



**Leibniz-Rechenzentrum**  
der Bayerischen Akademie der Wissenschaften

# Kapitel 8:

## Asymmetrische und hybride Kryptosysteme

- Asymmetrische Kryptosysteme
  - RSA
  - Sicherheit von RSA
  - Elliptische Kurven
- Schlüssellängen und Schlüsselsicherheit
- Hybride Kryptosysteme
- Elektronische Signatur
- Quantencomputer und quantensichere Kryptographie

- Jeder Partner besitzt Schlüsselpaar aus
  - persönlichem, geheim zu haltenden Schlüssel (private key)  
(wird NIE übertragen)
  - und öffentlich bekannt zu gebenden Schlüssel (public key)  
(kann über unsichere und öffentliche Kanäle übertragen werden)
- Protokoll:
  1. Alice und Bob erzeugen sich Schlüsselpaare:  $(k_e^A, k_d^A)$   $(k_e^B, k_d^B)$
  2. Öffentliche Schlüssel  $(k_e^A, k_e^B)$  werden geeignet öffentlich gemacht
  3. Alice will  $m$  an Bob senden; dazu benutzt sie Bobs öffentlichen Schlüssel
$$c = e(m, k_e^B)$$
  4. Bob entschlüsselt die Nachricht mit seinem privaten Schlüssel:
$$m = d(c, k_d^b) = d(e(m, k_e^b), k_d^b)$$
- Beispiele: RSA, DSA, ElGamal, ...

### Zielsetzung

- Effizienz / Performanz:
  - Schlüsselpaare sollen „einfach“ zu erzeugen sein.
  - Ver- und Entschlüsselung soll „schnell“ ablaufen.
- Veröffentlichung von  $k_e$  darf keine Risiken mit sich bringen
- Privater Schlüssel  $k_d$  darf nicht „einfach“ aus  $k_e$  ableitbar sein
  - D.h. Funktion  $f$  mit  $f(k_d) = k_e$  soll nicht umkehrbar sein („Einwegfunktion“)
- Einsatz zur **Verschlüsselung**:
  - Alice schickt Nachricht  $m$  mit Bobs Public Key verschlüsselt an Bob
  - Bob entschlüsselt den empfangenen Chiffretext mit seinem privaten Schlüssel
- Einsatz zur **elektronischen Signatur**:
  - Alice verschlüsselt ein Dokument mit ihrem privaten Schlüssel
  - Bob entschlüsselt das Dokument mit Alices öffentlichem Schlüssel

- Benannt nach den Erfindern: Rivest, Shamir, Adleman (1978)
- Sicherheit basiert auf dem **Faktorisierungsproblem**:
  - Geg. zwei große Primzahlen p und q (z.B. 200 Dezimalstellen):
  - $n=pq$  ist auch für große Zahlen einfach zu berechnen,
  - aber für gegebenes n ist dessen Primfaktorzerlegung sehr aufwendig
- Erfüllt alle Anforderungen an asymmetrisches Kryptosystem
- 1983 (nur) in USA patentiert (im Jahr 2000 ausgelaufen)
- Große Verbreitung, verwendet in:
  - TLS (Transport Layer Security)
  - PEM (Privacy Enhanced Mail) - S/MIME
  - PGP (Pretty Good Privacy)
  - GnuPG (GNU Privacy Guard)
  - SSH
  - ....

- Erzeugung eines Schlüsselpaars
- Verschlüsselung
- Entschlüsselung

# Erzeugung eines Schlüsselpaars

- Randomisierte Wahl von zwei ähnlich großen, unterschiedlichen Primzahlen, p und q
- $n = pq$  ist sog. RSA-Modul
- Euler'sche Phi-Funktion gibt an, wie viele positive ganze Zahlen zu n teilerfremd sind:  $\Phi(n) = (p - 1)(q - 1)$
- Wähle teilerfremde Zahl e mit  $1 < e < \Phi(n)$   
d.h. der größte gemeinsame Nenner von e und  $\Phi(n) = 1$ 
  - Für e wird häufig 65537 gewählt: Je kleiner e ist, desto effizienter ist die Verschlüsselung, aber bei sehr kleinen e sind Angriffe bekannt.
  - Der öffentliche Schlüssel besteht aus dem RSA-Modul n und dem Verschlüsselungsexponenten e.
- Bestimme Zahl d als multiplikativ Inverse von e bezüglich  $\Phi(n)$ 
$$d = e^{-1} \bmod \Phi(n)$$
  - Berechnung z.B. über den erweiterten Euklidischen Algorithmus
  - n und d bilden den privaten Schlüssel; d muss geheim gehalten werden

- Alice kommuniziert ihren öffentlichen Schlüssel  $(n, e)$  geeignet an Bob (Ziel hier: Authentizität von Alice, nicht Vertraulichkeit!)
- Bob möchte Nachricht  $M$  verschlüsselt an Alice übertragen:
  - Nachricht  $M$  wird als Integer-Zahl  $m$  aufgefasst, mit  $0 < m < n$   
d.h. Nachricht  $m$  muss kleiner sein als das RSA-Modul  $n$
  - Bob berechnet Ciphertext  $c = m^e \pmod{n}$
  - Bob schickt  $c$  an Alice
- Alice möchte Ciphertext  $c$  entschlüsseln
  - Alice berechnet hierzu  $m = c^d \pmod{n}$
  - Aus Integer-Zahl  $m$  kann Nachricht  $M$  rekonstruiert werden.

# Nomenklatur für kryptologische Verfahren

- Für Verschlüsselungsverfahren wird künftig die folgende Notation verwendet:

Ap	Öffentlicher (public) Schlüssel von A
As	Geheimer (secret) Schlüssel von A
Ap{m}	Verschlüsselung der Nachricht m mit dem öffentlichen Schlüssel von A
As{m} oder A{m}	Von A erstellte digitale Signatur von m
S[m]	Verschlüsselung von m mit dem symmetrischen Schlüssel S

### 1. Brute force:

- Ausprobieren aller möglichen Schlüssel
- Entspricht Zerlegung von  $n$  in die Faktoren  $p$  und  $q$
- Dauert bei großen  $p$  und  $q$  mit heutiger Technik hoffnungslos lange

### 2. Chosen-Ciphertext-Angriff (David 1982):

- Angreifer möchte Ciphertext  $c$  entschlüsseln, also  $m = c^d \pmod{n}$  berechnen
- Angreifer kann einen Ciphertext  $c'$  vorgeben und bekommt  $m'$  geliefert
- Angreifer wählt  $c' = s^e c \pmod{n}$ , mit Zufallszahl  $s$
- Aus der Antwort  $m' = c'^d \pmod{n}$  kann  $m = m' s^{-1}$  rekonstruiert werden.

## 1. Angriffe auf Signaturen (vgl. spätere Folien zur dig. Signatur)

- Multiplikativität von RSA  $m^e r^e = (mr)^e$  erlaubt die Konstruktion gültiger Signaturen für ein Dokument, das aus korrekt signierten Teildokumenten zusammengesetzt ist.

## 2. Timing-Angriff: [Kocher 1995]

- Überwachung der Laufzeit von Entschlüsselungsoperationen
- Über Laufzeitunterschiede kann privater Schlüssel ermittelt werden
- Gegenmaßnahme: Blinding; Alice berechnet statt  $c^d \pmod{n}$  mit einmaliger Zufallszahl  $r$

$$(r^e c)^d \pmod{n} = r c^d \pmod{n}$$

und multipliziert das Ergebnis mit der Inversen von  $r$ .

- Folge: Dauer der Entschlüsselungsoperationen hängt nicht mehr direkt nur von  $c$  ab, Timing-Angriff scheitert.

- Taktfrequenz-Seitenkanal für Timing Angriff auf SIKE (Supersingular Isogeny Key Encapsulation)
- HERTZBLEED bestimmt im Juni 2022 SIKE Schlüssel
  - Über Taktfrequenz und Timing-Informationen
  - Angriff entfernt möglich
  - Kompletter Schlüssel kann bestimmt werden
- SIKE (Supersingular Isogeny Key Encapsulation)
  - Aussichtsreicher Kandidat für Post-Quanten Krypto Ausschreibung der NIST
  - Im Juli 2022 vollständig gebrochen durch Castryck und Decru

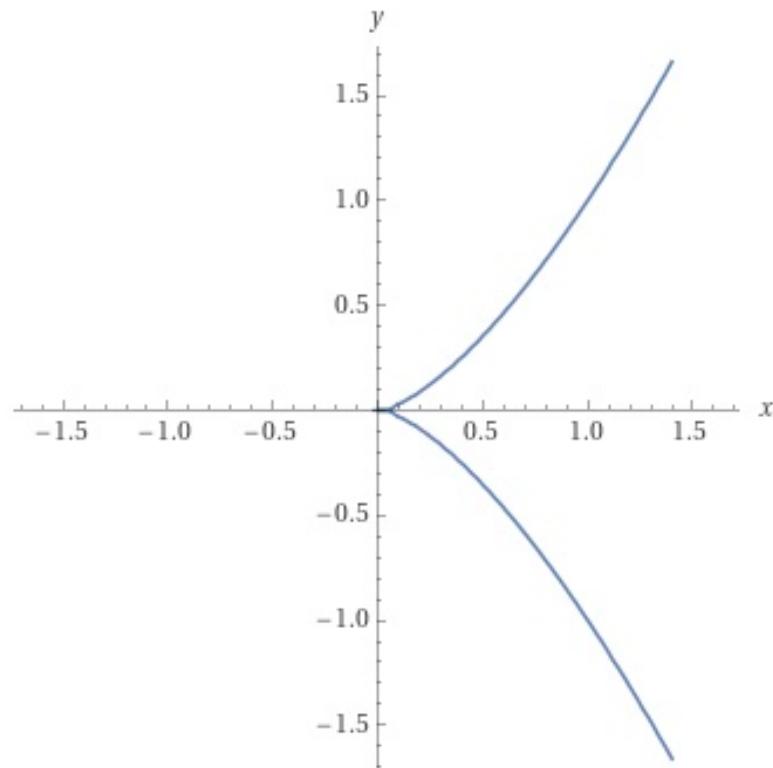
- Mathematische Angriffe lassen sich auf Faktorisierung zurückführen
- Schnellster bekannter Algorithmus: **General Number Field Sieve (GNFS)**, vgl. [Silv 01]
  - Laufzeitkomplexität  $L(N) = e^{(c+o(1)) \cdot \sqrt[3]{\log(N)} \cdot \sqrt[3]{\log(\log(N))^2}}$
  - Speicherplatzkomplexität  $\sqrt{L(N)}$
- Angriffe werden ggf. einfacher:
  - Wenn die Anzahlen der Ziffern von p und q große Unterschiede aufweisen (z.B.  $|p| = 10$  und  $|q| = 120$ )
  - Falls  $d < 1/3 \cdot \sqrt[4]{n}$ , kann d leicht berechnet werden
  - Die ersten  $m/4$  Ziffern oder die letzten  $m/4$  Ziffern von p oder q sind bekannt.
- Vgl. [Boneh 1999]: Twenty Years of attacks on the RSA cryptosystem,  
<http://crypto.stanford.edu/~dabo/pubs/papers/RSA-survey.pdf>

