

# Post-Quantum Cryptography for Engineers

[draft-ar-pquip-pqc-engineers-02 - Post-Quantum Cryptography for Engineers \(ietf.org\)](https://datatracker.ietf.org/draft-ietf/pquip-pqc-engineers-02)

PQUIP IETF 117

25<sup>th</sup> July 2023

Authors: Aritra Banerjee (Nokia), K Tirumaleswar Reddy (Nokia), Dimitrios Schoinianakis (Nokia), Tim Hollebeek (DigiCert)

# Why the draft is relevant to PQUIP?

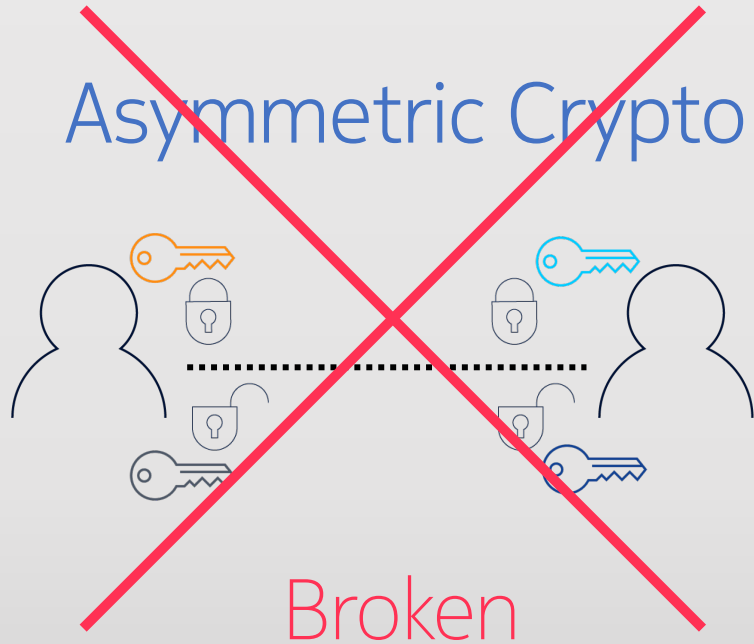
- The draft explains why engineers need to be aware of and understand post-quantum cryptography.
- It emphasizes the potential impact of Cryptographically Relevant Quantum Computers (CRQCs) on current cryptographic systems and the need to transition to post-quantum algorithms to ensure long-term security.
- Not much cryptographic math is discussed in this draft but rather an overview of post quantum use in protocols.

# Contents of the draft

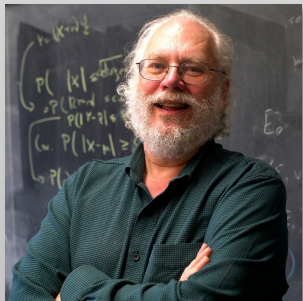
- Traditional Cryptographic Primitives that Could Be Replaced by PQC
- NIST PQC Algorithms
- Threat of CRQCs on Cryptography
- Timeline for transition (Mosca's model)
- Post-quantum cryptography categories
- KEMs, Signatures
- Recommendations for Security / Performance Trade-offs (PQC vs Classical)
- Post-Quantum and Traditional Hybrid Schemes
- Security Considerations

