Data: 27 Settembre 2024
Laureando: Tomas Lovato
Relatore: Prof. Luca Boldrin

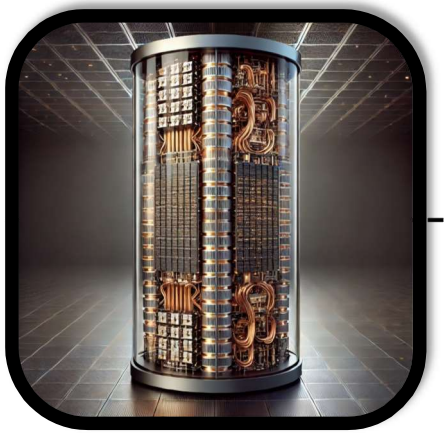
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

1222 · 2022
800
ANNI

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Analisi di Algoritmi di Firma Digitale Post-Quantum

1. Cybersicurezza



2. Ricerca nelle Quantum Tecnologies

3. L'identità digitale

4. I processi di standardizzazione

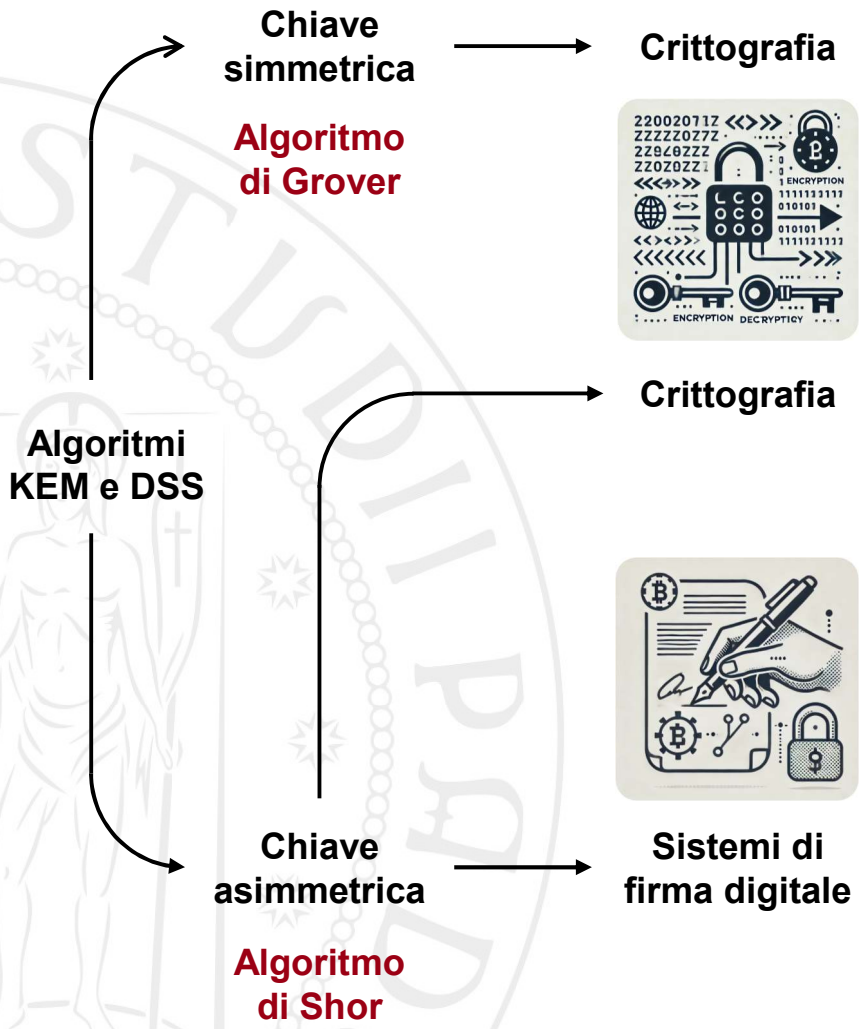
Attualmente, buona parte della sicurezza informatica sfrutta tecnologie come **RSA** ed **ECC**, parti di algoritmi di firma digitale che utilizziamo per garantire integrità e autenticità.

Tuttavia, tutto questo è destinato a cambiare radicalmente con l'avvento dei **computer quantistici**. Potrebbero infrangere gli algoritmi di sicurezza che oggi consideriamo sicuri in pochi secondi.

