SECURITY ANALYSIS OF CRYSTALS-KYBER

Deepa Thangavelu

CARIC 2023

21 April 2023

Agenda

» PKC

» PKC Under threat

» NIST PQC standardization (Round 4 & Alternatives)

» CRYSTALS-Kyber Decapsulation Mechanism

» Side-Channel Attacks on CRYSTALS-Kyber

» Chosen Ciphertext KEMs

» Full-Key Recovery

Public Key Cryptography (PKC)

PKC Primitives:

» Public-Key Encryption (PKE) - Confidentiality

» Key Encapsulation Mechanism (KEM) - Secret Key-Sharing

» Digital Signature Schemes (DSS) - Authenticity

PKC Primitives we use today:

» **Rivest-Shamir-Adleman (RSA)**

Security: **Prime Factorization** problem

» **Elliptic Curve Cryptography (ECC)**

Security: **Discrete Logarithm** problem

PKC Under Threat

Peter Shor in 1994 developed the **first quantum algorithm** that solves the factoring problem in **polynomial time**

Cryptosystem	Category	Key Size	Quantum Algorithm	# Logical Qubits Required	# Physical Qubits Required	Time Required to Break System
AES-GCM	Symmetric-Key Encryption	128	Grover's Algorithm	2,953	4.61×10^6	2.61×10^{12} years
		192		4,449	1.68×10^7	1.97×10^{22} years
		256		6,681	3.36×10^7	2.29×10^{32} years
RSA	Asymmetric-Key Encryption	1024	Shor's Algorithm	2,050	8.05×10^6	3.58 hours
		2048		4,098	8.56×10^6	28.63 hours
		4096		8,194	1.12×10^7	229 hours
ECC Discrete-log Problem	Asymmetric-Key Encryption	256	Shor's Algorithm	2,330	8.56×10^6	10.5 hours
		384		3,484	9.05×10^6	37.67 hours
		521		4,719	1.13×10^6	55 hours

Post Quantum Cryptography (PQC)

First NIST PQC Standards (US):

PKE / KEMs	Digital Signatures
Kyber	Dilithium
	FALCON
	SPHINCS+

BSI Recommendations:

PKE / KEMs	Digital Signatures
FrodoKEM	XMSS
Classic Mcelice	LMS

Lattice-based

Hash-based

Code-based

Classification of PQC finalists and alternative candidates

Lattice-based Cryptography

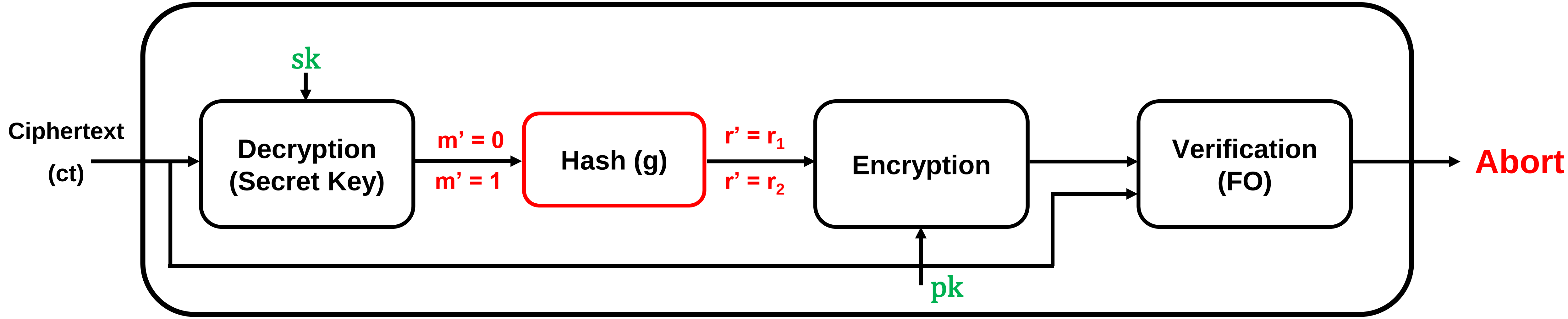
- **Public Key Encryption (PKE)/ Key Encapsulation Mechanisms (KEM)**
  - **LWE/LWR-based** (Kyber, SABER, Frodo)
  - **NTRU-based** (NTRU, NTRUPrime)
- **Digital Signature Schemes (DSS)**
  - **LWE, Fiat-Shamir with Aborts** (Dilithium)
  - **NTRU, Hash and Sign** (FALCON)