# Impact of Quantum Computers in Cryptography

## Asymmetric Crypto

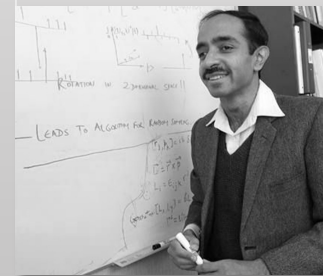
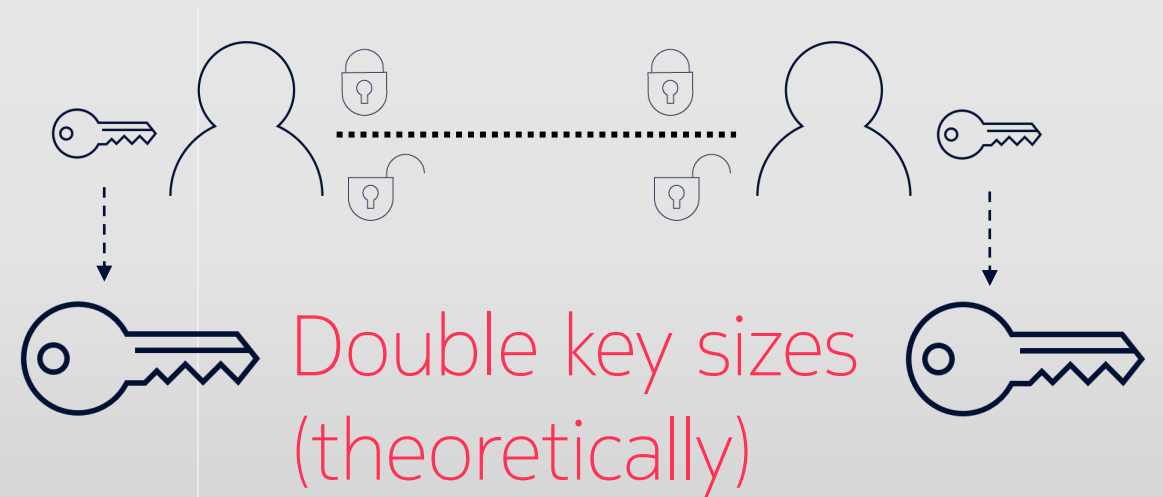


Broken



Peter Shor  
Algorithm for finding  
order of an element of a  
periodic group

## Symmetric Crypto



Lov Kumar Grover  
shows how to search in  $\sqrt{N}$

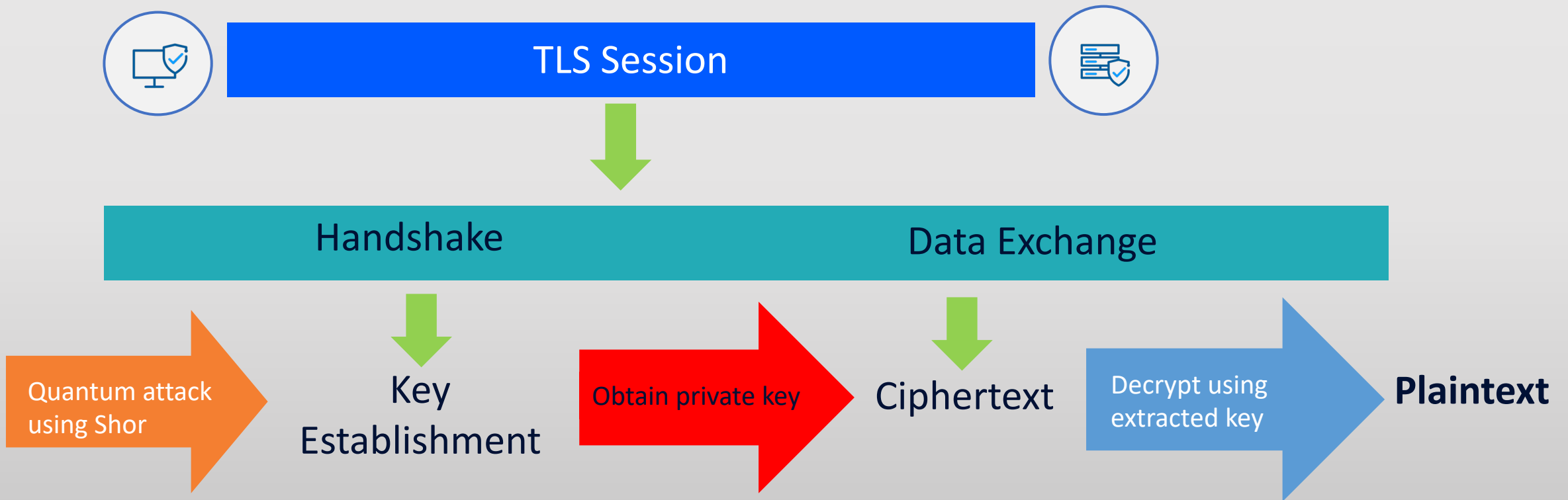
# Asymmetric Cryptography (Shor's Algorithm)

- Shor's algorithm helps find the order of an element of a periodic group solving the order-finding problem
- It solves an instance of the hidden subgroup problem (HSP) for finite abelian groups (multiplicative or additive)
- This has potential applications in breaking algorithms based on the hardness of prime factoring and the discrete log problem (finite field or elliptic curves).
- Affected algorithms include RSA, ECC, ECDH, etc.