FONTI:

ACN: https://www.acn.gov.it/portale/documents/20119/85999/ACN_Crittografia_Quantum_Safe.pdf

NIST: https://csrc.nist.gov/projects/post-quantum-cryptography



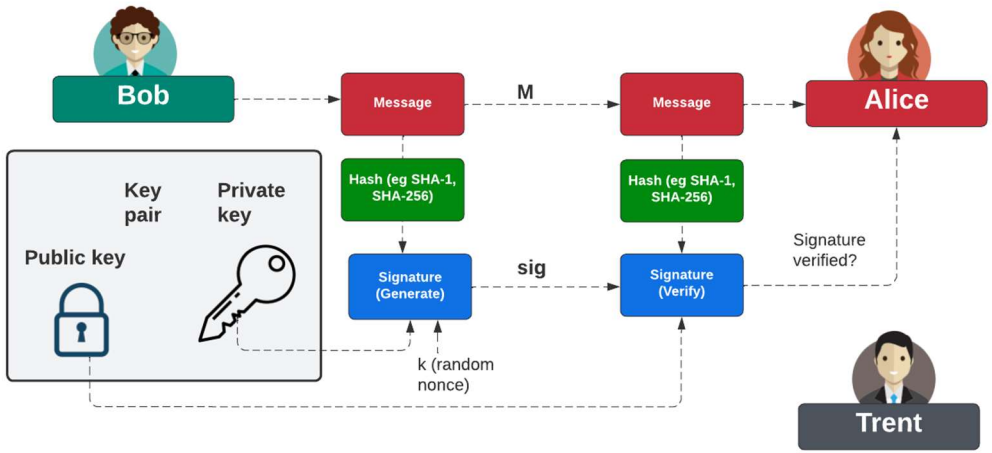
La **sicurezza** nelle comunicazioni è basata su tre proprietà:

1. **Autenticità**
2. **Integrità**
3. **Confidenzialità**

In questa ricerca vengono approfonditi i **DSS** (Digital Signature Systems), il cui scopo è garantire l'integrità e la confidenzialità di una comunicazione tra due o più entità.

La firma digitale necessita espressamente di due chiavi:

1. **Chiave privata**
2. **Chiave pubblica**



FONTI:

NATURE: https://www.nature.com/articles/nature23461

NIST: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

Un altro motivo per cui ci si sta concentrando sugli algoritmi a chiave asimmetrica è che sono i più colpiti: **l'algoritmo di Shor** minaccia fortemente la sicurezza degli attuali sistemi a chiave asimmetrica, in termini di tempi.

Security Strength	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
≤ 80	2TDEA[21]	$L = 1024$ $N = 160$	$k = 1024$	$f = 160\text{-}223$
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224\text{-}255$
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	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

Definizione dei Security Levels (NIST)

Algoritmo	Dimensione della chiave (in bit)	Livello di sicurezza in bit (computer attuale)	Livello di sicurezza in bit (computer quantistico)
RSA-1024	1024	80	∼0
RSA-2048	2048	112	∼0
ECC-256	256	128	∼0
ECC-384	384	192	∼0
AES-128	128	128	∼64
AES-256	256	256	∼128

Security Levels di RSA ed ECC considerando l'algoritmo di Shor

FONTI:

NATURE: https://www.nature.com/articles/nature23461

NIST: https://www.nist.gov/publications/status-report-third-round-nist-post-quantum-cryptography-standardization-process

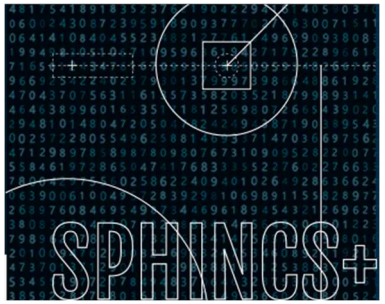
L'obiettivo della tesi è **verificare le performance** di parte dei candidati, specialmente i finalisti, e comprendere se essi sono veramente pronti per la standardizzazione e l'utilizzo nei futuri sistemi di crittografia e di **firma digitale**.

