Crypto Heute

Zuzana Trubini

cnlab Herbsttagung 2023: Bits & Qubits
Gleisarena, Zürich, 6. September 2023

cnlab

Crypto Heute

Welche Verfahren sind heute im Einsatz?

Worauf basiert ihre Sicherheit?

Wie wird die Schlüssellänge bestimmt?

Was ist wie stark von Quanten-Computern betroffen?

Sicherheitsempfehlungen

NIST 2020

Security Strength	Symmetric Key Algorithms	FFC (DSA, DH, MQV)	IFC* (RSA)	ECC* (ECDSA, EdDSA, DH, MQV)
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256\text{-}383$
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384\text{-}511$
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

* The security-strength estimates will be significantly affected when quantum computing becomes a practical consideration.

ECRYPT-CSA 2018

	Parameter	Legacy	Future System Use Near Term	Future System Use Long Term
Symmetric Key Size	k	80	128	256
Hash Function Output Size	m	160	256	512
MAC Output Size*	m	80	128	256
RSA Problem	$\ell(n) \geq$	1024	3072	15360
Finite Field DLP	$\ell(p^n) \geq$	1024	3072	15360
	$\ell(p), \ell(q) \geq$	160	256	512
ECDLP	$\ell(q) \geq$	160	256	512

parametrised by the finite field $\mathbb{F}_{p^n}$ and the subgroup size q

We note that the guidelines above, and indeed all analysis in this document, is on the basis that there is no breakthrough in the construction of quantum computers. If the development of quantum computers became imminent, then all this document's guidelines would need to be seriously reassessed.

BSI 2023

Block Cipher	MAC	RSA	DH $\mathbb{F}_p$	ECDH	ECDSA
128	128	3000[a]	3000[a]	250	250

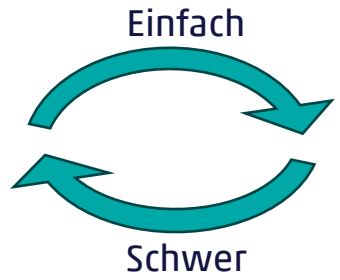
- The asymmetric mechanisms recommended in this Technical Guideline will become insecure in case of significant progress in the development of quantum computers.

Heutige Crypto

Crypto Primitiven (Bausteine)

- SHA
- AES
- Diffie Hellman (DH)
- Elliptic Curve DH (ECDH)
- DSA
- EC DSA
- RSA

- Schlüssellos
- Symmetrisch
- Asymmetrisch - basierend auf Einwegfunktionen



Crypto Protokolle

- TLS
- VPN
- ...

In Theorie können all diese Crypto Primitiven gebrochen werden.

In der Praxis müssen die Parameter so gewählt werden, dass dies praktisch unmöglich ist. Insbesondere **muss gelten**:

