- ICANN, Quantum Computing and the DNS, February 2022: Without massive and unexpected discoveries in both quantum physics and engineering for quantum computers, there is no chance that a cryptographically relevant quantum computer (CRQC) could be built in the next decade, and possibly not for many decades. Even after waiting a decade, there will clearly be years if not decades of warning before a CRQC will be built, and that amount of time will be more than sufficient for the DNSSEC community to adopt one or more appropriate signature algorithms based on post-quantum cryptography (PQC). "

Symmetric cryptography and Grover's algorithm

— While classical algorithms that test all keys are perfectly parallelizable, i.e., to increase the number of computers by a factor *N* gives a speed increase of *N*, an increase in the number of quantum computers by a factor *N* gives a speed increase of only $\sqrt{N}$.

— A CRQC that can crack RSA in a few hours would not pose any practical threat at all to symmetrical algorithms such as AES-128 and SHA-256.

— Using NIST's assumptions about performance, a huge cluster of one billion CRQCs would take a million years of uninterrupted calculation to find a single AES-128 key.

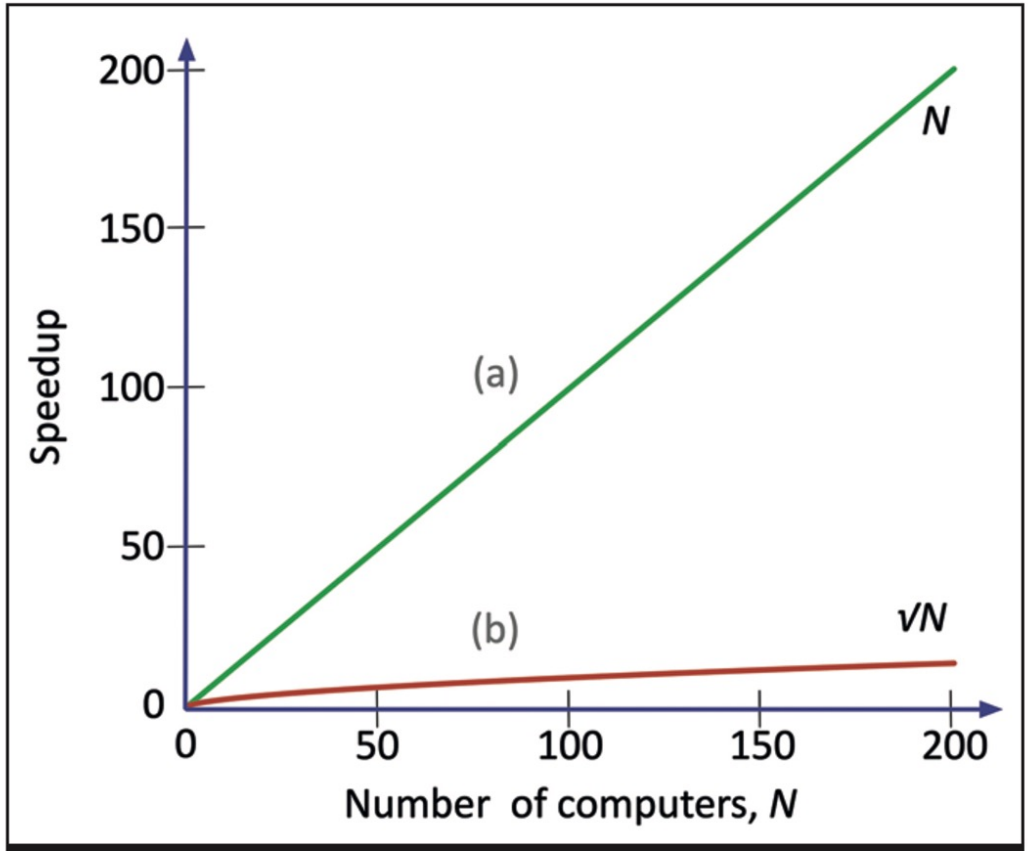
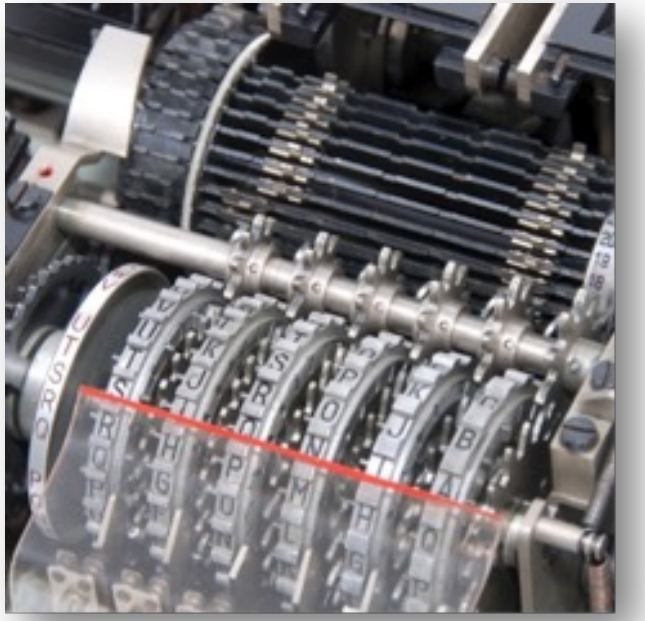
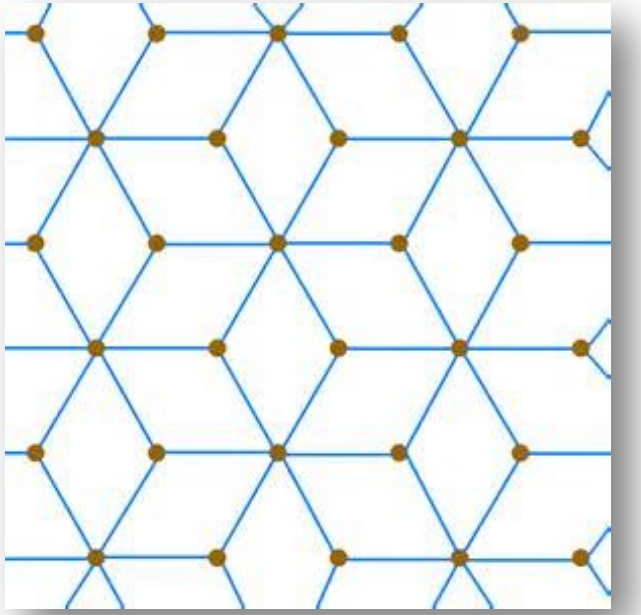


