Introduction to Cryptography

for Protexxa Students

By Marc Lijour

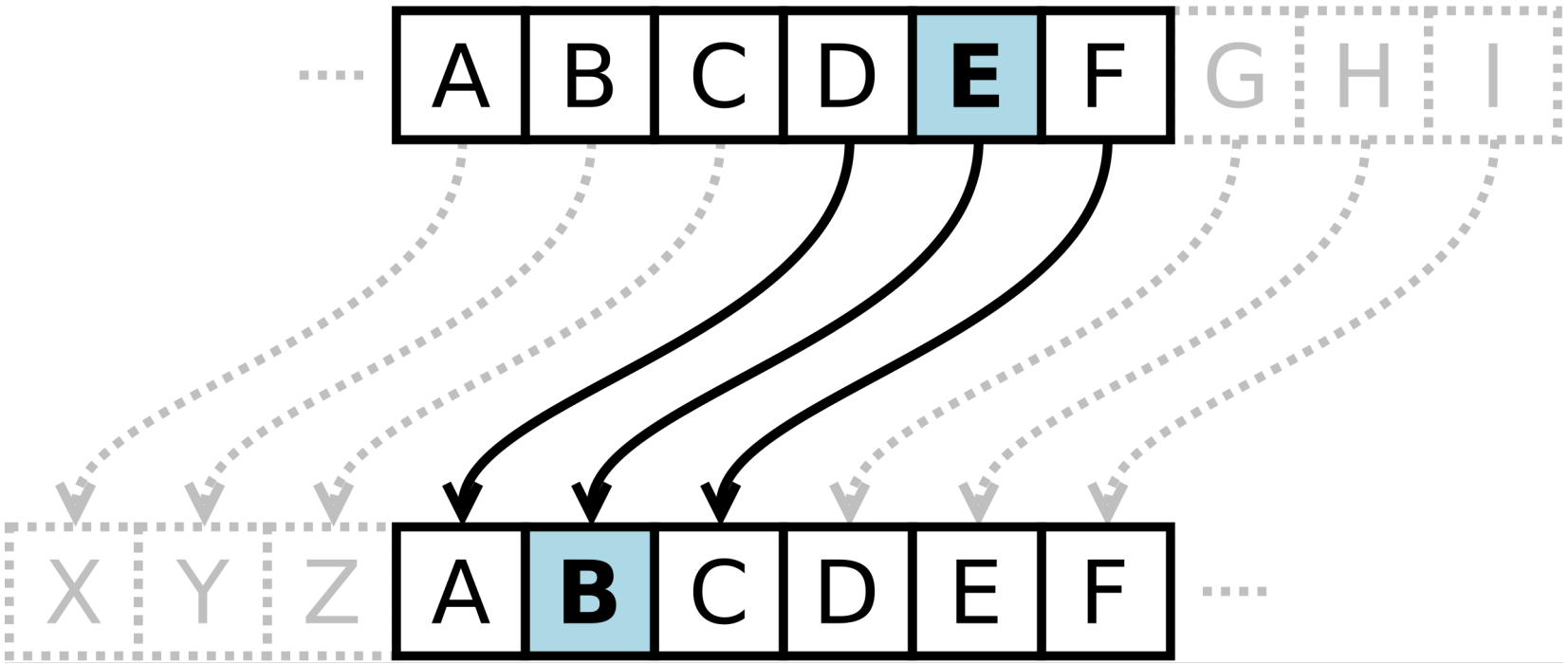
December 4, 2023

Table of Contents

1. Setting the stage
   - Origin Story
   - Some Definitions
2. Cryptography in practice
   - Cryptographic Primitives
   - Common Applications
   - Common Algorithms
3. The future of cryptography
   - Post-quantum Cryptography
   - Fully Homomorphic Encryption

Cryptography appeared as early as 1900 BC

- The first known use of cryptography was recorded in the tomb of *Khnumhotep II* in Egypt.
- Later used in Mesopotamia, in India, and among Hebrew scholars before its use was made popular in the latin world



The Caesar Cipher (circa 100 BC)

Introduction of encryption keys

- In the XVIth century, Giovan Battista Bellaso (1553) described the use of an encryption key
- Popularized as the Vigenère cipher, it was reputed unbreakable ("le chiffrage indéchiffrable")
- Charles Babbage ("father of the computer") is known to have broken this algorithm in 1854

The Vigenère cipher

A Vigenère cipher derivative: the One-Time Pad

- Gilbert Sandford Vernam, AT&T Bell Labs engineer, invents the One-Time Pad (OTP) in 1917
- Reputed **unbreakable** provided four conditions are met (the key is random, as long as the plaintext, used only once, and kept secret among the parties)

```
      h        e        l        l        o  message
   7 (h)    4 (e)   11 (l)   11 (l)   14 (o) message
+ 23 (X)   12 (M)    2 (C)   10 (K)   11 (L) key
= 30       16       13       21       25     message + key
=  4 (E)   16 (Q)   13 (N)   21 (V)   25 (Z) (message + key) mod 26
      E        Q        N        V        Z  → ciphertext
```

The OTP cipher (CC BY-SA 4.0, via Wikipedia, 2023)

Electric machinery

- The need for speed (real-time ideally) and the popularisation of electronic transmissions demanded better tools
- German engineer Arthur Scherbius invented the Enigma machine after WWI
- Polish mathematician and cryptologist Marian Rejewski breaks the cipher in 1932
- Alan Turing ("father of computer science") contributed to winning WWII

