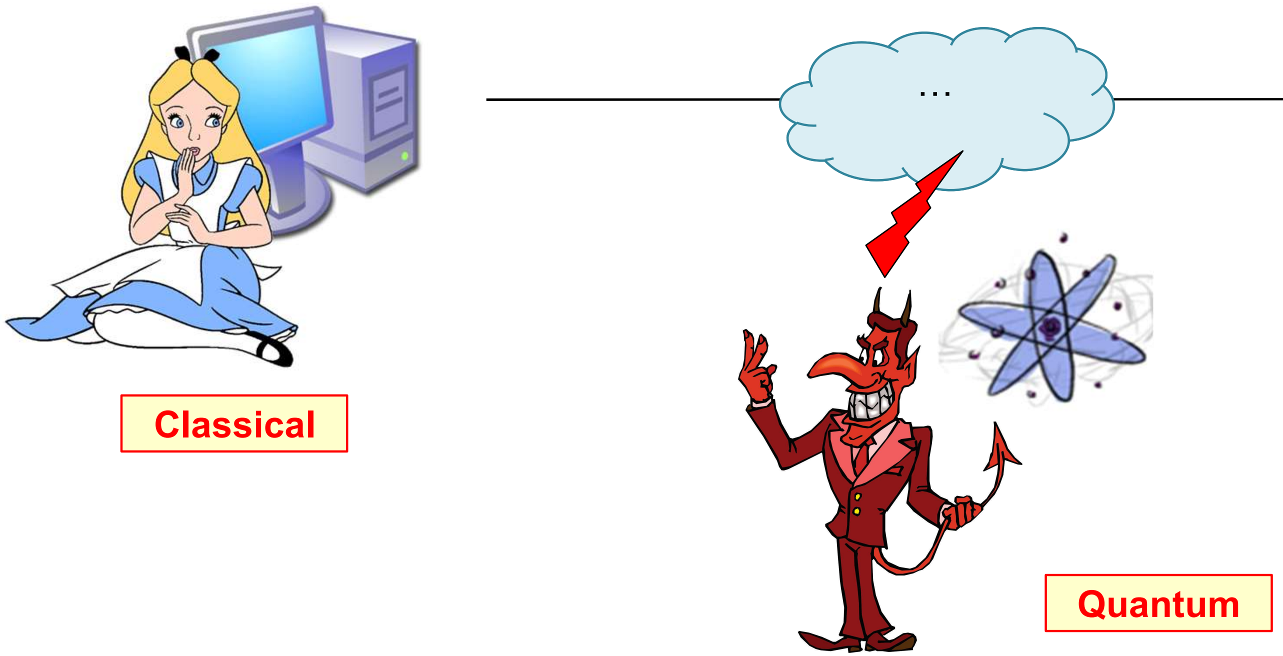
Introduction to post-quantum cryptography

Simona Samardjiska
Digital Security Group – Radboud University

Better solution: Post Quantum (PQ) Cryptography

Classical cryptosystems believed to be secure against quantum computer attacks



Post Quantum (PQ) Cryptography

Isogeny based cryptosystems – ***KEMs/NIKEs/signatures***
(Finding isogenies on supersingular elliptic curves)

Multivariate Quadratic cryptosystems – ***mainly signatures***
(Polynomial System Solving –PoSSo,
for quadratic polynomials - MQ problem)

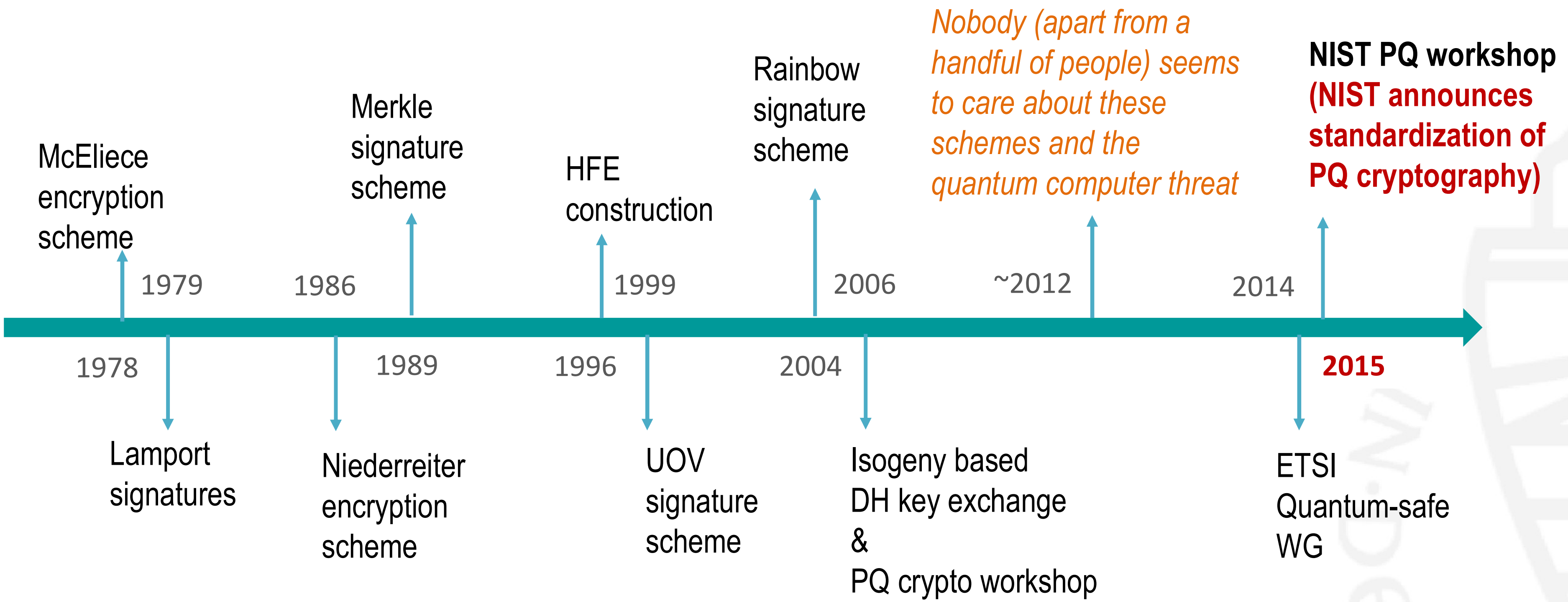
Hard Mathematical Problems

Code-based cryptosystems – ***mainly encryption/KEMs***
(decoding random linear codes)