Figure 1: Parallelisation of brute-force key search on classical (a) and quantum (b) computers.

Quantum-safe cryptography

- Quantum-Safe Cryptography is cryptography that is resistant to attacks from quantum computers. There are three types:
  - Classic symmetric cryptography (e.g., AES, SHA).
  - Classic asymmetric cryptography not affected by Shor's algorithm (e.g., lattice-based cryptography) (PQC).
  - Quantum cryptography (e.g., QKD) use quantum mechanics to perform cryptography.





Standardization of Post-Quantum Cryptography (PQC)

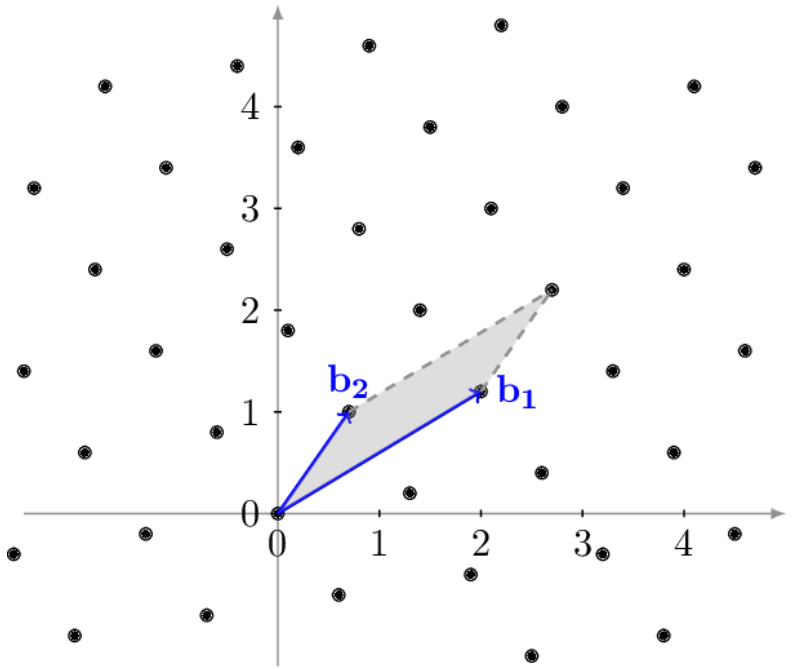
- IRTF CFRG and NIST have standardized stateful hash-based signatures. Only recommended if quantum-safe signatures are needed immediately and the risks of accidental one-time key reuse can be minimized.
- NIST Post-Quantum Cryptography project will standardize stateless algorithms for key exchange and digital signatures. The "winners" was announced in Q2 2022. Final standards within a few years.
- Hash-based signatures (1979) and lattice-based cryptography (1996) are well-studied but have not been used as they are slightly less efficient than RSA and ECC. Both run in software on classical computers.
- IETF and 3GPP are planning to introduce NIST PQC algorithms in the next few years. 3GPP products use many security protocols standardized by IETF (IPsec, (D)TLS, SSH, JOSE, X.509, etc.).
- US will use NIST PQC standards to protect TOP SECRET. Plans to update CNSA suite as soon as NIST announces the winning algorithms. TOP SECRET information often has to be kept secret for a very long time.
- ICANN has decided that DNSSEC does not need to be updated at this point in time. Will monitor quantum computer development and update to PQC when needed.
- Ongoing discussion on hybrid soliutions where both PQC and ECC are used.

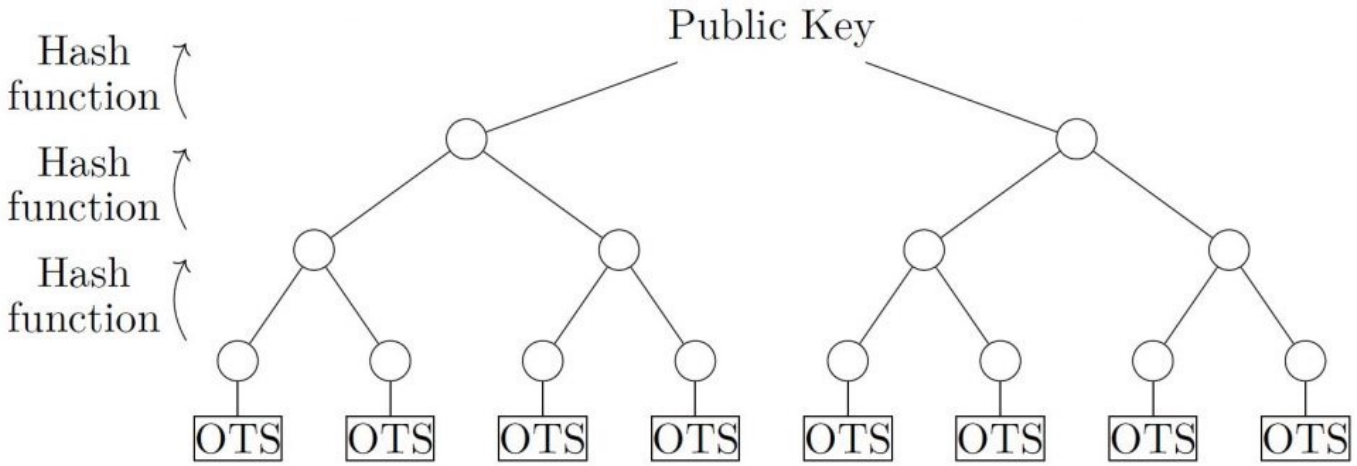
Post-Quantum Cryptography

— Post-Quantum Cryptography (PQC) or Quantum-Resistant (QR) Cryptography are secure even when/if Cryptographically Relevant Quantum Computers (CRQC) are built. A CRQC running Shor's algorithm would break RSA and ECC in a matter of hours. AES-128 and all other symmetric algorithms are Quantum-Resistant.

