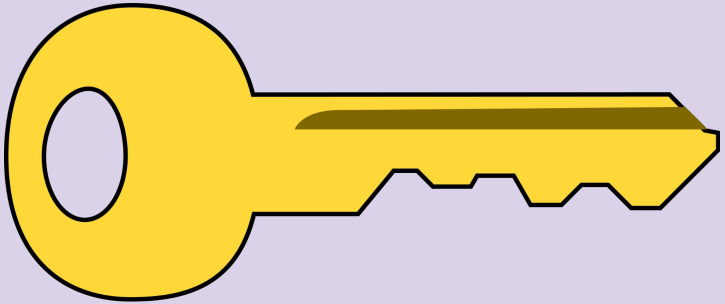
Quantum algorithms for discrete logarithms with applications to Diffie Hellman and ElGamal

Asya Dente

Quick Background for Cryptography

Symmetric Cryptography:

- Shared secret key (ex: one time pad)
- Security? Only the two users should have access to this shared secret key
- Issue: How to get the shared key



Asymmetric Cryptography:

- Public key and private key
- Security? Finding private key from public key requires solving a hard mathematical problem (like DLP)
- Issue: Secure until mathematical problem is no longer computationally difficult

DISCRETE LOGARITHM PROBLEM (DLP)

$$g^x \equiv h \pmod{p}$$

$$x \equiv log_g(h) \pmod{p}$$

Exponentiation:

Given g, p and x, compute h

Discrete Logarithm:

Given g, p and $h=g^x$, find the exponent x that produced h

p: large prime number (~2048 bits)

g: primitive root (generator) of Z_p^*

h: known element from group Z_p^*

WHY DLP IS HARD

The Structure:

- Group Z_p^* → Cyclicity
- Generator g
- Prime number p

Usually, the prime p is 2048 bits long, which means it has around $2^{2048} = 10^{616}$ possible values

---

With the “Baby-step Giant-step” algorithm, runtime: $O(\sqrt{p})$

This means around $\sqrt{(2^{2048})} = 2^{1024}$ steps… which is huge!

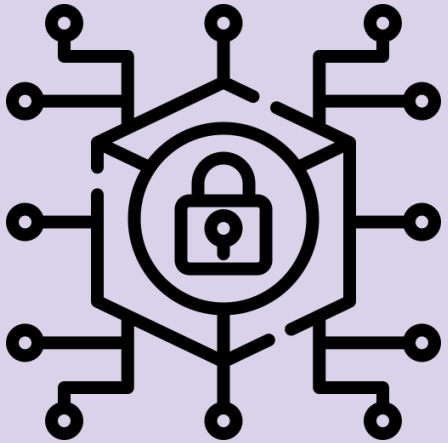
Current computers: 2^{100} operations… which already could take decades.

=> Impossible

THREAT TO DLP: SHOR'S

DLP → Period-finding Problem

The function is $f(x_1, x_2) = a^{sx_1+x_2} \bmod N$



- problem here would be: given a and $b = a^s$, what is s?
- still periodic, but period is a 2-tuple (ℓ, -sℓ), instead of one value for the factoring application. Important to find s
- r is smallest positive integer for which $a^r \bmod N = 1$

THREAT TO DLP: SHOR'S

DLP → Period-finding Problem

1. Superposition
2. U creates entanglement
3. Write in periodic structure
4. Apply inverse QFT to make interference happen
5. Hidden period starts showing
6. Continued fractions algorithm

$$\rightarrow \frac{1}{2^t} \sum_{x_1=0}^{2^t-1} \sum_{x_2=0}^{2^t-1} |x_1\rangle |x_2\rangle |0\rangle$$

$$\rightarrow \frac{1}{2^t} \sum_{x_1=0}^{2^t-1} \sum_{x_2=0}^{2^t-1} |x_1\rangle |x_2\rangle |f(x_1, x_2)\rangle$$