Hash-based signatures (only)
(only secure hash function needed)

Lattice-based cryptosystems – ***signatures/encryption/KEMs***
(many different hard problems – SIS, SVP, LWE)

Some fun history facts



- *McEliece is as old as RSA!*
- *As (almost) are hash-based signatures!*

- *The term "Post-quantum cryptography" coined*
- *in 2003 by Dan Bernstein*

- **Key sizes, signature sizes and speed**
  - Huge public keys, or signatures …. Or slow
  - ex. ECC 256b key vs McElliece 500KB key

The NIST call

Initial Timeline:

- *Fall 2016 – call for proposals*
- *November 2017 – deadline for submissions*
- *January 2019 – second round candidates*
- *July 2020 – Finalists!*
- *In a few months– results*
- *2 years later – Draft standard ready*
- *Deployment ?*

- *"NOT a competition"*
- *"more complicated than AES or SHA-3"*
- *"Ideally, several algorithms will emerge as **good choices**"*

- 82 submissions, 69 "complete and proper"
- 20 signatures
- 49 Key encapsulation mechanisms

Digital Security Group – Radboud University involved in 8 candidates

KEMs

- **Classic McEliece**
  - Code-based

Lattice based

- **CRYSTALS-KYBER**
- **NTRU-HRSS-KEM**
- **New Hope**
  - Implemented and tested by Google

- **SIKE**
  - Isogeny-based

Signatures

- **CRYSTALS-DILITHIUM**
  - Lattice based

- **SPHINCS+**
  - Hash based

- **MQDSS**
  - Multivariate

Chosen standards and what next

Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER

NIST COMPUTER SECURITY RESOURCE CENTER CSRC

PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates

July 05, 2022

Algorithms to be Standardized

Public-Key Encryption/KEMs	Digital Signatures
CRYSTALS-KYBER	CRYSTALS-Dilithium FALCON SPHINCS+

PQC Fourth Round Candidate Key-Establishment Mechanisms (KEMs)

Public-Key Encryption/KEMs
BIKE Classic McEliece HQC SIKE

New Call for Proposals: Digital Signature Algorithms with Short Signatures and Fast Verification

Submissions due by June 1, 2023.

Parameter set	Public key (bytes)	Secret key (bytes)	Ciphertext (bytes)	Keygen (Kcycles)	Encaps (Kcycles)	Decaps (Kcycles)
Classic-McEliece-348864	261120	6452	128	346550.8	44.4	134.6
Kyber512	800	1632	768	37	40.3	26.5
BIKE-L1	1541	5223	1573	586.4	79	1282.8
HQC-128	2249	2289	4481	151	252.6	443.2

		Size (bytes)		Relative time	
		Public key	**Signature**	**Verification**	**Signing**
Non PQ	**NIST P-256**	64	64	1 (baseline)	1 (baseline)
	RSA-2048	256	256	0.2	25
NIST finalists	**Dilithium2**	1,320	2,420	0.3	2.5
	Falcon512	897	666	0.3	5 *
NIST alternates*	**SPHINCS$^+$-128ss har.**	32	7,856	1.7	3,000
	SPHINCS$^+$-128fs har.	32	17,088	4	200
Others	**XMSS-SHAKE_20_128 ***	32	900	2	10 *

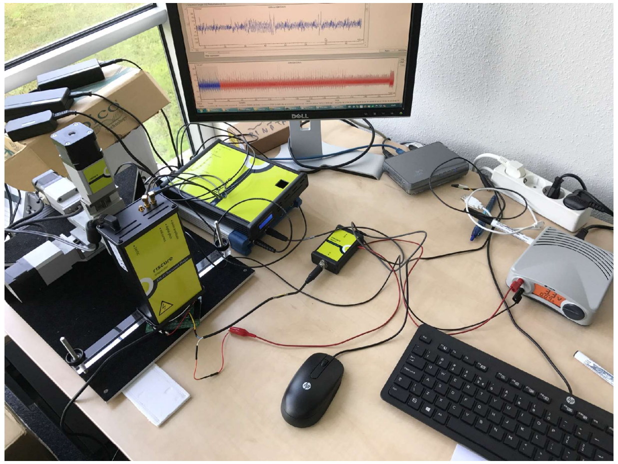
Post Quantum for embedded devices

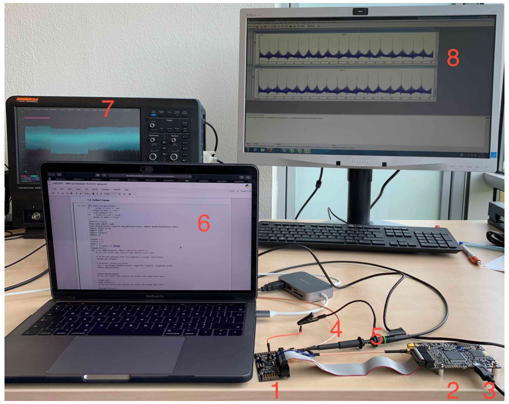
- **Arm® Cortex®-M4 recommended by NIST**,
- Other smaller embedded devices (Arm® Cortex®-M0, RISC-V)
  - Low clock speed (8-24MHz), ROM (16-32 KB) RAM (4-16KB),
  - floating point support?
  - multipliers?
- Long lived up to 30 years!
  - automotive and aviation industry
  - critical infrastructure
- ***Not addressed by the NIST process***
  - ***But critically needed now!***

- ***We organized (twice) a Lorentz workshop***
  - *Already > 6-7 highly influential publications*

- *Still a lot of work needed, but PQ crypto can be squeezed in embedded devices*

Physical security





- Devices running cryptography ***are not physically isolated***
  - Attacker may detect timing variations, power consumption, electromagnetic radiation – **Side channels**
  - **Attacker may use side channels to obtain secret data**
- **Currently one of the biggest challenges for PQ cryptography**
  - Understanding side channel attacks on PQ schemes
  - Providing cheap/reasonable **countermeasures**

  - In past 2 years – abundance of attacks on official implementations/finalists
  - ***Lattice based schemes particularly vulnerable***

Drop-in-replacement … is this realistic?

