

# LATTICE-BASED SIGNATURE SCHEMES AND THEIR SENSITIVITY TO FAULT ATTACKS



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



CROSSING

FDTC 2016  
08/16/2016

Nina Bindel

Johannes Buchmann

Juliane Krämer

TU Darmstadt

# SHOR'S ALGORITHM 1994

## Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer\*

Peter W. Shor<sup>†</sup>

### Abstract

A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

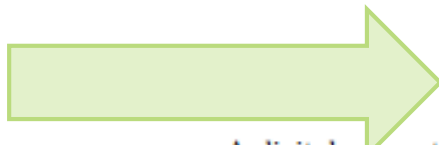
**Keywords:** algorithmic number theory, prime factorization, discrete logarithms, Church's thesis, quantum computers, foundations of quantum mechanics, spin systems, Fourier transforms

**AMS subject classifications:** 81P10, 11Y05, 68Q10, 03D10

# SHOR'S ALGORITHM 1994

Polynomial-Time Algorithms for Prime Factorization  
and Discrete Logarithms on a Quantum Computer\*

Peter W. Shor<sup>†</sup>



**RSA and ECDSA  
insecure**

A digital computer is generally considered a physical device; that is, it is believed that the time required for an increase in computation is proportional to the number of steps. It is true when quantum mechanics is taken into account. For factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

**Keywords:** algorithmic number theory, prime factorization, discrete logarithms, Church's thesis, quantum computers, foundations of quantum mechanics, spin systems, Fourier transforms

**AMS subject classifications:** 81P10, 11Y05, 68Q10, 03D10

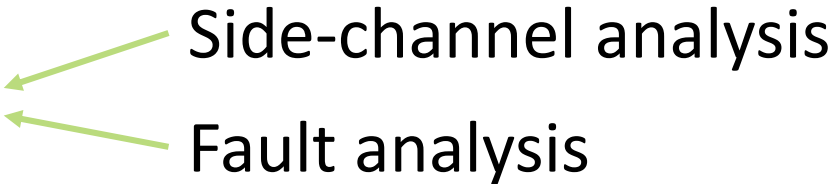
# QUANTUM COMPUTER REALISTIC?

- John Martinis (UCSB & Google Quantum Labs):  
until 2019 universal quantum computer
- Prediction by EU-commission:  
until 2035 universal quantum computer

# BETTER SAFE THAN SORRY

- NSA, 2015 : announcement about transition from classical to quantum-resistant crypto
- NIST, 2016: announcement to start standardization competition

# POST-QUANTUM CANDIDATES

- Quantum key distribution
- Multivariate Crypto
- Code-based Crypto
- Hash-based Crypto
- **Lattice-based Crypto** 

# CONTRIBUTION

- Analysis of LBSS: BLISS, GLP, ring-TESLA
- 1st order attacks
- Randomization, skipping, zeroing
- all-in-all 15 different attacks
- to 9 at least one scheme vulnerable
- Propose countermeasures

# VULNERABILITIES OF LBSS

Fault Attack	Changed Value or Op.	Algorithm	GLP	BLISS	ring-TESLA
Randomization	Secret	Sign	●	●	○
	Addition	Key Gen	●	●	●
Skipping	Addition	Sign	●	○	○
	Correctness check	Verify	●	●	●
	Size check	Verify	●	●	○
Zeroing	Secret	Key Gen	●	-	○
	Randomness	Sign	●	●	●
	Hash polynomial	Sign	●	●	●



# VULNERABILITIES OF LBSS

Fault Attack	Changed Value or Op.	Algorithm	GLP	BLISS	ring-TESLA
Randomization	Secret	Sign	●	●	○
	Addition	Key Gen	●	●	●
Skipping	Addition	Sign	●	○	○
	Correctness check	Verify	●	●	●
	Size check	Verify	●	●	○
Zeroing	Secret	Key Gen	●	-	○
	Randomness	Sign	●	●	●
	Hash polynomial	Sign	●	●	●

# NOTATION

- $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ , i.e., polys of degree  $n-1$  with coefficients in  $\left[-\frac{q}{2}, \frac{q}{2}\right]$
- Security assumption:    Learning with errors (R-LWE)  
                                  Short integer solution (R-SIS)