I FINALISTI DEL ROUND 3

FALCON
Fast-Fourier Lattice-based
Compact Signatures over NTRU

FALCON

CRYSTALS DILITHIUM



SPHINCS+

FONTI:
NIST PROGETTO PQC: https://csrc.nist.gov/projects/post-quantum-cryptography
ENISA: https://www.enisa.europa.eu/publications/post-quantum-cryptography-current-state-and-quantum-mitigation

Le caratteristiche fondamentali misurate:

1. **Lunghezza delle chiavi generate**
2. **Lunghezza delle firme generate**
3. **Tempi medi di generazione delle chiavi**
4. **Tempi medi di firma di un messaggio**
5. **Tempi medi di validazione della firma**

test.c (minificato)

```
def main(ITER=100, OUT, HASHING, HASH):
    for msg_len in range(64, 16MB):
        for i in range(ITER):
            start_timer ← time.now()
            keypair ← pqc_keygen()
            end_timer ← time.now()
            gen_times.append(end_timer - start_timer)
            pub_key_len ← len(keypair.public_key)
            priv_key_len ← len(keypair.private_key)
        end for  # Fine del loop sulle iterazioni
        avg_gen_time ← sum(gen_times) / ITER
        message ← generate_random_message(msg_len)
        for i in range(ITER):
            start_timer ← time.now()
            signature ← pqc_sign(keypair.private_key, message)
            end_timer ← time.now()
            sign_times.append(end_timer - start_timer)
            start_timer ← time.now()
            valid ← pqc_verify(keypair.public_key, signature, message)
            end_timer ← time.now()
            verify_times.append(end_timer - start_timer)
        end for
        avg_sign_time ← sum(sign_times) / ITER
        avg_verify_time ← sum(verify_times) / ITER
        print(format_output_record(message_len, pub_key_len, priv_key_len,
        ↪ avg_gen_time, avg_sign_time, avg_verify_time))
    end for
```

STRUMENTI E SOFTWARE

VERSIONING

SVILUPPO (IDE)

VIRTUALIZZAZIONE

LINGUAGGI DI PROGRAMMAZIONE

ESECUZIONE
PERFORMANCE
TEST

GENERAZIONE
ANALYTICS
E GRAFICI

DISPOSITIVI HARDWARE

PC#1
CPU: INTEL U9 14°
RAM: 16GB DDR5

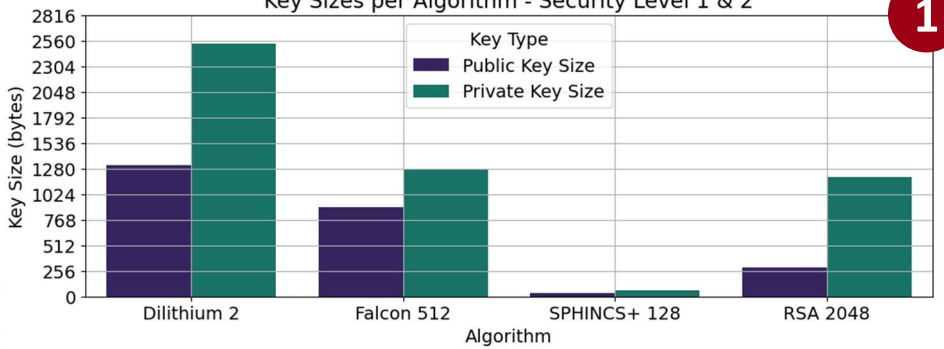
PC#2
CPU: INTEL i7 11°
RAM: 16GB DDR4

FONTI:

PROGETTO GITHUB: https://github.com/LovatoTomas/Analisi_di_algoritmi_di_firma_digitale_post-quantum

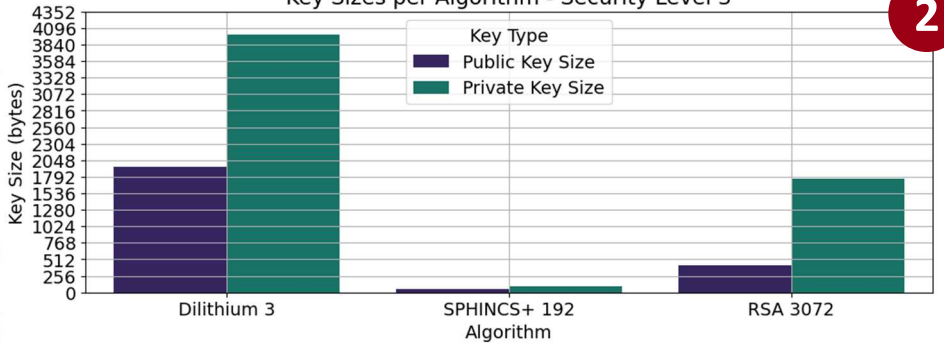
RISULTATI (1/3): KEY SIZES & SIGNATURE SIZES