- ***Protocols have constrains***
  - For ex. Data needs to fit into packets (not exceeding maximum transmission unit (MTU))
- In the case of **DNSSEC only Falcon can fit**

Prio	Requirement	Good	Accepted Conditionally
#1	Signature Size	≤ 1,232 bytes	–
#2	Validation Speed	≥ 1,000 sig/s	–
#3	Key Size	≤ 64 kilobytes	> 64 kilobytes
#4	Signing Speed	≥ 100 sig/s	–

  - **Still:** only one key/signature can be shipped at a time

- **In TLS?**
  - TLS has a handshake part for
    - key agreement (Diffie-Hellman) and
    - Authentication (signatures)
  - **A KEM can't be used as a drop-in-replacement**
    - Protocol needs to be changed
  - **Then why not use the KEM for authentication as well!**
    - **KEMTLS** (large scale experiments ongoing in collaboration with
    - Cloudflare)

***Post-Quantum* TLS 1.3 Handshake**

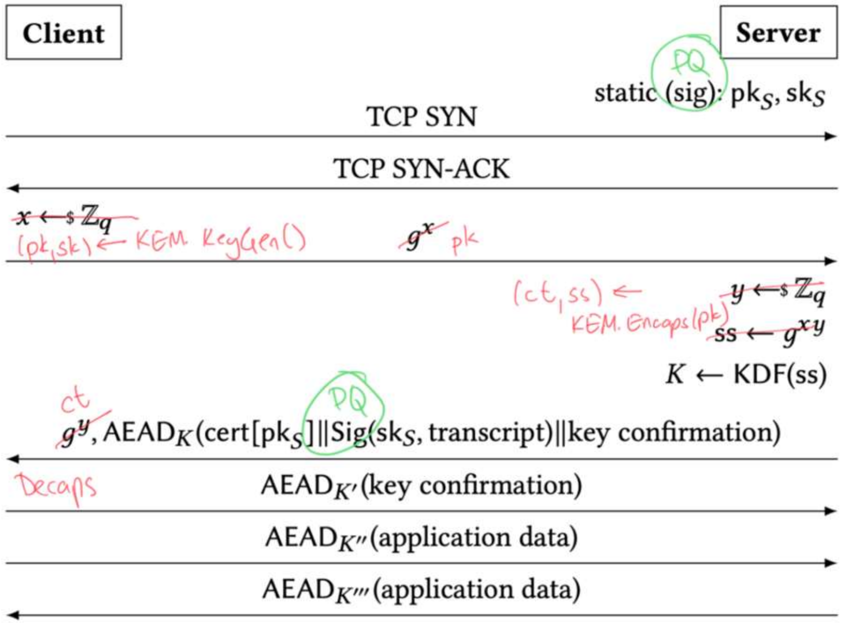
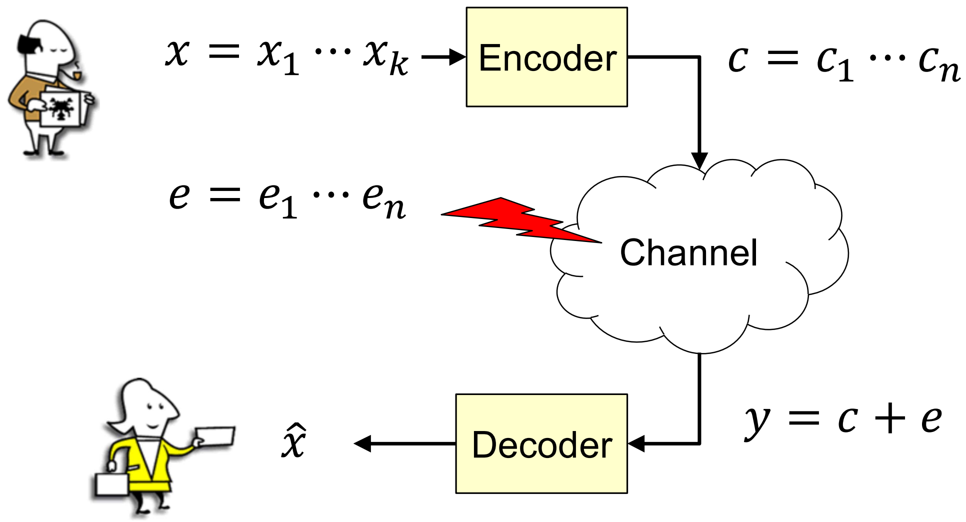