$$= \frac{1}{2^t \sqrt{r}} \sum_{\ell_2=0}^{r-1} \left[\sum_{x_1=0}^{2^t-1} e^{2\pi i (s\ell_2 x_1)/r} |x_1\rangle \right] \left[\sum_{x_2=0}^{2^t-1} e^{2\pi i (\ell_2 x_2)/r} |x_2\rangle \right] |\hat{f}(s\ell_2, \ell_2)\rangle$$

$$\rightarrow \frac{1}{\sqrt{r}} \sum_{\ell_2=0}^{r-1} |\widetilde{s\ell_2/r}\rangle |\widetilde{\ell_2/r}\rangle |\hat{f}(s\ell_2, \ell_2)\rangle$$

Discrete logarithms in **polynomial** time

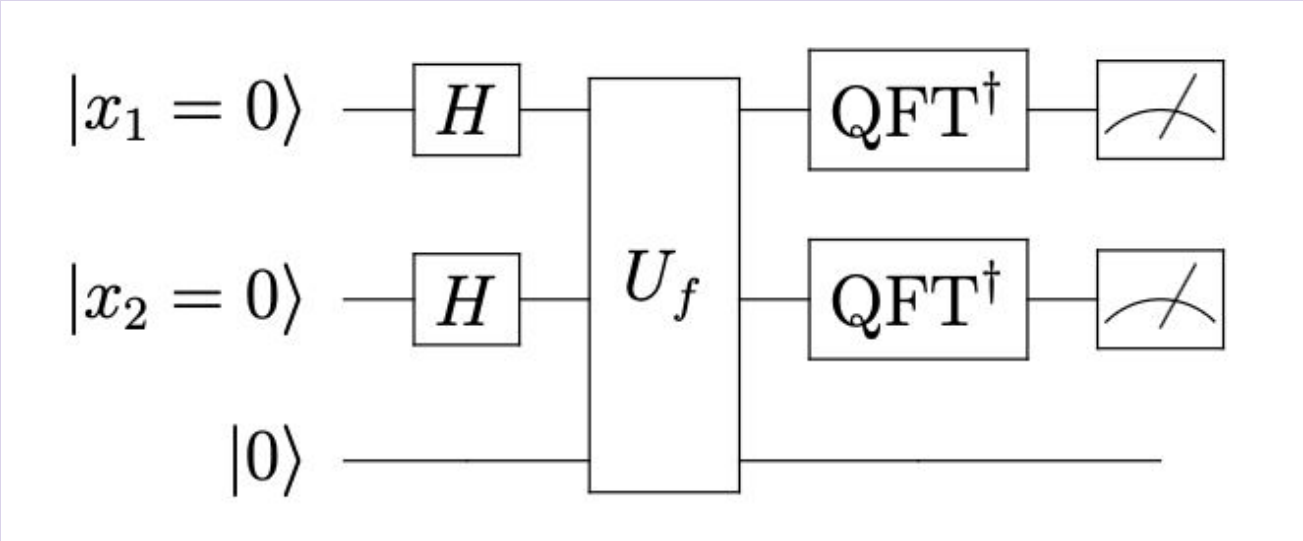
$$\rightarrow \left(\widetilde{s\ell_2/r}, \widetilde{\ell_2/r} \right)$$

DLP SOLVED CIRCUIT OVERVIEW

Uf computes f(x1,x2), encodes periodicity into phases

Inverse QFT “extracts” hidden period and turns it into measurable “peaks”

Measurement gives values that allow us to solve for s

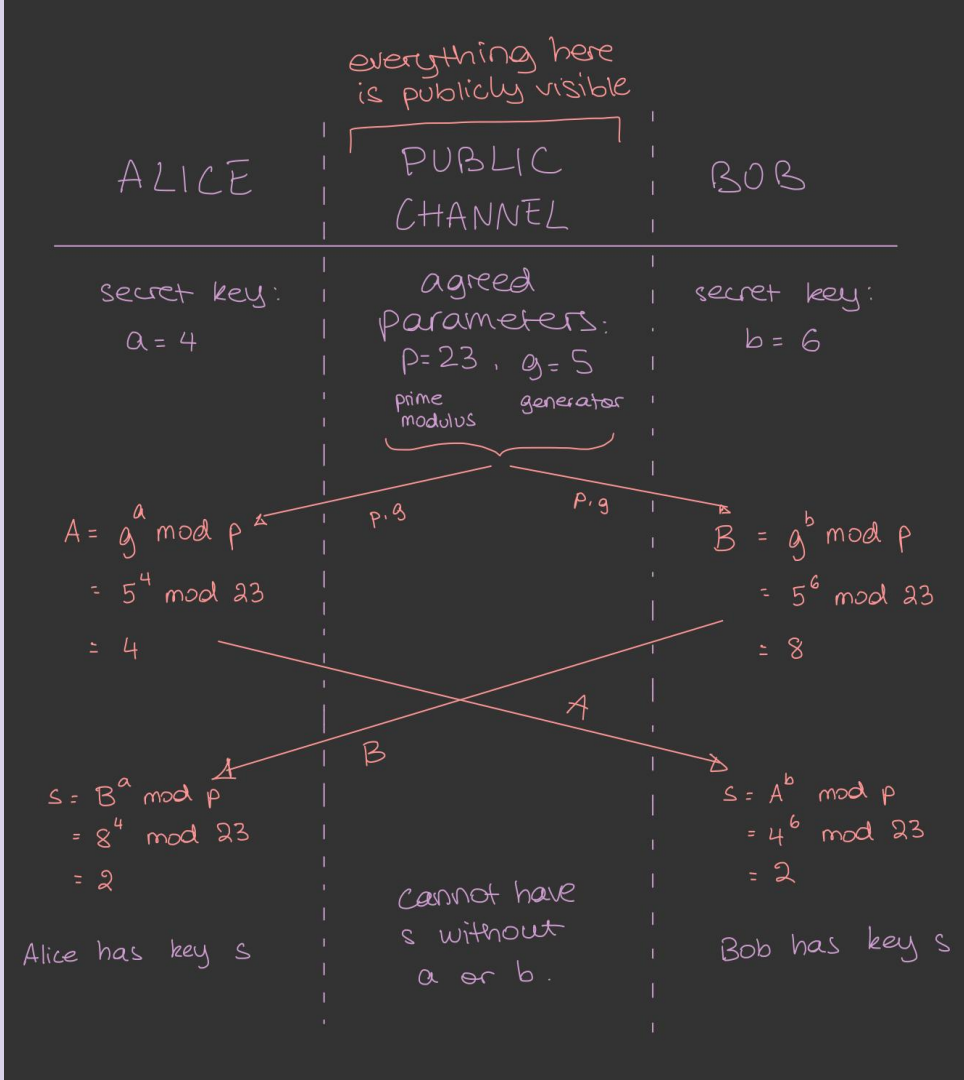


DIFFIE-HELLMAN

Method used to generate a symmetric key.

Symmetric key $s = g^{ab} \bmod p$

Assumptions:
1. The Discrete Logarithm Problem
2. The Computational Diffie–Hellman Problem (CDH)



SHOR'S ON DIFFIE HELLMAN

To break Diffie Hellman, use same algorithm as DLP, only with instance (if we want a):

$f(x1, x2) = g^{ax1+x2} \bmod p$

With:

base a (from book) = g modulus N = p

secret exponent s = a public value b (from book) = g^a

Once either a or b is known, attacker can just do the following (say he learns a):

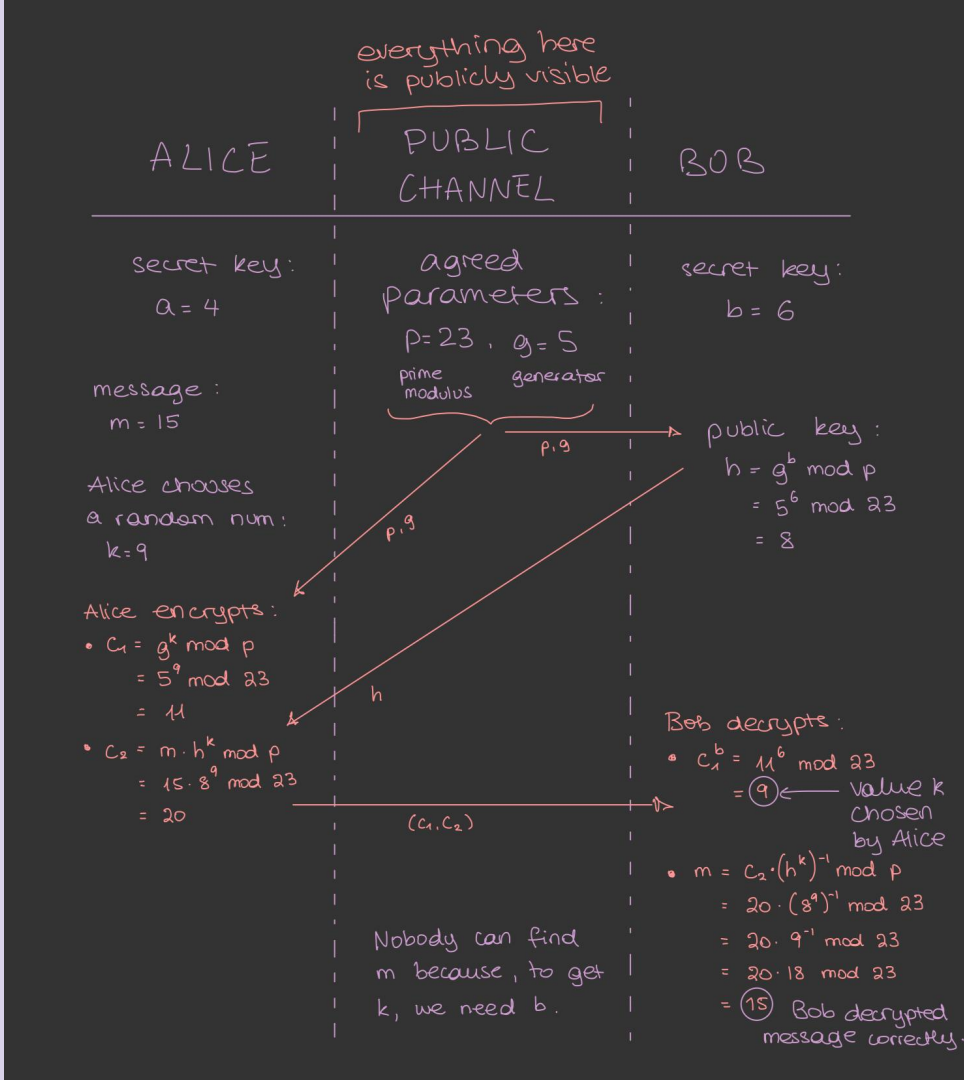
$s = g^{ab} \bmod p = (g^b)^a \bmod p = B^a \bmod p$

ELGAMAL

Public key encryption algorithm, based off Diffie Hellman, asymmetric

Ex: Alice wishes to send a message to Bob →

Assumptions:
1. The Discrete Logarithm Problem
2. The Computational Diffie–Hellman problem (CDH)
3. The Decisional Diffie Hellman assumption (DDH)



SHOR'S ON ELGAMAL

Run Shor's algorithm on the public key of Bob, to find his private key.
Bob's private key found → encryption broken → attacker can decrypt message.

