Fault Attack Resistance of Post Quantum Algorithms

Felipe Valencia, **Francesco Regazzoni**

Outline

- Post-quantum cryptography - Lattice-based cryptography
- Physical attacks - Faults attacks
- RLWE encryption
- Attacks on RLWE
- Results and conclusions

Quantum computing and cryptography

- Current cryptography is based on the hardness of the integer factorization and discrete logarithm problems
- Shor's algorithm is a quantum algorithm that can solve these problems in polynomial time

Quantum computing and cryptography

- Current cryptography is based on the hardness of the integer factorization and discrete logarithm problems
- Shor's algorithm is a quantum algorithm that can solve these problems in polynomial time

Quantum computer will break current cryptography

Quantum computing and cryptography

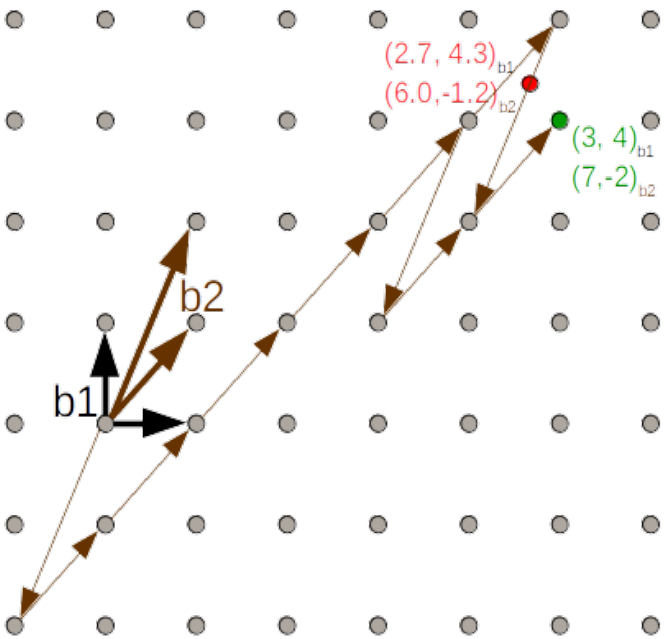
- Current cryptography is based on the hardness of the integer factorization and discrete logarithm problems
- Shor's algorithm is a quantum algorithm that can solve these problems in polynomial time

Quantum computer will break current cryptography

Cryptography based on lattice problems is presumable resistant against quantum attacks