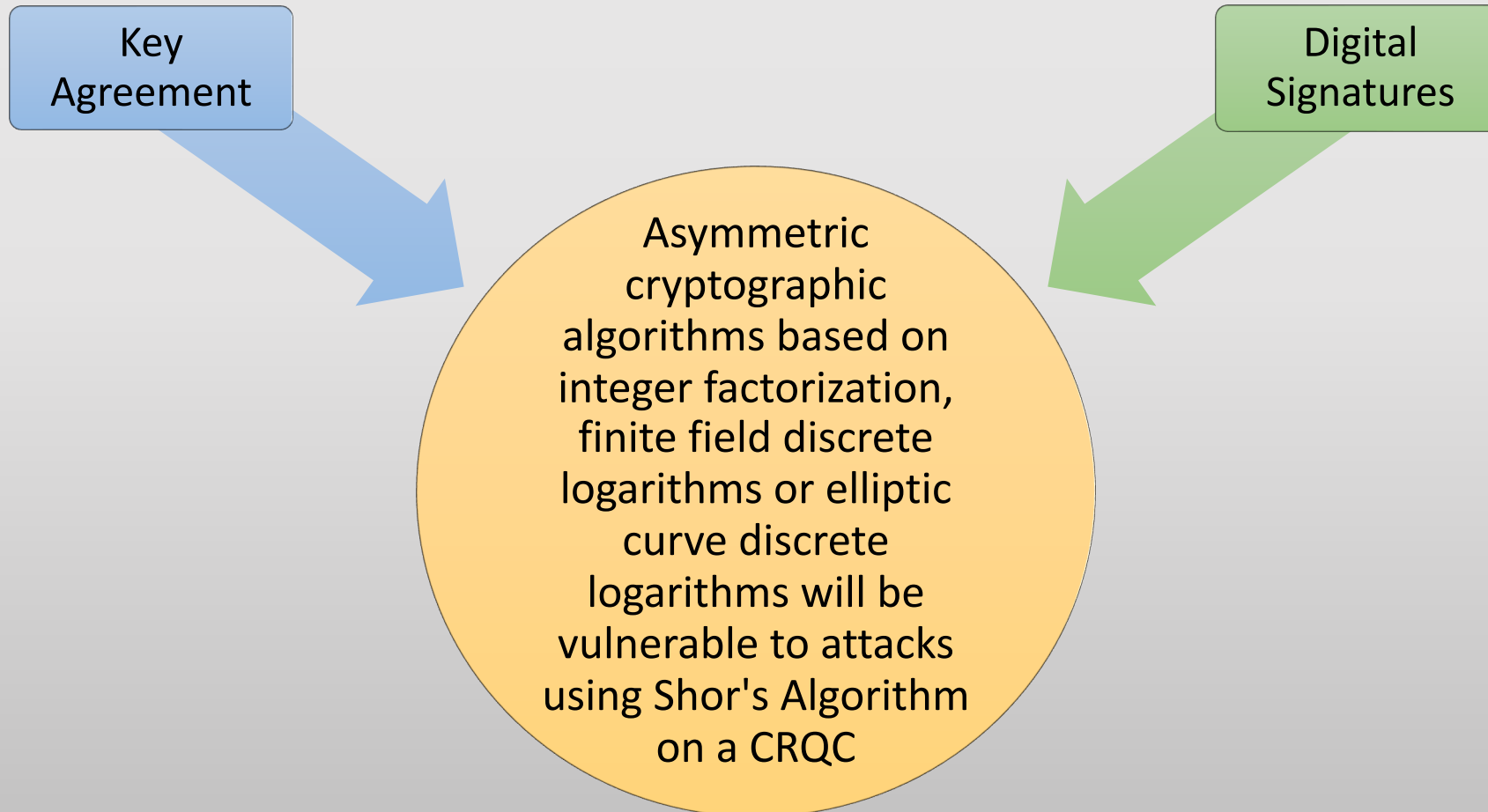
# Symmetric Cryptography (Grover's algorithm)

- Grover's algorithm theoretically requires us to double the key-size (AES128 -> AES256)
- This would affect all symmetric crypto algorithms that are used currently.
- Yet, this is a misconception:
  - Grover's algorithm is highly **non parallelizable**
  - Even thousands of QCs running in parallel would offer minimal gains in breaking symmetric keys.
- NIST still standardizes AES-128 [\(link\)](#).