1



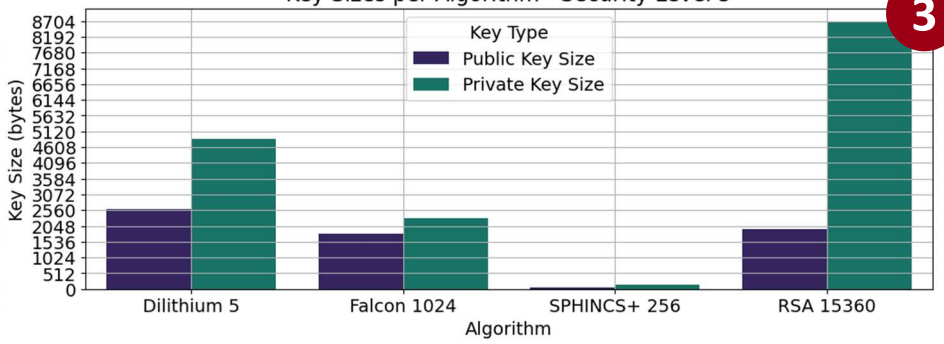
4

2

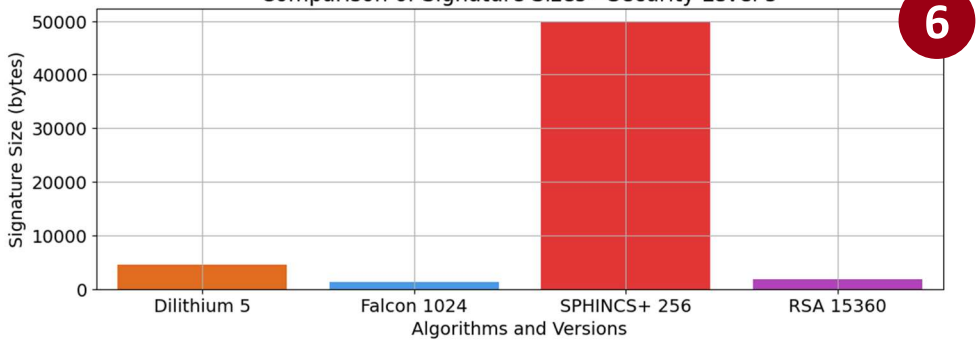


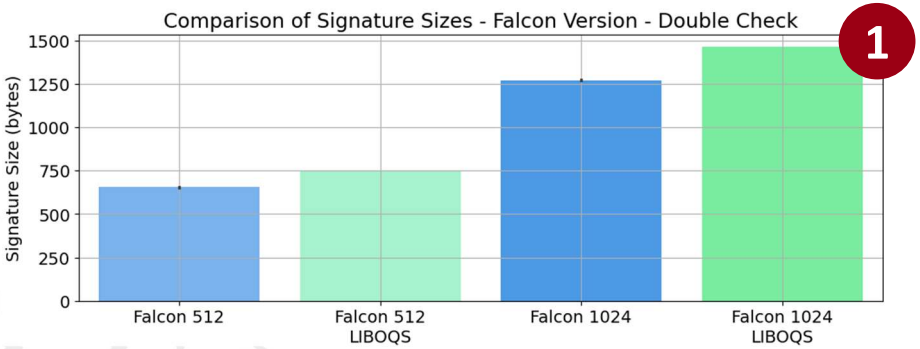
5

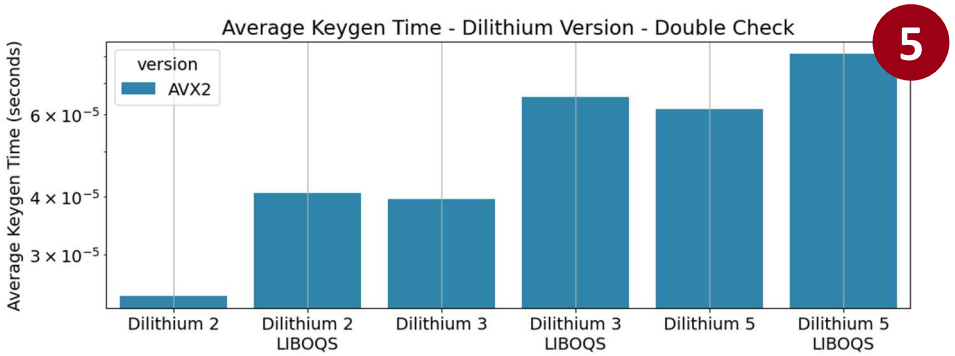
3

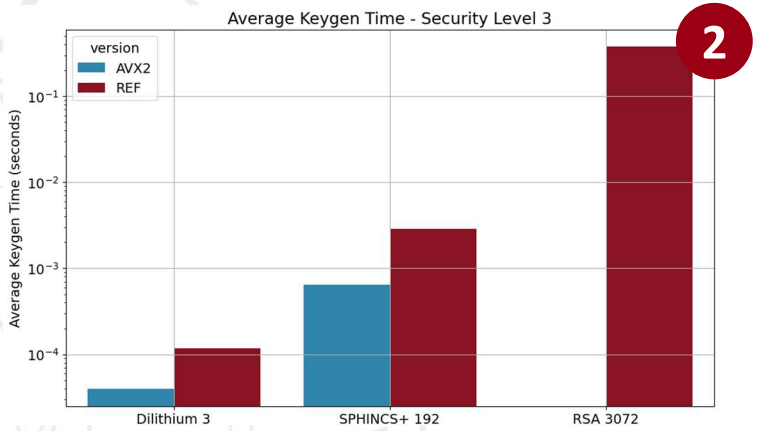


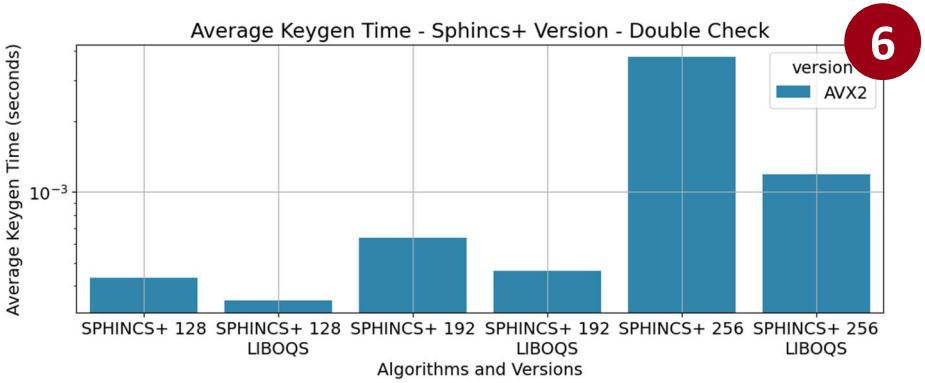