Lattice-based cryptography

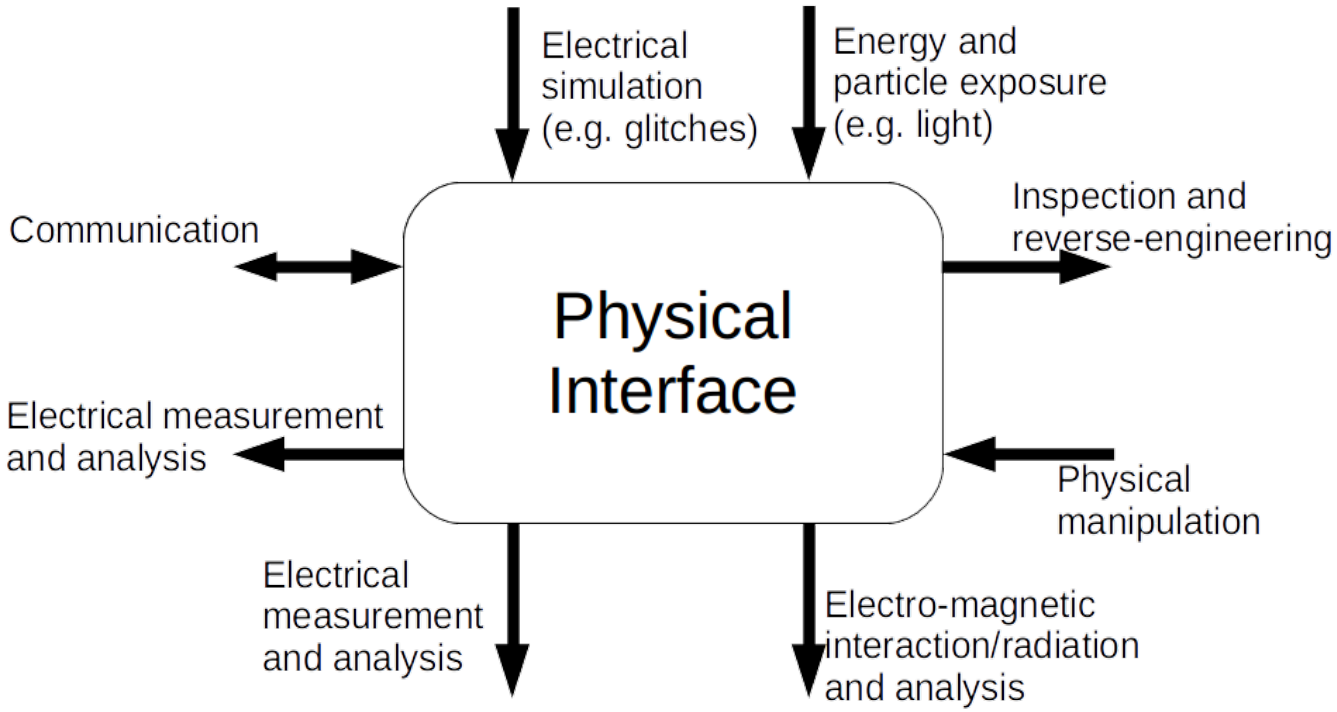
- A lattice L is a discrete set of points in the space $\mathbb{R}^n$ with periodic structure.
- Foundations problems are Shortest Vector Problem and Closes Vector Problem



Outline

- Physical attacks - Faults attacks
- RLWE encryption
- Attacks on RLWE
- Results and conclusions

Physical attacks



- Timing analysis
- Power analysis
- **Fault attacks**

Fault attacks

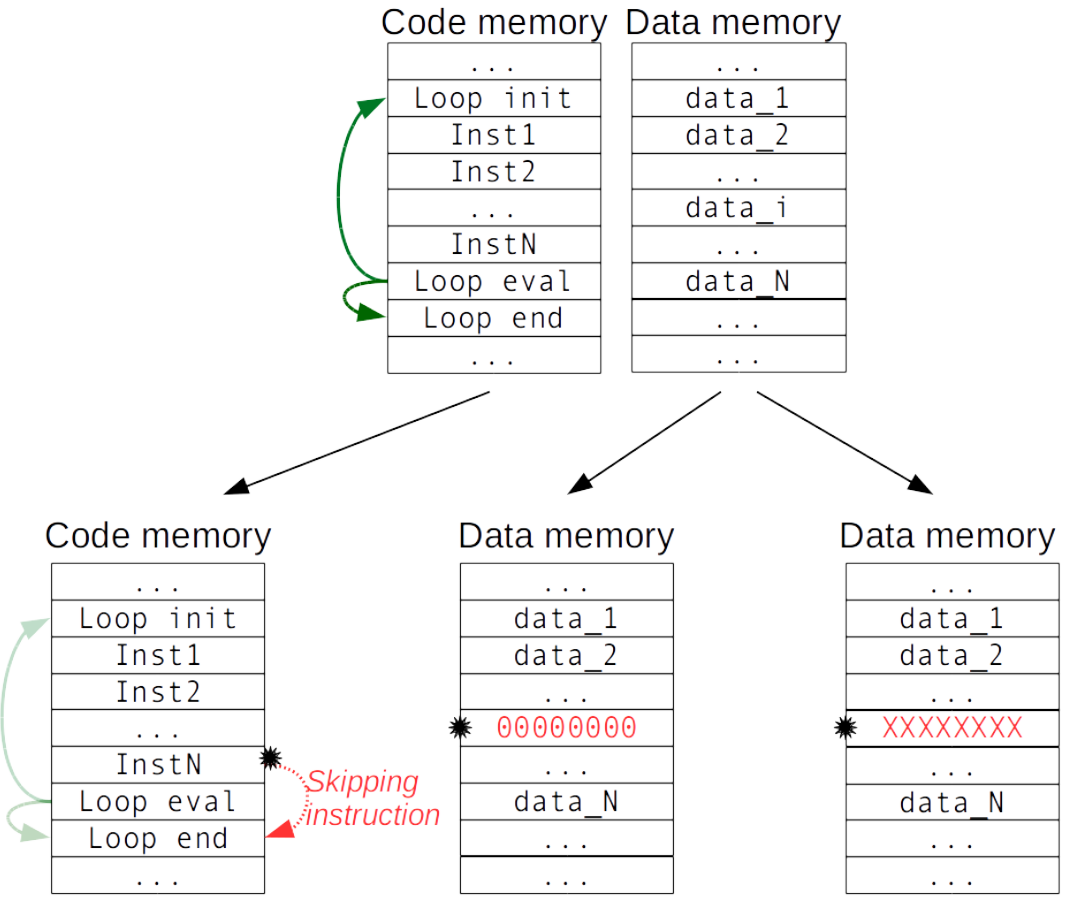
1. Malicious injection of a fault in a device running a cryptographic algorithm
2. Exploitation of the induced faulty behavior to gather information about the secret values

