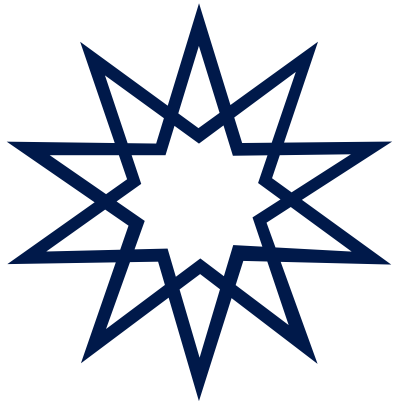


# QUANTUM COMPUTING AND THE FUTURE OF CYBER SECURITY

SIB5100 Information Security and Management

ÖNDER GÖRMEZ

21501035



**YTÜ** **YILDIZ TEKNİK**  
**ÜNİVERSİTESİ**

# Agenda

**I. Introduction**

**III. Conclusions**

**V. Q & A**

**II. PQC Algorithms**

**IV. References**

## Introduction - Terms and Abbreviations

<b>Kısaltma</b>	<b>Açılımı</b>
PQC	Post-Quantum Cryptography
HSM	Hardware Security Module
KMS	Key Management System
XMSS	Extended Merkle Signature System
LMS	Leighton-Micali Signatures
ML-KEM	Module-Lattice-Based Key-Encapsulation Mechanism Standard
ML-DSA	Module-Lattice-Based Digital Signature Standard
SLH-DSA	Stateless Hash-Based Digital Signature Standard

TABLE I

KULLANILAN KISALTMALAR LİSTESİ

## Introduction - Terms and Abbreviations

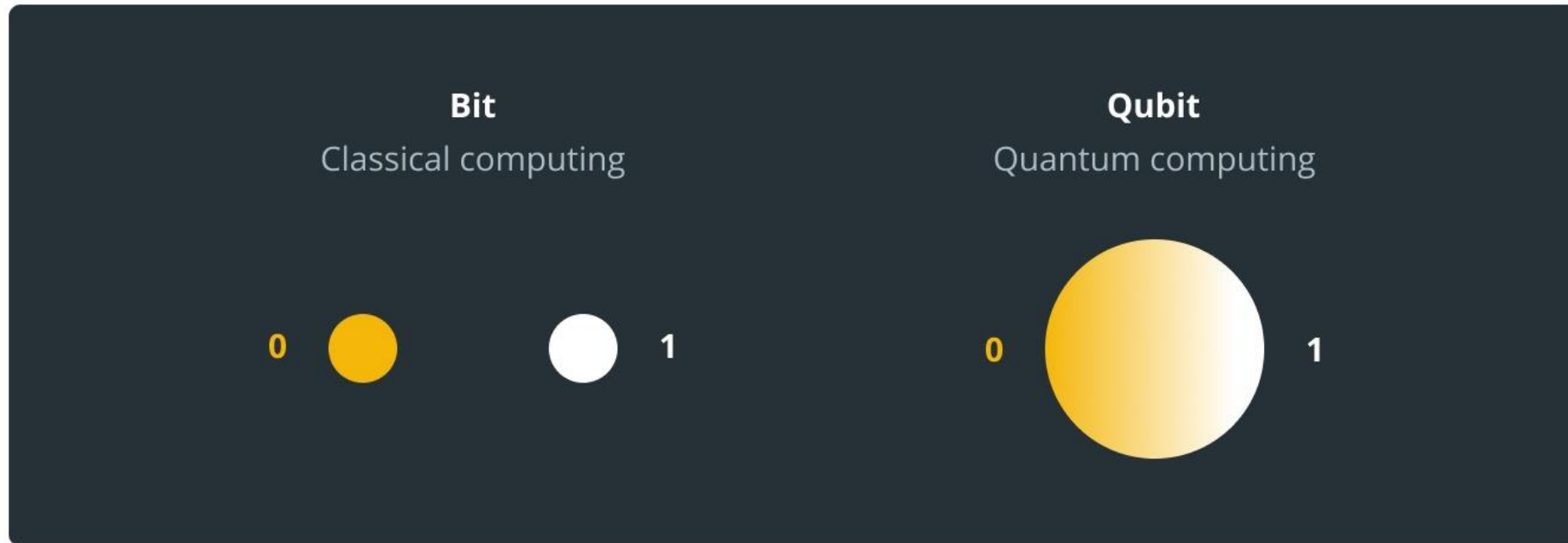
<b>İngilizce Terim</b>	<b>Türkçe Karşılığı</b>
Entangled quantum objects	Dolanık kuantum nesneleri
Quantum bits	Kuantum bitleri
Qubits	Kübitler
Superposition	Üst üste binme
Entanglement	Dolanıklık
Vulnerable Algorithms	Güvenlik Açığı Bulunan Algoritmalar