# Elliptische Kurven

- Funktionen der folgenden Form
  - $y^2 = x^3 + ax + b$

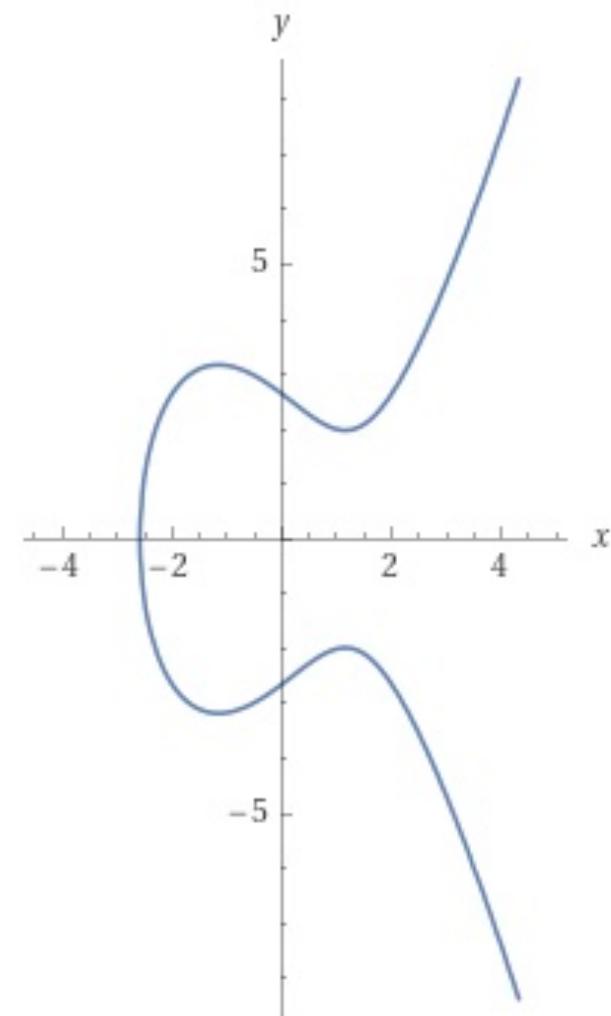
- Beispiel:  $a=b=0$

$$y^2 = x^3$$



# Elliptische Kurven

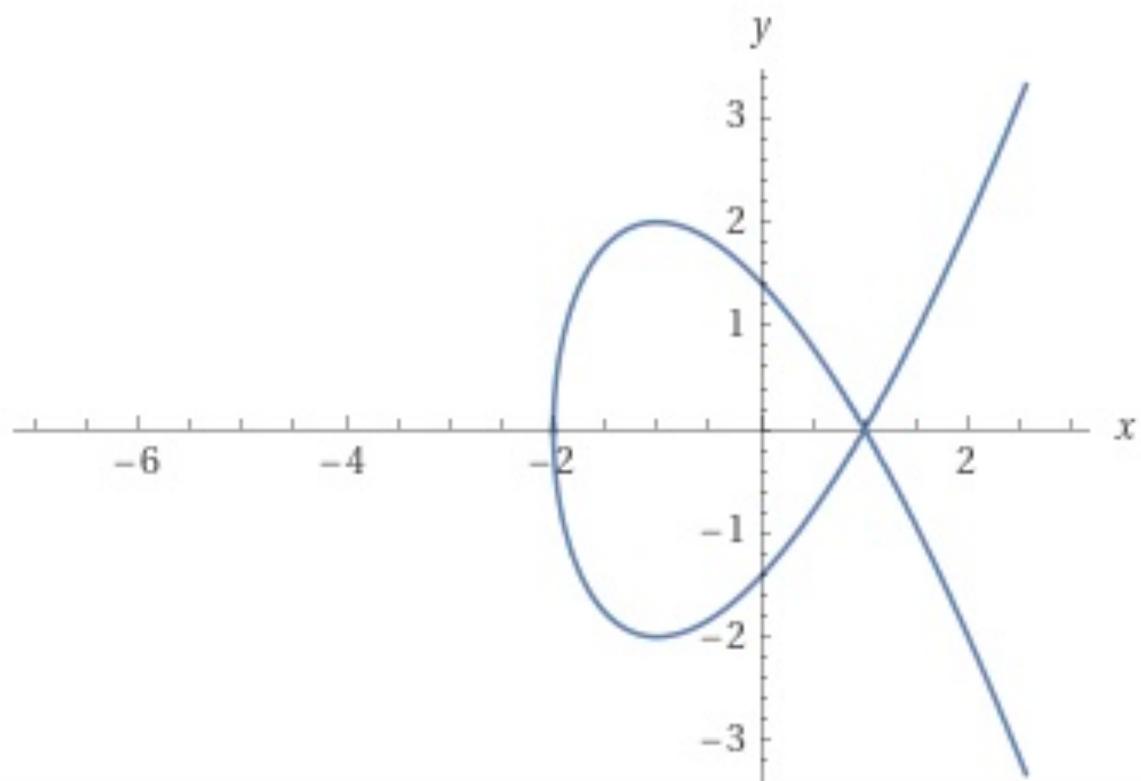
- Funktionen der folgenden Form
  - $y^2 = x^3 + ax + b$



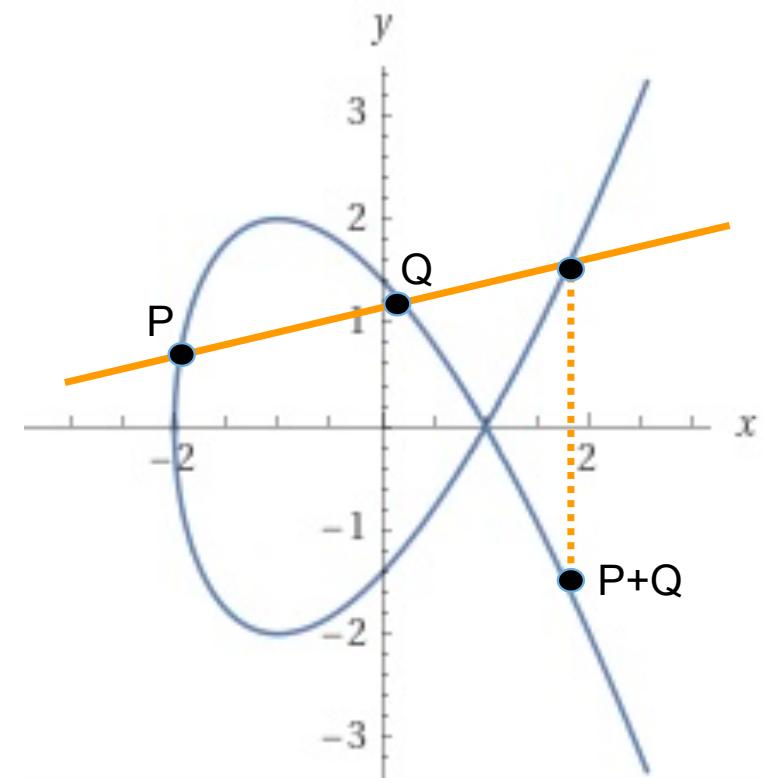
# Elliptische Kurven

- Funktionen der folgenden Form
  - $y^2 = x^3 + ax + b$

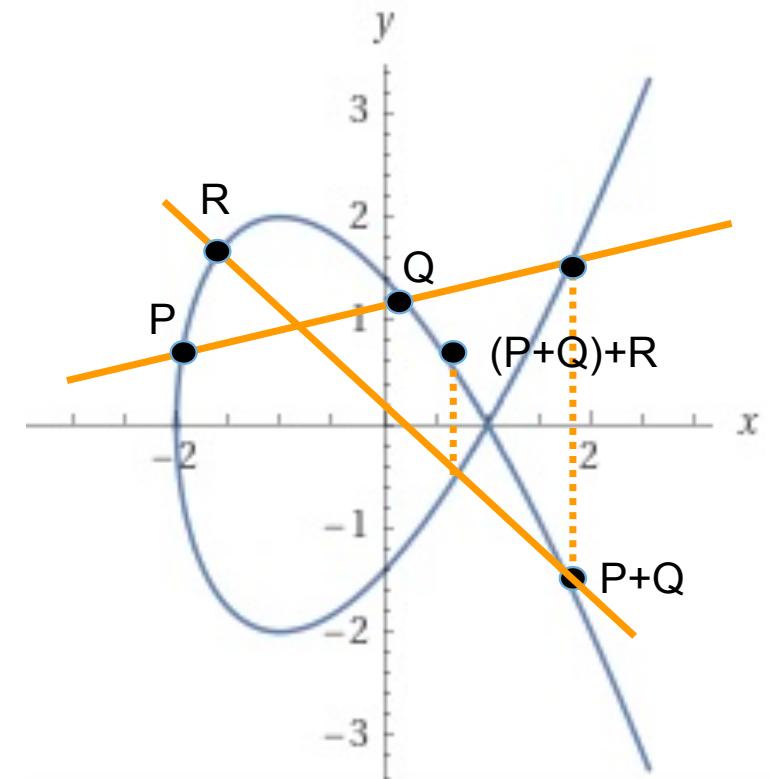
- Beispiel:
  - $y^2 = x^3 - 3x + 2$



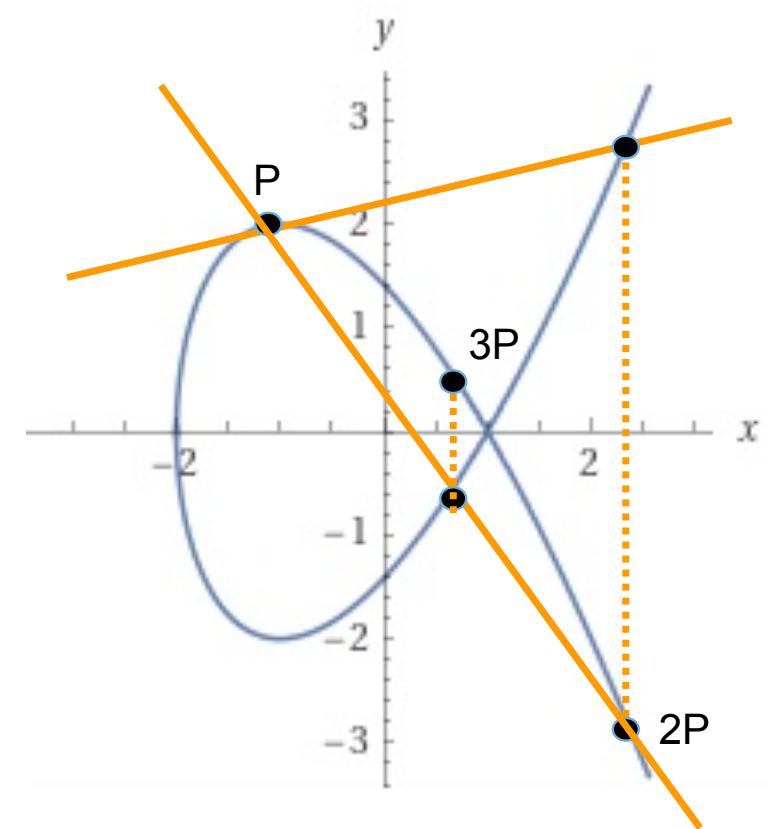
- Wähle zwei Punkt  $P$  und  $Q$  auf der Kurve
- Gerade durch  $P$  und  $Q$  schneidet Kurve in drittem Punkt
- Spiegelung des Punktes an der X-Achse liefert  $P+Q$



- Wähle zwei Punkt  $P$  und  $Q$  auf der Kurve
- Gerade durch  $P$  und  $Q$  schneidet Kurve in drittem Punkt
- Spiegelung des Punktes an der X-Achse liefert  $P+Q$
- Wähle Punkt  $R$  Gerade durch  $(P+Q)$  und  $R$  liefert  $(P+Q)+R$



- Wähle Tangente an Punkt  $P$
- Spiegelung am Schnittpunkt liefert  $(P+P)$
- Weitere Gerade von  $2P$  nach  $P$  liefert  $3P$
- Verfahren lässt sich beliebig fortsetzen:  $4P, 5P, 6P, 7P, \dots, NP$
- Gegeben sei  $NP$ . Wie aufwändig ist es  $N$  zu berechnen?
  - Linearer Aufwand  $O(N) = N$



## Komplexität erhöhen

- Elliptische Kurven sind Gruppen, d.h. Assoziativgesetz gilt
  - $4P = 3P+P = 2P+2P$
- NP Zerlegung in Zweierpotenzen, bzw. Darstellung von N als Binärzahl
  - $117P = 64P + 32P + 16 + 4P + 1$
  - Komplexität  $O(N) = \ln(N)$
  - Als Einwegfunktion nutzbar

- Alice und Bob einigen sich auf Punkt P
- Alice wählt geheimes A, Bob geheimes B
- Alice berechnet AP, Bob BP
- Öffentliche Schlüssel AP und BP können verteilt werden
- Alice berechnet mit geheimem Schlüssel A:  $BP^*A$
- Bob berechnet:  $AP^*B$
- Damit haben beide  $(BP)^*A = BA^*P = (AP)^*B$
- Dieses gemeinsame Geheimnis kann als symmetrischer Schlüssel verwendet werden (Elliptic Curve Diffie-Hellman)
- In der Kryptographie werden keine reellen Werte sondern ganzzahlige verwendet, d.h.  
 $y^2 = x^3 + a * x + b \text{ mod } p$

# Elliptic Curve Based Signature Algorithms

- ECDSA, ECGDSA und EC-Schnorr (s. [BSI-TR-03111](#))
- ECDSA Signatur (r,s)
- Input:
  - Starke Kryptographische Hash-Funktion
  - Privater Schlüssel von A
  - Parameter der Elliptischen Kurve (p,a,b,G,n,h)
    - p Primzahl
    - a,b Parameter der Kurvenfunktion
    - G Basispunkt als Generator des Körpers  $E(\mathbb{F}_p)$  (analog zu P aus vorigen Beispielen)
    - n Ordnung von G; Typischerweise Primzahl mit  $|n| \geq 224$  bit (Länge ab 224 bit)
    - h Cofaktor von G in  $E(\mathbb{F}_p)$ ;  $h = \frac{\#E(\mathbb{F}_p)}{n}$

## ECDSA Signatur (r,s) Berechnung

1.  $k = \text{Zufallszahl}(\{1, 2, \dots, n-2\})$

2.  $Q = kG$

3.  $r = OS2I(FE2OS(x_Q)) \pmod{n}$

IF  $r == 0$  goto 1.