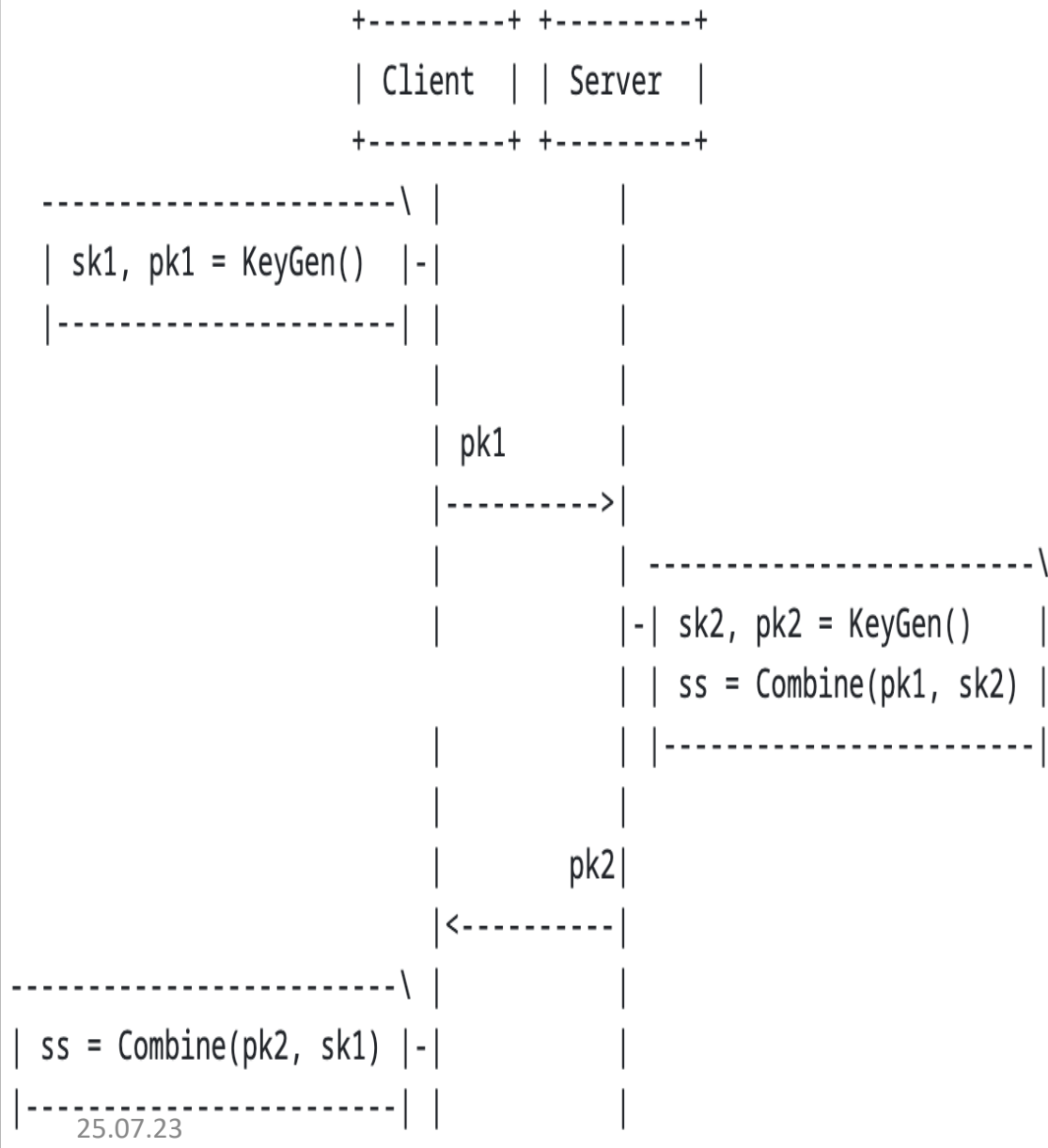
# A Harvest Now and Decrypt Later (HNDL) Attack on TLS



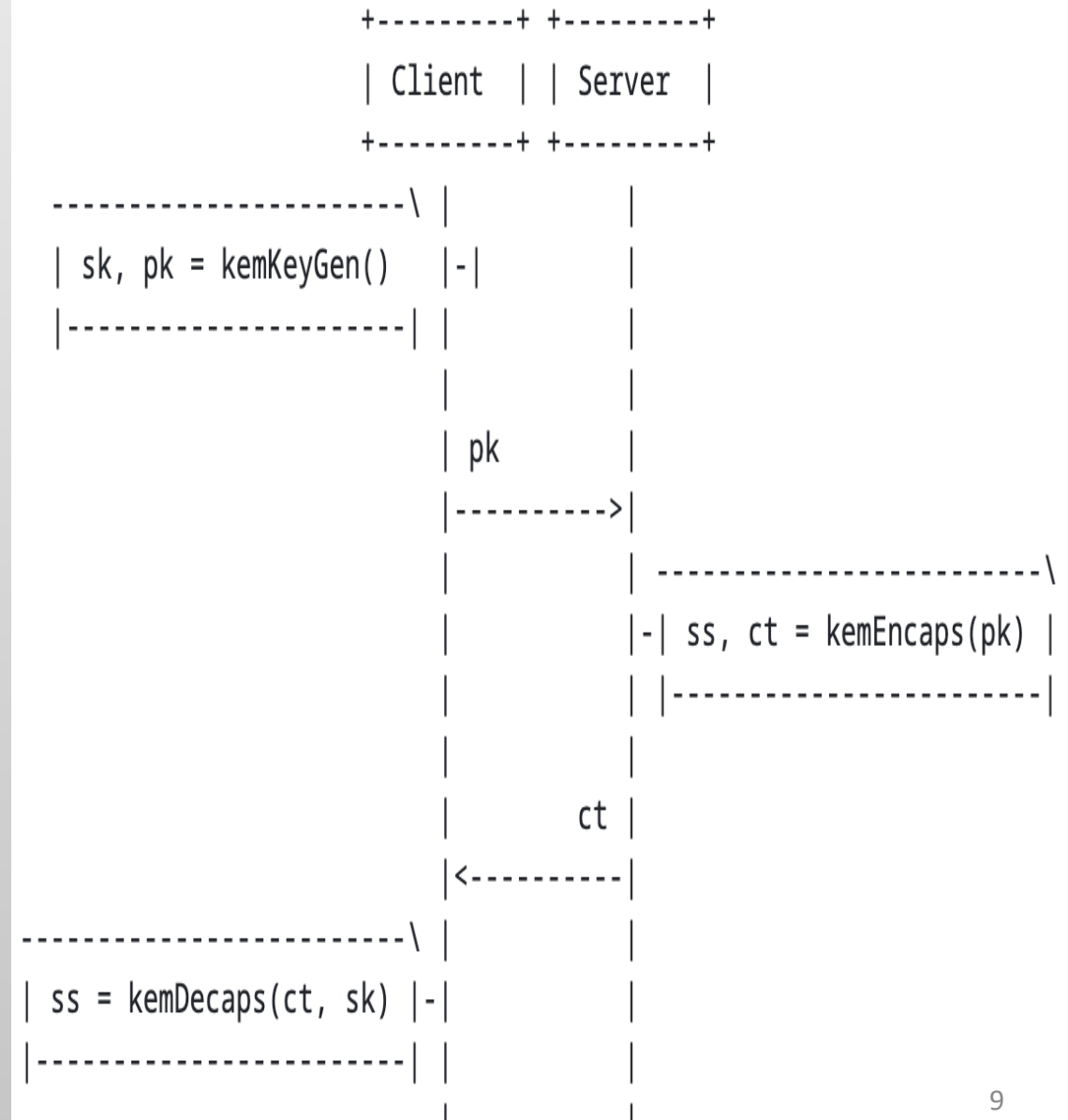
# Traditional Cryptographic Primitives that Could Be Replaced by PQC



# DH KEX



# KEM



# HPKE

- HPKE (Hybrid public key encryption) {RFC9180} deals with a variant of KEM which is essentially a PKE of arbitrary sized plaintexts for a recipient public key.
- It works with a combination of KEMs, KDFs and AEAD schemes (Authenticated Encryption with Additional Data).
- HPKE includes three authenticated variants, including one that authenticates possession of a pre-shared key and two optional ones that authenticate possession of a key encapsulation mechanism (KEM) private key.