The Enigma machine

Businesses adopt cryptography

- In the early 1970's, IBM clients start asking for encryption, leading Horst Feistel & the Crypto group to develop the Lucifer cipher
- It eventually becomes the Data Encryption Standard (DES) in 1976 (Simmons, 2023)
- Published by the US National Bureau of Standards, rebranded as the National Institute of Standards and Technology (NIST)
- Notably is was published (1977) allowing anyone to implement it in software code, still it was considered highly secure
- After 1997, DES was broken and replaced by the Advanced Encryption Standard (AES), more resistent to advanced compute and the availability of the Internet
- In 2000, the US Dept of Commerce simplified the rules to export commercial and open source software containing cryptography
- NIST remains a leading organization setting cybersecurity standards, among others such as the IEEE Standards Association

Cryptography becomes personal

- In 1991, Phil Zimmermann develops PGP (for Pretty Good Privacy), a simple solution to encrypt messages and to authenticate senders (establishing trust)
- GnuPG is a popular Free Software implementation introduced in 1997 on Linux and later on other platforms
- Key signing parties became popular in Free/Libre Open Source Software communities

Stevenfruitsmaak (2008), CC BY 2.5, via Wikimedia Commons

From the original Web to Web3

- Starting in 2008, many looked for a reboot of the financial system
- New cryptographic methods were invented to create the first massively scalable peer-to-peer electronic cash system: ₿

https://nymag.com/intelligencer/2021/09/occupy-wall-street-changed-everything.html

Lessons learned from history

- **The Kerckhoffs' Principle**: the security of a cryptosystem only on its keys, while everything else (including the algorithm) should be considered public knowledge (Petitcolas, 2011)
- In an increasingly digital world, cryptography has become an essential part of the toolkit for businesses and individuals, as much as for governments
- Today's systems are increasingly decentralized (e.g. IoT, Web3) and subject to rising cyberthreats demanding better cryptography
- Important applications include official IDs (e.g. driver license, passport), e-commerce, social media (e.g. sharing pictures online), privacy (e.g. protecting messages, medical data, financial transactions), and cryptocurrencies

More reading: see the nice article from Sidhpurwala (2023).

Definitions

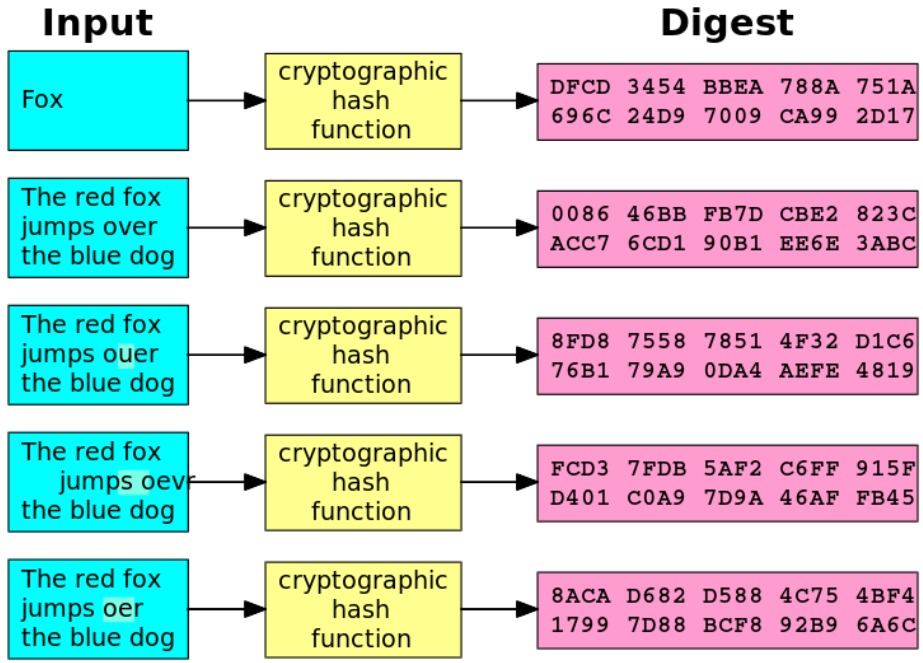
- **Plaintext**: unencrypted information
- **Cipher** (or Cypher in British English): a way to write a message in a way to hide its meaning from others
- **Ciphertext**: the result of applying an encrypting algorithm to plaintext
- **Algorithm**: a process to compute a specific result, such as encrypted or decrypted data
- **Encryption**: a cipher applied to digital information and signals
- **Decryption**: the reverse operation

Table of Contents

1. Setting the stage
   - Origin Story
   - Some Definitions
2. Cryptography in practice
   - Cryptographic Primitives
   - Common Applications
   - Common Algorithms
3. The future of cryptography
   - Post-quantum Cryptography
   - Fully Homomorphic Encryption