Image credit: Thom Wiggers

An overview of post-quantum families of cryptosystems

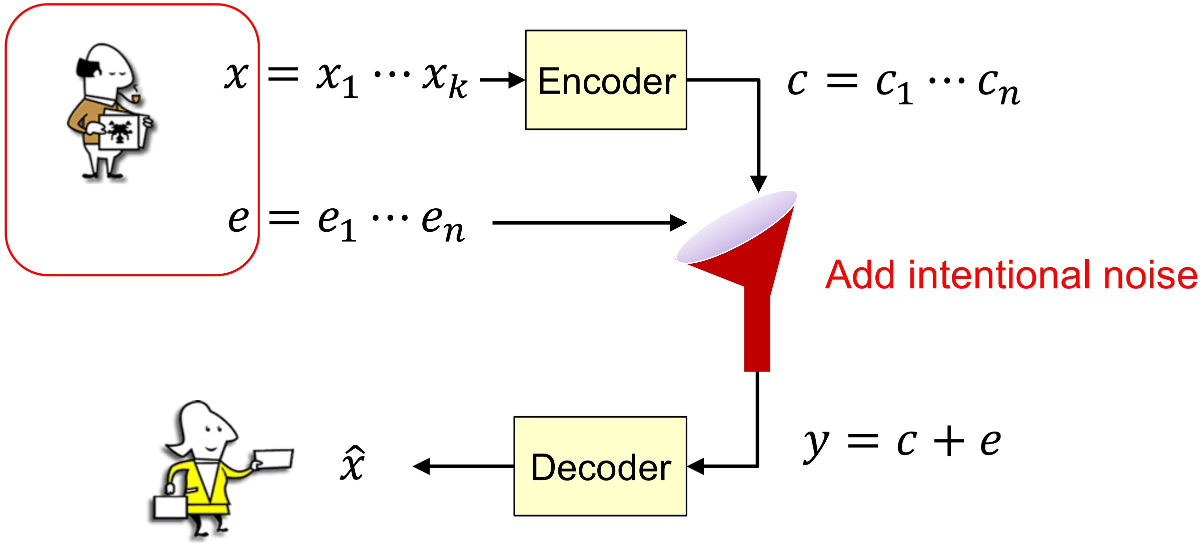
Code-based Cryptosystems

- Coding theory essentials
- Noisy channel communication:



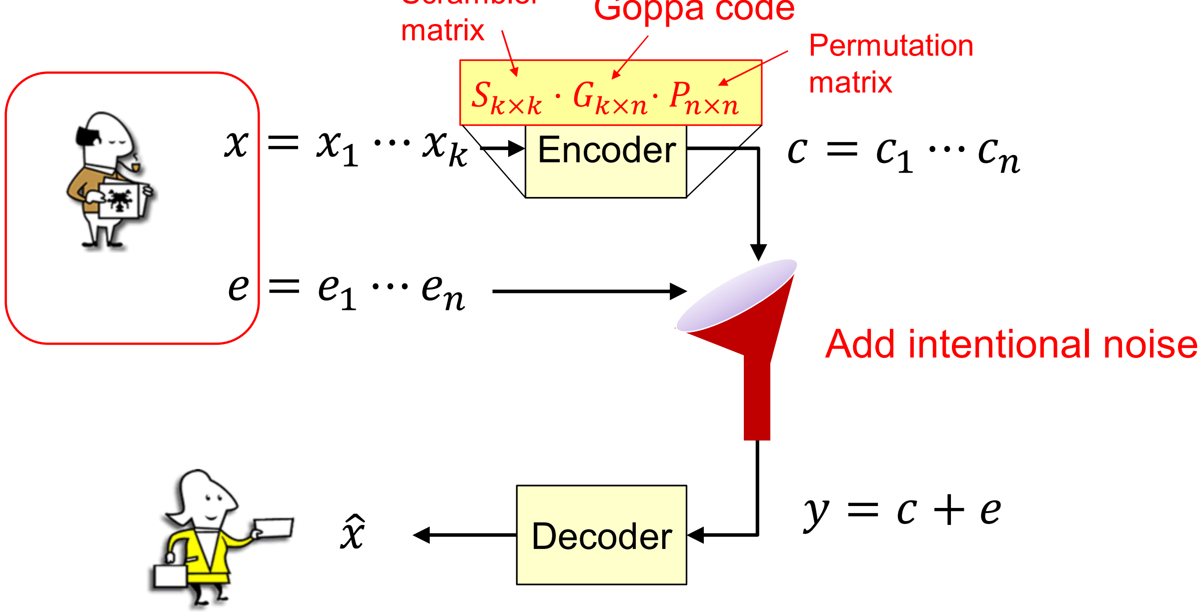
Code-based Cryptosystems

- Coding theory essentials
- In cryptography:



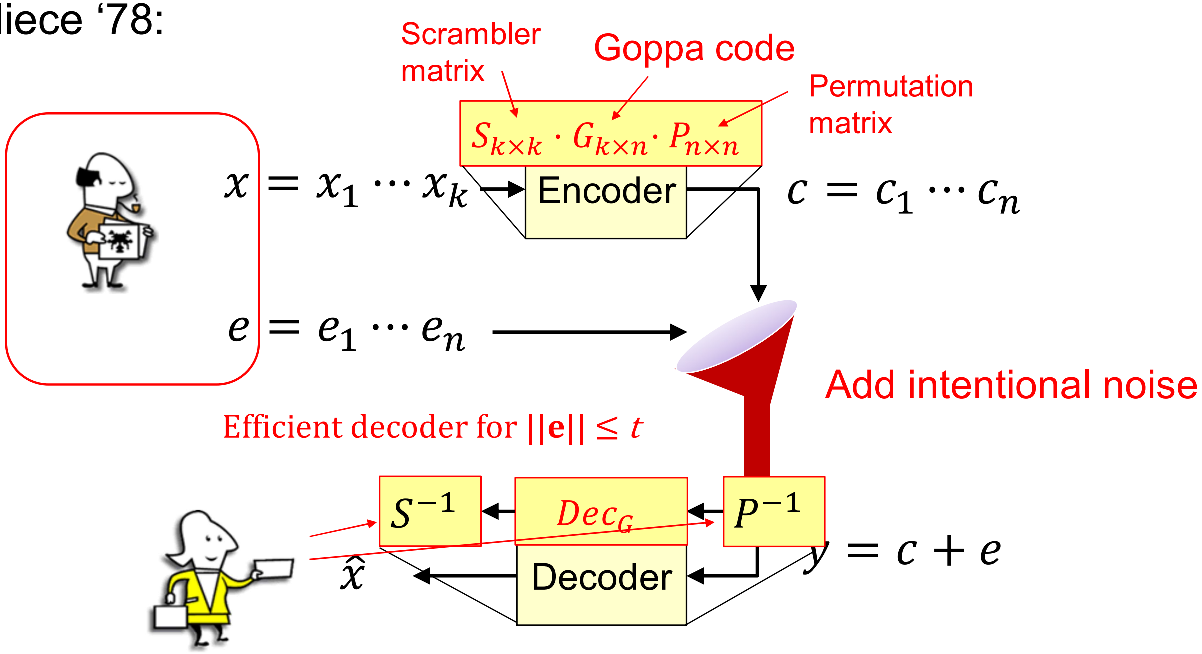
Code-based Cryptosystems

- Hard underlying problem (NP hard): Decoding random linear codes
- No reduction to the hard problem – instead, related problems believed to be hard
- Confidence in encryption schemes
- McEliece '78:



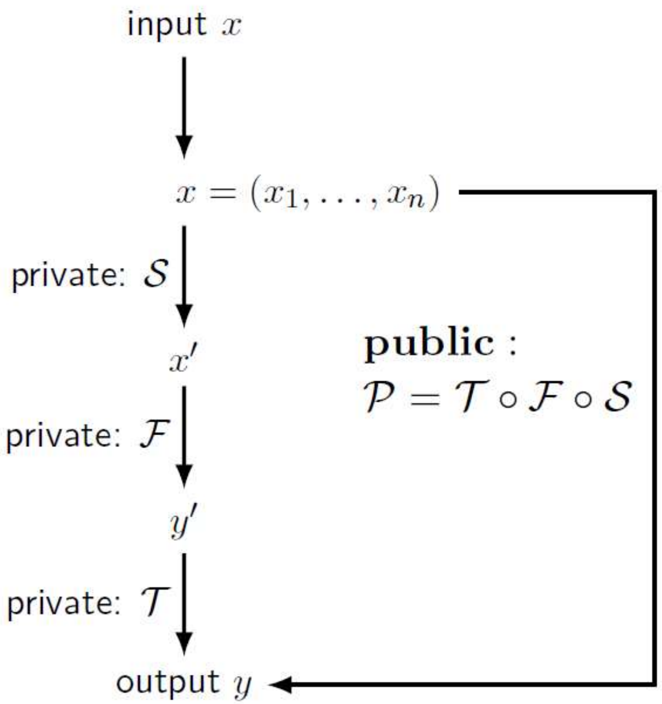
Code-based Cryptosystems

- Hard underlying problem (NP hard): Decoding random linear codes
- No reduction to the hard problem – instead, related problems believed to be hard
- Confidence in encryption schemes
- McEliece '78:



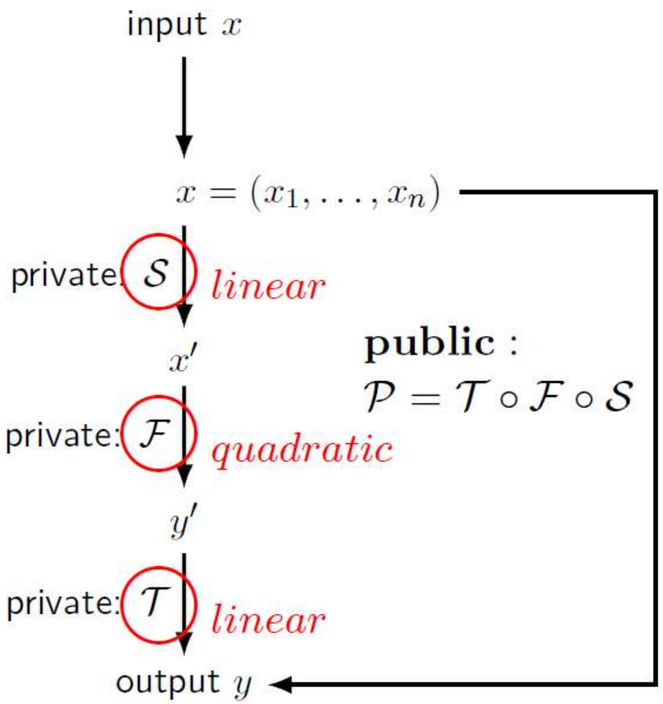
MQ (multivariate quadratic) Cryptosystems

- Hard underlying problem (NP hard): Polynomial system solving (PoSSo)
- (**Mainstream**) No reduction to the hard problem – related problems believed to be hard
- Confidence in signatures



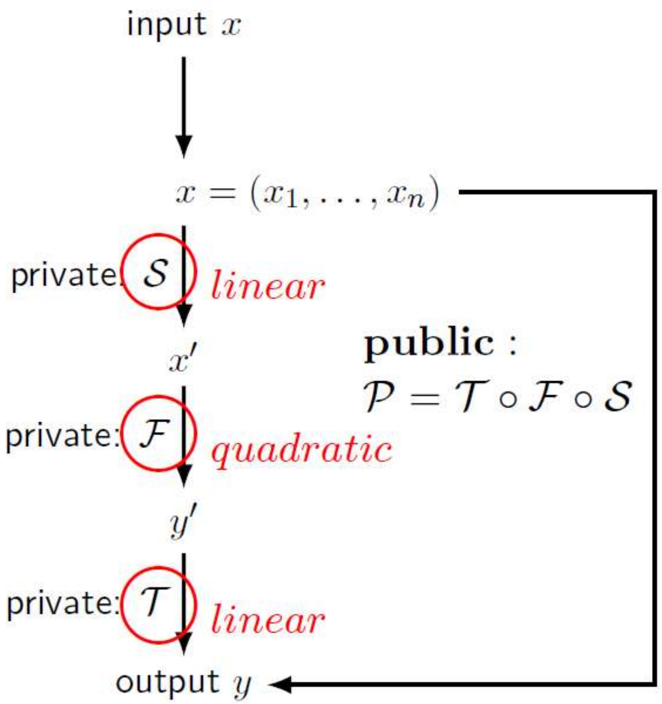
MQ (multivariate quadratic) Cryptosystems

- Hard underlying problem (NP hard): Polynomial system solving (PoSSo)
- (**Mainstream**) No reduction to the hard problem – related problems believed to be hard
- Confidence in signatures



MQ (multivariate quadratic) Cryptosystems

- Hard underlying problem (NP hard): Polynomial system solving (PoSSo)
- (**Mainstream**) No reduction to the hard problem – related problems believed to be hard
- Confidence in signatures