Categories and properties of fault attacks

- Non-invasive
- Semi-invasive
- Invasive

- Granularity: bit, byte, word, etc.
- Modification: stuck at, flip, random
- Duration: Transient, permanent, destructive

Attack models considered in this work



Outline

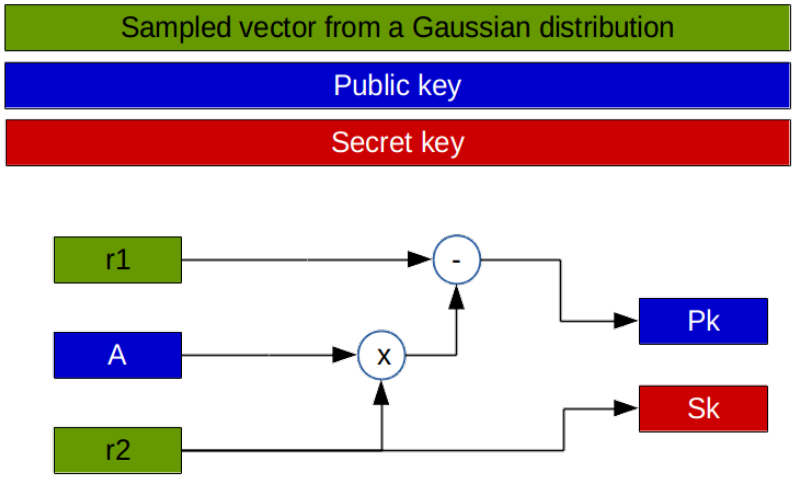
- RLWE encryption
- Attacks on RLWE
- Results and conclusions

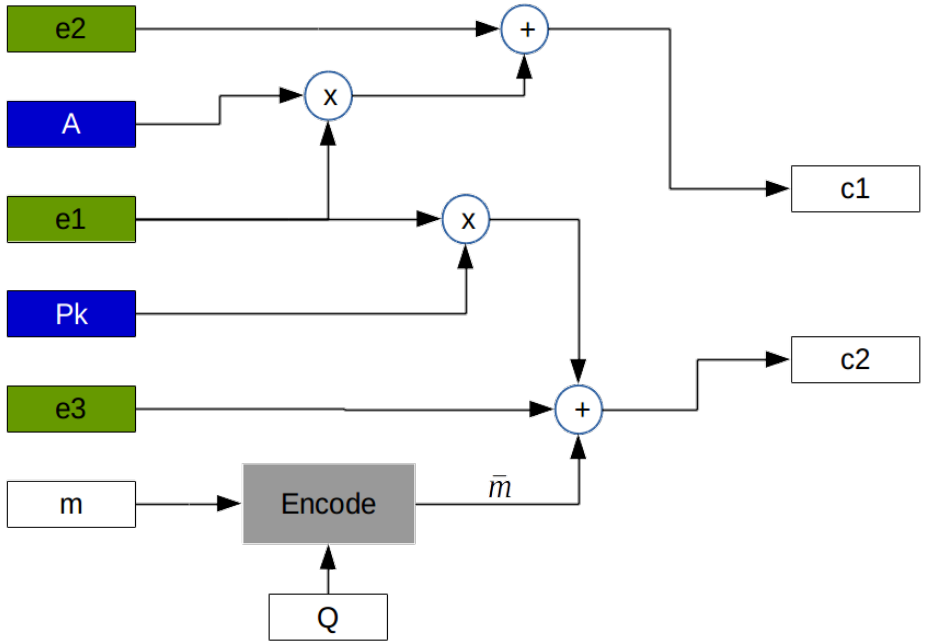
Definition

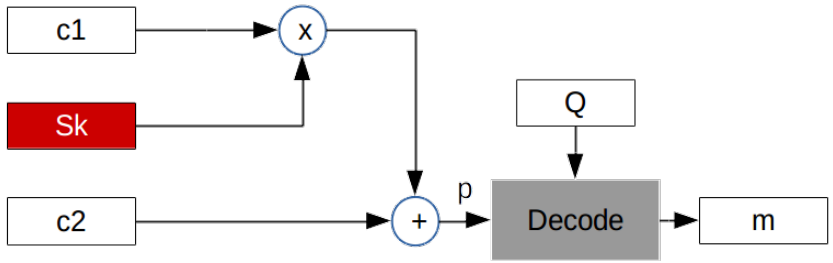
- RLWE (Ring Learning With Errors) encryption is a cryptosystem based on the Learning With Errors problem on Ring. It is parameterized by the length N, an integer Q and a distribution with variance σ

- The Number Theoretic Transform is a Fourier transform performed in a ring instead of $\mathbb{C}$
- It speeds up the RLWE encryption because it reduces the complexity of the polynomial multiplication from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$