Since we want b:

$f(x1, x2) = g^{bx1+x2} \bmod p$

With:

base a (from book) = g modulus N = p

secret exponent s = b public value b (from book) = $h = g^b$

POST QUANTUM CRYPTOGRAPHY (PQC)

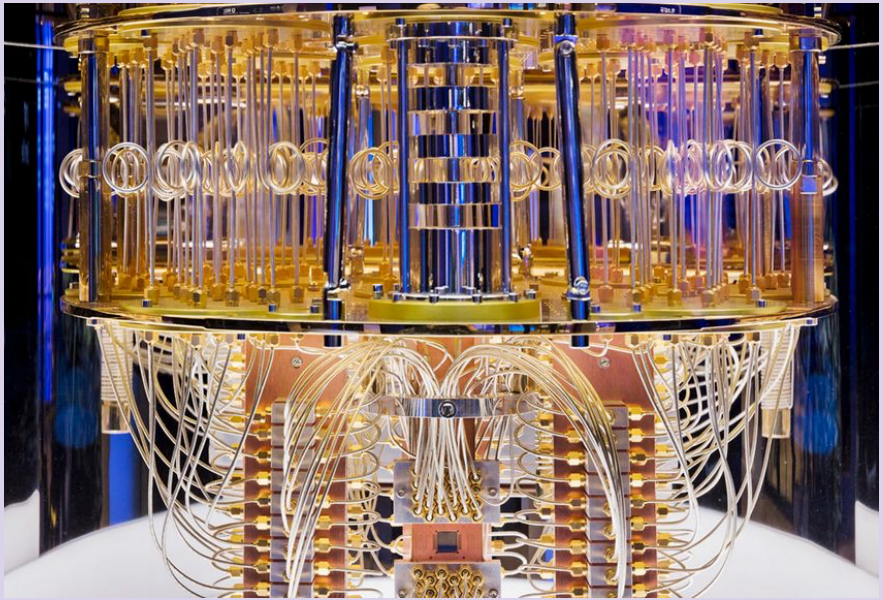
CRYSTALS-Kyber (ML-KEM)

for key exchange

CRYSTALS-Dilithium (ML-DSA), FALCON, and SPHINCS+ (SLH-DSA)

for digital signatures

CONCLUSION



REFERENCES

[1] M. A. Nielsen and I. L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, 2010. Relevant sections: Section 5.4.2 (p. 238), Section 10.10.2 (p. 432), and Appendix 5 (p. 640).
[2] A. M. Childs. Lecture Notes on Quantum Algorithms - Chapter 5. Available at: https://www.cs.umd.edu/~amchilds/qa/qa.pdf.
[3] A. Mandl. Quantum Algorithms for the Discrete Logarithm Problem. Bachelor's Thesis, TU Wien, 2021. Available at: https://repositum.tuwien.at/bitstream/20.500.12708/18830/1/Mandl%20Alexander%20-%202021%20-%20Quantum%20algorithms%20for%20the%20discrete%20logarithm%20problem.pdf.
[4] Lecture Notes:
Lecture 10 — Quantum Fourier Transform;
Lecture 11 — Quantum Phase Estimation;
Lecture 12 — Shor's Algorithm.
[5] W. Di!e and M. E. Hellman. "New Directions in Cryptography," IEEE Transactions on Information Theory, IT-22(6): 644–654, 1976. Available at: https://ieeexplore.ieee.org/document/1055638.
[6] ChatGPT (OpenAI). Used as a supplementary explanation tool for clarifying di!cult concepts and verifying intermediate steps where needed.
[7] Di!e–Hellman Key Exchange. Available at: https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange.
[8] Di!e–Hellman Key Exchange and Perfect Forward Secrecy, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/computer-networks/diffie-hellman-key-exchange-and-perfect-forward-secrecy/.
18[9] Discrete Logarithm Problem Explained, YouTube. Available at: https://www.youtube.com/watch?v=za9azzh4v9A.
[10] Shor's Algorithm and Discrete Logs, YouTube. Available at: https://www.youtube.com/watch?v=yKNTfeqSEhc.
[11] ElGamal Encryption Algorithm, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/computer-networks/elgamal-encryption-algorithm/.
[12] DDH, Available at: https://en.wikipedia.org/wiki/Decisional_Diffie%E2%80%93Hellman_assumption.
[13] https://www.quora.com/What-is-the-largest-number-that-has-been-factored-by-a-classical-computer-How-long-did-it-take