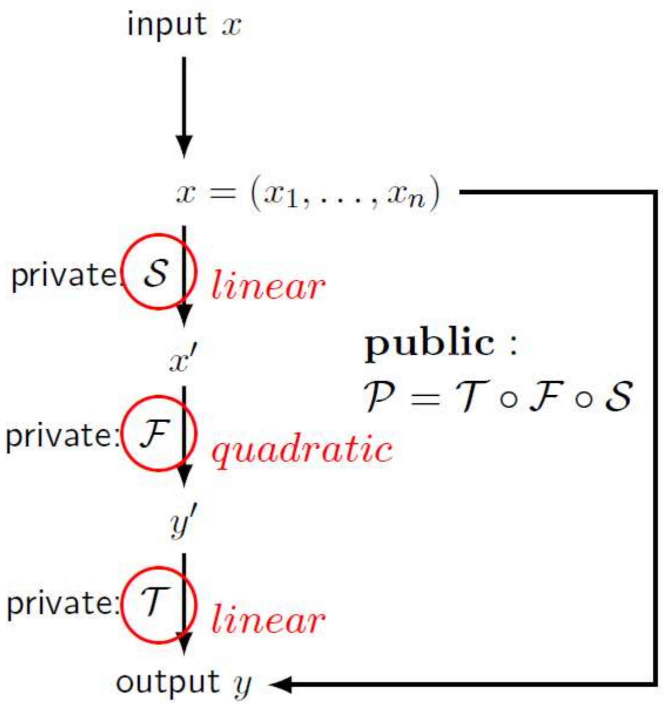


Public $\mathcal{P}$

$$p_1(x_1, \ldots, x_n)$$
$$p_2(x_1, \ldots, x_n)$$
$$\cdots$$
$$p_m(x_1, \ldots, x_n)$$

MQ (multivariate quadratic) Cryptosystems

- Hard underlying problem (NP hard): Polynomial system solving (PoSSo)
- (**Mainstream**) No reduction to the hard problem – related problems believed to be hard
- Confidence in signatures



PoSSo:

Input:

$p_1, p_2, \ldots, p_m \in \mathbb{F}_q[x_1, \ldots, x_n]$

Question:

Find - if any - $(u_1, \ldots, u_n) \in \mathbb{F}_q^n$ st.

$$\begin{cases} p_1(u_1, \ldots, u_n) = & 0 \\ p_2(u_1, \ldots, u_n) = & 0 \\ \quad \ldots & \\ p_m(u_1, \ldots, u_n) = & 0 \end{cases}$$

MQDSS

IDS

$\mathcal{P}$		$\mathcal{V}$
$\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk})$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\mathsf{ch}_1}$	$\mathsf{ch}_1 \leftarrow_R \mathsf{ChS}_1(1^k)$
$\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1)$	$\xrightarrow{\mathsf{resp}_1}$	
	$\xleftarrow{\mathsf{ch}_2}$	$\mathsf{ch}_2 \leftarrow_R \mathsf{ChS}_2(1^k)$
$\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2)$	$\xrightarrow{\mathsf{resp}_2}$	
		$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$

↓ ↓ ↓

FS signature

Signer

$\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk})$

$\mathsf{ch}_1 \leftarrow H_1(m, \mathsf{com})$

$\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1)$

$\mathsf{ch}_2 \leftarrow H_2(m, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1)$

$\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2)$

output : $\sigma = (\mathsf{com}, \mathsf{resp}_1, \mathsf{resp}_2)$

Verifier

$\mathsf{ch}_1 \leftarrow H_1(m, \mathsf{com})$

$\mathsf{ch}_2 \leftarrow H_2(m, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1)$

$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$

output : b

Lattice-based Cryptosystems

- Encryption, signatures, key exchange
- Many different hard problems

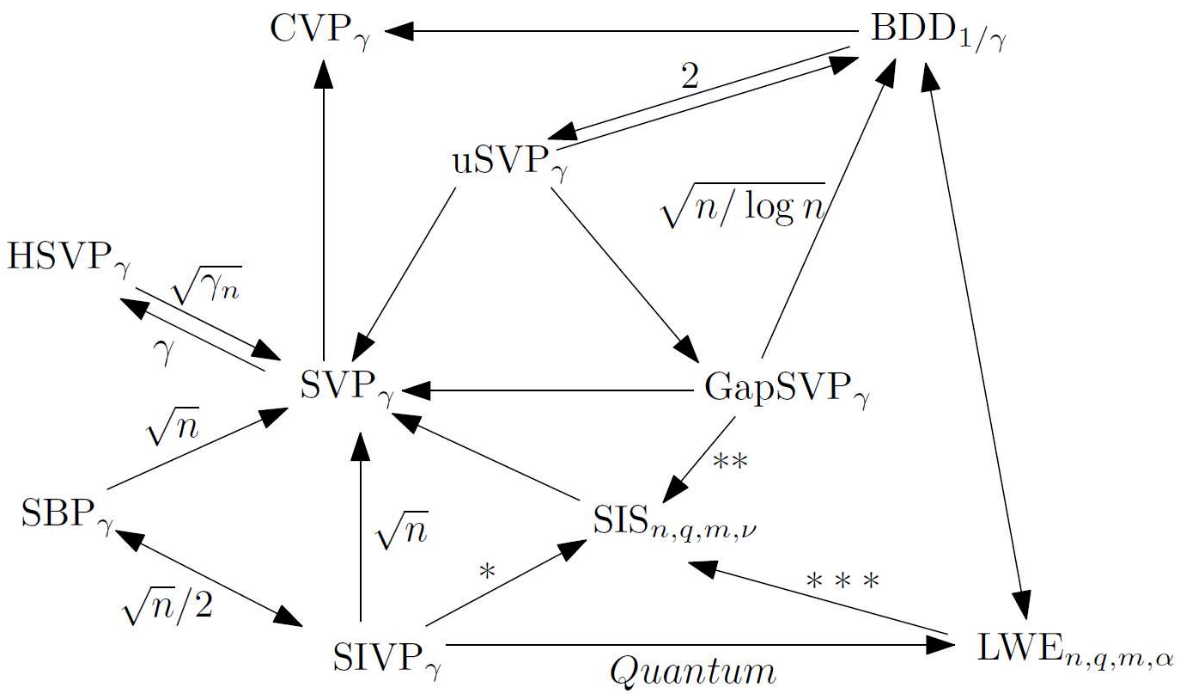


Fig. from Joop van de Pol's MSc-thesis

Lattice-based Cryptosystems

- Learning with errors (LWE)
- Variants R-LWE, Module-LWE, LPN, …
  - Additional structure undermines security claims

- Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$
- Let χ be an *error distribution* on $\mathcal{R}_q$
- Let $\mathbf{s} \in \mathcal{R}_q$ be secret
- Attacker is given pairs $(\mathbf{a}, \mathbf{as} + \mathbf{e})$ with
  - $\mathbf{a}$ uniformly random from $\mathcal{R}_q$
  - $\mathbf{e}$ sampled from χ
- Task for the attacker: find $\mathbf{s}$
- Common choice for χ: discrete Gaussian

Lattice-based Cryptosystems

- Learning with errors (LWE)
- Variants R-LWE, Module-LWE, LPN, …
  - Additional structure undermines security claims

- Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$
- Let χ be an *error distribution* on $\mathcal{R}_q$
- Let $\mathbf{s} \in \mathcal{R}_q$ be secret
- Attacker is given pairs $(\mathbf{a}, \mathbf{as} + \mathbf{e})$ with
  - $\mathbf{a}$ uniformly random from $\mathcal{R}_q$
  - $\mathbf{e}$ sampled from χ
- Task for the attacker: find $\mathbf{s}$
- Common choice for χ: discrete Gaussian

Alice (server)		Bob (client)
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$		$\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{as} + \mathbf{e}$	$\xrightarrow{\mathbf{b}}$	$\mathbf{u} \leftarrow \mathbf{as}' + \mathbf{e}'$
	$\xleftarrow{\mathbf{u}}$	

Alice has	$\mathbf{v}$	$= \mathbf{us}$	$= \mathbf{ass}' + \mathbf{e}'\mathbf{s}$
Bob has	$\mathbf{v}'$	$= \mathbf{bs}'$	$= \mathbf{ass}' + \mathbf{es}'$

Lattice-based Cryptosystems

- Learning with errors (LWE)
- Variants R-LWE, Module-LWE, LPN, …
  - Additional structure undermines security claims

- Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$
- Let χ be an *error distribution* on $\mathcal{R}_q$
- Let $\mathbf{s} \in \mathcal{R}_q$ be secret
- Attacker is given pairs $(\mathbf{a}, \mathbf{as} + \mathbf{e})$ with
  - $\mathbf{a}$ uniformly random from $\mathcal{R}_q$
  - $\mathbf{e}$ sampled from χ
- Task for the attacker: find $\mathbf{s}$
- Common choice for χ: discrete Gaussian

Alice (server)		Bob (client)
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$		$\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{as} + \mathbf{e}$	$\xrightarrow{\mathbf{b}}$	$\mathbf{u} \leftarrow \mathbf{as}' + \mathbf{e}'$
	$\xleftarrow{\mathbf{u}}$	

$$
\begin{array}{llll}
\text{Alice has} & \mathbf{v} & = \mathbf{us} & = \mathbf{ass}' + \mathbf{e}'\mathbf{s} \\
\text{Bob has} & \mathbf{v}' & = \mathbf{bs}' & = \mathbf{ass}' + \mathbf{es}'
\end{array}
$$

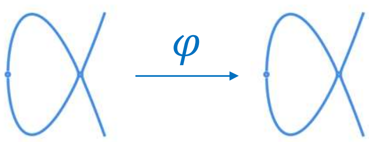
approximately same

small

Isogeny-based cryptography (slides credit Krijn Reijnders)

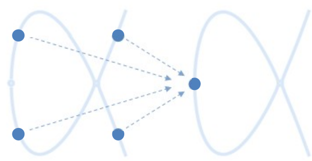
Isogeny

‘Nice’ map between two elliptic curves



Degree

Isogeny φ of degree N maps N points to 1



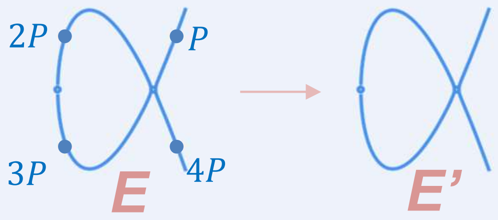
Hardness

Given two curves, find φ of degree N

How to compute an isogeny?

A small isogeny

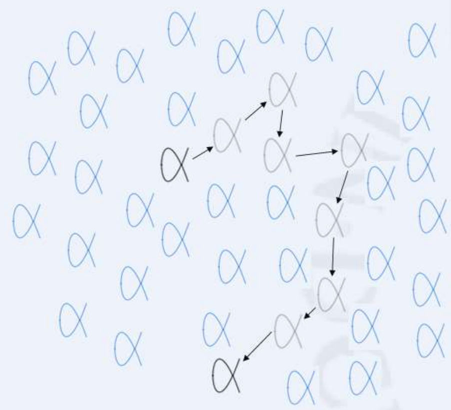
Use Vélu's formulas (degree N)



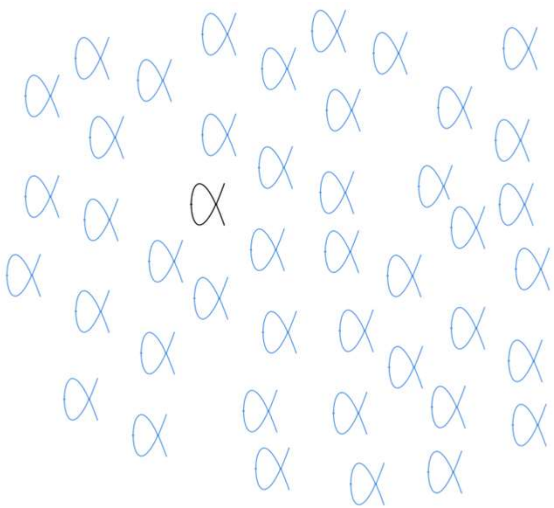
1. Find torsion point P of order N on E
2. Calculate $P, 2P, 3P, \ldots$
3. Use these points to compute E'