Features of CRYSTALS-Kyber

Key Encapsulation Mechanism (KEM)

- **Modules Learning with Errors (MLWE) Problem**
- **Prime modules q=3329**
- **Kyber.CCAKEM.KeyGen**
- **Kyber.CCAKEM.Enc**
  - Encapsulate a (secret) message m
  - Session key derived from m
- **Kyber.CCAKEM.Dec**
  - Decapsulate ciphertext using long term secret key
  - Fujisaki-Pkamoto transform for IND-CCA security

CRYSTALS-Kyber Decapsulation Mechanism



$$m' = \text{Decrypt}(sk, ct)$$
$$r' = \mathcal{G}(m', pk)$$
$$ct' = \text{Encrypt}(pk, m', r')$$
$$\text{If}(ct = ct')$$
$$K = \mathcal{H}(r' \| ct')$$
$$\text{Else}$$
$$K = \mathcal{H}(z \| ct')$$

Physical Attacks on CRYSTALS-Kyber

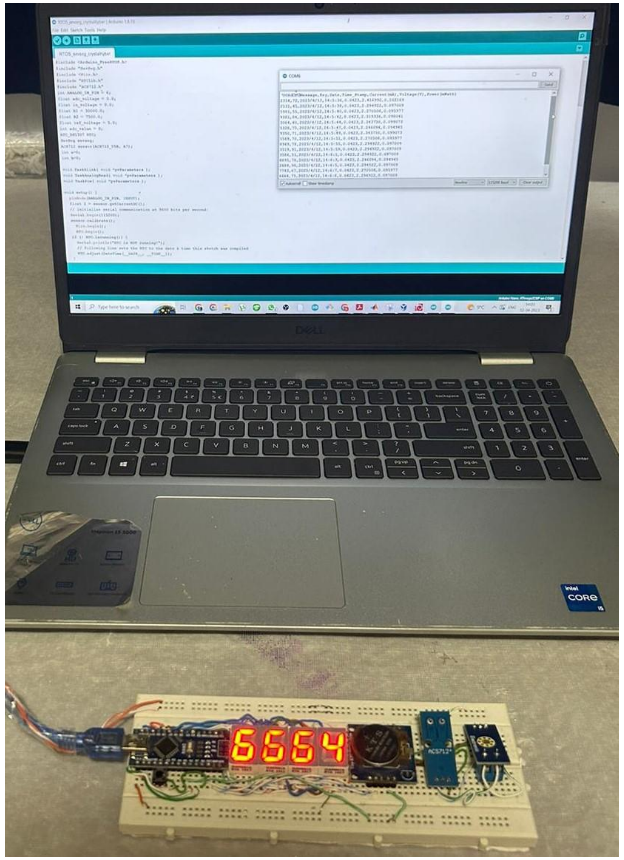
Side Channels Attack (SCA)

- ↘ Reveals sensitive data.
- ↘ Observes device’s physical signature during its operation for cryptanalysis.

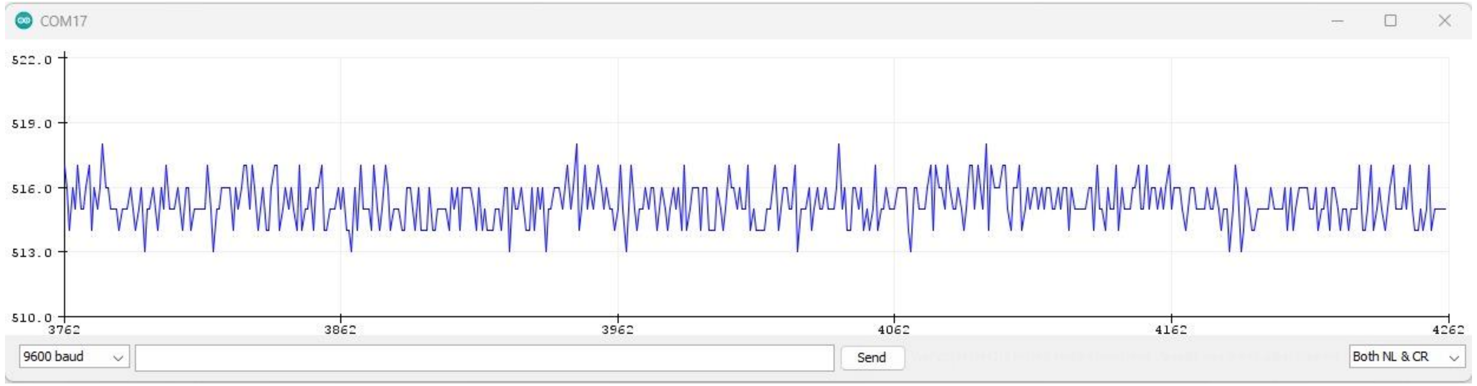
Side-Channel Attack Vectors

- ↘ Timing
- ↘ Power Consumption
- ↘ Electromagnetic Emanation (EM)

Experimental Set-up

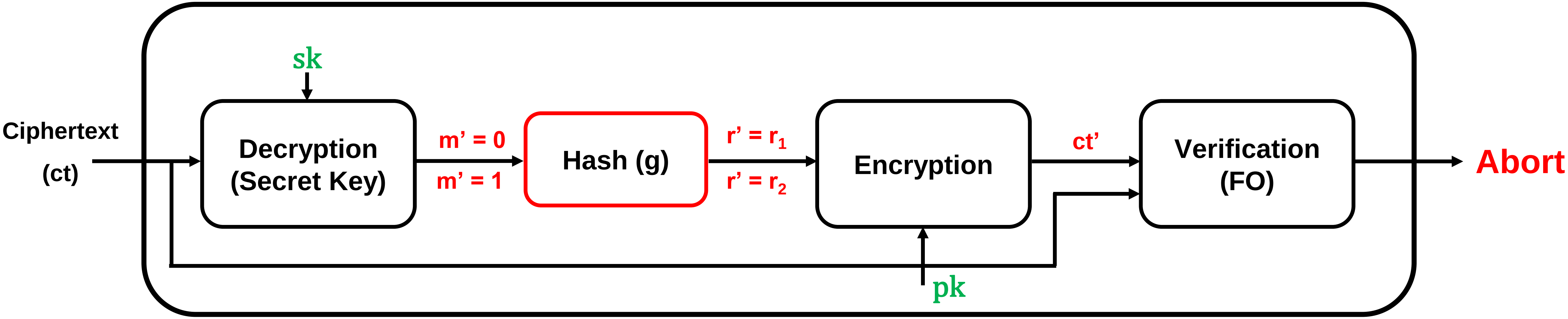


- Perform all the experiments on the most optimized implementations of the targeted schemes present in the pqm4 library, power consumption trace analysis on the AT328 microcontroller.
- Clock Speed of 16 MHz;
- The ACS712, a series of current sensor integrated circuits (Ics)
- Voltage sensor measures 0-2.5V
- equipment set-up is capable of capturing power traces of the target device

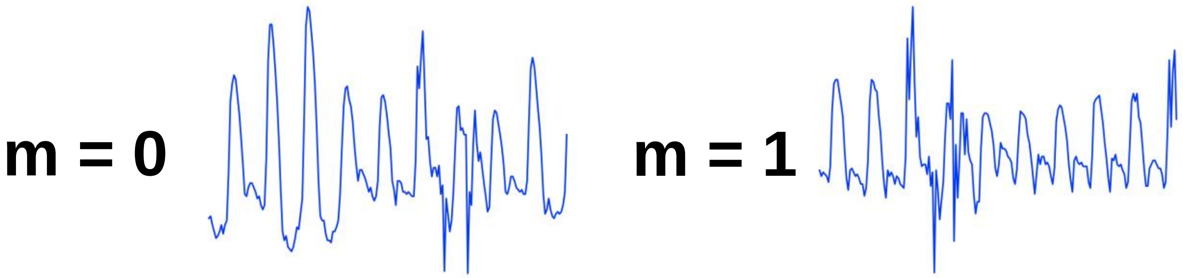


Leakage Traces

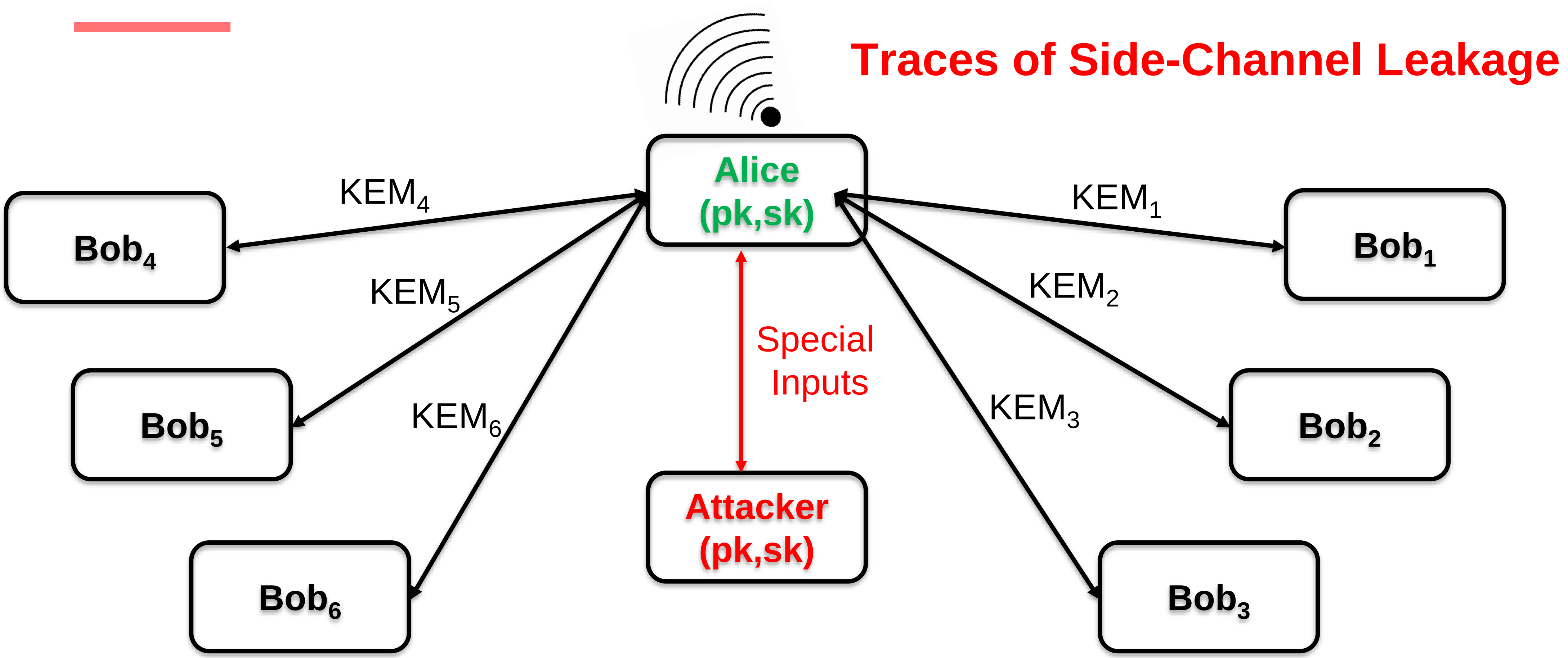
Realizing a Side-Channel based PC Oracle



Message = Function (Single Secret Coefficient)



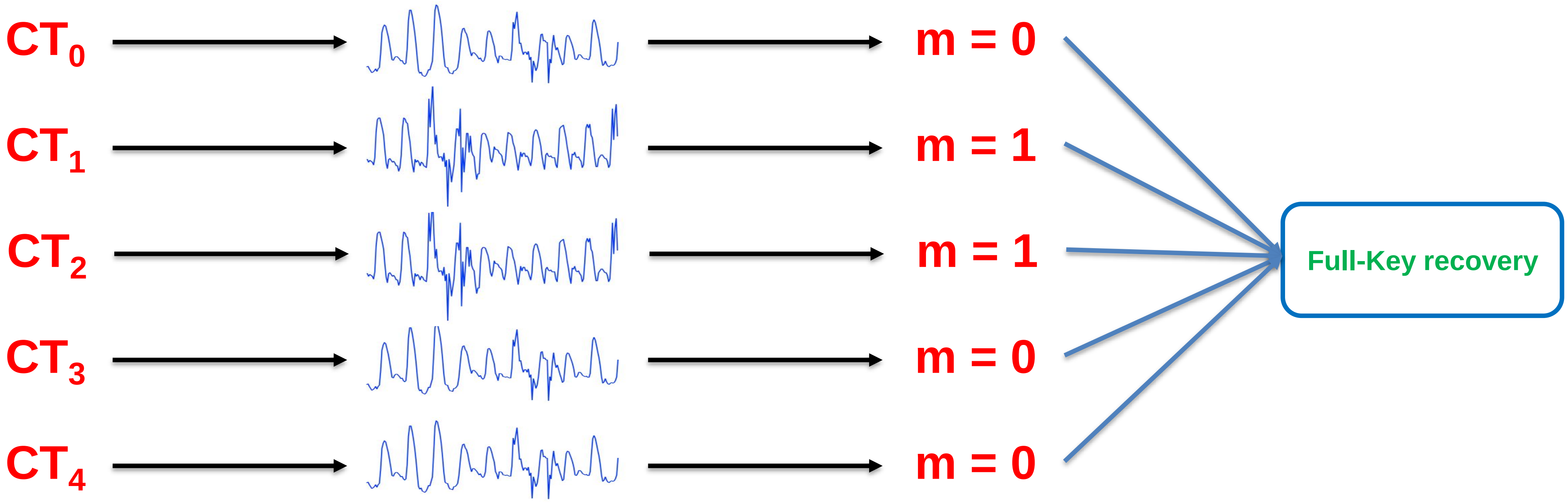
Chosen Cipher-text KEMs



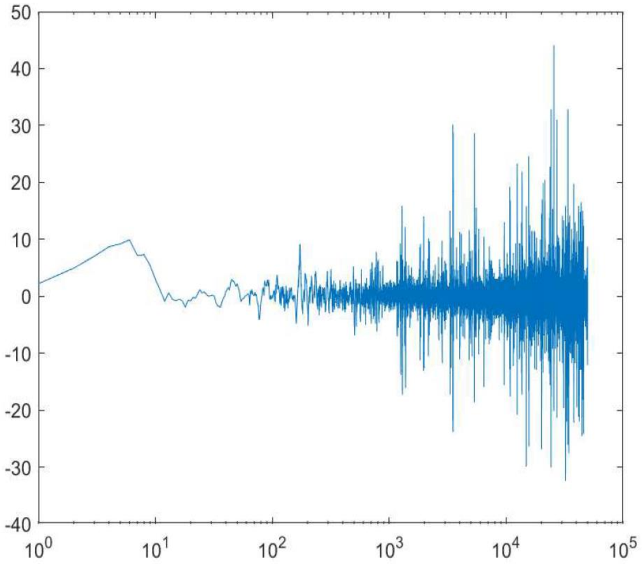
Main Target: Decapsulation Procedure

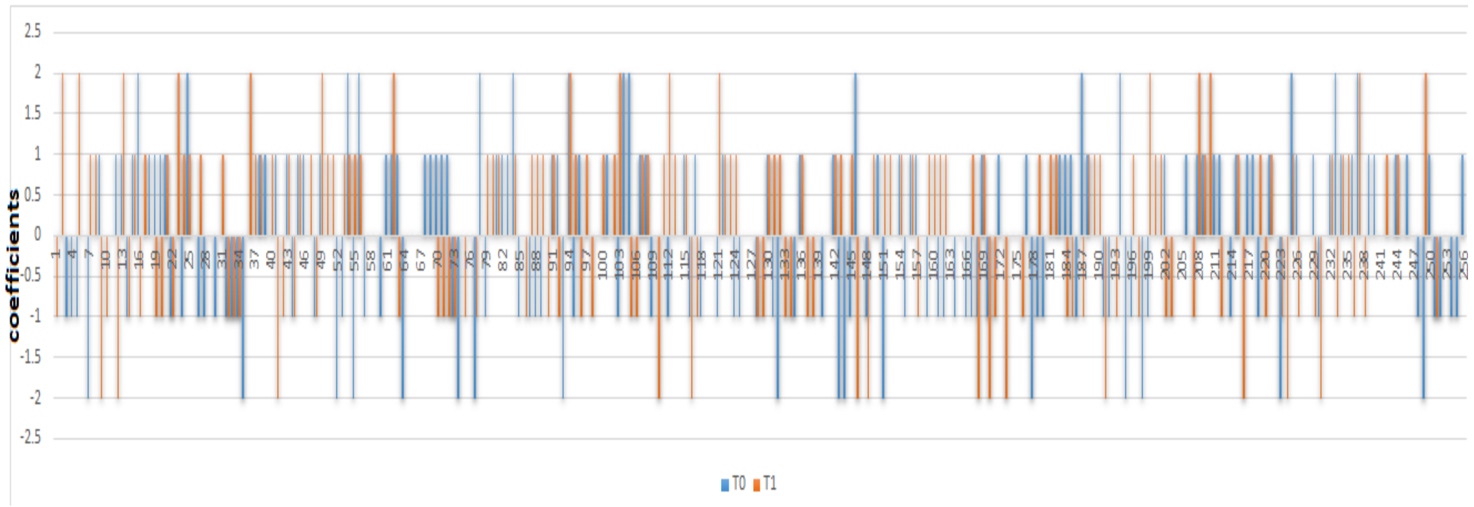
concordia.ab.ca

Key Recovery Analysis



Output from Matlab Calculations

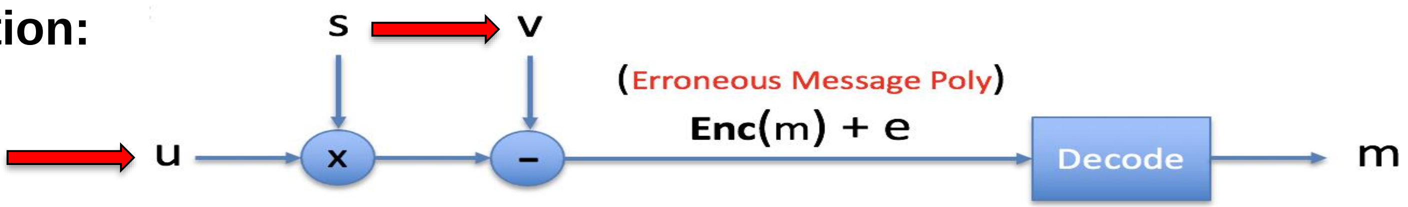




TVLA Leakage

Constructing Malicious Ciphertexts

Decryption:



Chosen u	k	0	0	0	0	0	0
u.s	$k.s_0$	$k.s_1$	$k.s_2$	$k.s_3$	$k.s_4$	$k.s_5$	$k.s_6$
Chosen v	p	0	0	0	0	0	0
m’ = u.s - v	$k.s_0$-p	$k.s_1$	$k.s_2$	$k.s_3$	$k.s_4$	$k.s_5$	$k.s_6$
m = Decode(**m**)	$f(s_0)$	0	0	0	0	0	0
	m_0	m_1	m_2	m_3	m_4	m_5	m_6

Constructing Malicious Ciphertexts

- Polynomial multiplication in polynomial rings have special rotational properties.

$$R_q = \mathbb{Z}_q[x] \bmod (x^n - 1) \quad R_q = \mathbb{Z}_q[x] \bmod (x^n + 1)$$

- Multiplication of a polynomial with x^i **"rotates"** the polynomial by "i" positions (cyclic or anti-cyclic)

Chosen u	k	0	0	0	0	0	0
Chosen v	p	0	0	0	0	0	0
m'=Decode(**m'**)	$f(s_0)$	0	0	0	0	0	0

Recover s_0 using knowledge of O/X

Constructing Malicious Ciphertexts

- Polynomial multiplication in polynomial rings have special rotational properties.

$$R_q = \mathbb{Z}_q[x] \bmod (x^n - 1) \quad R_q = \mathbb{Z}_q[x] \bmod (x^n + 1)$$

- Multitplication of a polynomial with x^i **"rotates"** the polynomial by "i" positions (cyclic or anti-cyclic)

Chosen u	0	k	0	0	0	0	0
Chosen v	p	0	0	0	0	0	0
m'=Decode(**m'**)	$f(s_{n-1})$	0	0	0	0	0	0

Recover s_{n-1} using knowledge of O/X

Constructing Malicious Ciphertexts

- ❑ Polynomial multiplication in polynomial rings have special rotational properties.

$$R_q = \mathbb{Z}_q[x] \bmod (x^n - 1) \quad R_q = \mathbb{Z}_q[x] \bmod (x^n + 1)$$

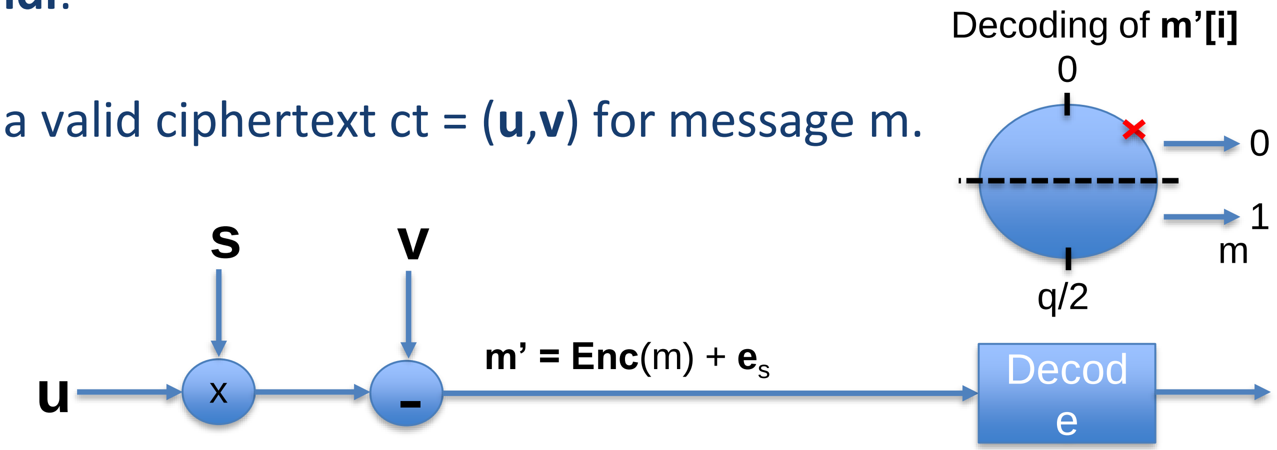
- ❑ Multitplication of a polynomial with x^i **"rotates"** the polynomial by "i" positions (cyclic or anti-cyclic)
- ❑ No Rotation property in schemes based on Standard LWE/LWR (FrodoKEM) - But, attack still works…
- ❑ Location of non-zero bit of message changes (depending upon secret coefficient to recover)

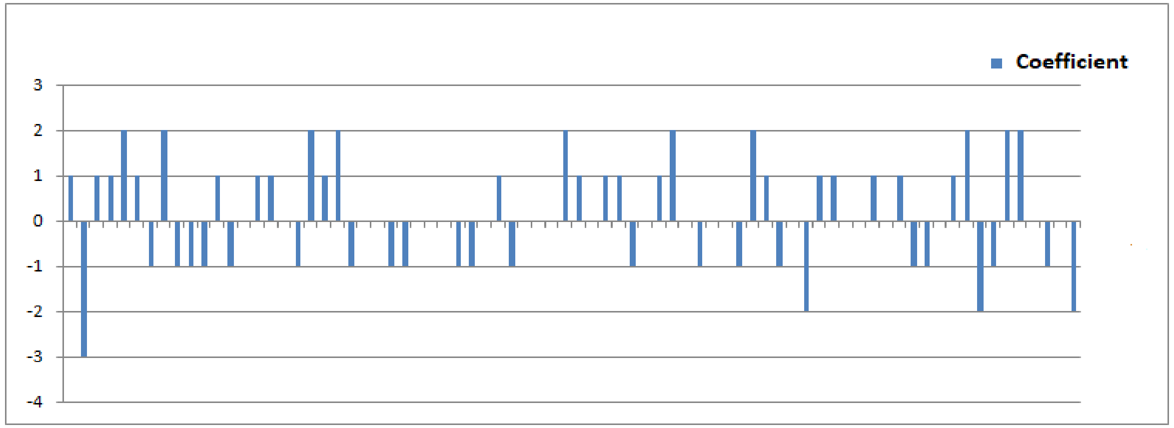
Chosen u	k	0	0	0	0	0	0
Chosen v	0	0	p	0	0	0	0
m'=Decode(**m'**)	0	0	$f(s_2)$	0	0	0	0

Recover s_{n-1} using knowledge of O/X

Allocation of values

- **Modus Operandi**:
  - Construct a valid ciphertext ct = (**u**,**v**) for message m.





Full-Key Recovery


Thank you!

Q and A