— IRTF CFRG and NIST has standardized stateful hash-based signature algorithms XMSS and LMS

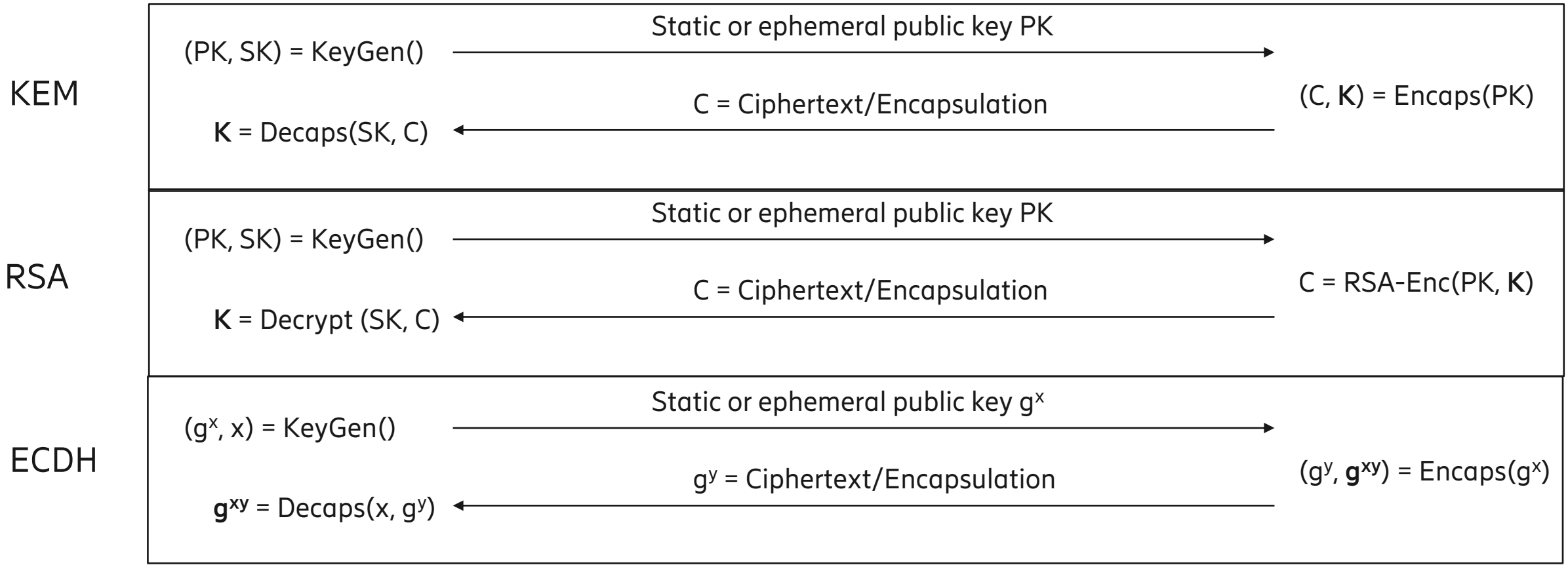
— NIST PQC project is planning to standardize Kyber, Dilithium, and Falcon based on structured lattices as well as the stateless hash-based signature algorithm SPHINX+





Key Encapsulation Mechanism (KEM)

— NIST will specify (the non-signature) PQC algorithms using a KEM interface. KEM can be seen as a limited asymmetric key encryption algorithm where the symmetric key is generated by the KEM.
In a Diffie-Hellman KEM, the Ciphertext/Encapsulation g^y is a public key.