(mit  $x_Q$  X-Koordinate von Q)

OS2I = Octet string to Integer Conversion

FE2OS = Finite Field Element to Octet String

4.  $k_{inv} = k^{-1} \pmod{n}$

5.  $s = k_{inv} * (r * d_A + OS2I(H_\tau(M))) \pmod{n}$

IF  $s == 0$  goto 1.

$H_\tau$  Hashfunktion liefert die  $\tau$  ersten Bits ( $\tau = \lceil \ln(n) \rceil$ ),  $d_A$  private key von A

6. Output (r,s)

- Asymmetrische Kryptosysteme
  - RSA
  - Sicherheit von RSA
  - Elliptische Kurven
- Schlüssellängen und Schlüsselsicherheit
- Hybride Kryptosysteme
- Elektronische Signatur
- Quantencomputer und quantensichere Kryptographie

## Einflussfaktoren

- Symmetrisches oder asymmetrisches Verfahren ?
- Algorithmus
  
- PC-/softwarebasierter Angriff, oder
- Angriff mit dedizierter Hardware
  - Angriff mit integrierter Schaltung (ASIC, application specific integrated circuit)
  - Angriff mit programmierbarer integrierten Schaltung (FPGA, field programmable gate array)
  - GPGPU (General-purpose computing on graphics processing units)
  
- Kosten und Ressourcenbedarf

# Angriffe auf symmetrische Kryptosysteme

- Brute-Force Angriff
  - Durchsuchen des gesamten Schlüsselraums
  - Im Mittel ist halber Schlüsselraum zu durchsuchen
- Referenzahlen; Größenordnungen (gerundet)

	Größenordnung
Sekunden in einem Jahr	$3 * 10^7$
Alter des Universums in Sekunden	$4 * 10^{17}$
Schlüsselraum bei 64 Bit Schlüssellänge	$2 * 10^{19}$
Masse des Mondes [kg]	$7 * 10^{22}$
Masse der Erde [kg]	$6 * 10^{24}$
Masse der Sonne [kg]	$2 * 10^{30}$
Schlüsselraum bei 128 Bit Schlüssellänge	$3 * 10^{38}$
Anzahl Elektronen im Universum	$10^{77} - 10^{79}$

# RSA Schlüssellängen

- RSA Challenge: Belohnung für das Brechen von RSA Schlüsseln, z.B. durch Faktorisierung

Dezimalstellen	Bits	Datum	Aufwand	Algorithmus
100	332	April 1991	7 Mips Jahre	Quadratisches Sieb
110	365	April 1992	75 Mips J.	
120	398	Juni 1993	830 Mips J.	
129	428	April 1994	5000 Mips J.	
130	431	April 1996	1000 Mips J.	General Number Field Sieve (GNFS)
140	465	Februar 1999	2000 Mips J.	
155	512	August 1999	8000 Mips J.	
160	530	April 2003	k.A.	GNFS(Lattice Sieve)
174	576	Dez. 2003	k.A.	GNFS(Lattice/Line Sieve)
193	640	Nov. 2005	30 2,2-GHz-Opteron-Jahre	GNFS

## RSA Schlüssellängen (Forts.)

- RSA Challenge wurde 2007 eingestellt
  - rund \$30.000 Preisgeld ausbezahlt
- RSA-768 wurde 2009 von Kleinjung et al. „geknackt“
  - hätte \$50.000 Preisgeld eingebracht

Dezimalstellen	Bits	Datum	Aufwand	Algorithmus
232	768	Dez. 2009	1/2 Jahr 80 CPUs (Vorauswahl)	GNFS

- Bereits 2007 wurde von Kleinjung et al. die 1039. Mersenne-Zahl (1039-Bit-Zahl) faktorisiert
  - war allerdings nicht Bestandteil der RSA Challenge

## Symmetrisch vs. Asymmetrisch

- Verschiedene Institutionen geben Vergleiche heraus  
Bits of Security (äquiv. Schlüssellänge symmetrischer Verfahren)
  - NIST (National Institute of Standards and Technology) 2007:

Bits of Security	80	112	128	192	256
Modullänge (pq)	1024	2048	3072	7680	15360

- NESSIE (New European Schemes for Signatures, Integrity and Encryption) (2003)

Bits of Security	56	64	80	112	128	160
Modullänge (pq)	512	768	1536	4096	6000	10000

# BSI Empfehlungen für Schlüssellängen



- BSI-TR-02102-1; Technische Richtlinie vom 02.02.2024:  
[https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?\\_\\_blob=publicationFile&v=9](https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?__blob=publicationFile&v=9) Richtet sich an Entwickler die ab 2024 neue Systeme planen
  - TR wird jährlich überprüft und ggf. angepasst
- Bis 2022 Sicherheitsniveau von 100 Bit danach mind. 120 Bit

<b>Symmetrische Verfahren</b>		<b>Asymmetrische Verfahren</b>		
Ideale Blockchiffre	Idealer MAC	RSA	DSA/DLIES	ECDSA/ECIES
120	120	2800	2800	240

- Ab 2023 werden 128 Bit bzw.  $\geq 3.000$  Bit gefordert

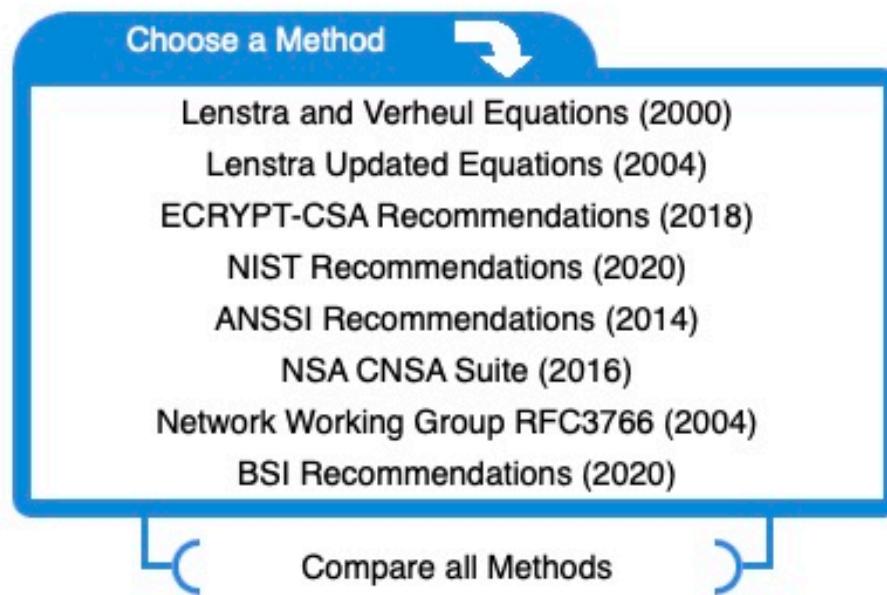
Blockchiffre	MAC	RSA	DH $\mathbb{F}_p$	ECDH	ECDSA
128	128	3000	3000	250	250

**Tabelle 1.2:** Empfohlene Schlüssellängen für verschiedene kryptographische Verfahren.

[[https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?\\_\\_blob=publicationFile](https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?__blob=publicationFile)]

- Vergleich der verschiedenen Empfehlungen:

<https://www.keylength.com/>



## 1 Reference for the comparison

You can enter the year until when your system should be protected and see the corresponding key sizes or you can enter a key/hash/group size and see until when you would be protected.

Enter a year:

2024

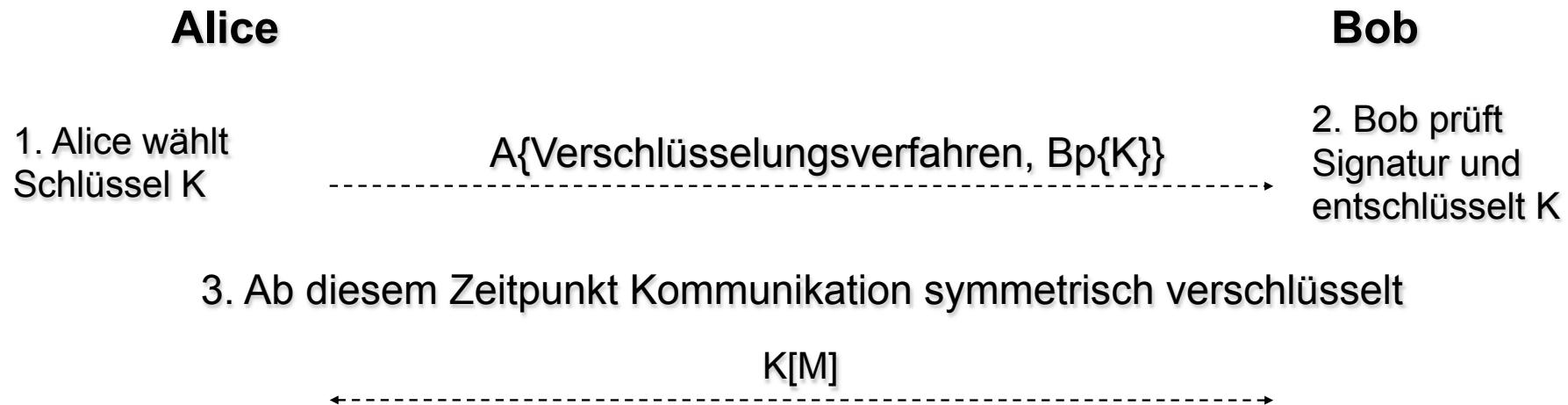
## 2 Compare

Method	Date	Symmetric	Factoring Modulus	Discrete Logarithm Key	Elliptic Curve	Hash
[1] Lenstra / Verheul <a href="#">?</a>	2024	89	2113 1696	157	2113	167 177
[2] Lenstra Updated <a href="#">?</a>	2024	84	1507 1756	168	1507	168 168
[3] ECRYPT	2018 - 2028	128	3072	256	3072	256 256
[4] NIST	2019 - 2030	112	2048	224	2048	224 224
[5] ANSSI	2021 - 2030	128	2048	200	2048	256 256
[6] NSA	-	256	3072	-	-	384 384
[7] RFC3766 <a href="#">?</a>	-	-	-	-	-	-
[8] BSI	2023 - 2026	128	3000	250	3000	250 256

- Asymmetrische Kryptosysteme
  - RSA
  - Sicherheit von RSA
  - Elliptische Kurven
- Schlüssellängen und Schlüsselsicherheit
- Hybride Kryptosysteme
- Elektronische Signatur
- Quantencomputer und quantensichere Kryptographie

# Hybride Kryptosysteme

- Vereinen Vorteile von symmetrischen und asymmetrischen Verfahren
- Asymmetrisches Verfahren zum Schlüsselaustausch
- Symmetrisches Verfahren zur Kommunikationsverschlüsselung

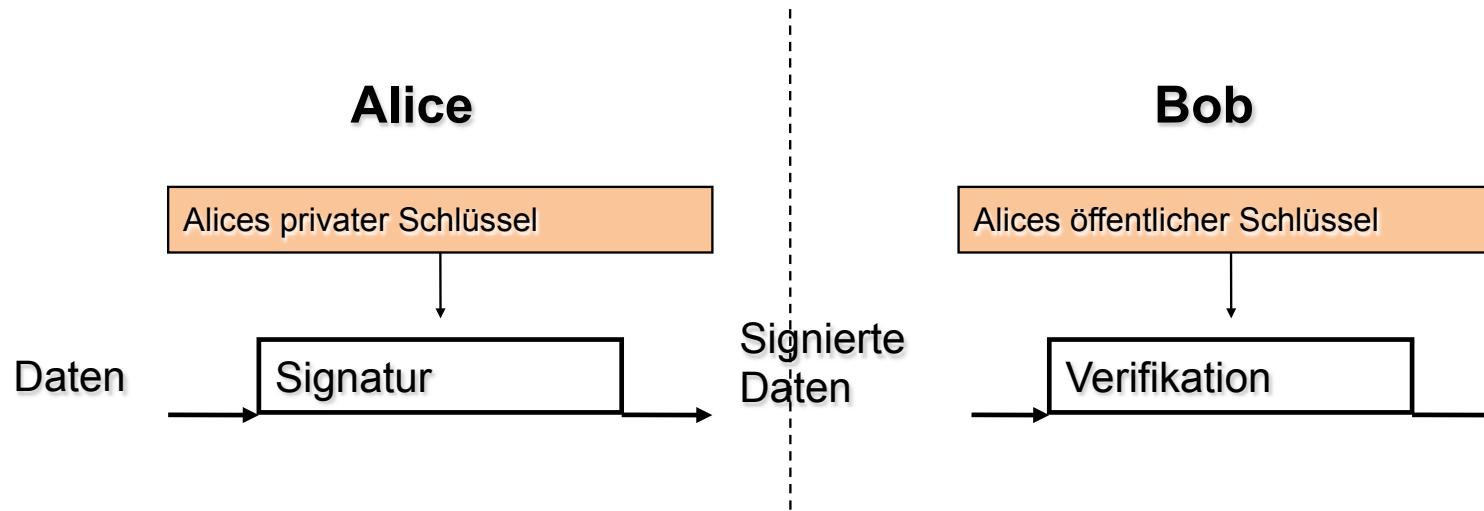


- Beispiele für hybride Verfahren: SSL/TLS, PGP, SSH,...
  - Oftmals neuer Schlüssel pro Nachricht oder Zeiteinheit, aus K abgeleitet