# Note on Kyber

- Kyber, which is a PQC KEM does not support the static-ephemeral key exchange that allows HPKE based on DH based KEMs its (optional) authenticated modes as discussed in Section 1.2 of [I-D.westerbaan-cfrg-hpke-xyber768d00-02](#).

# NIST Candidates Selected for Standardization/4th Round Candidates



# Security property for KEM and Signatures

## IND-CCA2

- IND-CCA2 (Indistinguishability under adaptive Chosen-Ciphertext Attack) is an advanced security notion for encryption schemes. It ensures the confidentiality of the plaintext, resistance against chosen-ciphertext attacks, and prevents the adversary from forging new ciphertexts.
- Kyber, BIKE, Classic McEliece provide IND-CCA2 security

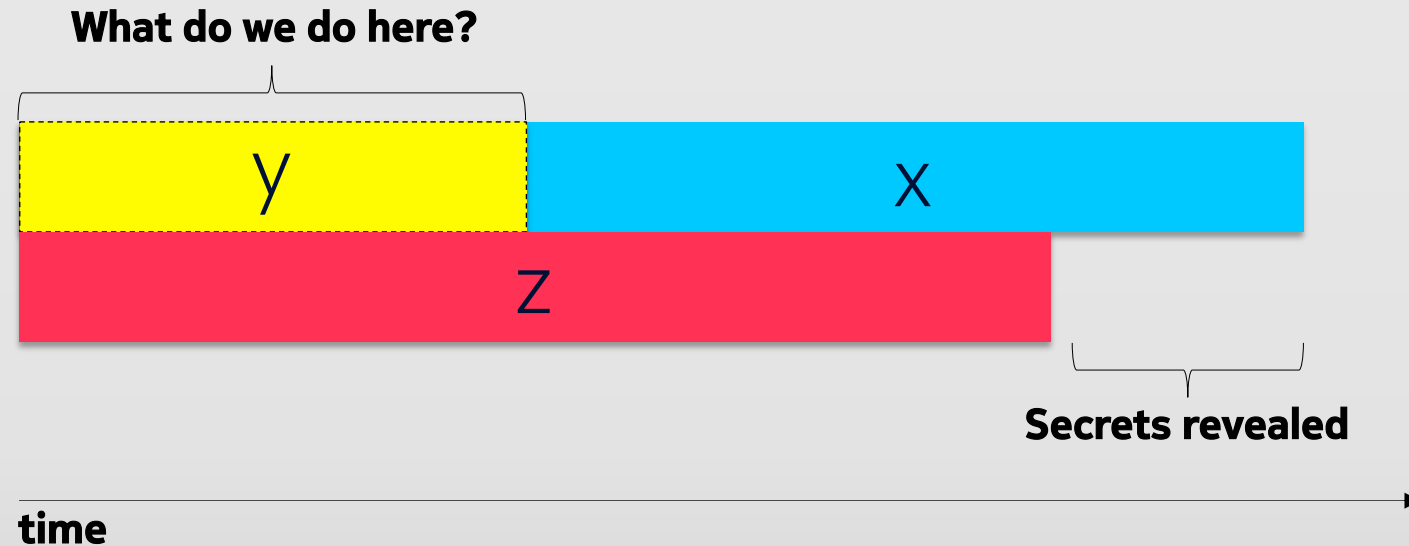
## EUFCMA

- EUFCMA (Existential Unforgeability under Chosen Message Attack) [GMR88] is a security notion for digital signature schemes. It guarantees that an adversary, even with access to a signing oracle, cannot forge a valid signature for an arbitrary message. EUFCMA provides strong protection against forgery attacks, ensuring the integrity and authenticity of digital signatures by preventing unauthorized modifications or fraudulent signatures.
- Dilithium, Falcon and Sphincs+ provide EUFCMA security.

# PQC standardization timeline



# Mosca's theorem of cybersecurity in the quantum era<sup>1</sup>



If  $x + y > z$ , then start worrying

- x: time we want to keep our systems secure**
- y: time to deploy a quantum-safe migration plan**
- z: time to build a large-scale quantum computer (2030s?)**

[1] <https://eprint.iacr.org/2015/1075.pdf>

# Details of XMSS and LMS

- PQC: XMSS (RFC8391) and LMS (RFC8554) are stateful hash-based signature schemes.
- Reusing a secret key state compromises cryptographic security guarantees.
- Signing a potentially large but fixed number of messages
- The number of signing operations depends upon the size of the tree.
- Increasing the number of layers reduces key generation time exponentially and signing time linearly at the cost of increasing the signature size linearly.

# Hash-then-Sign vs Sign-then-Hash

- Hash-then-Sign: Fixed size digest of the message is signed.
- Rely on the collision-resistance of the hash function.
- Reduces the size of signed messages.
- Protocols like TLS 1.3 and DNSSEC use the Hash-then-Sign paradigm.
- PQC Signature schemes internally apply hash functions.

# Security levels (PQC Algorithms: NIST)

PQ Security Level	AES/SHA3 hardness	PQC Algorithm
1	Find optimal key in AES-128	Kyber512, Falcon512, Sphincs+SHA256 128f/s
2	Find optimal collision in SHA3-256	Dilithium2
3	Find optimal key in AES-192	Kyber768, Dilithium3, Sphincs+SHA256 192f/s
4	Find optimal collision in SHA3-384	No algorithm tested at this level
5	Find optimal key in AES-256	Kyber1024, Falcon1024, Dilithium5, Sphincs+SHA256 256f/s

**Key takeaway:** Users can leverage the required algorithm based on the security level based on their use case. The security is defined as a function of resources required to break AES and SHA3 algorithms, i.e., optimal key recovery for AES and optimal collision attacks for SHA3.

# Key & ciphertext/signatures of PQC Algorithms on different security levels

PQ Security Level	Algorithm	Public key size (in bytes)	Private key size (in bytes)	Ciphertext/Signature size (in bytes)
1	Kyber512	800	1632	768
1	Falcon512	897	1281	666
2	Dilithium2	1312	2528	2420
3	Kyber768	1184	2400	1088
5	Falcon1024	1793	2305	1280
5	Kyber1024	1568	3168	1588

# SPHINCS+ and its many variants (Simple only)

SPHINCS+ algorithm security levels for different categories i.e., (f) for fast verification and (s) for compactness/smaller. Both SHA256 and SHAKE-256 parametrisation output the same signature sizes, so both have been included.

PQ Security Level	Algorithm	Public key size (in bytes)	Private key size (in bytes)	Signature size (in bytes)
1	SPHINCS+--{SHA2,SHAKE}-128f	32	64	17088
1	SPHINCS+--{SHA2,SHAKE}-128s	32	64	7856
3	SPHINCS+--{SHA2,SHAKE}-192f	48	96	35664
3	SPHINCS+--{SHA2,SHAKE}-192s	48	96	16224
5	SPHINCS+--{SHA2,SHAKE}-256f	64	128	49856
5	SPHINCS+--{SHA2,SHAKE}-256s	64	128	29792

# Falcon vs Dilithium vs Sphincs+

- Dilithium is known for its relatively fast signature generation, while Falcon can provide more efficient signature verification.
- Falcon also has lower key and signature sizes as compared to Dilithium.
- SPHINCS+ offers smaller key sizes, larger signature sizes, slower signature generation, and slower verification when compared to Dilithium and Falcon.

# Challenges in Falcon's Signing Operations

- Falcon's signing operations require constant-time, 64-bit floating point operations to avoid catastrophic side channel vulnerabilities. Doing this correctly (which is also platform-dependent to an extreme degree) is very difficult, as NIST's report noted.
- Providing a masked implementation of Falcon also seems impossible, per the authors at the RWPQC 2023 symposium earlier this year.

# PQC vs Traditional KEMs/KEEs

PQ Security Level	Algorithm	Public key size (in bytes)	Private key size (in bytes)	Ciphertext size (in bytes)
Traditional	P256_HKDF_SHA256	65	32	65
Traditional	P521_HKDF_SHA512	133	66	133
Traditional	X25519_HKDF_SHA256	32	32	32
1	Kyber512	800	1632	768
3	Kyber768	1184	2400	1088
5	Kyber1024	1568	3168	1588

# PQC vs Traditional Signatures

PQ Security Level	Algorithm	Public key size (in bytes)	Private key size (in bytes)	Signature size (in bytes)
Traditional	RSA2048	256	256	256
Traditional	P256	64	32	64
1	Falcon512	897	1281	666
2	Dilithium2	1312	2528	768
3	Dilithium3	1952	4000	3293
5	Falcon1024	1793	2305	1280

# PQ/T Hybrid Confidentiality

- Protect from protect from "Harvest Now, Decrypt Later" attack
- Concatenate hybrid key agreement scheme
  - Hybrid key exchange in TLS 1.3 <https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/>
    - It provides hybrid confidentiality but does not address hybrid authentication
    - the client's key-share contains two component public keys, one for a post-quantum algorithm and one for a traditional algorithm (ECDH ephemeral key-share)
    - For the server's share, concatenation of ct (ciphertext) and ephemeral key-share (ECDH)
    - hybrid secret by concatenating the two shared secrets
- Cascade hybrid key agreement scheme
  - Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2) <https://datatracker.ietf.org/doc/rfc9370/>
  - Allows the negotiation of one or more PQC algorithms to exchange data, in addition to the existing (EC)DH key exchange data.

# PQ/T Hybrid Authentication

- Protect from on-path attacker using CRQC
- Authentication through a PQ/T hybrid scheme or a PQ/T hybrid protocol, as long as at least one component algorithm remains secure to provide the intended security level.
- The frequency and duration of system upgrades and the time when CRQCs will become widely available need to be weighed in to determine whether and when to support the PQ/T Hybrid Authentication property.
- Discussions in LAMPS WG to use PQ/T Hybrid Certificate

# Hybrid Mechanisms

- It is possible to use more than two algorithms together in a hybrid scheme, and there are multiple possible ways those algorithms can be combined.
- For the purposes of a post-quantum transition, the simple combination of a post-quantum algorithm with a single classical algorithm is the most straightforward, but the use of multiple post-quantum algorithms with different hard math problems has also been considered.
- When combining algorithms, it is possible to require that both algorithms validate (the so-called "and" mode) or that only one does (the "or" mode), or even some more complicated scheme.
- Schemes that do not require both algorithms to validate only have the strength of the weakest algorithm, and therefore offer little or no security benefit.

# Hybrid Mechanisms (Cont.)

- When combining keys in an "and" mode, it may make more sense to consider them to be a single composite key, instead of two keys.
- This generally requires fewer changes to various components of PKI ecosystems, many of which are not prepared to deal with two keys or dual signatures.
- To an implementer, a "composite" algorithm composed of two other algorithms is simply a new algorithm, and support for adding new algorithms generally already exists.

# Hybrid Mechanisms (Cont.)

- All that needs to be done is to standardize the formats of how the two keys from the two algorithms are combined into a single data structure, and how the two resulting signatures are combined into a single signature.
- The answer can be as simple as concatenation, if the lengths are fixed or easily determined.
- **Many of these points are still being actively explored and discussed, and the consensus may change over time**

# Security Considerations - Cryptanalysis

- Classical cryptanalysis exploits weaknesses in algorithm design, mathematical vulnerabilities, or implementation flaws, whereas quantum cryptanalysis harnesses the power of CRQCs to solve specific mathematical problems more efficiently.
- Both pose threats to the security of cryptographic algorithms, including those used in PQC
- Developing and adopting new cryptographic algorithms resilient against these threats is crucial for ensuring long-term security in the face of advancing cryptanalysis techniques

# Security Considerations - Cryptographic Agility

- Cryptographic agility is relevant for both classical and quantum cryptanalysis as it enables organizations to adapt to emerging threats, adopt stronger algorithms, comply with standards, and plan for long-term security in the face of evolving cryptanalytic techniques and the advent of CRQCs.
- Several PQC schemes are available that need to be tested; cryptography experts around the world are pushing for the best possible solutions, and the first standards that will ease the introduction of PQC are being prepared

# Hybrid Key Exchange : Bridging the Gap Between Post-Quantum and Traditional Cryptography

- Post-quantum algorithms selected for standardization are relatively new and they have not been subject to the same depth of study as traditional algorithms.
- In addition, certain deployments may need to retain traditional algorithms due to regulatory constraints, for example FIPS compliance.
- Hybrid key exchange enables potential security against "Harvest Now, Decrypt Later" attack while not fully abandoning traditional cryptosystems.

# Contributing to this document

- Comments and Suggestions are welcome
- The document is being collaborated on: [tiredy2/pqc-for-engineers \(github.com\)](https://github.com/tiredy2/pqc-for-engineers)
- E-mail archive: [pqc \(ietf.org\)](https://www.ietf.org/pqc)