# LATTICE-BASED HARDNESS ASSUMPTION

R-LWE



$$a \cdot s + e = b \pmod{q}$$

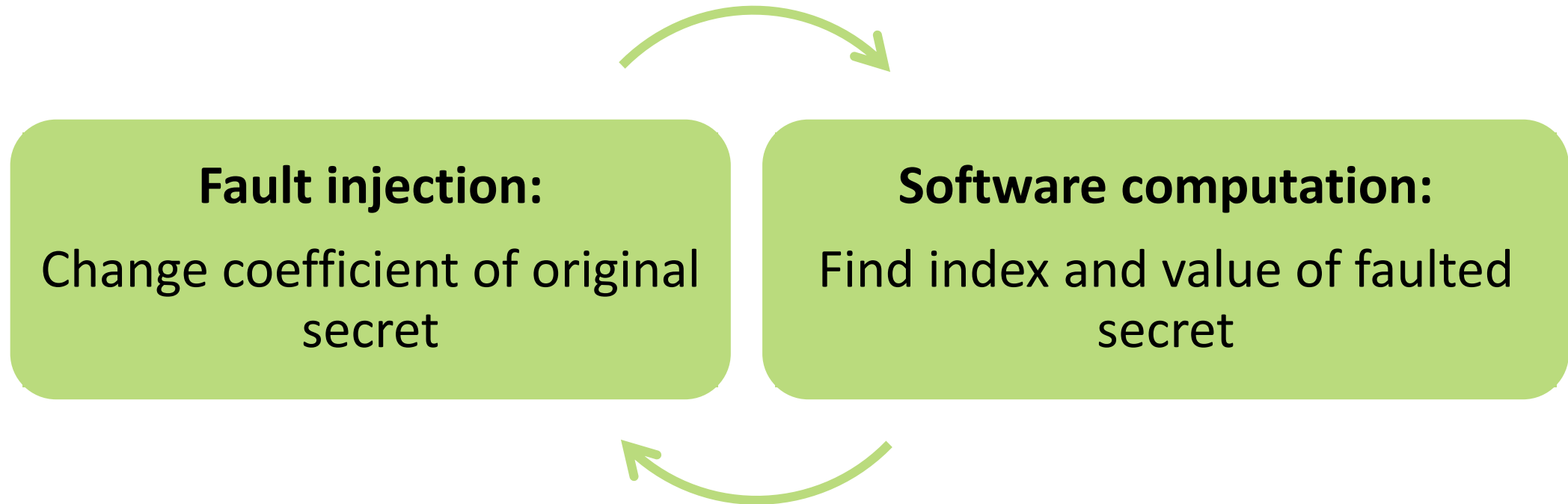
$$a \xleftarrow{\$} R_q$$

$$s_i, e_i \leftarrow D_\sigma \text{ or "small"}$$

Secret key  
Public key

# IDEA RANDOMIZATION ATTACK

- Based on Bao et al. [BDHJNN96]



# DESCRIPTION KEY GENERATION OF GLP SCHEME

Key Generation

Input:  $1^\kappa$

Output:  $pk, sk$

1.  $s, e \leftarrow$  poly with coeffs  $\in \{-1, 0, 1\}$
2.  $a \xleftarrow{\$} \mathbb{Z}_q[x]/(x^n + 1)$
3.  $b \leftarrow as + e \pmod q$
4.  $sk = s, pk = (a, b)$
5. Return  $(pk, sk)$

# DESCRIPTION OF GLP SCHEME

## Signature Generation

Input:  $sk = (s, e), \mu$

Output:  $\sigma = (z_1, z_2, c)$

1.  $y_1, y_2 \leftarrow \$$
2.  $c \leftarrow H(ay_1 + y_2, \mu)$
3.  $z_1 \leftarrow y_1 + sc$
4.  $z_2 \leftarrow y_2 + ec$
5. Return  $(z_1, z_2, c)$  with some probability

## Verification

Input:  $\sigma, \mu, pk = (a, b)$

Output:  $\{0,1\}$

1. Check size of  $z_1, z_2$
2. Check  $c = H(az_1 + z_2 - bc, \mu)$
3. If both checks okay: accept
4. Otherwise: reject

# STRUCTURE ATTACK



**1st Insert fault:** change one coeff.  $s_i \in \{-1,0,1\}$  to  $s_i' \in \{-1,0,1\}$

Assumption 1

Assumption 2:  
coeffs. saved in 2 bit



**2nd Software computation:** find index  $i$  and determine value of  $s_i$

● **1st** find  $s_i - s_i'$  at index  $i$

● **2nd** compute  $s_i$

# FAULTED SIGNATURE

Signature Generation

Input:  $sk = (s, e), \mu$

Output:  $\sigma = (z_1, z_2, c)$

1.  $y_1, y_2 \leftarrow \$$
2.  $c \leftarrow H(ay_1 + y_2, \mu)$
3.  $z_1 \leftarrow y_1 + s'c$
4.  $z_2 \leftarrow y_2 + ec$
5. Return  $(z_1, z_2, c)$  with some probability

During verification check  $c = H(az_1 + z_2 - bc, \mu)$

Instead check  $c = H(\alpha z_1 + z_2 - bc - \alpha x^i c, \mu)$  for values  
 $\alpha \in \{-2, -1, 0, 1, 2\}$  and  $i \in \{0, \dots, n - 1\}$



# FINDING INDEX AND VALUE

For which values  $\alpha \in \{-2, -1, 0, 1, 2\}$  and  $i \in \{0, \dots, n - 1\}$  does the equation ...