- Asymmetrische Kryptosysteme
  - RSA
  - Sicherheit von RSA
  - Elliptische Kurven
- Schlüssellängen und Schlüsselsicherheit
- Hybride Kryptosysteme
- Elektronische Signatur
- Quantencomputer und quantensichere Kryptographie

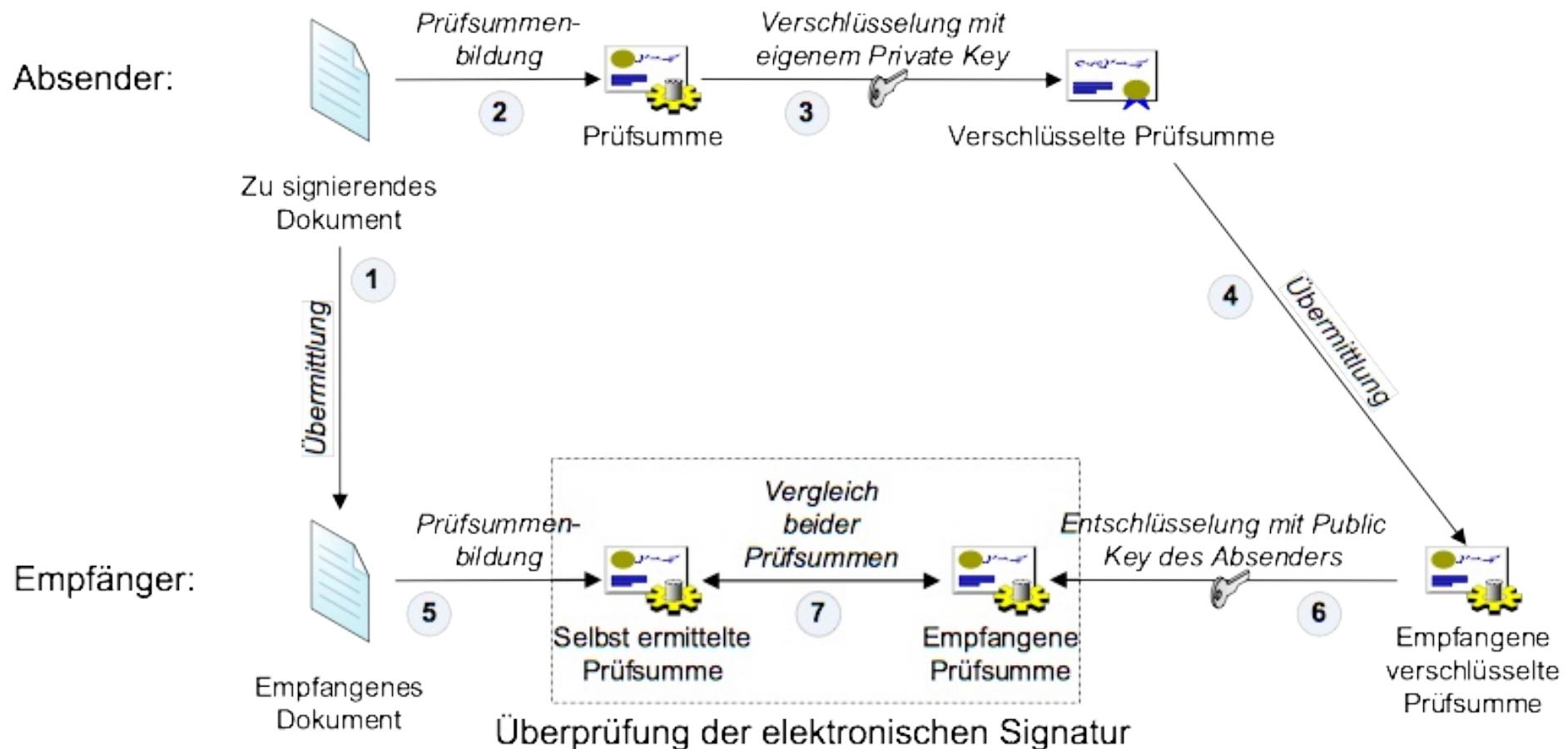
# Elektronische Signatur

- Alice „signiert“ Daten mit ihrem privaten Schlüssel
- Jeder kann die Signatur mit Alices öffentlichem Schlüssel verifizieren



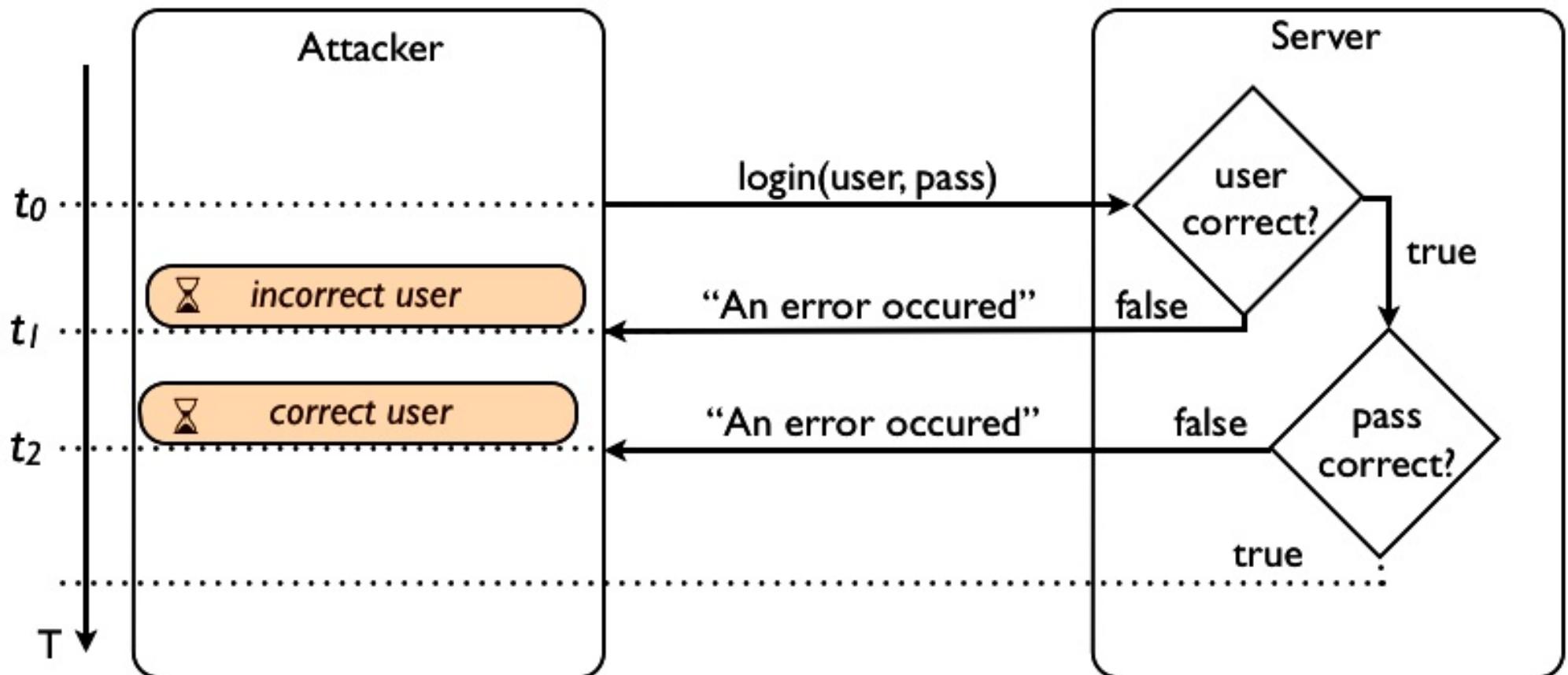
- Asymmetrische Verfahren sind im Vergleich sehr langsam
- Daher i.d.R. nicht Signatur der gesamten Daten
- **Lediglich kryptographischer Hash-Wert der Daten wird signiert** (digitaler Fingerabdruck der Daten)

# Signatur und deren Verifikation



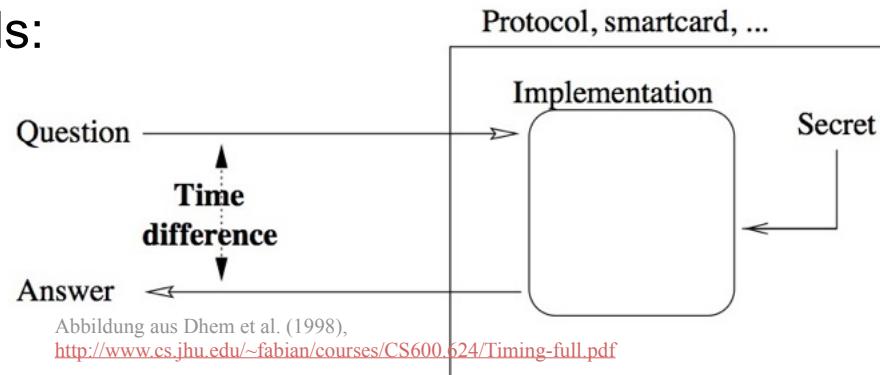
# Timing-Angriffe

- Zunächst am Beispiel Webanwendungen:

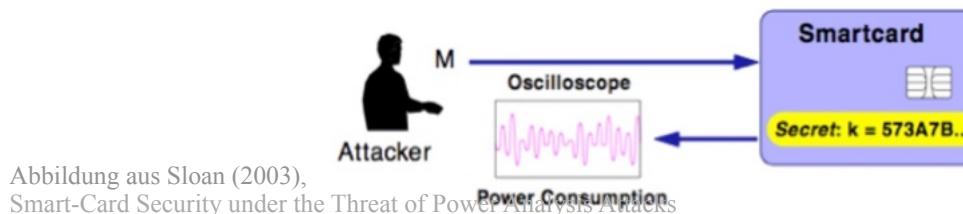


# Timing-Angriffe

## ■ Auf RSA-Smartcards:



- Different power consumption when operating on logical ones vs. logical zeroes.



## Analogie zur Unterschrift

- Zentrale Anforderungen an die (analoge) Unterschrift:
  1. Perpetuierungsfunktion: Dokument und Unterschrift sind dauerhaft.
  2. Echtheitsfunktion: Die Unterschrift ist authentisch.
  3. Die Unterschrift kann nicht wiederverwendet werden.
  4. Abschlussfunktion: Unterschrift ist räumlicher Abschluss des Dokuments; dieses kann später nicht verändert werden.
  5. Beweisfunktion: Unterzeichner kann seine Unterschrift später nicht leugnen.
- Weitere Anforderungen?
- Bei der Unterschrift auf Papier ist keine dieser Anforderungen vollständig erfüllt! Trotzdem wird die Unterschrift im Rechtsverkehr akzeptiert. Ihre Funktion wird durch Rahmenbedingungen gesichert.

## Erfüllung oder Anforderungen?

1. Perpetuierungsfunktion: Fälschungssicher und dauerhaft
  2. Echtheitsfunktion: Authentizität sichergestellt
  3. **Wiederverwendbarkeit: Wie gewünscht nicht gegeben**
  4. Abschlussfunktion: Nicht veränderbar
  5. Beweisfunktion: Unterschrift ist nicht zu leugnen
- 
1. Solange privater Schlüssel geheim gehalten wird.
  2. Abhängig von zweifelsfreier Zuordnung des Schlüsselpaares zu einer Identität (Zertifizierung, CA)
  3. Digitale Signatur „beinhaltet“ den Dateninhalt
  4. vgl. 3.
  5. Jeder kann Signatur bzw. Echtheit mit öffentlichem Schlüssel des Unterzeichners verifizieren.