**TABLE II**  
**KULLANILAN TERİMLER LİSTESİ**

## Introduction - What is a quantum computer?



## Introduction - What is a qubit?



 | cointelegraph.com

source: *Medium*

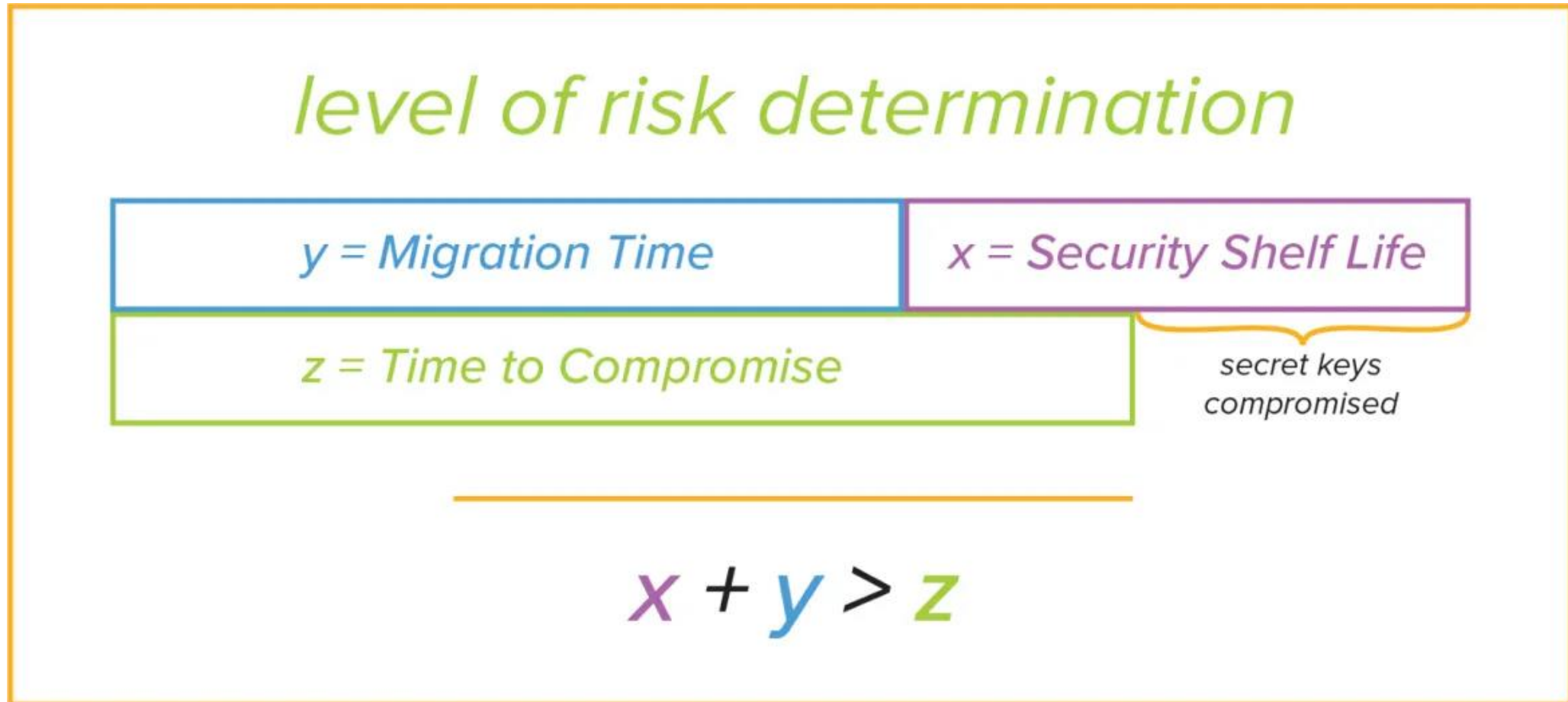
Source: <https://cointelegraph.com/learn/articles/post-quantum-threats-to-proof-of-work-cryptocurrencies>

## Introduction - After quantum computer?

Güvenliği sorgulanacak olan sektörler:

- Crypto Currencies (PoW)
  - Elliptic Curve Digital Signature Algorithm (ECDSA) -> Schor's algorithm
  - Cryptographic hash functions (SHA-256) -> Grover's algorithm
    - Mitigation SHA-256 to SHA-512
- Entire Internet Infrastructure
- Telecommunication
- ....