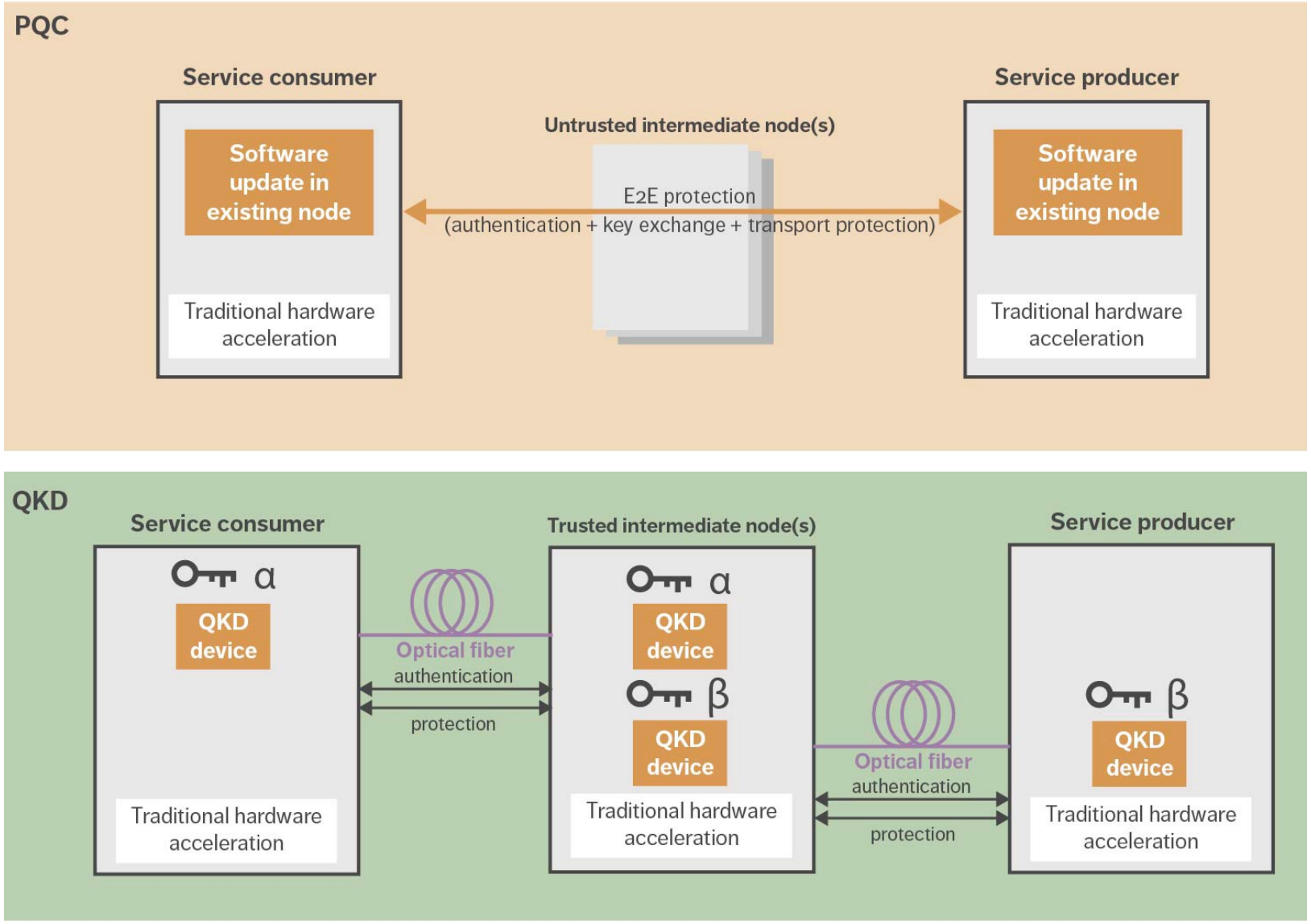
NIST PQC - Performance and Sizes

- Algorithms based on structured lattices are as fast or faster than ECC, but with sizes larger than RSA.
- Rainbow and SIKE was broken late in the PQC project. Falcon has smaller signatures (666 bytes) but the additional complexity makes is unsuitable for constrained IoT devices. Current algorithms are too big and will not work in many LPWAN.

Algorithm	Generate key pair	Sign	Verify	Encapsulate	Decapsulate	Public key size	Signature/ciphertext size
LMS/XMSS/HSS/MXSS^MT (hash-based PQC)	43 - 165000 ms	0.51 - 6.0 ms	0.096 - 5.3 ms			50 - 100 B	700 - 8000 B
Picnic picnic3-L1 (symmetric primitives)	0.001 ms	7.272 ms	5.508 ms			35 B	12492 B
Rainbow 1a (multivariate PQC)	401.505 ms	0.020 ms	0.014 ms			152097 B	64 B
Dilithium-1024x768-AES (lattice-based PQC)	0.018 ms	0.098 ms	0.025 ms			1184 B	2044 B
EdDSA Ed25519	0.018 ms	0.019 ms	0.066 ms			64 B	64 B
ECDSA P-256	0.041 ms	0.053 ms	0.113 ms			64 B	64 B
RSA-3072	232.704 ms	3.542 ms	0.033 ms	0.038 ms	3.530 ms	384 B	384 B
NIST P-256	0.094 ms			0.231 ms	0.231 ms	32 B	32 B
Curve25519	0.050 ms			0.054 ms	0.054 ms	32 B	32 B
Kyber-512-90s (lattice-based PQC)	0.006 ms			0.010 ms	0.008 ms	800 B	736 B
Classic McEliece 348864 (code-based PQC)	18.494 ms			0.015 ms	0.048 ms	261120 B	128 B
SIKE p503 (isogeny-based PQC)	4.292 ms			7.041 ms	7.511 ms	378 B	402 B

Quantum Key Distribution (QKD)

- Modern infrastructure is implemented with zero trust principles where cryptography is used for confidentiality, integrity protection, and authentication on many of the layers of the network stack, often all the way to software in the cloud.
- QKD is an unauthenticated physical point-to-point key exchange protocol that requires new hardware and trusted relays.
- A quantum internet makes sense to connect quantum computers or quantum sensors but has nothing to do with security.



What security researchers and governments think about QKD

- Bruce Schneier:
  - *"Quantum cryptography: as awesome as it is pointless. It's a clever idea, but basically useless in practice".*
- France (ANSSI):
  - "deployment constraints specific to QKD hinder large-scale deployments with high practical security"
  - *"not to be considered as the next step for secure communications"*
  - *"QKD on point-to-point links can nevertheless be considered as a defense-in-depth measure"*
- UK (GCHQ):
  - *" QKD protocols do not provide authentication, they are vulnerable to physical man-in-the-middle attacks"*
  - *"does not endorse the use of QKD for any government or military applications, and cautions against sole reliance on QKD for business-critical networks"*
  - *"advice is that the best mitigation against the threat of quantum computers is quantum-safe cryptography."*
- US (NSA):
  - *"Quantum key distribution is only a partial solution"*
  - *" requires special purpose equipment"*
  - *" increases infrastructure costs and insider threat risks"*
  - *" securing and validating QKD is a significant challenge"*
  - *"increases the risk of denial of service"*
  - *"does not support the usage of QKD or QC to protect communications in National Security System"*

Summary

- Too much talk about quantum technologies and security (compared to other relevant security problems).
- If large robust quantum computers are ever built, they will break RSA and ECC in a few hours.
- Quantum attacks will have no practical effect on symmetric crypto (like AES-128).
- NIST will soon standardize new quantum secure (PQC) algorithms. For most uses cases there is no panic to update, but it depends on how long you want protection.
- Quantum crypto like QKD completely lacks practical uses today. It is very limited, not needed, not practically secure, and not ready for use.