- Asymmetrische Kryptosysteme
  - RSA
  - Sicherheit von RSA
  - Elliptische Kurven
- Schlüssellängen und Schlüsselsicherheit
- Hybride Kryptosysteme
- Elektronische Signatur
- Quantencomputer und quantensichere Kryptographie

# Kapitel 8:

# Quantencomputing

- Was ist ein Quantencomputer
- Qubits, Quantegatter
- MQV: Quantencomputer am LRZ
- Faktorisierungsproblem und Shor-Algorithmus
- Quantensichere Kryptographie

# Qubit

- Qubit kann wie ein klassisches Bit zwei Zustände annehmen:
  - $|0\rangle$  und  $|1\rangle$
  - Dirac Schreibweise, spricht sich cat(0) und cat(1)
- Superposition heißt: Qubit Psi kann beliebige Überlagerungszustände annehmen
  - $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$
  - $\alpha$  und  $\beta$  bezeichnen die Amplitude, d.h. den „Anteil“ mit dem die beiden Zustände in der Superposition enthalten sind
  - z.B.  $|\Psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$
  - Das Quadrat der Amplitude beschreibt die Wahrscheinlichkeit mit der der Wert gemessen wird, d.h.  $|\alpha|^2 + |\beta|^2 = 1$
  - $|\alpha|^2$  ist die Wahrscheinlichkeit 0 zu messen; im Bsp.  $(\frac{1}{\sqrt{2}})^2 = 0,5$  d.h. 50 % Wahrscheinlichkeit

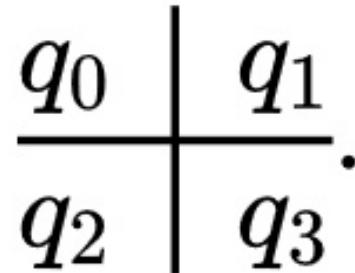
## Qubit - Messung

- Bei der Messung kollabiert die Superposition, d.h.
  - $|\Psi\rangle = 1|0\rangle + 0|1\rangle = |0\rangle = 0$  oder
  - $|\Psi\rangle = 0|0\rangle + 1|1\rangle = |1\rangle = 1$
- d.h. nach der Messung gilt  $|\Psi\rangle = |1\rangle = 1$  oder  $|\Psi\rangle = |0\rangle = 0$

- Im klassischen Rechner verwandeln sog. Gatter Eingangsbitfolgen in Ausgangsbitfolgen
- Quanten-Gatter
  - kann logische Operationen (und weitere) abbilden
  - Verschränkung von Qubits:
    - Qubits sind so miteinander verknüpft, dass Änderung an Einem zu Änderung am Anderen führt - egal wie weit sie voneinander entfernt sind
  - Wegen der **Superposition**:
    - **Alle** Zustände der Eingangsbitfolge kann ein Quanten-Gatter
    - **gleichzeitig** in **alle** Zustände der Ausgangsbitfolge umwandeln
  - Quantenparallelismus - das ist der 

## Beispiel: Sudoku

- 2x2 Sudoku: Jede Zahl darf in jeder Spalte und jeder Zeile nur ein mal vorkommen
- 16 Möglichkeiten



- Wie löst das ein klassischer Rechner?
- Genau wie Sie:  
Nacheinander alle Möglichkeiten prüfen.

0   0	1   0	0   1	1   1
0   0	0   0	0   0	0   0
0   0	1   0	0   1	1   1
1   0	1   0	1   0	1   0
0   0	1   0	0   1	1   1
0   1	0   1	0   1	0   1
0   0	1   0	0   1	1   1
1   1	1   1	1   1	1   1

## Beispiel: Sudoku mit Quantencomputer

- Jede mögliche Kombination lässt sich mit 4 Qubits darstellen:

$$|\psi\rangle = \frac{|q_0\rangle}{|q_2\rangle} \begin{array}{c|c} & |q_1\rangle \\ \hline |q_3\rangle & \end{array}$$

- Und alle können in Superposition (dargestellt als  $|+\rangle$ ) gebracht werden:

$$|\psi\rangle = \frac{|+\rangle}{|+\rangle} \begin{array}{c|c} & |+\rangle \\ \hline |+\rangle & \end{array} = “\frac{0}{0}|0\rangle + \frac{1}{0}|0\rangle + \dots + \frac{0}{1}|1\rangle + \dots + \frac{1}{1}|1\rangle”$$

- Damit lassen sich alle Kombinationen gleichzeitig darstellen und berechnen
- Wie viele Schritte braucht der QC?
- 1

## Qubits und Rechenleistung

- Mit 4 Qubits sind  $2^4$  d.h. 16 Zustände möglich
- Jedes weitere Qubit erhöht die Anzahl der Zustände um den Faktor 2 -> exponentielles Wachstum
- **ABER** viele Herausforderungen:
  - Qubits und Quantenzustände sind sehr störungsanfällig
  - Kohärenzzeit (Zeit in der Qubits in Superposition oder verschränkt sind) aktuell noch sehr kurz
  - Messung des Ergebnisses
  - Ergebnis ist wieder Superposition, Messung liefert einen Zustand der Superposition

# Technologien für QC

- Supraleitende Schaltkreise
  - Googles Sycamore 53 Qubits (2019)
  - IBM - Osprey 433 Qubits (2022)
- Ionenfallen
- Neutral-Atome
- Spin-Qubits
- Photonen
- Munich Quantum Valley:



(kommt)



(kommt)

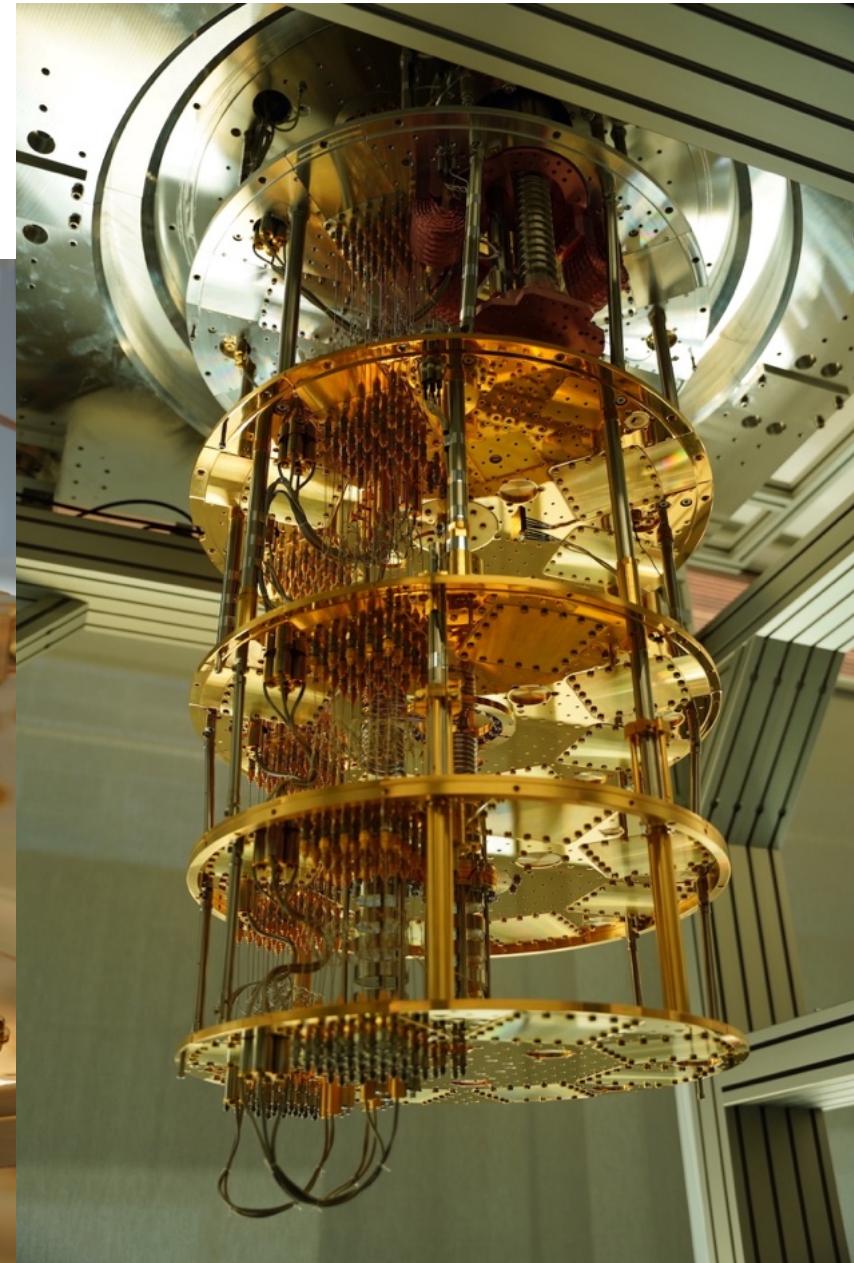
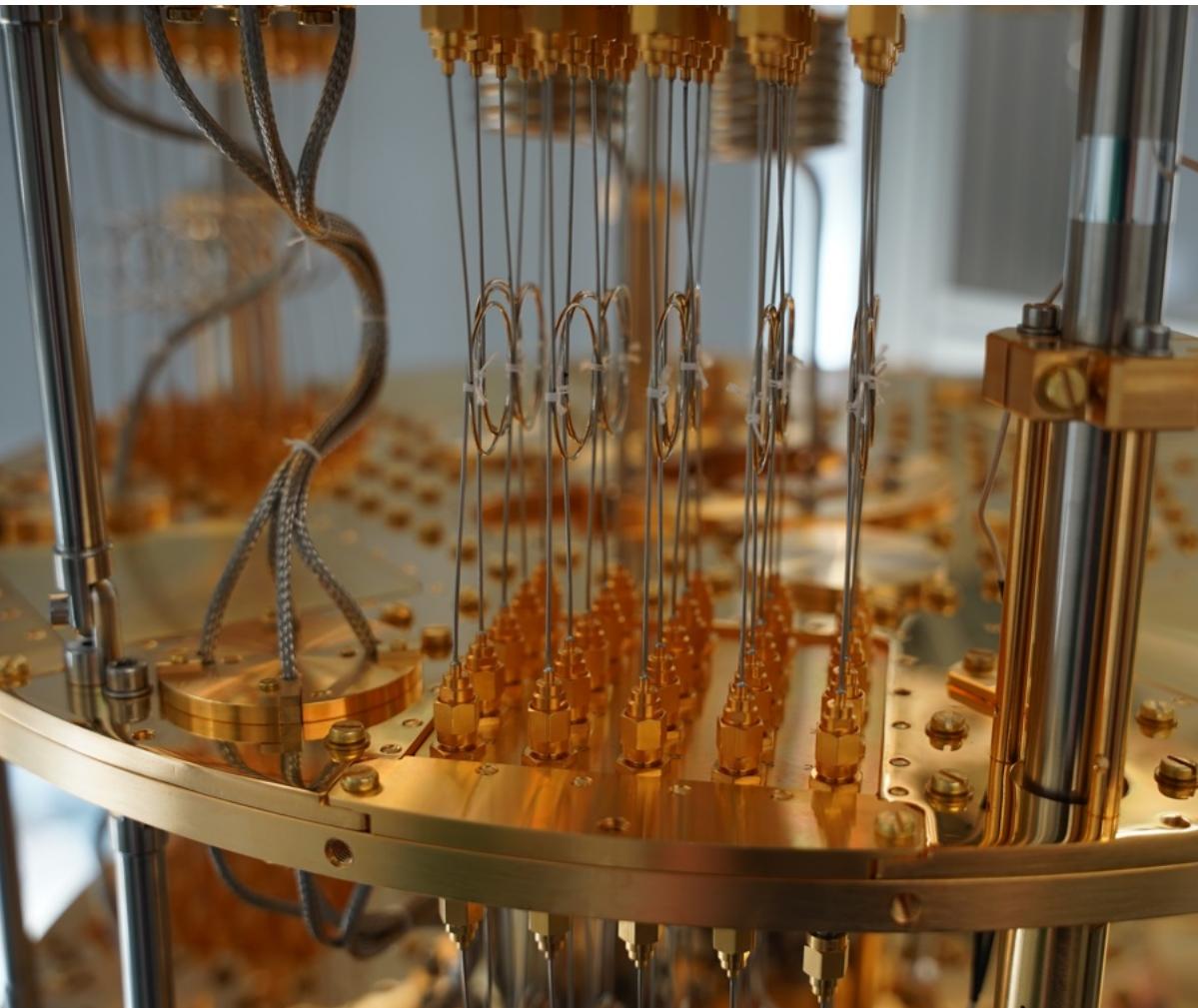


<https://www.munich-quantum-valley.de/de/forschung/forschungsbereiche>

# Quantencomputer am LRZ

## Kryostat

lrz



LRZ: Digitalisierungspartner der Wissenschaft  
LRZ Quantum Integration Centre (QIC)



**Quantum am LRZ**

- Integration von Supercomputing mit Quantencomputing
- Integration von Quantencomputing-Diensten in das LRZ-Portfolio
- Integration in der Quanten-Community: Das LRZ als Schnittstelle zwischen Entwickler:innen, Unternehmen, Anwender:innen

Als fester Bestandteil und in enger Kooperation mit:

Munich Quantum Valley logo, featuring a stylized infinity symbol.  
Walther Meißner Institut logo, featuring three concentric circles.