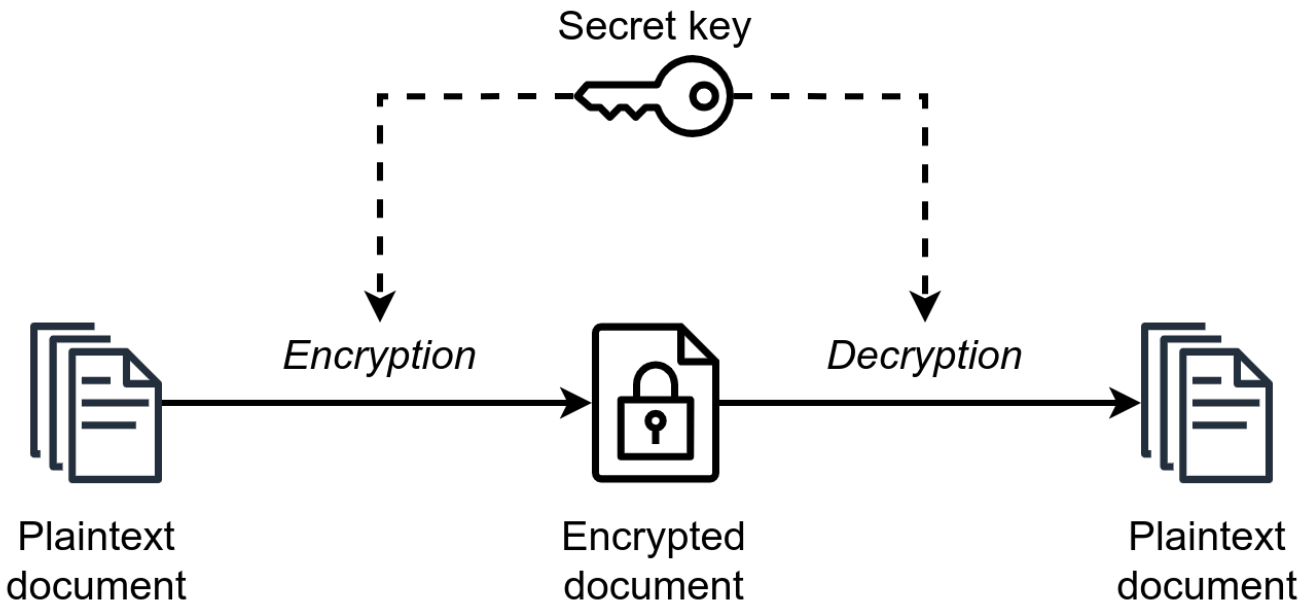
Hashing

- Used for fingerprinting digital documents, checksums, and storing login information among other things
- One-way hash functions have interesting properties used in computer science
- Small changes in plaintext creates a visible big change in ciphertext
- Hashes have a fixed length and are spread across the universe of possibilities (good for indexing)



Encrypting with symmetric-key cryptography

- A secret (a single key) is *shared* among participants
- The key is used for both encrypting and decrypting
- Efficient for bulk encryption
- Keeping the key secret can be challenging (e.g. in communications)



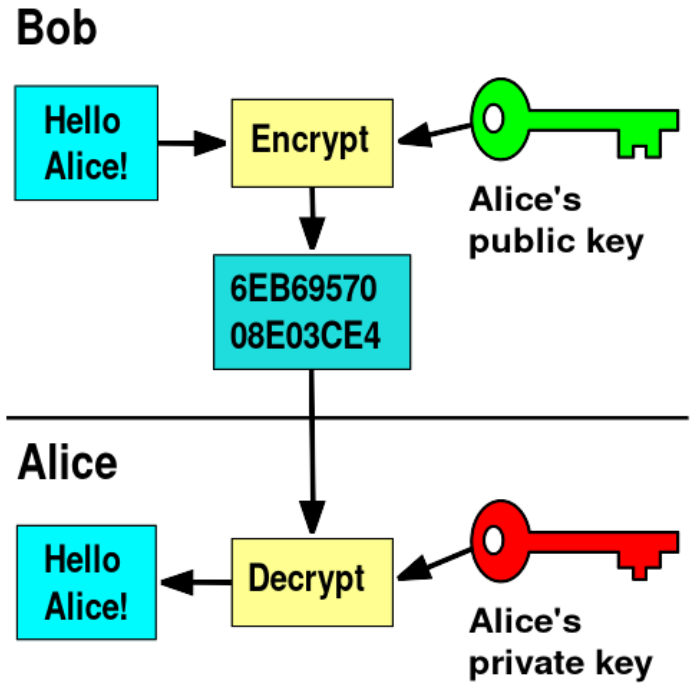
MarcTOK and JGraph (2023), CC BY-SA 4.0, via Wikimedia Commons

See also Henninger and Mashatan (2019)

Public-key cryptography

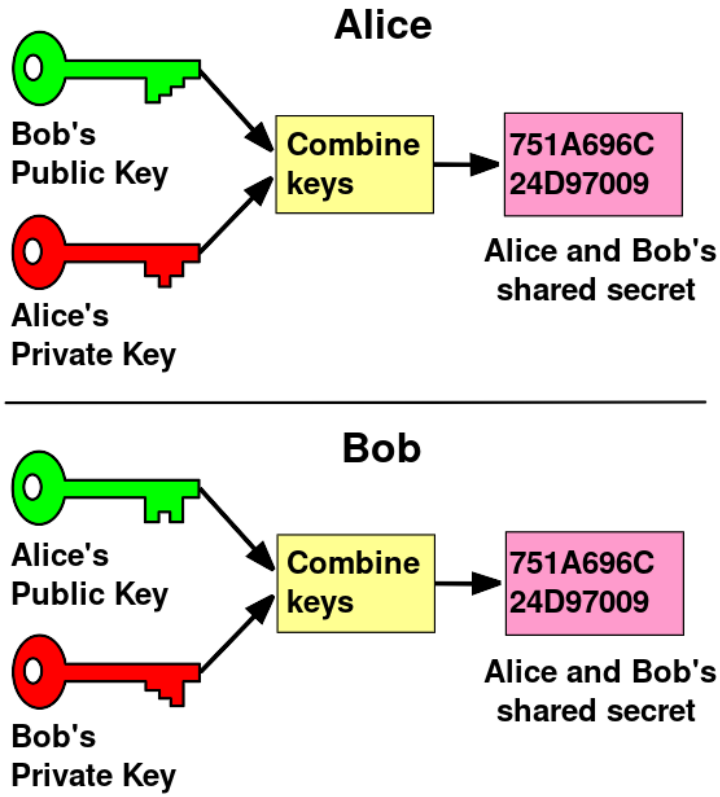
A better way to share keys is to use asymmetric encryption.