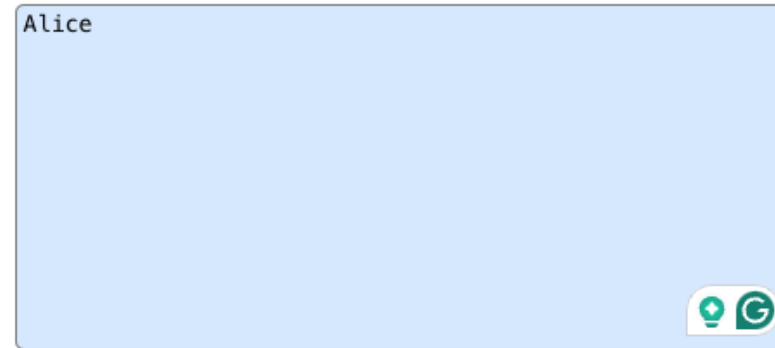
## Introduction - Mosca's Theorem



Source: <https://www.redhat.com/en/blog/post-quantum-cryptography-introduction>



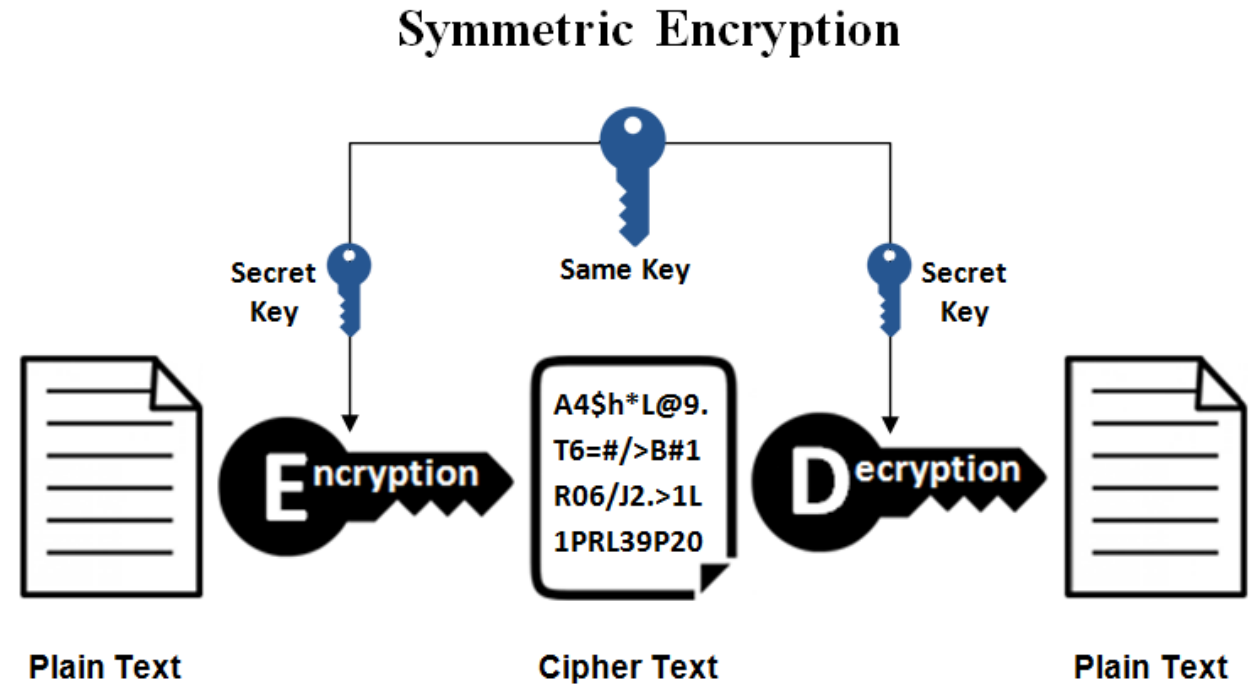
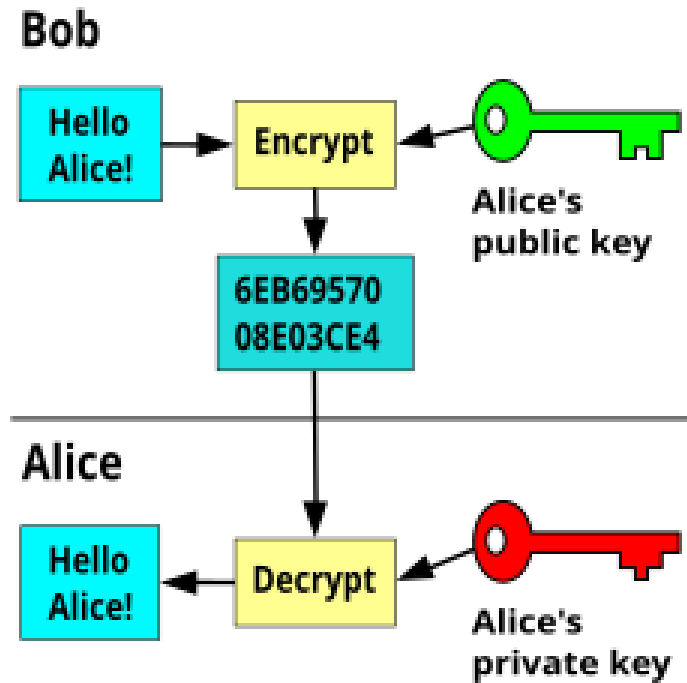
## Introduction - What is Cryptography?