nötiger Rechenaufwand $\gg$ **verfügbare Rechenleistung**

Sicherheit & Schlüssellänge

Def: ≈ 2^k Operationen nötig zum Brechen

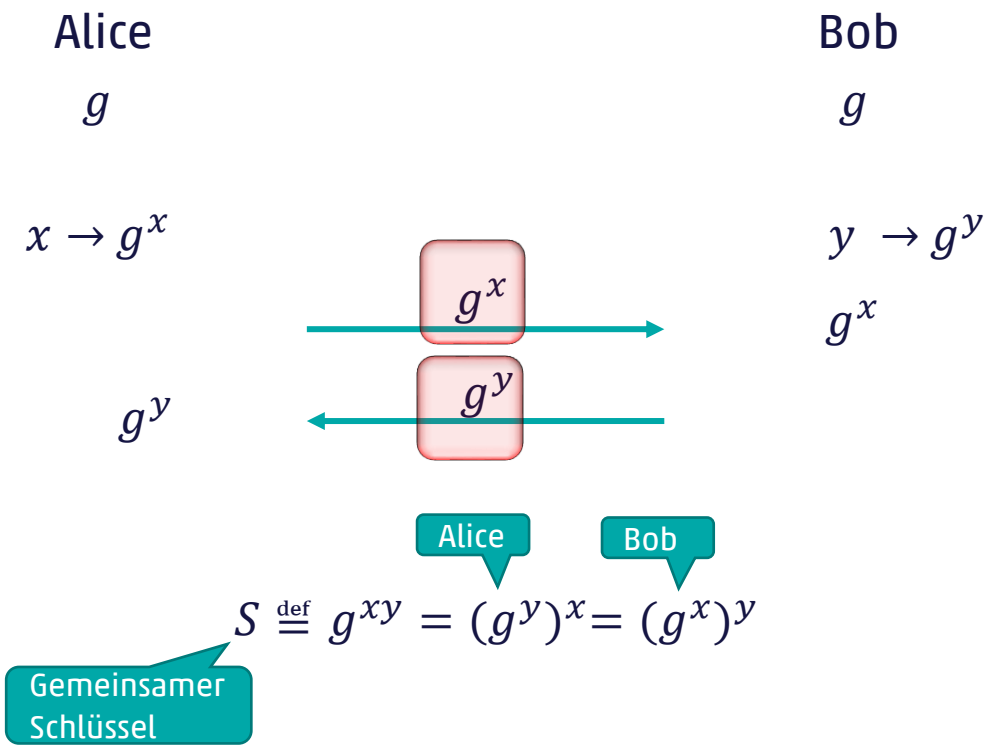
- Bester Alg – Exhaustive Search
- k Bit-Key -> 2^k Möglichkeiten

Time Frame	**Security Strength** k	**Symmetric Key** **Bitlänge vom Key**	**DH, DSA, DLIES** **(Finite Field Crypto, FFC)** **In UG von $\mathbb{Z}_p^*$ oder $\mathbb{F}^*(p^n)$** **Bitlänge von PK/SK (Bitlänge der Ordnung von G/UG)**	**ECDH, ECDSA, ECIES** **(Elliptic Curve Crypto, ECC)** **Auf Elliptischen Kurven** **Bitlänge des SK (Bitlänge der Ordnung des Basispunktes)**	**RSA** **In $\mathbb{Z}_N$** **Bitlänge des Modulus N**
Legacy	80	80	1024 / 160	160	1024
Near Term	128	128	3072 / 256	256	3072
Long Term	256	256	15360 / 512	512	15360

Exp viel einfacher als √ und log

- Es gibt bessere Algorithmen als Exhaustive Search
- Schlüsselwahl – s.d. Aufwand zum Brechen ≈ 2^k 3DES Operationen

Diffie-Hellman Key-agreement (DH, ECDH)



Angreifer ???
- Kennt: g^x, g^y
- Braucht: g^{xy}

= DHP (Diffie Hellman Problem)

DHP ist höchstens so schwer wie x aus g^x berechnen

DLP (Discrete Logarithm Problem)

Vermutung: DHP ist gleich schwer wie DLP.

Berechnungen in einer zyklischen Gruppe mit Generator g:
- Untergruppe von $\mathbb{Z}_p^*$ oder $\mathbb{F}^*(p^n)$ -> DH
- Elliptische Kurve -> ECDH

Komplexität von DLP & Schlüssellänge von DH

Gegeben: Generator g einer zyklischen Gruppe G (mit n Elementen) und g^x

Gesucht: x (zwischen 1 und n)

b_x = die Bitlänge von x
(d.h. $b_x = \log_2 n$)

Komplexität hängt von der Gruppe ab:

- **Generische Algorithmen** für beliebige Gruppen mit n Elementen
  - Ex.Search: Laufzeit $\approx n = 2^{b_x}$
  - Schneller: Baby-Step-Giant-Step, Pollard-Rho, ...

Für Sicherheitslevel 128

Laufzeit $\approx \sqrt[2]{n} = \sqrt[2]{2^{b_x}} = 2^{\frac{b_x}{2}}$. Damit gleich 2^{128}, muss $\frac{b_x}{2} = 128$ also $b_x = 128 \cdot 2 = 256$.

In jedem Fall:
Sicherheitslevel 128
⇩
Bitlänge des SK $\geq 2 \cdot 128$

- **Spezifische Algorithmen** für zyklischen Untergruppen von $\mathbb{Z}_p^*$ (d.h. g und g^x haben Bitlänge $b_p = \log_2 p$)
  - Am schnellsten: Number Field Sieve (NFS) mit Laufzeit $L_p(\frac{1}{3}; 1.92)$

$$L_p\left(\frac{1}{3}; 1.92\right) = \mathcal{O}\left(e^{(1.92+o(1))\,(\ln p)^{\frac{1}{3}}(\ln\ln p)^{\frac{2}{3}}}\right) \approx 2^{9\cdot\sqrt[3]{b_p}} \overset{!}{=} 2^{128} \Rightarrow 9 \cdot \sqrt[3]{b_p} = 128 \text{ also } b_p = \left(\frac{128}{9}\right)^3 \approx 3000.$$

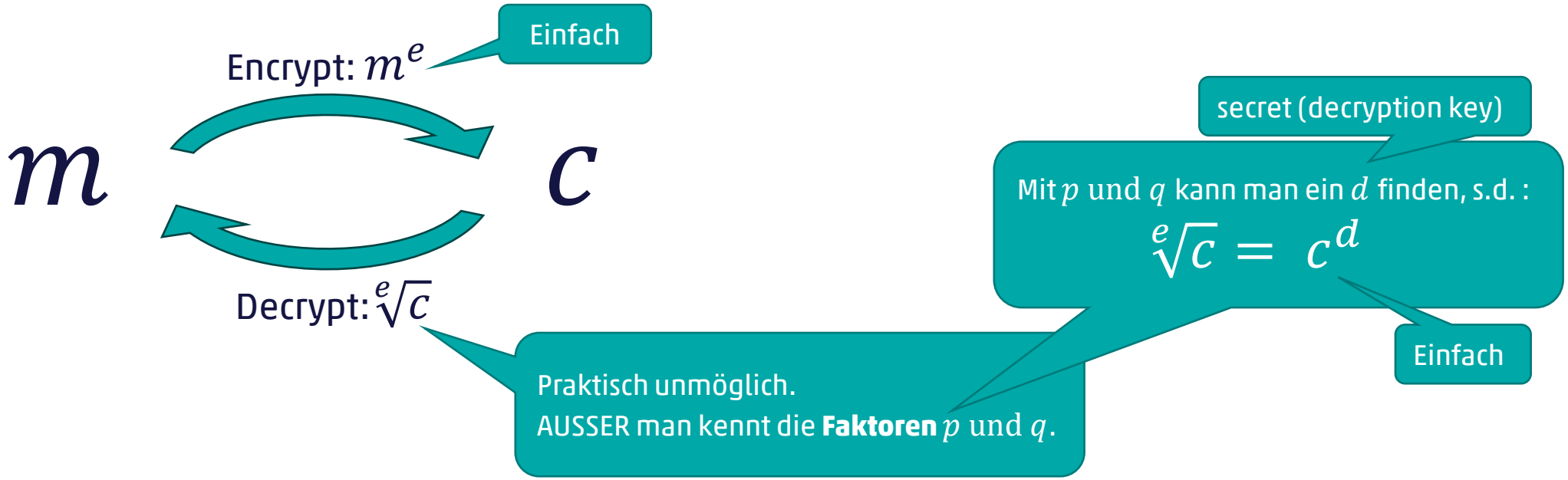
Näherung für p mit Bitlänge 1000-5000, mit Berücksichtigung von Laufzeit-Messungen:
NFS für 512-bit Zahl entspricht dem Rechenaufwand von etwa 2^{50} DES-Operationen

In $\mathbb{Z}_p^*$:
Sicherheitslevel 128
⇩
Bitlänge des PK ≥ 3000

RSA

- Verschlüsselungs- & Signaturverfahren
- Berechnungen in $\mathbb{Z}_N$ mit $N = pq$ (p, q grosse Primzahlen)

- Verschlüsselung von $m \in \mathbb{Z}_N$ mit public (encryption) key e:



Komplexität der Faktorisierung & Länge des RSA-Modulus

Gegeben: RSA Modulus $N = pq$

Gesucht: Faktoren p und q

Komplexität:

- Ex.Search: Laufzeit $\approx \sqrt[2]{N}$
- Schnellster Algorithmus: Number Field Sieve (NFS) mit Laufzeit $L_N(\frac{1}{3}; 1.92)$

$$L_N\left(\frac{1}{3}; 1.92\right) = \mathcal{O}\left(e^{(1.92+\sigma(1))\,(\ln N)^{\frac{1}{3}}\,(\ln\ln N)^{\frac{2}{3}}}\right) \approx 2^{9\cdot\sqrt[3]{b_N}} \overset{!}{=} 2^{128} \Rightarrow 9\cdot\sqrt[3]{b_N} = 128 \text{ also } b_N = \left(\frac{128}{9}\right)^3 \approx 3000.$$

Sicherheitslevel 128 ⇩ Bitlänge von $N \geq 3000$

Näherung für N mit Bitlänge 1000-5000, mit Berücksichtigung von Laufzeit-Messungen: NFS für 512-bit Zahl entspricht dem Rechenaufwand von etwa 2^{50} DES-Operationen

Heutige Crypto in der Welt von Morgen

In Zeiten von Cryptographisch relevanten Quanten Computern (CRQC)

Crypto Primitiven	Bsp	Bester Klassischer Algorithmus	Heutige Schlüssellänge für $k = 128$	Bester Quanten Algorithmus	Schlüssellänge für $k = 128$ im Fall von CRQC
Symmetrisch	AES	Ex.Search $\approx n = 2^{b_x}$	128	Grovers (Ex.Search) $\approx \sqrt{n} = 2^{\frac{b_x}{2}}$	256
EC DLP	ECDSA ECDH	Pollard-Rho $\approx \sqrt{n} = 2^{\frac{b_x}{2}}$	256	Shor	✗
DLP	DSA, DH ElGamal	NFS $L_p\left(\frac{1}{3}; 1.92\right) \approx 2^{c \cdot \sqrt[3]{b_p}}$	3000	Shor	✗
Faktorisierung	RSA	NFS $L_N\left(\frac{1}{3}; 1.92\right) \approx 2^{c \cdot \sqrt[3]{b_N}}$	3000	Shor	✗

Heutige Crypto in der Welt von Morgen

Ziel / Crypto Verfahren	Keyless	Symmetrisch	Asymmetrisch
Hashfunktionen	SHA-2, SHA-3, Whirlpool		
Key Agreement		✗	DH, ECDH
Vertraulichkeit (Verschlüsselung)		AES	RSA, ElGamal, ECIES
Integrität, Authentizität (Signaturen & MACs)		HMAC, CMAC	RSA, DSA, ECDSA

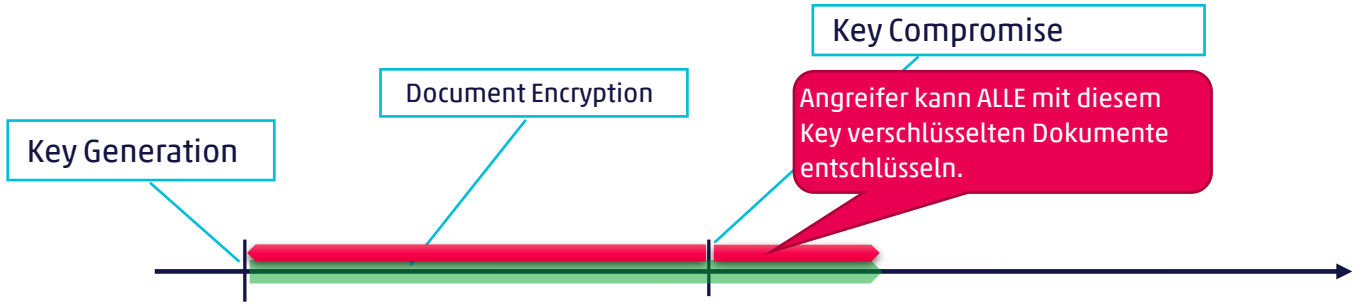
Keine Nichtabstreitbarkeit

Basierend auf DLP und Faktorisierung

Bedroht durch CRQC

Verschlüsselung & Signaturen (im Laufe der Zeit)