- Two keys are generated by an algorithm based on one-way functions theory
- One of the keys is shared publicly (the *public key*)
- The other key is to be kept private at all cost (the *private key*)
- Both keys togeter are called a *key pair*



Diffie–Hellman key exchange (DHE)

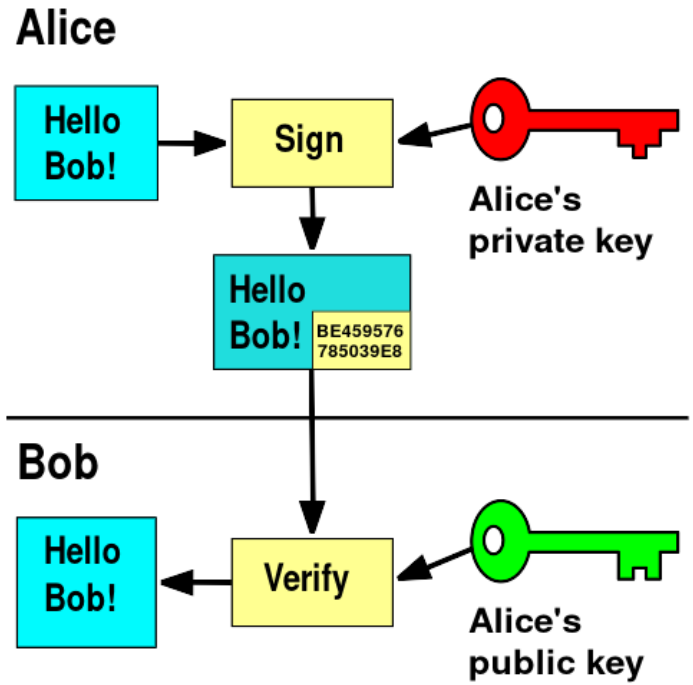
- Each party generates a key pair
- Both parties obtain an *authentic* copy of each other's public key
- Each party computes a shared secret that can be used as a *symmetric cipher*



Signing

Diffie and Helman wrote about *digital signatures* as early as 1976, just before the RSA algorithm was invented by Ronald Rivest, Adi Shamir, and Len Adleman.

- This time Alice's uses her private key to compute a signature
- Anyone can use her public key to verify that Alice was the one who signed
- Applications include asserting the origin of a message and non-repudiation



Private key signing (CC BY-SA 4.0, via FlippyFlink, 2019)