Calculate Hashes Copy to clipboard [\(undo\)](#)

NTLM	A26A81E411BA2D307679C9C560326B85	MD2	42b8fbd3f576de58bebe7d9c3b2c297b	MD4	f872eb19e2ef9f11f25b4a71b516f9e8
MD5	64489c85dc2fe0787b85cd87214b3810	MD6-128	8afd98e8c3750640734166b7f870b3c1	MD6-256	fbf2ad62dee939cfbd58a0142c55451c55cffe51a3'
MD6-512	387ccd91e43fefed1b5f9ef90976b6b336b2bdfdd3	RipeMD-128	25b880972ed1dd1059594be76393e1a9	RipeMD-160	b417ed99b95ce21448b7d789c50009e4f088e3a7
RipeMD-256	aa937768ee9ceee5a0b53234bdd81e7be8eb1d5c	RipeMD-320	fe7c2e34982faabd43a17db85b834238814d5db61	SHA1	35318264c9a98faf79965c270ac80c5606774df1
SHA3-224	119acc8d36094e71ece854096b2c91d4788c2d3d	SHA3-256	ada0018bcd09ed8fc81b323331950a89541d2416	SHA3-384	0cbd5b30ed604cb3ff2ea3b0d77451f7fae30baa02
SHA3-512	c763cb065ab61e8277d781313f3653093ce4cfa5f	SHA-224	6874ecdadb214ee888e37c8c983e2f1c9c0ed169c	SHA-256	3bc51062973c458d5a6f2d8d64a023246354ad7e
SHA-384	d4b960a3af641fda19285d49b5fd31fb8640165ecf	SHA-512	299403b3d6b5c6244fc0ec6f278cb8c233734f0c1f	CRC16	9819
CRC32	e64f9343	Adler32	056001df	Whirlpool	3f10d2da3a275ed8fa2c5e140dc10b35fd22111c6f