6

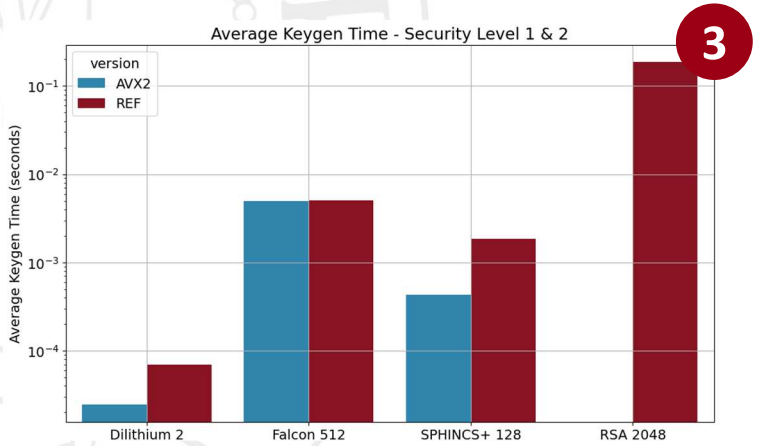


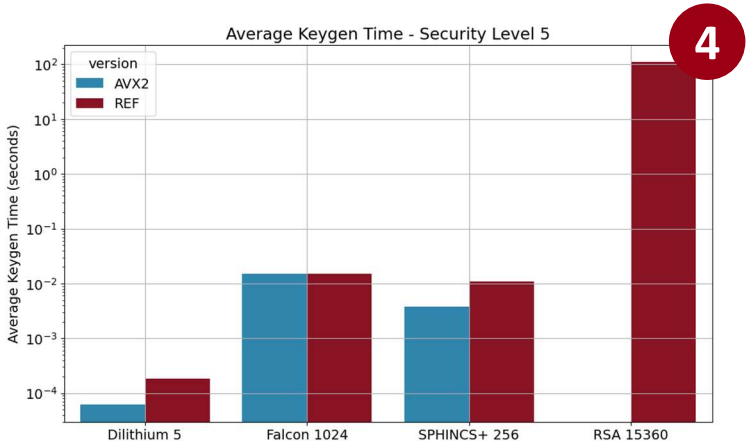

Comparison of Signature Sizes - Falcon Version - Double Check
1
Signature Size (bytes)
1500
1250
1000
750
500
250
0
Falcon 512
Falcon 512 LIBOQS
Falcon 1024
Falcon 1024 LIBOQS