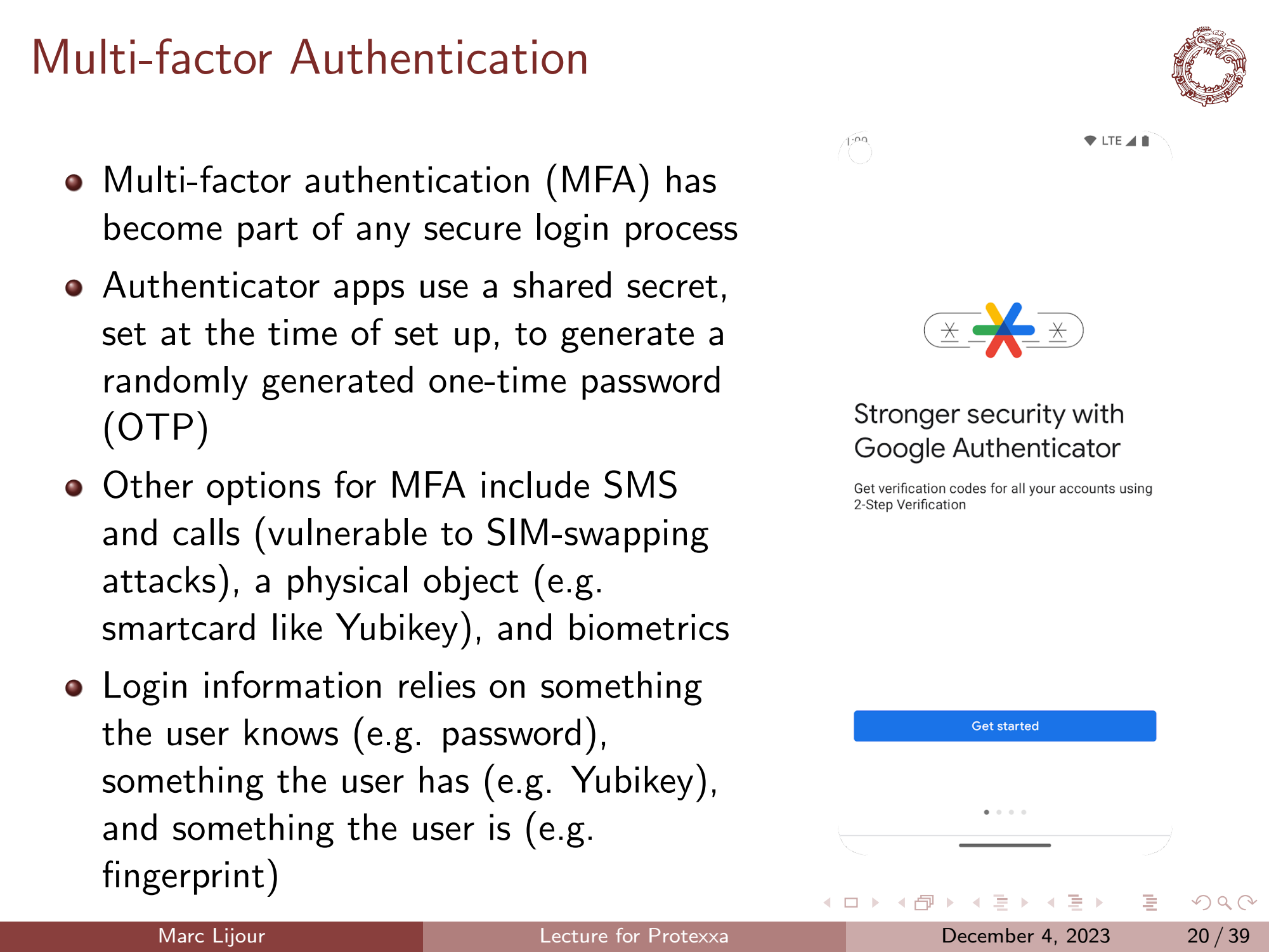
Managing keys

- Public keys can be published on key servers such as https://keys.openpgp.org and others
- Private keys need to be stored with the highest security care (limited access, read-only, passphare-protected, etc)
- Added level of security are provided by full-disk encryption (e.g. computer hard drive) and smartcards (e.g. Yubikey)
- Both the public and private keys need to be carefully managed (The Free Software Foundation, 1999)

Installing software

- Installing software is open to vulnerability, but cryptography can help
- Making sure the binaries (i.e. executable file) have not been swaped by a malicious version
- Verify the signature of the checksum published on the website
- Compute a hash of the downloaded binaries and compare it with the server's published checksum
- See for example https://ubuntu.com/tutorials/how-to-verify-ubuntu
- Modern systems (e.g. App stores) perform these tasks behind the scene

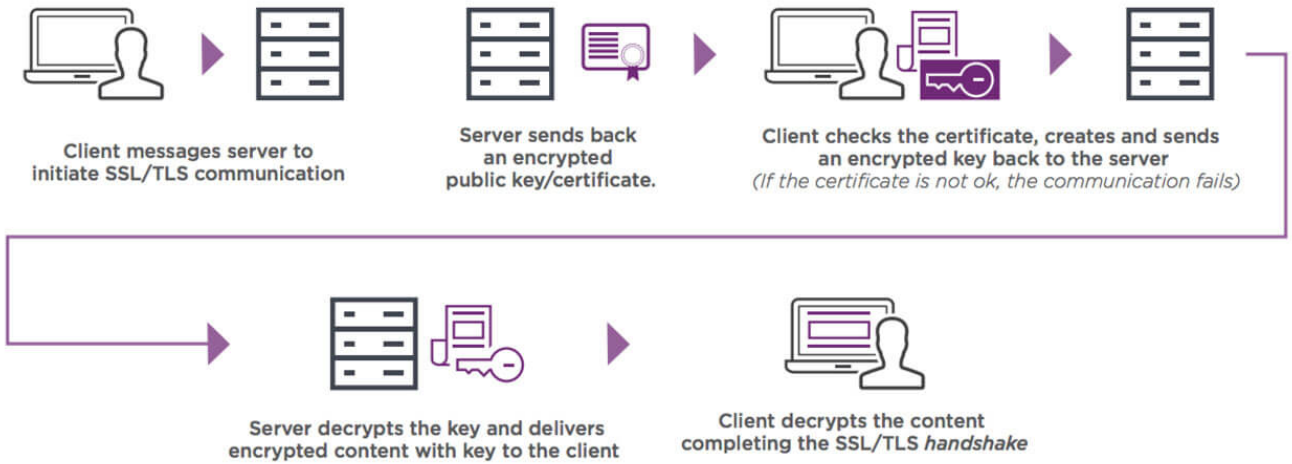
Multi-factor Authentication



- Multi-factor authentication (MFA) has become part of any secure login process
- Authenticator apps use a shared secret, set at the time of set up, to generate a randomly generated one-time password (OTP)
- Other options for MFA include SMS and calls (vulnerable to SIM-swapping attacks), a physical object (e.g. smartcard like Yubikey), and biometrics
- Login information relies on something the user knows (e.g. password), something the user has (e.g. Yubikey), and something the user is (e.g. fingerprint)

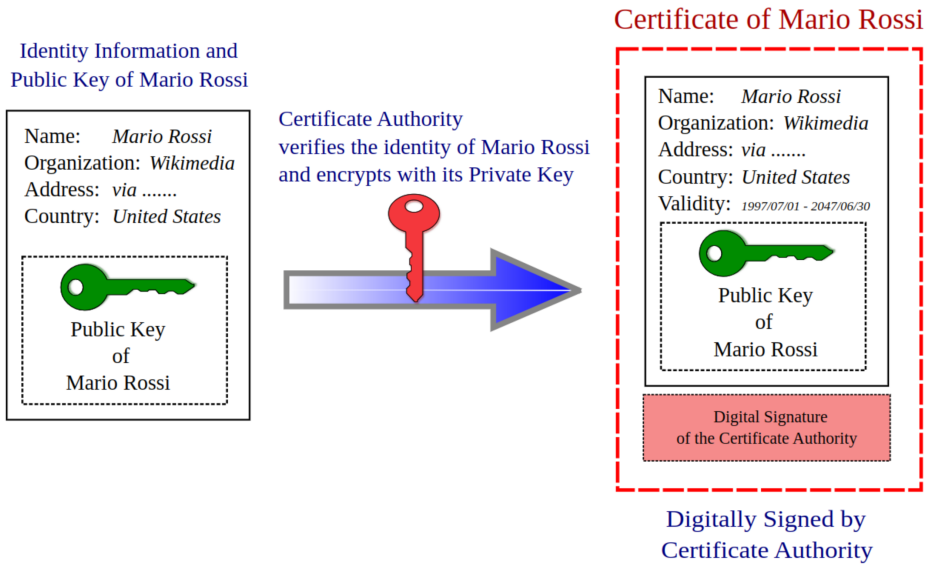
Securing Web communications (HTTPS)

- The HTTP protocol is sending messages in plaintext
- HTTPS refers to the practice of establishing an encrypted tunnel between the browser and the Web server before sending any Web content (preventing eavesdropping and man-in-the-middle attacks)
- The SSL protocol is now deprecated, and replaced by TLS (Transport Layer Security)
- As of June 2021, TLS 1.1 has been deprecated to the benefit of TLS 1.2 or later



HTTPS relies on X.509 digital certificates

- To start encrypted tunnels securely, trust in the public (encrypting) key must be established
- An X.509 certificate must have been issued by a recognized *Certificate Authority*
- Once the certificate has been verified, the browser can use the public key to generate a short-term session key to encrypt communications (symmetric cipher)



Public key certificate diagram (CC BY-SA 3.0, via Giaros, 2007)

Other applications

- End-to-end encryption
- Virtual Private Network (VPN)
- SSH

SHA (Secure Hash Algorithm)

- SHA 1 algorithms have been discontinued
- SHA 256 (for a length of 256 bits) is a part of the SHA 2 family of algorithms
- Other less common lengths include 224, 256, 384 and 512 bits (SHA 224, etc)
- Used for digital signature verification, password hashing (for secure storage), SSL handshakes, integrity checks

MD5 (message-digest algorithm)

- An older hashing algorithm only used to verify data integrity nowadays
- Invented by Ronald Rivest (from RSA) in 1991
- In 2011, serious weaknesses were found in collision resistance (the ability to find plaintexts leading to the same hash) using common laptops

CRC (Cyclic Redundancy Check) codes

- A CRC code is used to detect errors during data transmissions in digital networks or storage devices
- Adds a few bits to the transmission for error checking
- Algorithm based on the remainder of polynomial divisions
- Commonly implemented in binary hardware

AES (Advanced Encryption Standard)

- A symmetric cipher variant of the Rijndael block cipher developed Joan Daemen and Vincent Rijmen included in the ISO/IEC 18033-3 standard
- Based on a substitution–permutation network
- AES was announced by the NIST in 2001
- Very popular including with the US government: known as the first (and only) publicly accessible cipher approved by the NSA
- It is considered to be quantum resistant

RSA (Rivest–Shamir–Adleman)

- Ron Rivest, Adi Shamir and Leonard Adleman made the algorithm public in 1977
- RSA is based on the difficulty of factoring prime numbers
- Relatively heavy compute i.e. slow
- Mostly used to transmit shared secrets to enable symmetric ciphers (e.g. via TLS, OpenSSL), i.e. encrypting and signing
- Highly secure if the key size is large enough (at least 2048 bits)
- Henninger and Mashatan summarize the latest recommendations from NIST (2019)

DSA (Digital Signature Algorithm)

- The DSA algorithm is based on modular exponentiation and the discrete logarithm problem
- NIST proposed DSA for use in their Digital Signature Standard (DSS) in 1991, adopted in 1994
- Designed for digital signatures (and verification)
- Best suited for signing and decryption
- Can use smaller keys than RSA (1024 or 2048 bits)

ECDSA (Elliptic Curve Digital Signature Algorithm)

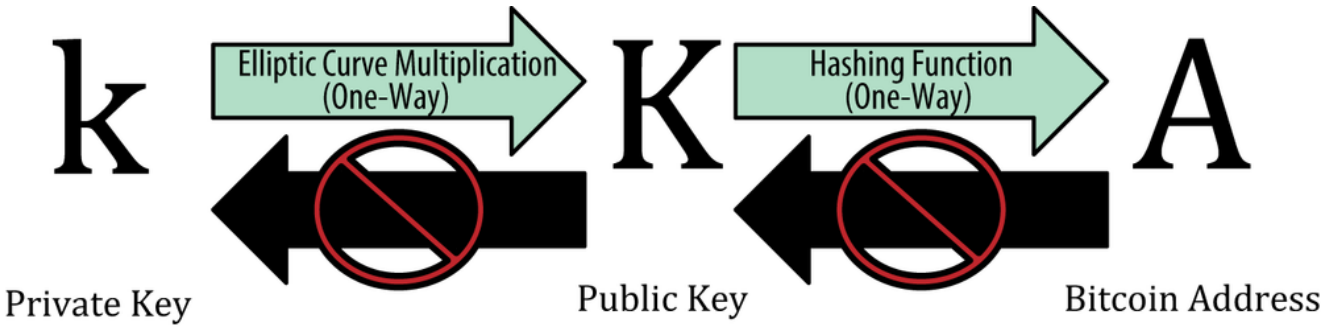
- A modern variant of DSA using elliptic-curve cryptography
- Faster and more secure than RSA and DSA for signing thanks to smaller keys (256 or 384 bits)
- Resistant to quantum attacks

EdDSA (Edwards-curve Digital Signature Algorithm)

- A variant of Schnorr signature based on twisted Edwards curves (planed models of elliptic curves)
- ed25519 is used in OpenSSH
- Less susceptible to weak random number generation as in DSA/ECDSA

Example: applications in Bitcoin

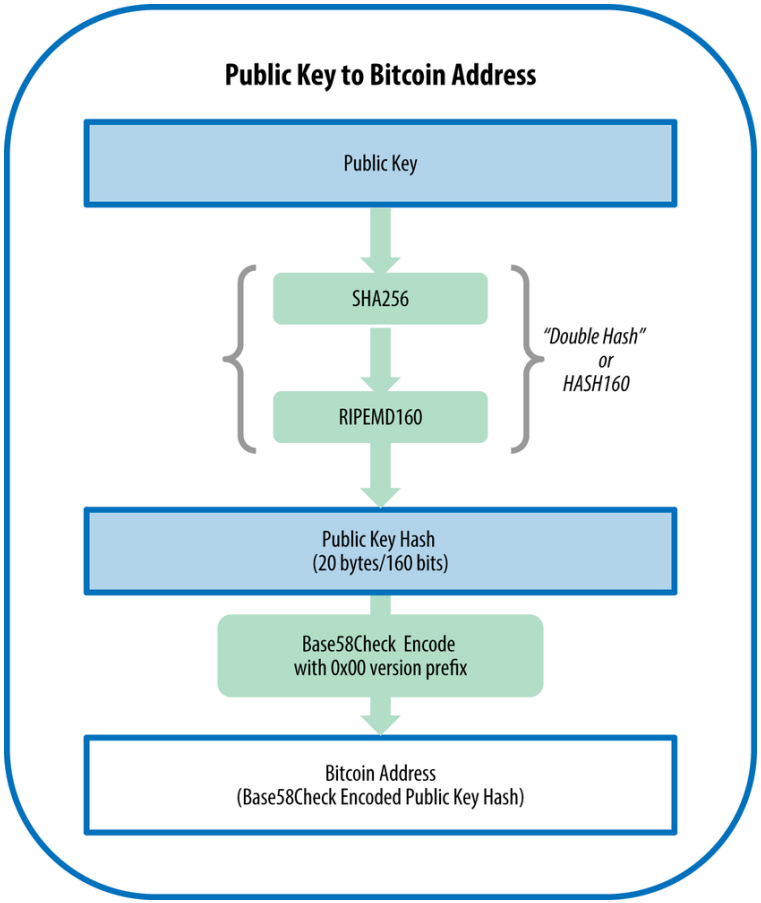
- Bitcoin uses the secp256k1 elliptic curve for its ECDSA implementation libsecp256k1
- The private key is selected at random, from which the public key is derived through multiplication
- The reverse operation from the multiplication is called "finding the discrete logarithm", a hard problem making ECDSA a one-way application in practice



Extract from Mastering Bitcoin (Antonopoulos, 2014)

Example: applications in Bitcoin

- From the public key, the Bitcoin software applies the SHA256 hashing algorithm, and then the *RACE Integrity Primitives Evaluation Message Digest* (RIPEMD160) on the result to produce a 160-bit hash
- Finally, it encodes the public key hash in Base58Check format, which is derived from Base58 encoding plus a checking code using SHA256 to produce a checksum



Public key to bitcoin address: conversion of a public key into a bitcoin address –Extract from Mastering Bitcoin (Antonopoulos, 2014)

Table of Contents

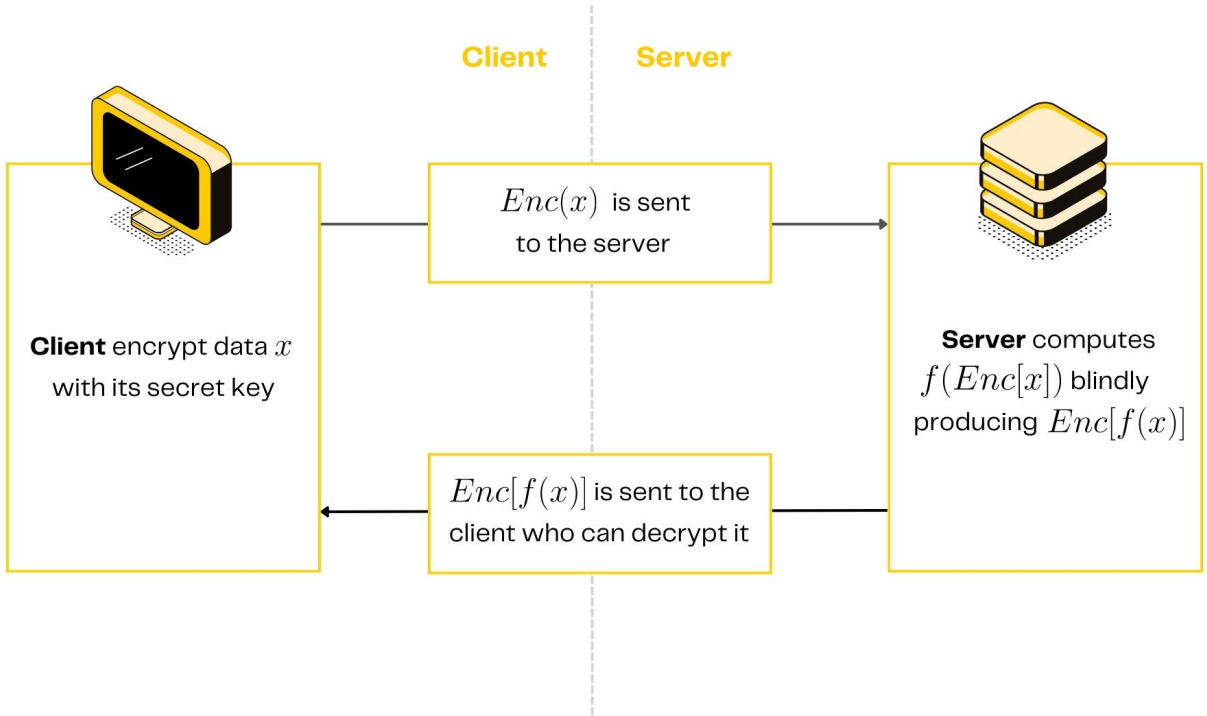
1. Setting the stage
   - Origin Story
   - Some Definitions
2. Cryptography in practice
   - Cryptographic Primitives
   - Common Applications
   - Common Algorithms
3. The future of cryptography
   - Post-quantum Cryptography
   - Fully Homomorphic Encryption

Post-quantum cryptography

- With the rise of Quantum computing, conventional cryptographic algorithms are at risk
- Both content in flight and at rest is concerned
- Researchers are working on quantum-resistant algorithms
- Government agencies are actively searching for solutions e.g. NIST https://csrc.nist.gov/projects/post-quantum-cryptography

Fully Homomorphic Encryption (FHE)

- Data seldom need to be decrypted to be processed on servers
- Hardware-based solutions such as *hardware enclaves* and *hardware security modules* (HSM) provide trusted execution environments
- Fully Homomorphic Encryption (FHE) is another approach albeit compute-intensive and slower
- Part of the secure multi-party computation field of cryptography



Homomorphic Encryption (FHE) enables encrypted data processing (Hindi, 2023)

Let's chat further!

Contact Details

@ marc@creative-emergy.com
Marc Lijour
@marclijour
Calendly

IBU International Business University

IEEE BLOCKCHAIN CANADA

LA FRENCH TECH TORONTO

BLOCKZERO ADVISORS

References I

Antonopoulos, A. M. (2014). *Mastering bitcoin: Unlocking digital crypto-currencies* (1st). O'Reilly Media, Inc.

FlippyFlink. (2019). Private key signing. https://upload.wikimedia.org/wikipedia/commons/7/78/Private_key_signing.svg

Giaros. (2007). Public key certificate diagram. https://upload.wikimedia.org/wikipedia/commons/6/65/PublicKeyCertificateDiagram_It.svg

Henninger, A., & Mashatan, A. (2019). Standardized cryptographic algorithms. https://www.torontomu.ca/content/dam/tedrogersschool/cybersecurity-research-lab/CRL101/PDFs/Standardized_cryptography_101.pdf

Hindi, R. *Https://www.zama.ai/post/private-smart-contracts-using-homomorphic-encryption*. Presentation at EDCON 2023, Montenegro. 2023, May. https://www.zama.ai/post/private-smart-contracts-using-homomorphic-encryption

IEEE Standards Association. (2023). IEEE Standards & Projects for Cybersecurity. https://standards.ieee.org/practices/foundational/cybersecurity-standards-projects/

MarcTOK & JGraph. (2023). Simple symmetric encryption. https://commons.wikimedia.org/wiki/File:Simple_symmetric_encryption.png

National Institute of Standards and Technology (NIST). (2023). Cybersecurity Framework. https://www.nist.gov/cyberframework

References II

Petitcolas, F. A. P. (2011). Kerckhoffs' principle. In H. C. A. van Tilborg & S. Jajodia (Eds.), *Encyclopedia of cryptography and security* (pp. 675–675). Springer US. https://doi.org/10.1007/978-1-4419-5906-5_487

Sidhpurwala, H. (2023). A brief history of cryptography. https://www.redhat.com/en/blog/brief-history-cryptography

Simmons, G. J. (2023). Data encryption standard. In *Encyclopedia britanica*. https://www.britannica.com/topic/Data-Encryption-Standard#ref1101034

Stevenfruitsmaak. (2008). FOSDEM 2008 key signing party. https://commons.wikimedia.org/wiki/File:FOSDEM_2008_Key_signing_party.jpg

The Free Software Foundation. (1999). Key Management. In *The GNU Privacy Handbook*. https://www.gnupg.org/gph/en/manual/c235.html

Wikipedia. (2023). One-time pad. https://en.wikipedia.org/wiki/One-time_pad