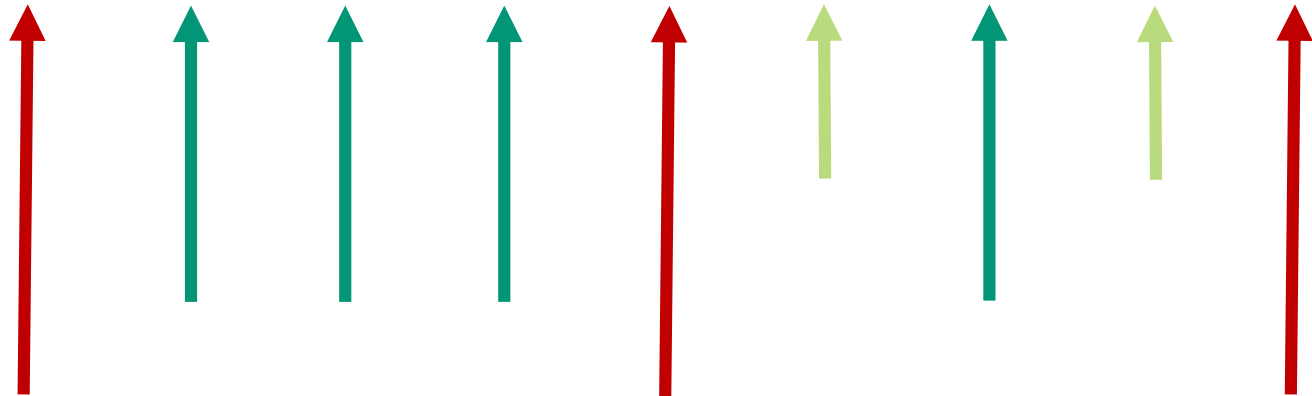
$$\begin{aligned}c &= H(a\mathbf{z}_1 + z_2 - bc - a\alpha x^i c, \mu) \\ &= H(a(y_1 + \mathbf{s}'c) + y_2 + ec - (as + e)c - a\alpha x^i c, \mu) \\ &= H(ay_1 + y_2 + a(\mathbf{s}' - s - \alpha x^i)c, \mu)\end{aligned}$$

... hold?

# DETERMINATION OF COEFFICIENT

$s_i$	0	0	0	1	1	1	-1	-1	-1
$s_i'$	0	1	-1	0	1	-1	0	1	-1
$\alpha = s_i - s_i'$	<b>0</b>	<b>-1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>2</b>	<b>-1</b>	<b>-2</b>	<b>0</b>

Probability to  
uniquely determine  
coeff.: 2/9



# NUMBER OF NEEDED FAULTS

Number of secret coefficients:  $n = 512$

→ plain expected number of faults:  $\frac{9}{2} \cdot 512 \approx 2304$

Reduce number of faults:

Hybrid approach of **fault attacks** and **mathematical cryptanalysis of LWE**

Enough to determine **118** of the secret coefficients

→ expected number of faults:  $\frac{9}{2} \cdot 118 \approx 531$

# HYBRID APPROACH

- LWE gets easier when part of the secret known
- Software Computation time: 1 day
- Lattice cryptanalysis [LP10]: 118 coefficients necessary
- Coefficients by fault attacks: 118
- Coefficients by lattice-based cryptanalysis: 396

# GENERALIZATIONS

- change more than one coefficient per fault
  - decreases number of expected faults
  - increases run time to find coefficients
- apply similar approach to BLISS
  - coeffs chosen in small interval
- not feasible for ring-TESLA
  - coeffs chosen Gaussian distributed

→ One countermeasure: use  
LWE with Gaussian distribution

# COUNTERMEASURE

1.  $y_1, y_2 \leftarrow \mathcal{S}$
2.  $c \leftarrow H(ay_1 + y_2, \mu)$
3.  $b' = as' + e \pmod q$
4.  $z_1 \leftarrow a^{-1}(b - b')c + s'c + y_1$
5.  $z_2 \leftarrow y_2 + ec$
6. Return  $(z_1, z_2, c)$

$$\begin{aligned}z_1 &= a^{-1}(b - b')c + s'c + y_1 \\ &= a^{-1}(as + e - as' - e) + s'c + y_1 \\ &= a^{-1}a(s - s')c + s'c + y_1 \\ &= sc + y_1\end{aligned}$$

Disadvantage:

- Additional computation:  $a^{-1}, b'$
- Additional input:  $b$

# FUTURE WORK

- implement and run attack in praxis
- implement countermeasures and evaluate their effectiveness