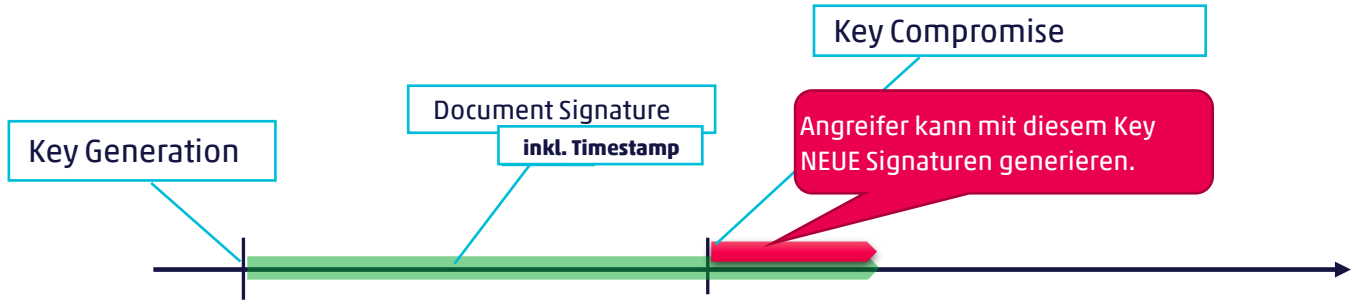
- Verschlüsselung



Verschlüsselung:

- "Store now, decrypt later"
- Nachträglicher Schutz nicht möglich
- Risiko muss bereits jetzt berücksichtigt werden

- Signaturen



Signaturen:

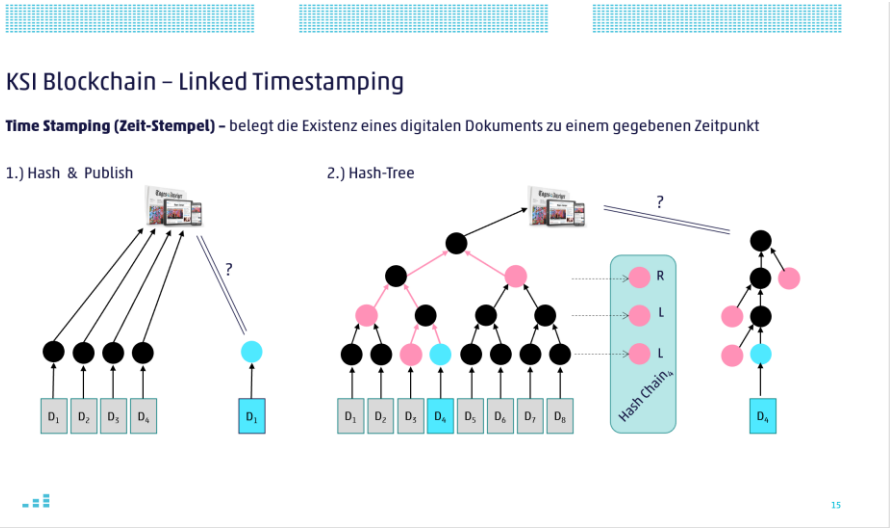
- Alte Signaturen nicht gefährdet...
- ... solange Timestamp gültig