# Quantum Computing am LRZ



**On-premise  
Quantencomputing-Systeme**

**Quantum Computing  
Dienste**

**High-performance  
Quantum Computing  
(HPCQC)**

**User Community  
Aus- & Weiterbildung**



## Hardware Quantum Simulators

### Eviden (Atos) Qaptiva (QLM):

- Simulates up to 38 qubits
- Includes noise models
- Available now for access

## Software Quantum Simulators

### On SuperMUC-NG (pure CPU):

#### IntelQS simulator

- Simulates up to 42 qubits
- Available now for access

#### Qiskit

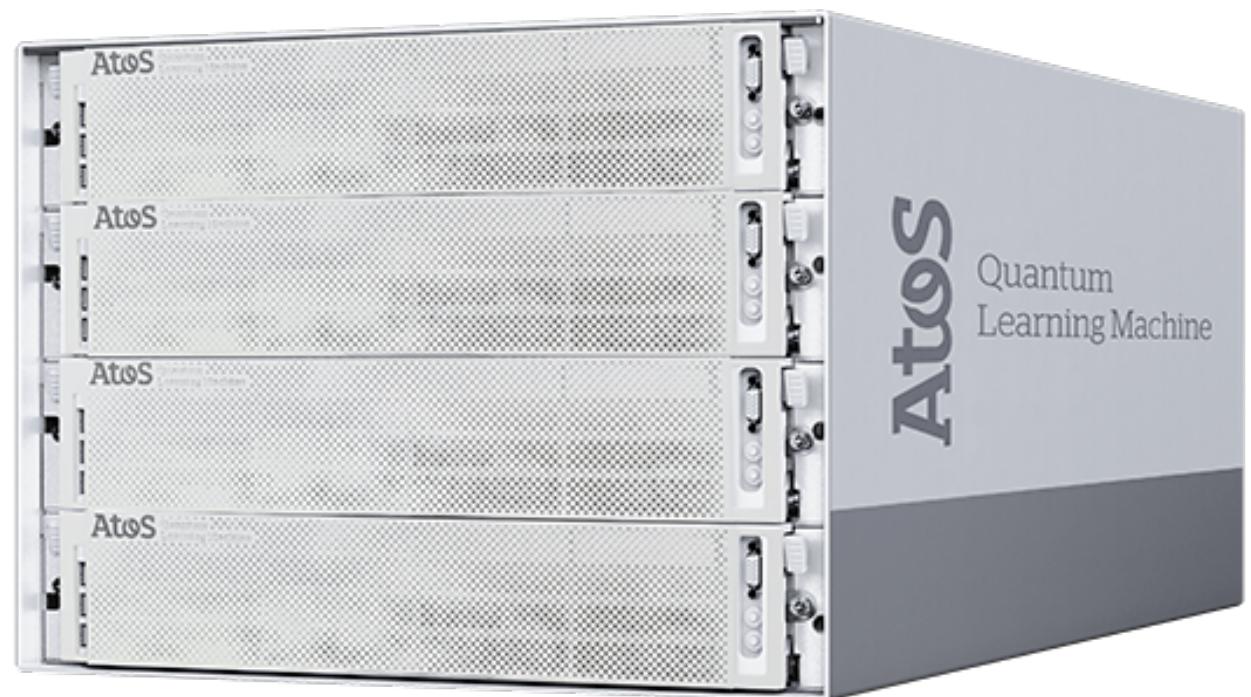
- MPI support on SuperMUC-NG (in prep)

### On SuperMUC-NG Phase 2 (Intel GPUs):

- Xanadu PennyLane (in prep)

### On GPU cluster (Nvidia GPUs):

- CUDA Quantum (in prep)



User documentation is in preparation and a pilot consulting service is launching in H1-2024.



## Quantum Systems at LRZ

### Superconducting:

System 1: 5 qubits (research system)

System 2: 20 qubits (research)

System 3: 20 qubits

Accessible: H1-2024 to Bavarian,  
German and European users in pilot  
phase initially, then wider access

System 4: 50+ qubits

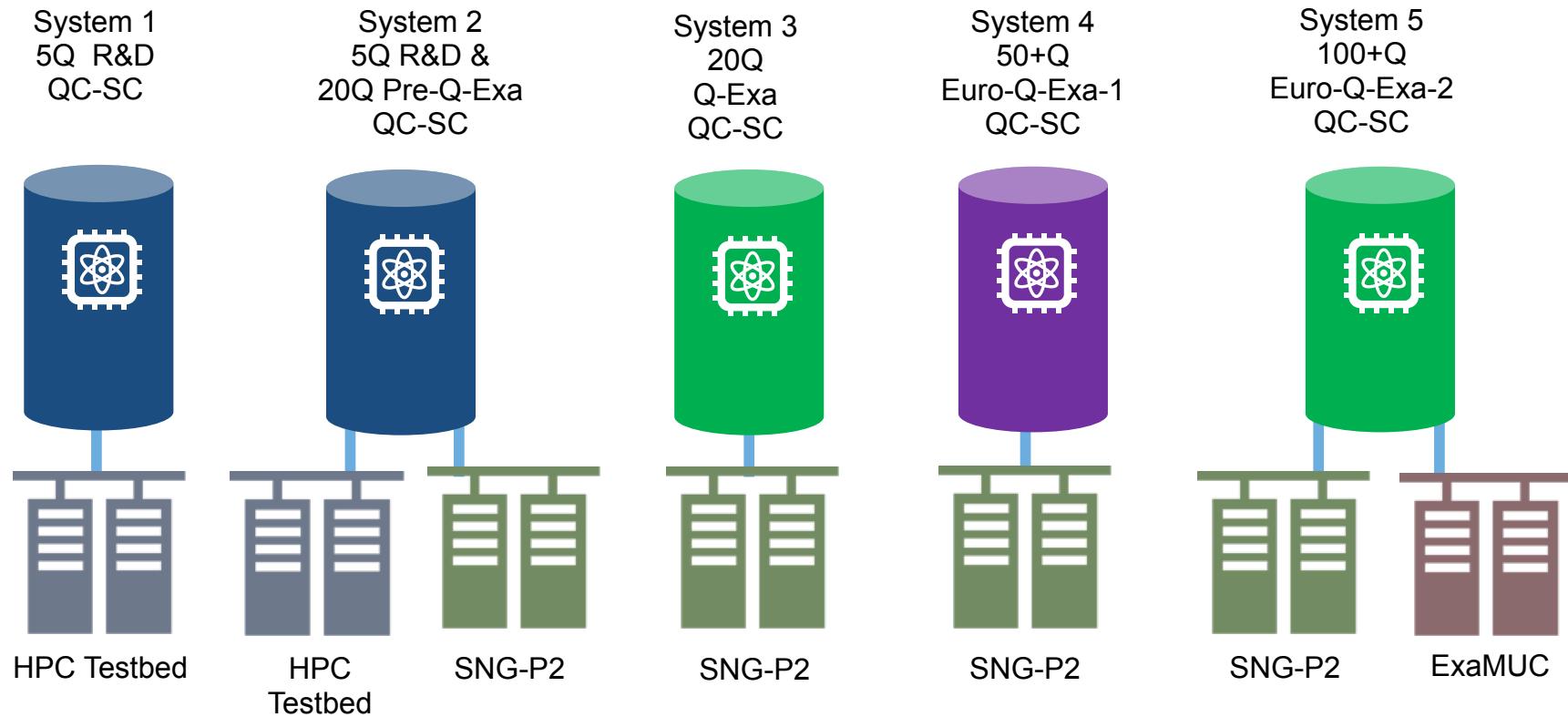
Accessible: end of H1-2025 to Bavarian,  
German and European users

System 5: 100+ qubits

Accessible: end of H2-2026 to Bavarian,  
German and European users



## Superconducting Systems Roadmap



## Quantum Systems at LRZ

### Ion-Trap:

System 1: 20+ qubits  
target system for software development initially with users from Munich Quantum Valley users next, later opened to Bavarian users and select research partners.

### Neutral Atom:

System 1:  
Base prototype arriving in 2024 for environmental telemetry study only. System for target software development and then general use coming 2026 onwards.



Announcement of award to AQT  
embargoed until December 5, 2023, 10am

## Erster hybrider Quantencomputer am LRZ

- Kopplung des QC mit dem Supercomputer
- Erstmals gezeigt im Juni 2024
- <https://www.quantum.lrz.de/de/bits-von-qubits/detail/deutschlands-erster-hybrider-quantencomputer-am-leibniz-rechenzentrum>



# Mosca's inequality

- Michele Mosca, Professor at University of Waterloo (CA)



Z = Time until QCs break crypto

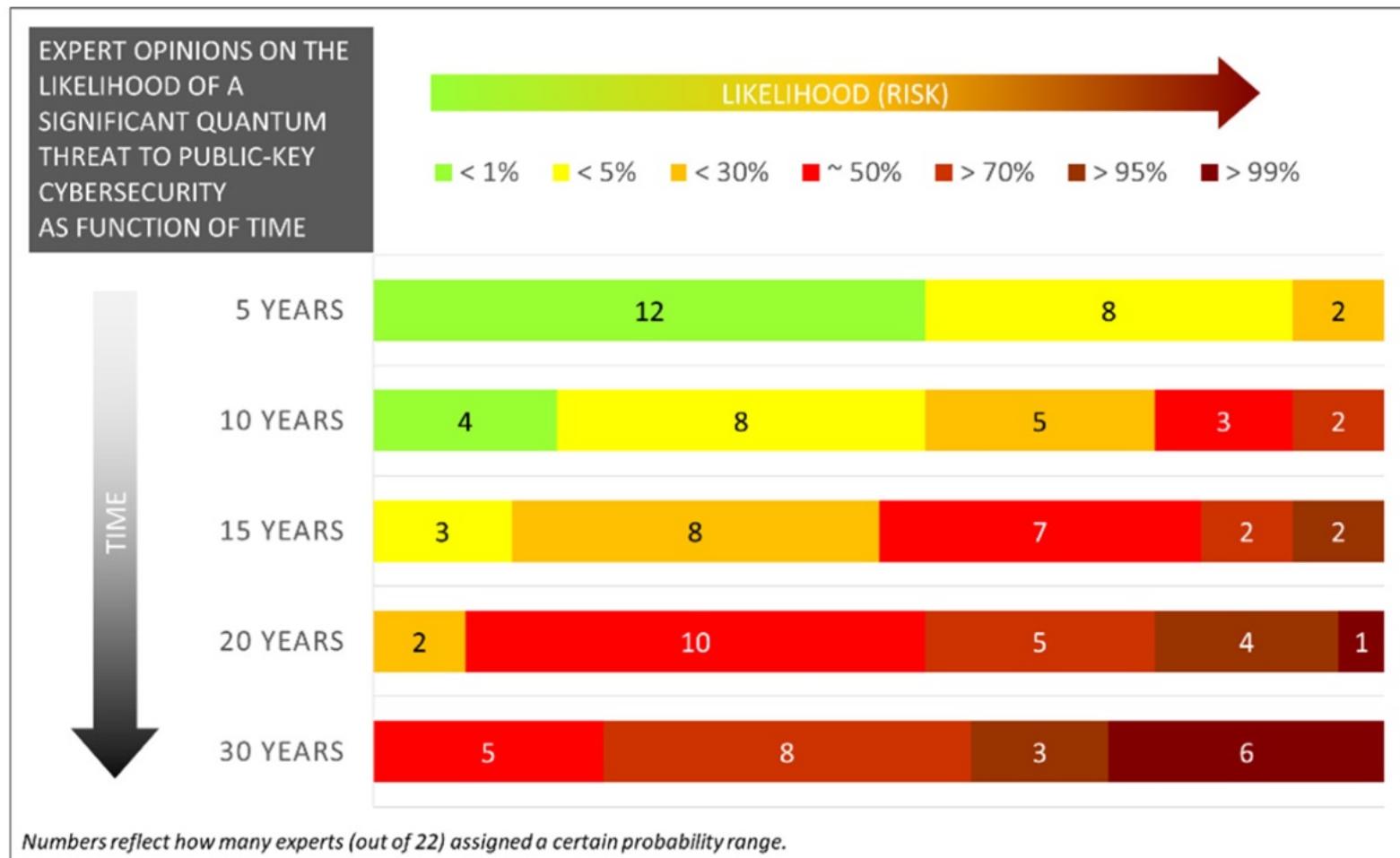
X = Time for migration to new crypto    Y = Time that data should be secret

Z = Time until QCs break crypto

If  $X + Y \leq Z$     ✓ We're safe.

If  $X + Y \geq Z$     ✗ We're in trouble.