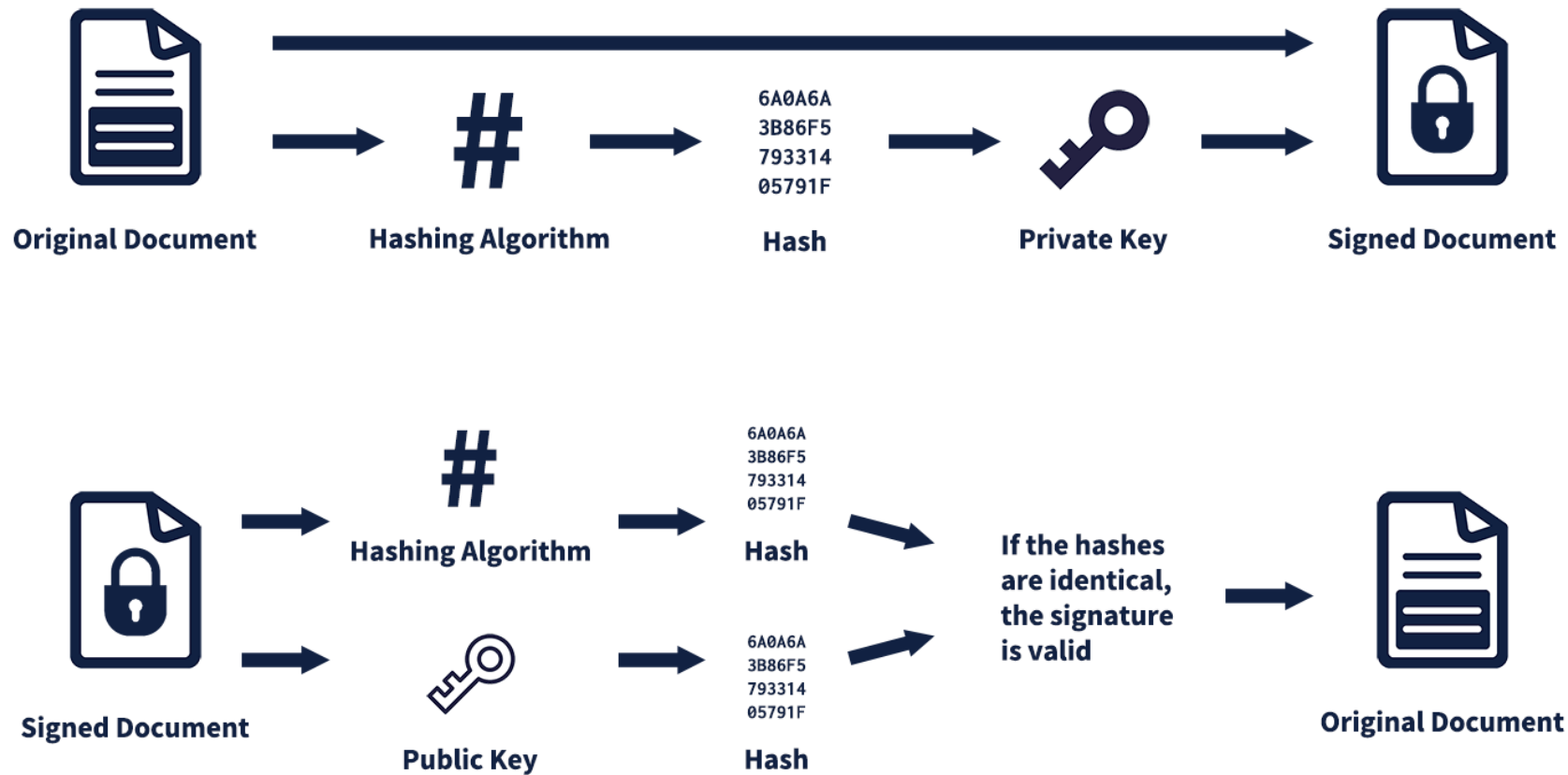
## Introduction - What is Cryptography?



Source: [https://upload.wikimedia.org/wikipedia/commons/thumb/f/f9/Public\\_key\\_encryption.svg/250px-Public\\_key\\_encryption.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/f/f9/Public_key_encryption.svg/250px-Public_key_encryption.svg.png)

Source: <https://www.ssl2buy.com/wp-content/uploads/2015/12/Symmetric-Encryption.png>

## Introduction - What is Cryptography?



Source: [https://techterms.com/img/xl/digital\\_signature\\_796.png](https://techterms.com/img/xl/digital_signature_796.png)

© TechTerms.com

## Introduction - What is a PQC?



Source: <https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards>

# Agenda

**I. Introduction**

**III. Conclusions**

**V. Q & A**

**II. PQC Algorithms**

**IV. References**

## PQC - Introduction

Aksiyon planı aşağıdakine benzer olmalıdır.