A large isogeny

Chain together small isogenies



DH-style post quantum key exchange



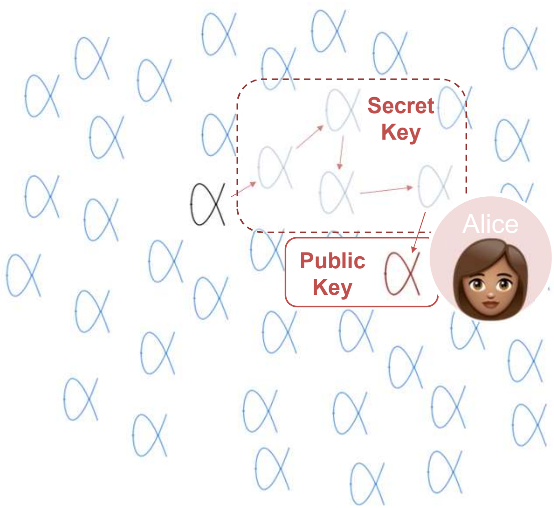
CSIDH
Post quantum key exchange

Alice and Bob perform **long** walks in isogeny graphs

A long walk is composed of **a lot** of small isogenies (≈400)

Problem with CSIDH: **slow** compared to other post quantum key exchanges

DH-style post quantum key exchange



CSIDH

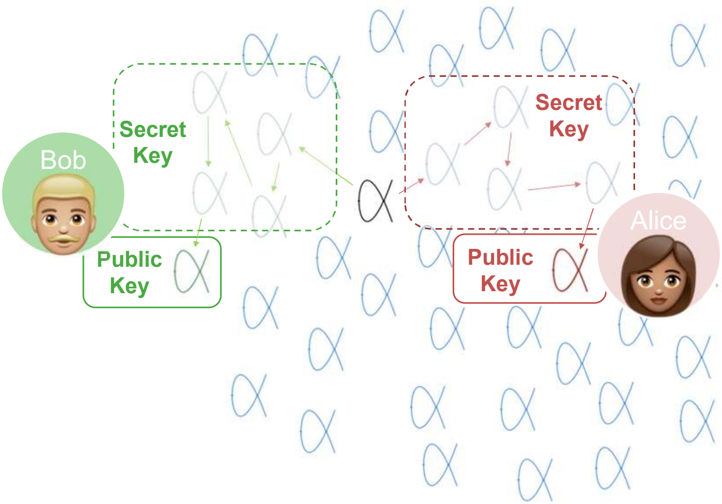
Post quantum key exchange

Alice and Bob perform **long** walks in isogeny graphs

A long walk is composed of **a lot** of small isogenies (≈400)

Problem with CSIDH: **slow** compared to other post quantum key exchanges

DH-style post quantum key exchange



CSIDH

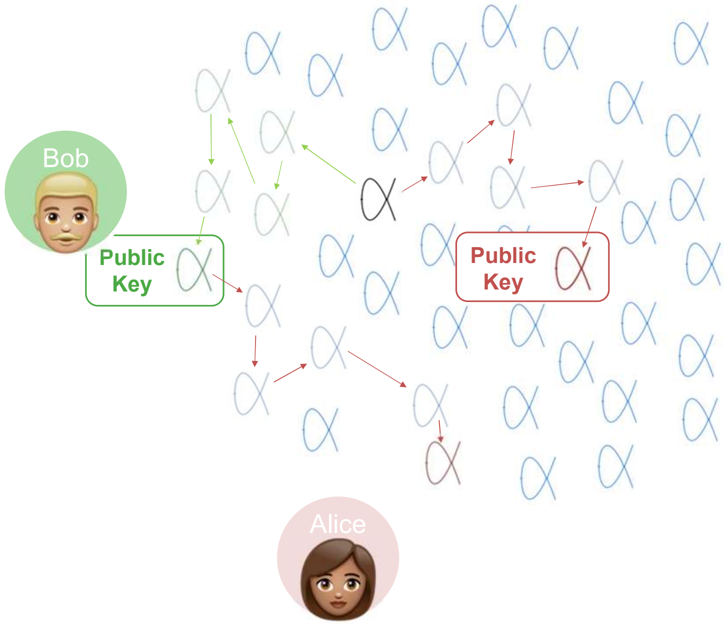
Post quantum key exchange

Alice and Bob perform **long** walks in isogeny graphs

A long walk is composed of **a lot** of small isogenies (≈400)

Problem with CSIDH: **slow** compared to other post quantum key exchanges

DH-style post quantum key exchange



CSIDH

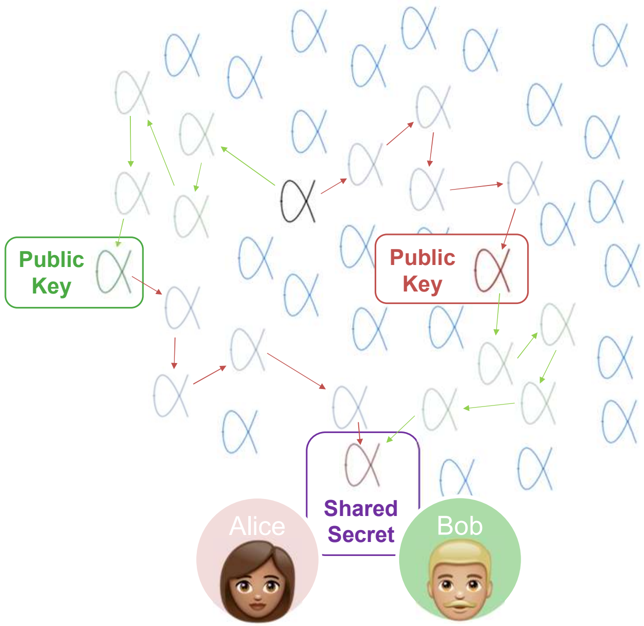
Post quantum key exchange

Alice and Bob perform **long** walks in isogeny graphs

A long walk is composed of **a lot** of small isogenies (≈400)

Problem with CSIDH: **slow** compared to other post quantum key exchanges

DH-style post quantum key exchange



CSIDH

Post quantum key exchange

Alice and Bob perform **long** walks in isogeny graphs

A long walk is composed of **a lot** of small isogenies (≈400)

Problem with CSIDH: **slow** compared to other post quantum key exchanges

Thank you for listening!

?