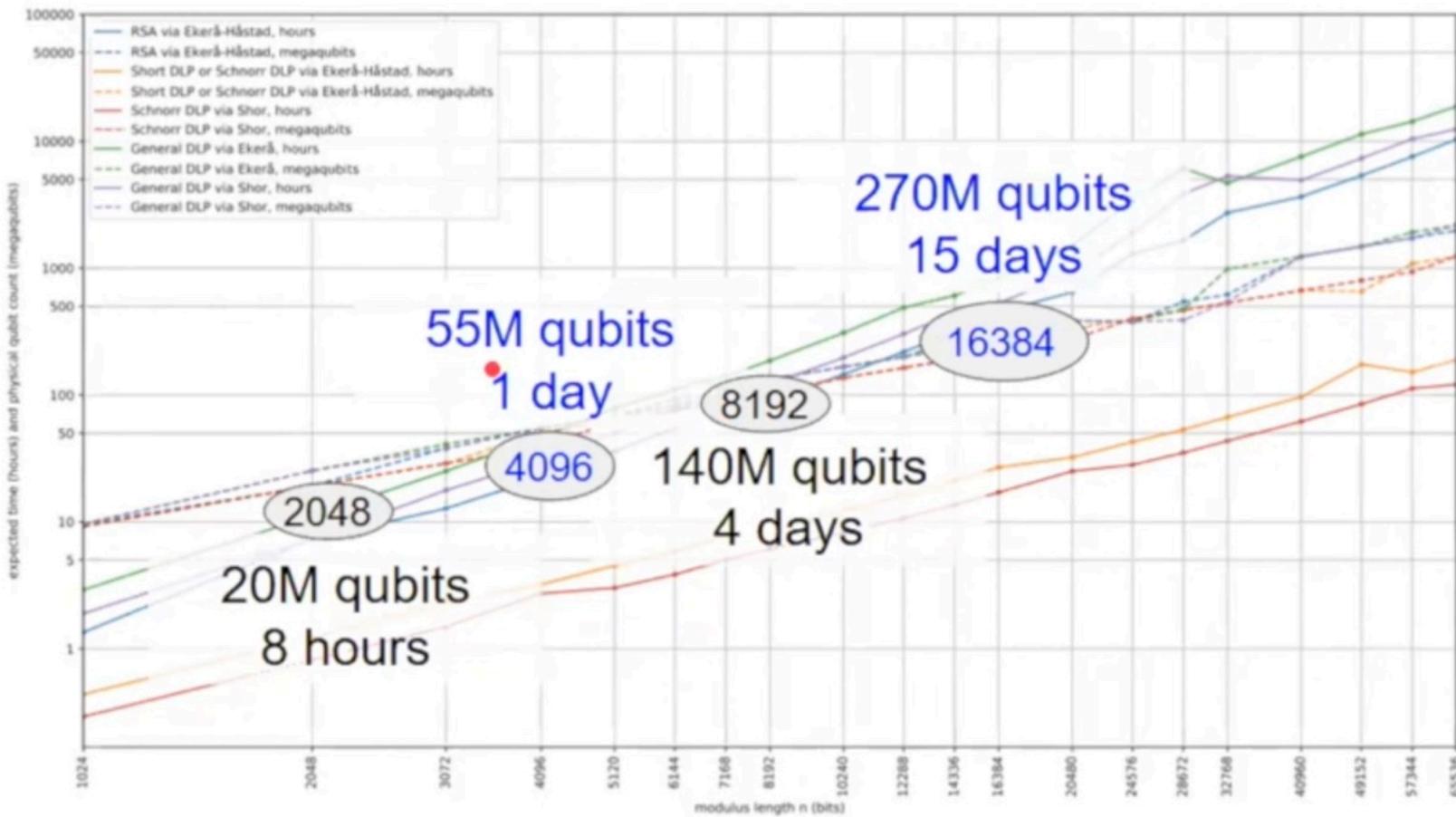
# Expert opinions



## Shor Algorithmus zur Faktorisierung (1997)

- probabilistischer Algorithmus, Eingabe:  $n$ , Ausgabe: ein Teiler von  $n$
- Laufzeit:  $O((\log n)^3)$
- Benötigte Qubits zur Faktorisierung einer Zahl  $n$  mit  $N$  Binärstellen, d.h.  $n < 2^N$ 
  - Ursprünglicher Algorithmus:  $3N$  Qubits
  - bester bekannter Algorithmus:  $2N+3$  Qubits
  - gilt nur für einen **fehlerfreien** Quantencomputer
- Auf realem QC wird ein Faktor  $M$  mehr Qubits zur Fehlerkorrektur benötigt
- Für  $N=2048$  ist die Annahme, dass  $\sim 20$  Mio. physikalische Qubits benötigt werden
- Aktuell ist Shor deshalb noch nicht anwendbar, aber
  - Fortschritte bei der Fehlerkorrektur
  - Fortschritte bei QC
- IBM hat 2001 auf einem 7 Qubit Rechner die Zahl 15 faktorisiert, d.h.  $N=4, M=3$

# Increasing RSA key sizes doesn't help that much



Craig Gidney & Martin Ekerå: How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits  
<https://quantum-journal.org/papers/q-2021-04-15-433/>

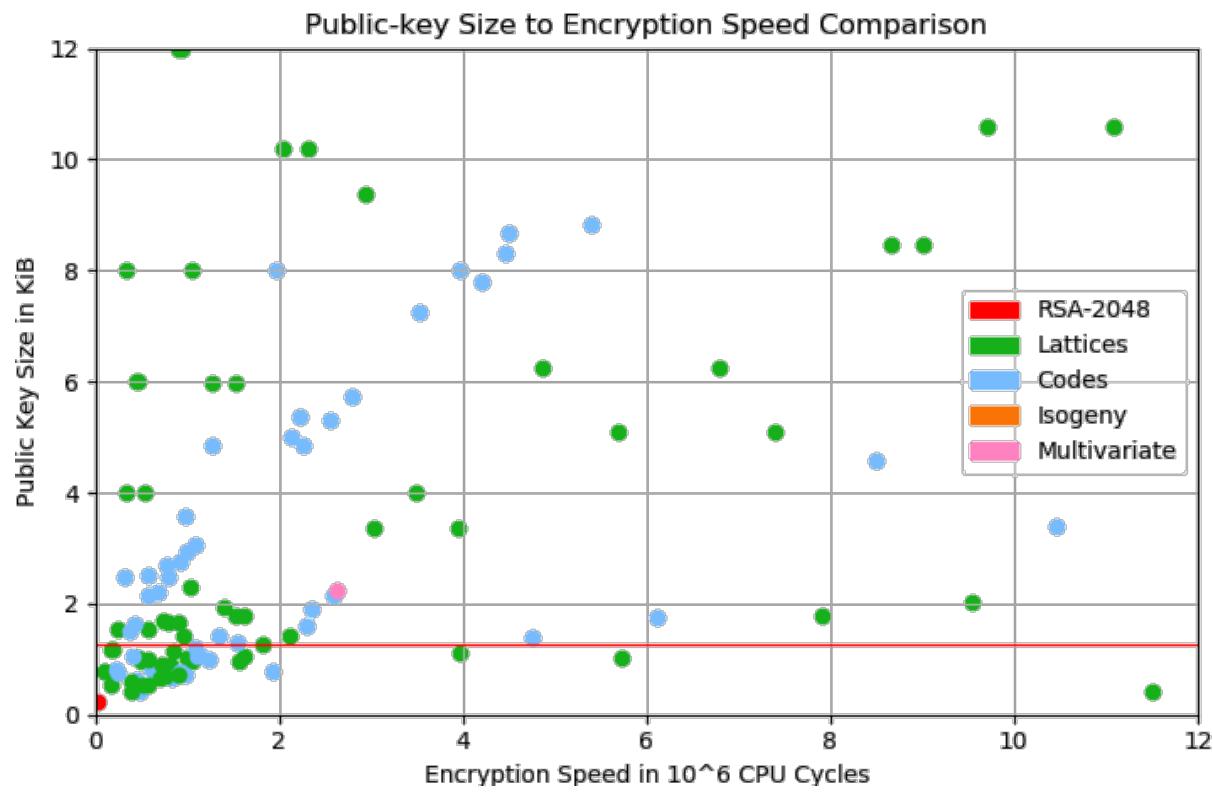
# NIST process: Post quantum cryptography (PQC)

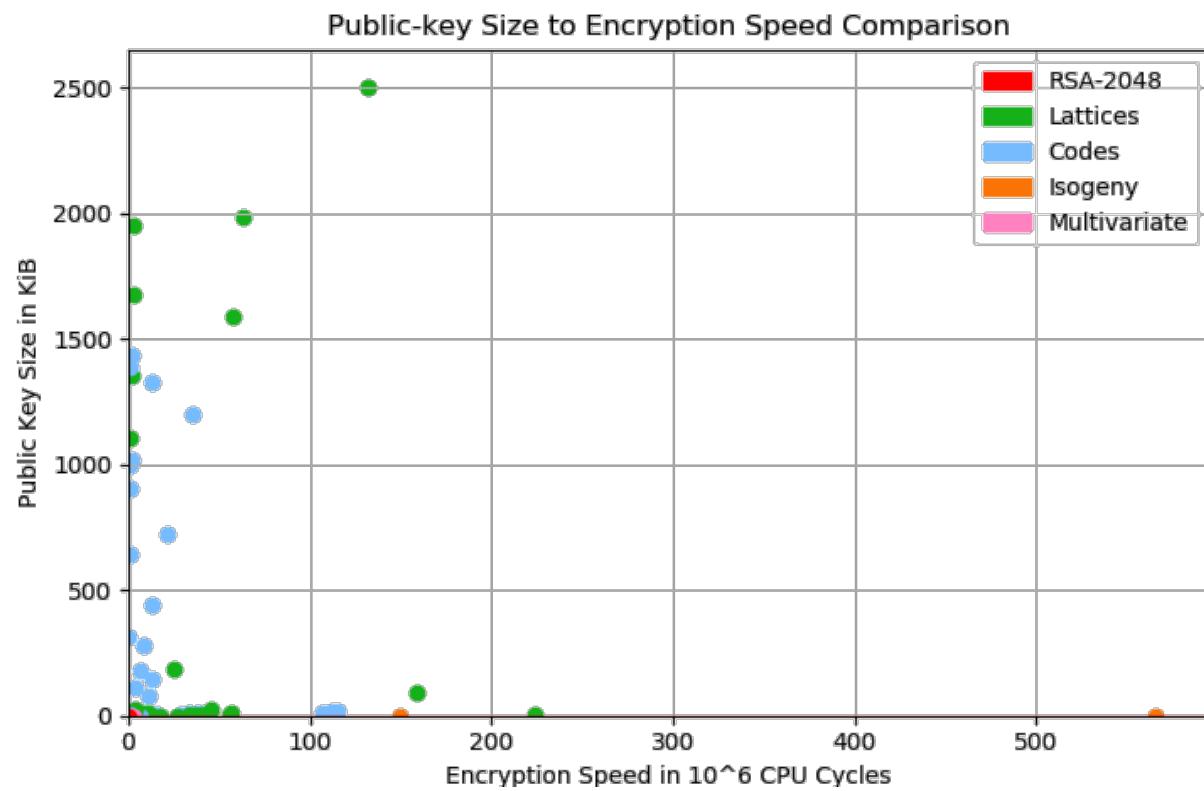
- Research on PQC started in the 2000s
- Efforts in research, application and standardization, especially:

The US-american „National Institute of Standards and Technology“ started a process to determine the best post-quantum crypto-schemes to be used in the near future.

- Why a standard? → So there is some common ground for everyone.
- Goal: Standardization by 2023
- Covers Key exchange, Encryption and Signing

The screenshot shows the NIST Computer Security Resource Center website. The header includes the NIST logo, the Information Technology Laboratory, and the Computer Security Resource Center. Below the header, there are two buttons: 'PROJECTS' and 'POST-QUANTUM CRYPTOGRAPHY'. The main content area features a section titled 'Post-Quantum Cryptography PQC' with a sub-section 'Post-Quantum Cryptography Standardization'. It mentions the announcement of Round 3 candidates on July 22, 2020, and the availability of the NISTIR 8309 Status Report on the Second Round. It also links to guidelines for submitting tweaks for Third Round Finalists and Candidates. At the bottom, there is a 'Call for Proposals Announcement' (information retained for historical purposes-call closed 11/30/2017) and a detailed description of the process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms.