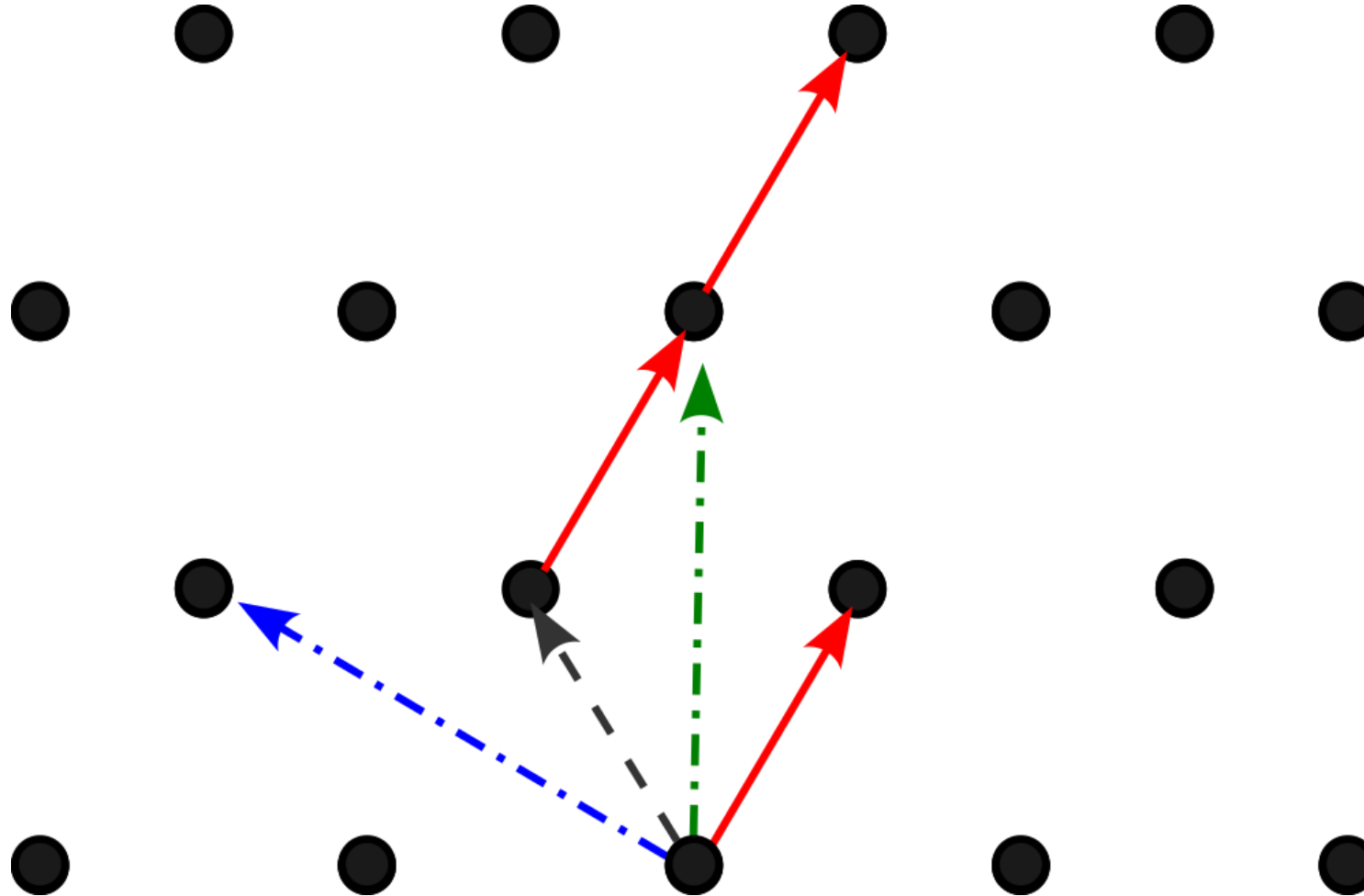
- RSA, DSA, ECC, DH – vulnerable
- TLS, SSH, S/MIME, PGP, IPSEC protocols may be depends on it
- VPNs, Kerberos may be depends on it
- Browsers, encrypted messaging, disk encryption, authentication schemes, ... applications may be depends on it

## PQC - Hash-based Signatures

Value to sign	Private key	Public key
0	sk0 (randomly generate)	pk0(=HASH(sk0))
1	sk1 (randomly generate)	pk1(=HASH(sk1))
2	sk2 (randomly generate)	pk2(=HASH(sk2))
3	sk3 (randomly generate)	pk3(=HASH(sk3))

Source: <https://www.redhat.com/en/blog/post-quantum-cryptography-hash-based-signatures>

## PQC - Lattice-based Cryptography





## PQC - Lattice-based Cryptography

- Klasik encryption yöntemlerinin yerine geçecek.

The other consideration is how you represent the vectors. As with classical cryptography, you can map these vectors into finite fields (either prime fields or binary fields). The main difference between the fields used in lattice operations and those used in Diffie-Helman

or Elliptic Curves is one of size. Lattice fields usually fit within the computer's word size, so there is no need to use special big integer arithmetic libraries. The security doesn't come from the size of our vector values, but in the size of the vectors and the number of vectors in our basis.

## PQC - Code-based Cryptography

- Ana algoritma lattice-based cryptography
- Bu backup senaryosu
- Just in case
- Harberleşme altyapılarında kullanılan error correction code tabanlı bir yöntem.
  - Hamming Codes
  - Goppa Codes

## PQC - Implementation Started

### Post-quantum cryptography support

With the PKCS #11 API, you can also perform [post-quantum cryptographic](#) operations. Traditional cryptography relies on complicated mathematical problems that are difficult for classical computers to solve. However, with the computing capabilities, quantum computers can solve these problems. Post-quantum cryptography is considered to be resistant to cryptanalytic attacks from quantum computers. It usually uses asymmetric algorithms and has multiple approaches.

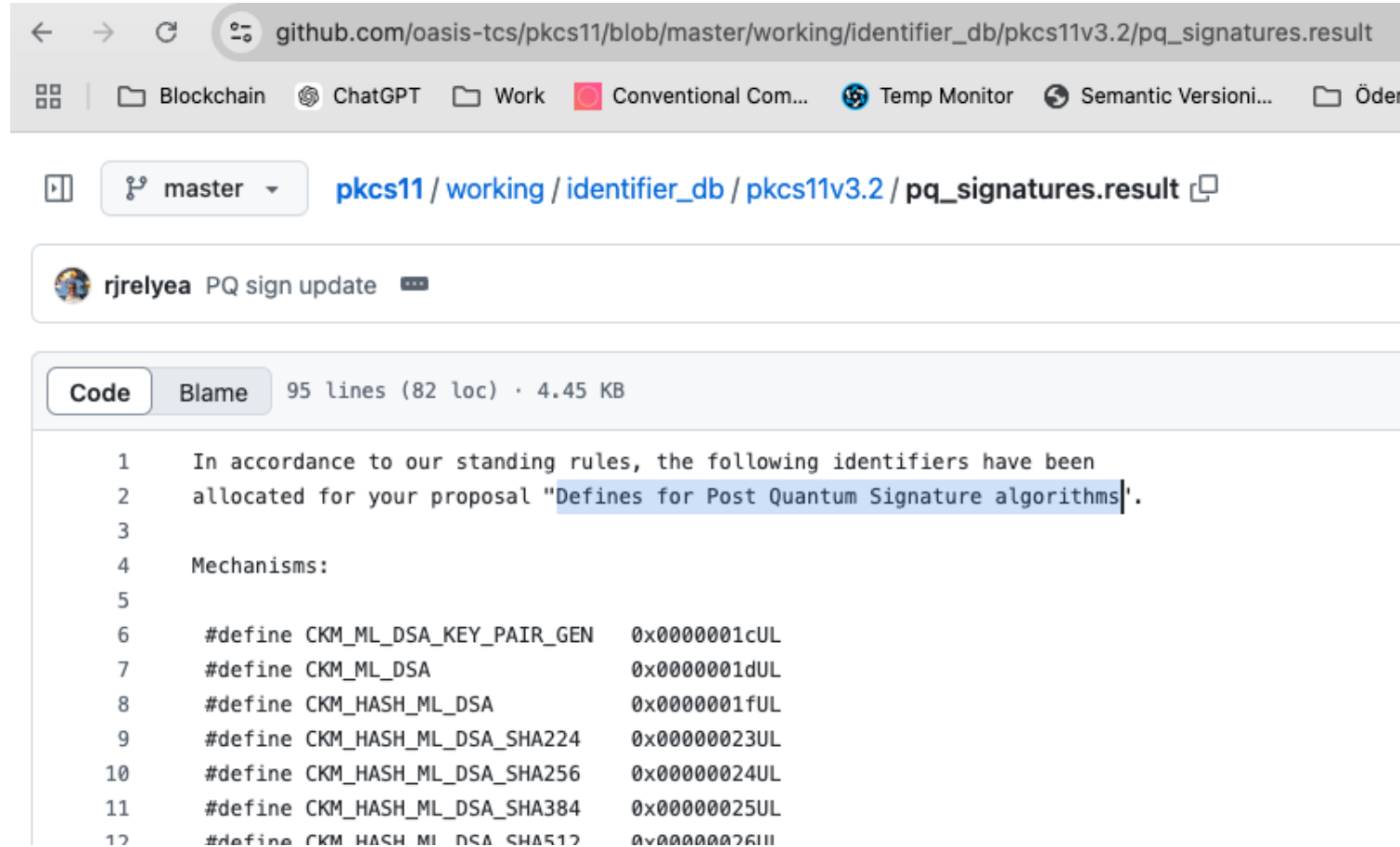
The PKCS #11 API provides the [Dilithium algorithm](#) for post-quantum cryptography. It is a lattice-based digital signature scheme and can be used for signature generation and verification. Currently, only the [high-security version of round 2 Dilithium](#) is supported and it is not available for `C_SignUpdate` and `C_VerifyUpdate` operations.

**Note:** The Dilithium algorithm is supported only by the IBM 4769 crypto card, also referred to as Crypto Express 7S (CEX7S). If you create your instances in Virtual Private Cloud (VPC) based regions, where the CEX7S crypto cards are used, you can use the Dilithium algorithm for post-quantum cryptography with the PKCS #11 API. For a list of VPC-based regions, see [Regions and locations](#).

For more information about Dilithium algorithm support in PKCS #11, see [PKCS #11 API reference](#). You can also find Dilithium algorithm code examples in the [GitHub sample repository](#).