Average Keygen Time - Dilithium Version - Double Check
5
version
AVX2
Average Keygen Time (seconds)
6×10^{-5}
4×10^{-5}
3×10^{-5}
Dilithium 2
Dilithium 2 LIBOQS
Dilithium 3
Dilithium 3 LIBOQS
Dilithium 5
Dilithium 5 LIBOQS

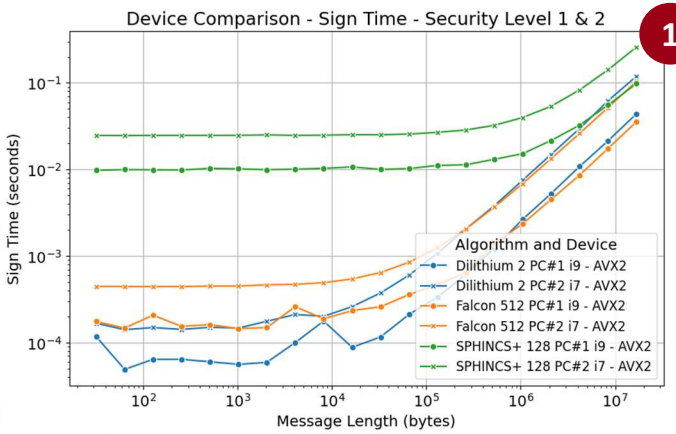

Average Keygen Time - Security Level 3
2
version
AVX2
REF
Average Keygen Time (seconds)
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Dilithium 3
SPHINCS+ 192
RSA 3072


Average Keygen Time - Sphincs+ Version - Double Check
6
version
AVX2
Average Keygen Time (seconds)
10^{-3}
SPHINCS+ 128
SPHINCS+ 128 LIBOQS
SPHINCS+ 192
SPHINCS+ 192 LIBOQS
SPHINCS+ 256
SPHINCS+ 256 LIBOQS
Algorithms and Versions

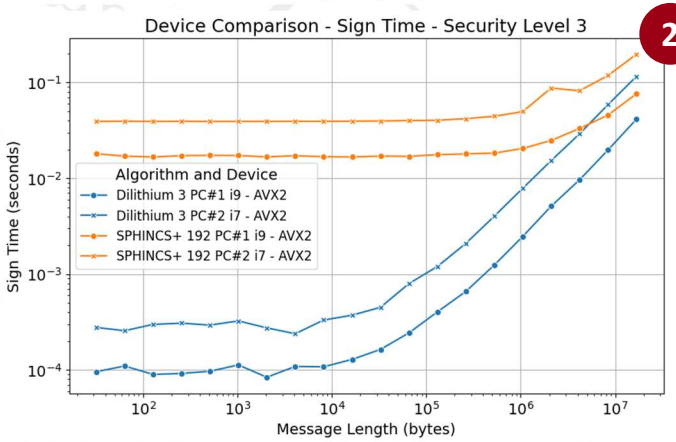

Average Keygen Time - Security Level 1 & 2
3
version
AVX2
REF
Average Keygen Time (seconds)
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Dilithium 2
Falcon 512
SPHINCS+ 128
RSA 2048


Average Keygen Time - Security Level 5
4
version
AVX2
REF
Average Keygen Time (seconds)
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Dilithium 5
Falcon 1024
SPHINCS+ 256
RSA 15360

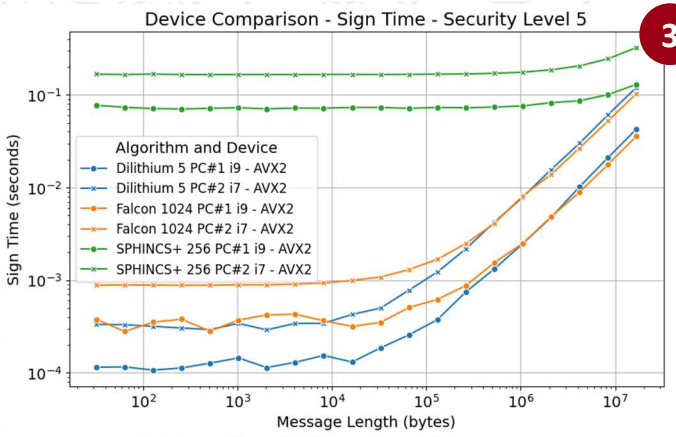
1


Device Comparison - Sign Time - Security Level 1 & 2
Sign Time (seconds)
Message Length (bytes)
Algorithm and Device
Dilithium 2 PC#1 i9 - AVX2
Dilithium 2 PC#2 i7 - AVX2
Falcon 512 PC#1 i9 - AVX2
Falcon 512 PC#2 i7 - AVX2
SPHINCS+ 128 PC#1 i9 - AVX2
SPHINCS+ 128 PC#2 i7 - AVX2