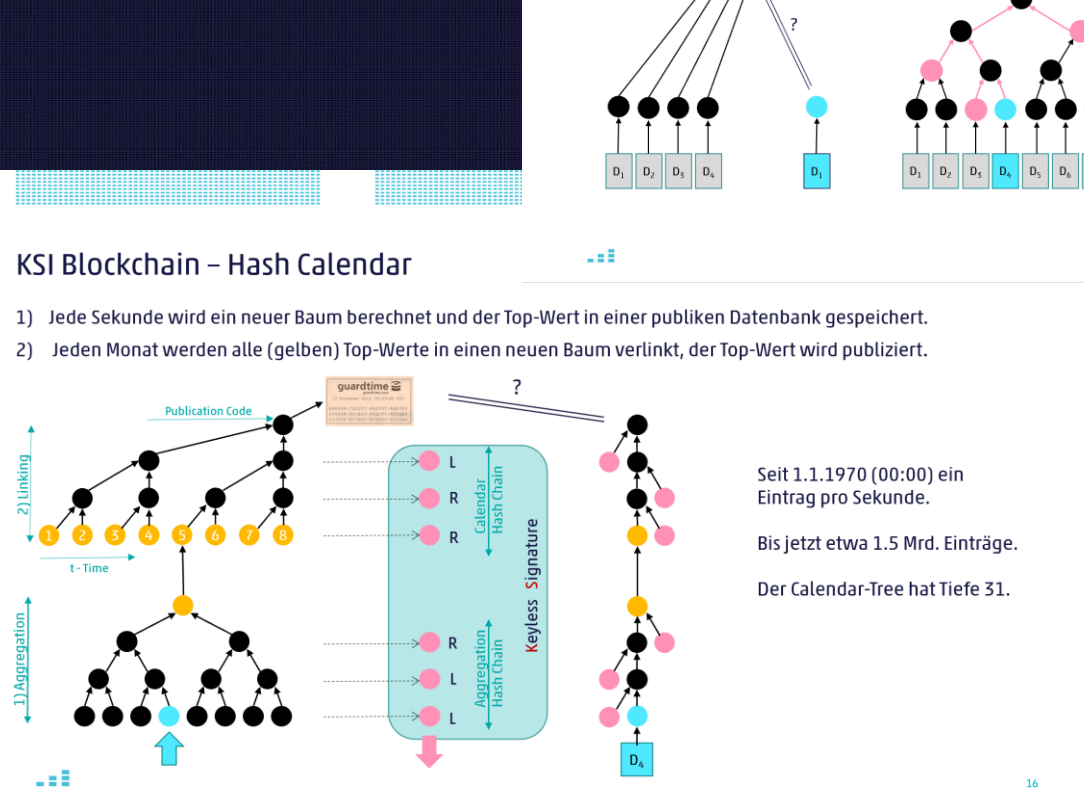
Muss bei Bedarf erneuert werden (BSI TR-ESOR-03125: Beweiserhaltung kryptographisch signierter Dokumente)

Insbesondere falls ebenfalls basierend auf asymmetrischer Crypto

Quantensicheres Timestamping: KSI Blockchain

KSI Blockchain – Quantensicheres Timestamping





LONG-TERM

KSI timestamps can be stored and verified indefinitely, without the need for complex crypto-lifecycle management. KSI timestamps are immune to quantum computing attacks, which makes them ideal for long term archiving and future-oriented projects.

eIDAS

Guardtime's KSI® Blockchain Timestamping Service is compliant with the eIDAS regulation and included in the European Trusted List.

KSI is the first blockchain-based technology to receive an eIDAS accreditation and marks an important step in the evolution of digital trust technologies. Accreditation was conducted by TÜV Nord, Germany.

https://guardtime.com/timestamping

TLS

- Sichere Kommunikation
  - Geheim: Key-Agreement und Verschlüsselung
  - Authentisch: Client/Server-Authentisierung und Daten-Authentisierung
- Verwendete Crypto-Primitiven:
  - Key Agreement: DH oder ECDH
  - Signaturen für Client/Server Authentisierung: RSA oder ECDSA
  - Hashfunktionen: SHA
  - Authentisierte Encryption: AES-GCM, AES-CCM
- Mit Forward Secrecy (basiert auf der Sicherheit von DH/ECDH)
- Key Agreement & Signaturen müssen durch **quanten-resistente Verfahren** ersetzt werden
  - **Wann ?**: **Signaturen** – nur für Authentisierung, keine Gefahr rückwirkend -> hat noch Zeit

    Key Agreement – unter der Annahme, dass es in 20 Jahren einen **CRQC** geben könnte ist es schon zu spät...

Für Kommunikation die in 20 Jahren noch geheim sein sollte.

Vielen Dank für Ihre Aufmerksamkeit_

Zuzana Trubini

info@cnlab-security.ch
+41 55 214 33 40

cnlab security AG
Obere Bahnhofstrasse 32b
CH-8640 Rapperswil-Jona
Switzerland

cnlab