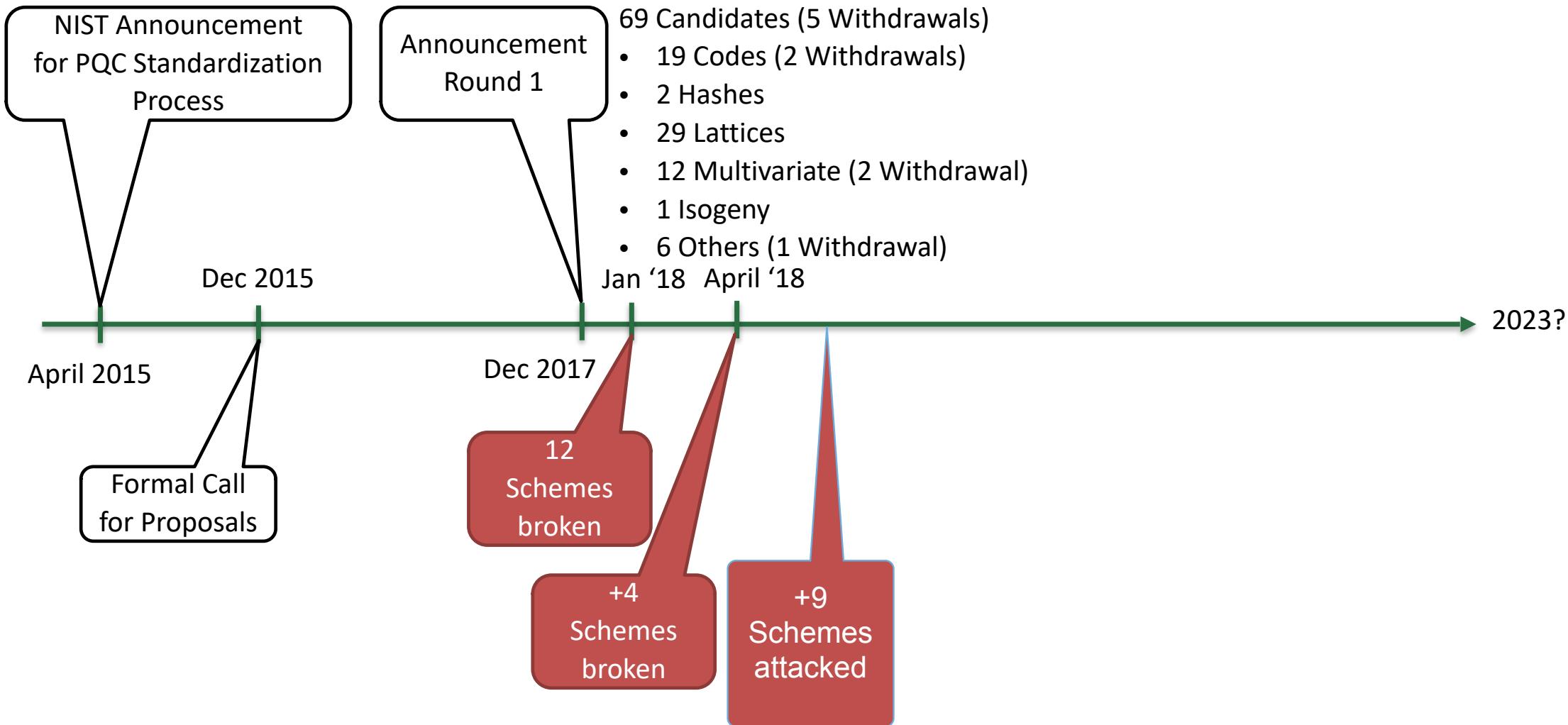




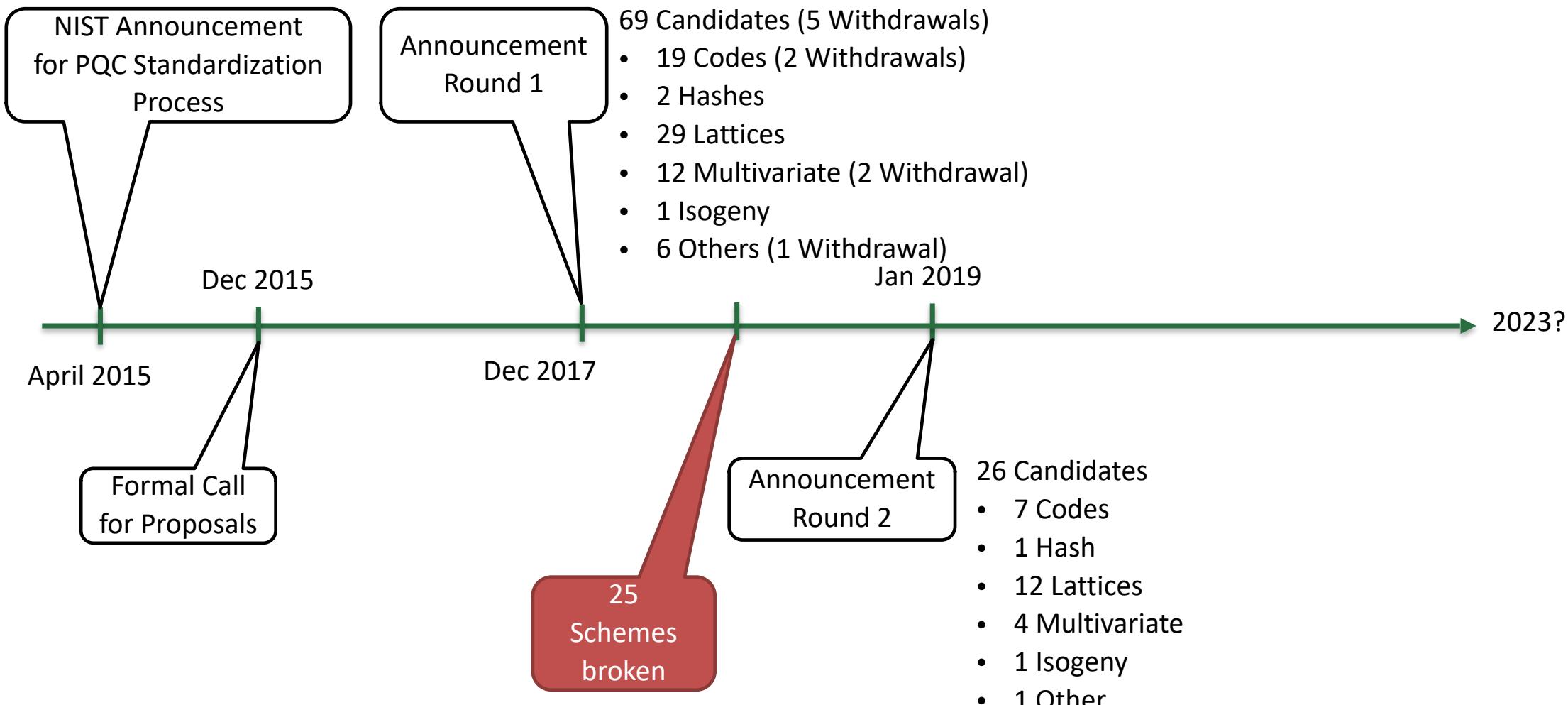
# NIST PQC Standardization process: Security

Level	Classical Security	Quantum Security	Asymmetric Security (DiscLog, RSA)	Examples
I	128 bits	64 bits	3,024 bits	At least as hard to break as AES128 (exhaustive key search)
II	128 bits	80 bits		At least as hard to break as SHA256 (collision search)
III	192 bits	96 bits	8,192 bits	At least as hard to break as AES192 (exhaustive key search)
IV	192 bits	128 bits		At least as hard to break as SHA384 (collision search)
V	256 bits	128 bits	15,336 bits	At least as hard to break as AES256 (exhaustive key search)

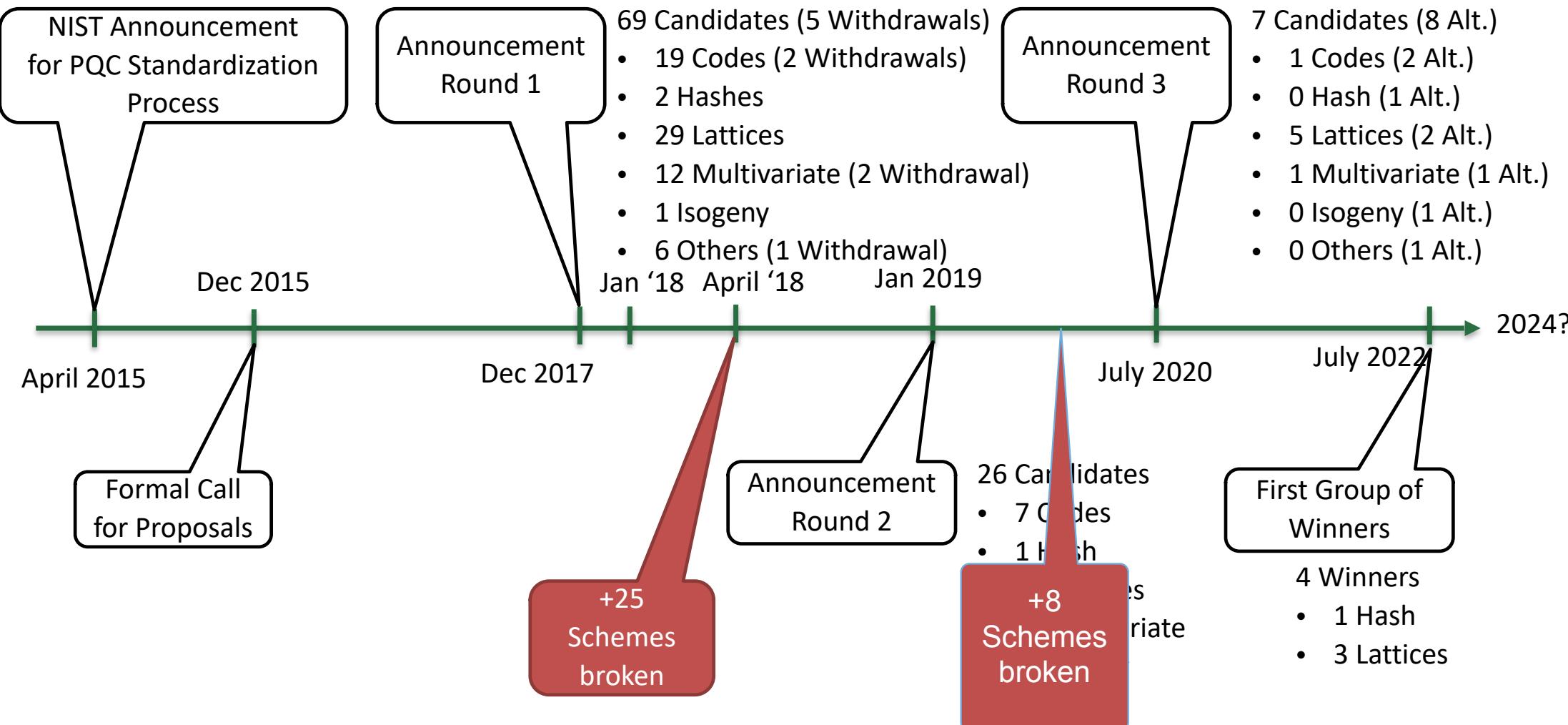
# NIST PQC Standardization process



# NIST PQC Standardization process



# NIST PQC Standardization process



- Few Weeks before NIST makes final decision
  - 4 more Candidates are attacked
  - Rainbow (most popular) completely broken (normal laptop in just 2 days)
  - 3 finalists (Cyber, Saber Dilithium) weekend with a new technology that might work against some other algorithms as well (broken by Israel Center of Encryption and Information Security)
- Selection process of NIST works properly
- July 5th 2022: NIST announced the first group of winners

Type	PKE/KEM	Signature
Lattice	Crystals-Kyber	Crystals-Dilithium Falcon
Hash based		Sphincs+

- July 5th 2022: NIST announced the first group of winners and a Round 4
- Round 4

Type	PKE/KEM
Code-based	Bike Classic McEliece HQC
Supersingular elliptic curve isogeny	SIKE

- SIKE broken August 5, 2022 on a classical computer by Wouter Castryck and Thomas Decru

## Patent und Intellectual Property Probleme

- After NIST announced the first group of winners
- Concerns about intellectual property rights, surrounding lattice-based schemes
  - Kyber
  - New-Hope
- NIST indicates there are two patent-portfolios relating to CRYSTALS-KYBER:  
<https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/selected-algos-2022/nist-pqc-license-summary-and-excerpts.pdf>
- NIST claims to take this under consideration

# NIST PQC Round 3 Comparison – Security Level 1

Method	Scheme	Secret Key Sizes	Public Key Sizes	Ciphertext/Signature Sizes (32Byte Plaintext)
Encryption / Key Encapsulation	McEliece (Code)	6 KB	260 KB	128 B
	Crystals-Kyber (Lattice)	1.6 KB	800 B	768 B
	NTRU (Lattice)	1.2 KB	1 KB	1 KB
	SABER (Lattice)	1.5 KB	672 B	736 B
Signatures	Crystals-Dilithium (Lattice)	n/A	1.4 KB	2 KB
	Falcon (Lattice)	n/A	897 B	666 B
	Rainbow (Multivariate)	100 KB	157 KB	66 B

- Additional call for digital signature proposals until June 2023
- NIST Projekt: Migration to Post-Quantum Cryptography  
<https://csrc.nist.gov/pubs/pd/2021/08/04/migration-to-postquantum-cryptography/final>

### Standardisierung in der IETF

- IETF - Internet Engineering Task Force - Standardisierung von Internet-Protokollen
  - IETF Draft - kann jeder einbringen, wird diskutiert, verbessert und dann vielleicht zu
  - RFC - Request for Comments (deFakto Standard)
  - STD - „echter“ IETF Standard
- IETF und IRTF (Internet Research Task Force) und Quanten-Krypto
  - Verschiedene Working Groups zu verschiedenen Themen, z.B. Post Quantum Cryptography (PQC) transition
  - <https://wiki.ietf.org/group/sec/PQCAgility>
- [RFC 8554](#) - Leighton-Micali Hash-Based Signatures
- [RFC 8391](#) - XMSS: eXtended Merkle Signature Scheme