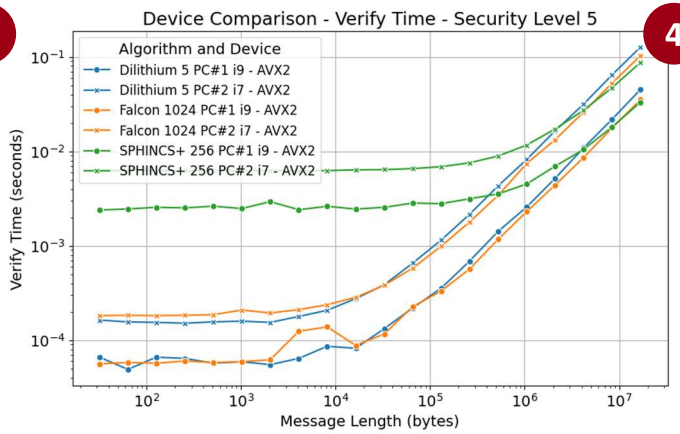
2


Device Comparison - Sign Time - Security Level 3
Sign Time (seconds)
Message Length (bytes)
Algorithm and Device
Dilithium 3 PC#1 i9 - AVX2
Dilithium 3 PC#2 i7 - AVX2
SPHINCS+ 192 PC#1 i9 - AVX2
SPHINCS+ 192 PC#2 i7 - AVX2

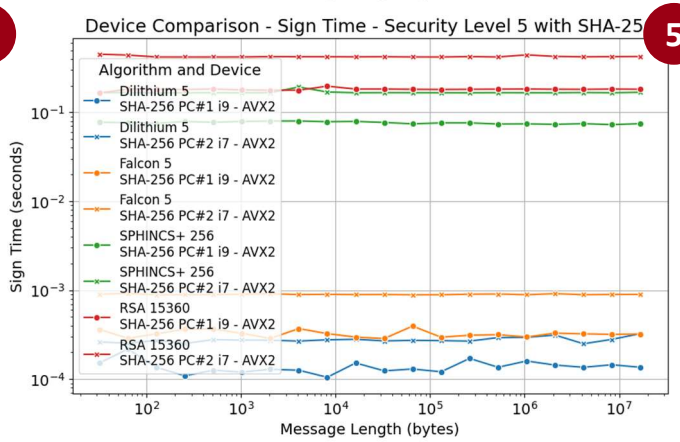
3


Device Comparison - Sign Time - Security Level 5
Sign Time (seconds)
Message Length (bytes)
Algorithm and Device
Dilithium 5 PC#1 i9 - AVX2
Dilithium 5 PC#2 i7 - AVX2
Falcon 1024 PC#1 i9 - AVX2
Falcon 1024 PC#2 i7 - AVX2
SPHINCS+ 256 PC#1 i9 - AVX2
SPHINCS+ 256 PC#2 i7 - AVX2

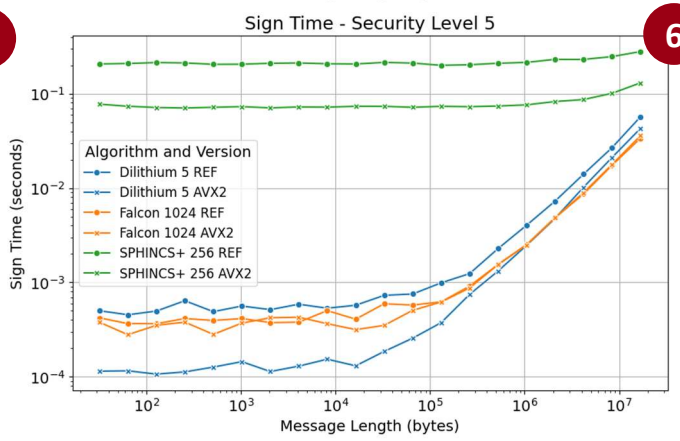
4


Device Comparison - Verify Time - Security Level 5
Verify Time (seconds)
Message Length (bytes)
Algorithm and Device
Dilithium 5 PC#1 i9 - AVX2
Dilithium 5 PC#2 i7 - AVX2
Falcon 1024 PC#1 i9 - AVX2
Falcon 1024 PC#2 i7 - AVX2
SPHINCS+ 256 PC#1 i9 - AVX2
SPHINCS+ 256 PC#2 i7 - AVX2