Key generation, Encryption, Decryption





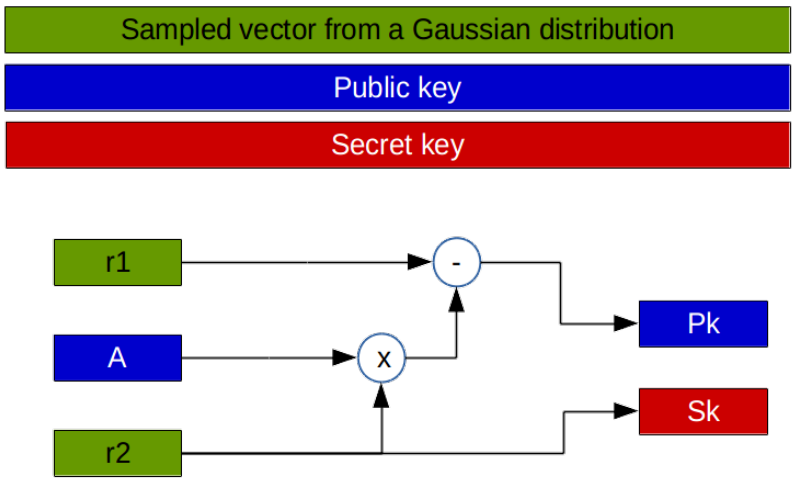


Outline

- Attacks on RLWE
- Results and conclusions

Key generation

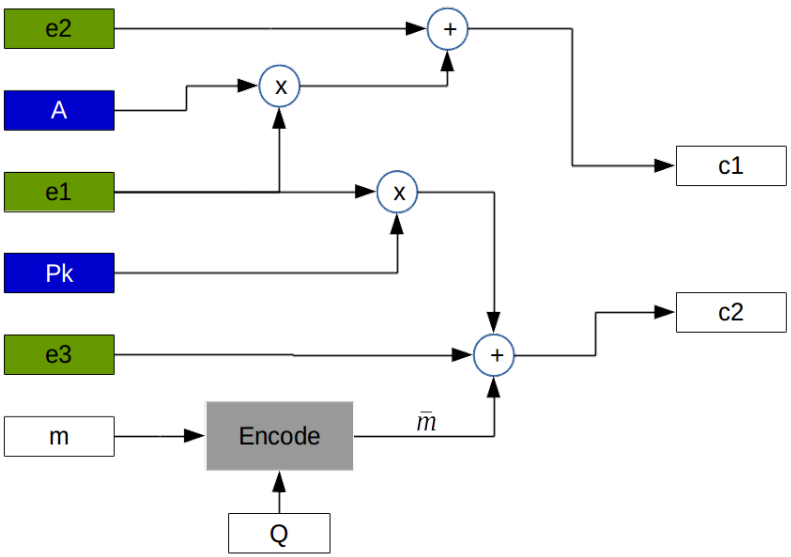
Create a weak key for which the system still works



- $pk = ar_2$, $sk = r_2$, $r_1 = 0$ – It is easy to compute r_2 from pk.
- $pk = r_1$, $sk = 0^n$ – As in this case the secret key consists of zeros, the scheme can be easily broken.
- $pk = r_1$, $sk = r_2$; $pk = random$, $sk = r_2$; $pk = p$, $sk = random$; $pk = 0$, $sk = r_2$ – These faults produce an incorrect result, thus would not be exploitable by an attacker.

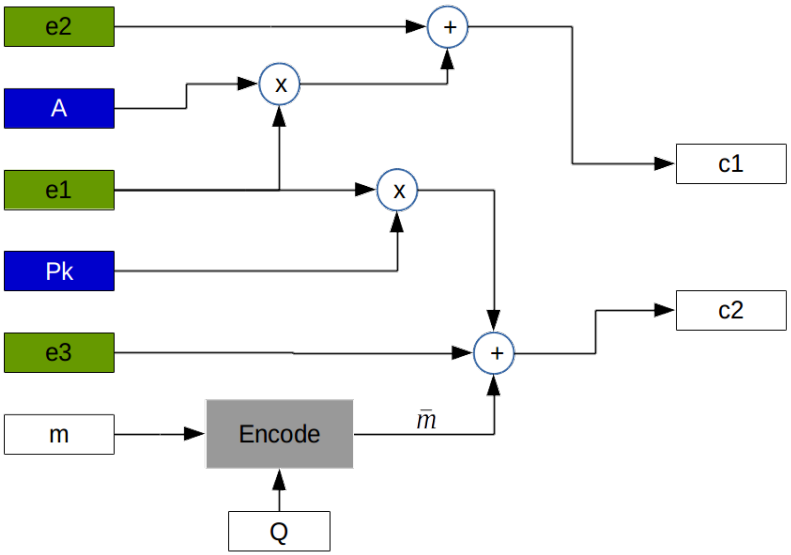
Encryption

Recover the encrypted message



- $c_1 = ae_1$, $c_2 = pke_1 + e_3 + enc(m)$, $e_2 = 0$ – The message can be recovered by computing e_1 from c_1. With e_1, e_3 can be eliminated with a threshold function.
- $c_1 = e_2, c_2 = e_3 + enc(m), e_1 = 0$ – The message can be recovered from c_2 eliminating the e_3 with a threshold function.
- $c_1 = ae_1 + e_2, c_2 = e_3 + enc(m) - e_3$ can be eliminated with a threshold function.

Recover the encrypted message



- $c_1 = e_2, c_2 = pke_1 + e_3 + enc(m)$ – This situation destroys the encryption scheme.
- $c_1 = random, c_2 = pke_1 + e_3 + enc(m)$ – This case is a generalization of the previous one and therefore leads to the same conclusion.
- $c_1 = ae_1 + e_2, c_2 = pke_1 + e_3$ or $c_1 = ae_1 + e_2, c_2 = random$ – This destroys information about the message.

Decryption

Recover the secret key

- Zeroing the key.
- Zeroing the ciphertext.
- Zeroing during the NTT.
- Randomization of the key.

Zeroing the key

```
for (int s=0;s<n;s++){ ** Skipped
  for (int c=1;c<n;c++){
    idx=(s+c) % n;
    value=sk[s]*c1[c];
    if(s+c>n){
      res[idx]=(res[idx]-val) % q;
    }else{
      res[idx]=(res[idx]+val) % q;
    }
  }
}
```

For $j = (c + s) \bmod n$ we that $(sk * c_1)(j)$ is equal to

$$\sum_{s=0}^{n-1} \sum_{c=0}^{n-1} (sk(s) \cdot c_1(c)) \bmod q$$

$sk'_1 = [A\ 0\ 0\ ...\ 0]$

$sk'_2 = [A\ B\ 0\ ...\ 0]$

...

sk recovered completely

Zeroing the cipher

```
for (int c=1;c<n;c++){** Skipped
  for (int s=0;s<n;s++){
    idx=(s+c) % n;
    value=sk[s]*c1[c];
    if(s+c>n){
      res[idx]=(res[idx]-val) % q;
    }else{
      res[idx]=(res[idx]+val) % q;
    }
  }
}
```

For $j = (c + s) \bmod n$ we that $(sk * c_1)(j)$ is equal to

$$\sum_{s=0}^{n-1} \sum_{c=0}^{n-1} (sk(s) \cdot c_1(c)) \bmod q$$

$c_1 = [A\ 0\ 0\ ...\ 0]$

$c_1 = [A\ B\ 0\ ...\ 0]$

...

This is equivalent to a cipher-chosen attack

Zeroing in NTT

- In the NTT domain, polynomial multiplication corresponds to component-wise product between vectors.
- In this case, zeroing a section of the key is equivalent to zeroing the same section of the ciphertext c_1
- $mes'(i) = Decode(Offset + Const \cdot Sk(m) + c_2(i)$
- For every component, there is a linear equation with known offset and known slope.
- With 1 equation is it possible to constrained the range of values for 1 component. With n equation is possible to recover the complete key

Outline

- Results and conclusions

Summary of the attacks

Phase	**Fault**	**Result**
Key Generation	$r_1 = 0$	Weak key generated
Key Generation	$r_2 = 0$	Weak key generated
Encryption	$e_1 = 0$	Message recovery
Encryption	$e_2 = 0$	Message recovery
Encryption	$pke_1 = 0$	Message recovery
Decryption	Zeroing secret key	Secret key recovery
Decryption	Zeroing the cipher text	Secret key recovery
Decryption	Zeroing during the NTT	Secret key recovery
Decryption	Randomization of the key	Secret key recovery

Countermeasures

- Measuring statistics
- Redundancy loop
- Protection against CCA2 attacks

Conclusions

- We systematically analyzed the vulnerability of R-LWE to fault attacks.
- Attacks on the decryption are more attractive for attackers because it allows to recovery the secret key.
- R-LWE can be attacked using fault attacks.
- Some fault attacks are comparable to chosen-ciphertext attacks. Thus, the same countermeasure can applied for both.

Questions

Thank you for your attention