Source: <https://cloud.ibm.com/docs/hs-crypto?topic=hs-crypto-pkcs11-intro>

## PQC - Implementation Started



github.com/oasis-tcs/pkcs11/blob/master/working/identifier\_db/pkcs11v3.2/pq\_signatures.result

Blockchain ChatGPT Work Conventional Com... Temp Monitor Semantic Versioni... Öder

master pkcs11 / working / identifier\_db / pkcs11v3.2 / pq\_signatures.result

rjrelyea PQ sign update

Code Blame 95 lines (82 loc) · 4.45 KB

```
1 In accordance to our standing rules, the following identifiers have been
2 allocated for your proposal "Defines for Post Quantum Signature algorithms".
3
4 Mechanisms:
5
6 #define CKM_ML_DSA_KEY_PAIR_GEN 0x0000001cUL
7 #define CKM_ML_DSA 0x0000001dUL
8 #define CKM_HASH_ML_DSA 0x0000001fUL
9 #define CKM_HASH_ML_DSA_SHA224 0x00000023UL
10 #define CKM_HASH_ML_DSA_SHA256 0x00000024UL
11 #define CKM_HASH_ML_DSA_SHA384 0x00000025UL
12 #define CKM_HASH_ML_DSA_SHA512 0x00000026UL
```

Source: [https://github.com/oasis-tcs/pkcs11/blob/master/working/identifier\\_db/pkcs11v3.2/pq\\_signatures.result](https://github.com/oasis-tcs/pkcs11/blob/master/working/identifier_db/pkcs11v3.2/pq_signatures.result)

# Agenda

**I. Introduction**

**III. Conclusions**

**V. Q & A**

**II. PQC Algorithms**

**IV. References**

## Conclusions

While there have been no substantive changes made to the standards since the draft versions, NIST has changed the algorithms' names to specify the versions that appear in the three finalized standards, which are:

- **Federal Information Processing Standard (FIPS) 203** (<https://csrc.nist.gov/pubs/fips/203/final>), intended as the primary standard for general encryption. Among its advantages are comparatively small encryption keys that two parties can exchange easily, as well as its speed of operation. **The standard is based on the CRYSTALS-Kyber algorithm, which has been renamed ML-KEM, short for Module-Lattice-Based Key-Encapsulation Mechanism.**
- **FIPS 204** (<https://csrc.nist.gov/pubs/fips/204/final>), intended as the primary standard for protecting digital signatures. **The standard uses the CRYSTALS-Dilithium algorithm, which has been renamed ML-DSA, short for Module-Lattice-Based Digital Signature Algorithm.**
- **FIPS 205** (<https://csrc.nist.gov/pubs/fips/205/final>), also designed for digital signatures. **The standard employs the SpHincS+ algorithm, which has been renamed SLH-DSA, short for Stateless Hash-Based Digital Signature Algorithm.** The standard is based on a different math approach than ML-DSA, and it is intended as a backup method in case ML-DSA proves vulnerable.

Similarly, **when the draft FIPS 206 standard built around FALCON is released, the algorithm will be dubbed FN-DSA, short for FFT (fast-Fourier transform) over NTRU-Lattice-Based Digital Signature Algorithm.**

Source: <https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards>

## Conclusions

### Selected Algorithms:

- FIPS 203 -> ML-KEM -> Module-Lattice-Based Key-Encapsulation Mechanism Standard
- FIPS 204 -> ML-DSA - Module-Lattice-Based Digital Signature Standard
- FIPS 205 -> SLH-DSA - Stateless Hash-Based Digital Signature Standard

Source: <https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards>

## Conclusions

- IBM Quantum Computer ve Post Quantum Cryptography üzerine çalışıyor.
- PKCS11 komitesinde bir Red Hat çalışanı, dolaylı olarak IBM var.

## Future Works

- FIPS 203, FIPS 204, FIPS 205 , FIPS 206 (draft) incelenebilir.



# Agenda

**I. Introduction**

**II. PQC Algorithms**

**III. Conclusions**

**IV. References**

**V. Q & A**

## References

- [1] [An overview of post-quantum threats to proof-of-work cryptocurrencies](#)
- [2] [Post-quantum cryptography: An introduction](#)
- [3] [Post-quantum cryptography: Hash-based signatures](#)
- [4] [Post-quantum cryptography: Lattice-based cryptography](#)
- [5] [Post-quantum cryptography: Code-based cryptography](#)
- [6] [NIST Releases First 3 Finalized Post-Quantum Encryption Standards](#)

## References



### Robert Relyea

Principal Programmer



Relyea has worked in crypto security on the Network Security System code used in Mozilla browsers since 1996. He joined Red Hat in 2006. He is also the co-chair of the OASIS PKCS #11 Technical Committee.

# Agenda

**I. Introduction**

**III. Conclusions**

**V. Q & A**

**II. PQC Algorithms**

**IV. References**

## Q & A



**THANK YOU FOR LISTENING...**