5

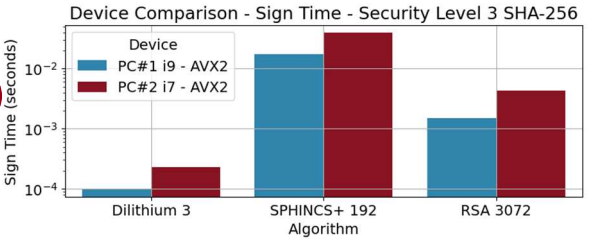

Device Comparison - Sign Time - Security Level 5 with SHA-25
Sign Time (seconds)
Message Length (bytes)
Algorithm and Device
Dilithium 5 SHA-256 PC#1 i9 - AVX2
Dilithium 5 SHA-256 PC#2 i7 - AVX2
Falcon 5 SHA-256 PC#1 i9 - AVX2
Falcon 5 SHA-256 PC#2 i7 - AVX2
SPHINCS+ 256 SHA-256 PC#1 i9 - AVX2
SPHINCS+ 256 SHA-256 PC#2 i7 - AVX2
RSA 15360 SHA-256 PC#1 i9 - AVX2
RSA 15360 SHA-256 PC#2 i7 - AVX2

6


Sign Time - Security Level 5
Sign Time (seconds)
Message Length (bytes)
Algorithm and Version
Dilithium 5 REF
Dilithium 5 AVX2
Falcon 1024 REF
Falcon 1024 AVX2
SPHINCS+ 256 REF
SPHINCS+ 256 AVX2

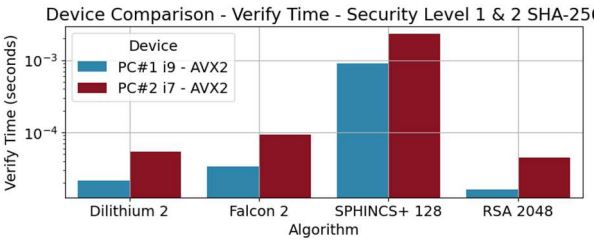
7


Device Comparison - Sign Time - Security Level 1 & 2 SHA-256
Sign Time (seconds)
Device
PC#1 i9 - AVX2
PC#2 i7 - AVX2
Dilithium 2
Falcon 2
SPHINCS+ 128
RSA 2048
Algorithm


Device Comparison - Sign Time - Security Level 3 SHA-256
Sign Time (seconds)
Device
PC#1 i9 - AVX2
PC#2 i7 - AVX2
Dilithium 3
SPHINCS+ 192
RSA 3072
Algorithm


Device Comparison - Sign Time - Security Level 5 SHA-256
Sign Time (seconds)
Device
PC#1 i9 - AVX2
PC#2 i7 - AVX2
Dilithium 5
Falcon 5
SPHINCS+ 256
RSA 15360
Algorithm

8


Device Comparison - Verify Time - Security Level 1 & 2 SHA-256
Verify Time (seconds)
Device
PC#1 i9 - AVX2
PC#2 i7 - AVX2
Dilithium 2
Falcon 2
SPHINCS+ 128
RSA 2048
Algorithm

I FINALISTI DEL ROUND 3

FALCON
Fast-Fourier Lattice-based Compact Signatures over NTRU

PRO
1. Efficienza in termini di dimensione
2. Velocità nelle operazioni di firma
3. Robustezza teorica

CONTRO
1. Difficoltà di implementazione
2. Prestazioni meno stabili, soprattutto con hardware limitati

DILITHIUM

PRO
1. Efficienza computazionale
2. Semplicità di implementazione
3. Livelli di sicurezza elevati
4. Versatilità nelle applicazioni

CONTRO
1. Dimensione delle chiavi
2. Limitazioni hardware

SPHINCS+

PRO
1. Sicurezza elevata
2. Hash-Based Stateless
3. Robustezza crittografica

CONTRO
1. Performance
2. Dimensione delle firme
3. Maggiore complessità operativa

13 Agosto 2024 → Rilasciati i primi standard PQC da parte del NIST: CRYSTALS Kyber (KEM); CRYSTALS Dilithium (**DSS**); SPHINCS+ (DSS).

FONTI:
ENISA: https://www.enisa.europa.eu/publications/post-quantum-cryptography-current-state-and-quantum-mitigation
NIST NEW STANDARDS ARTICLE: https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards