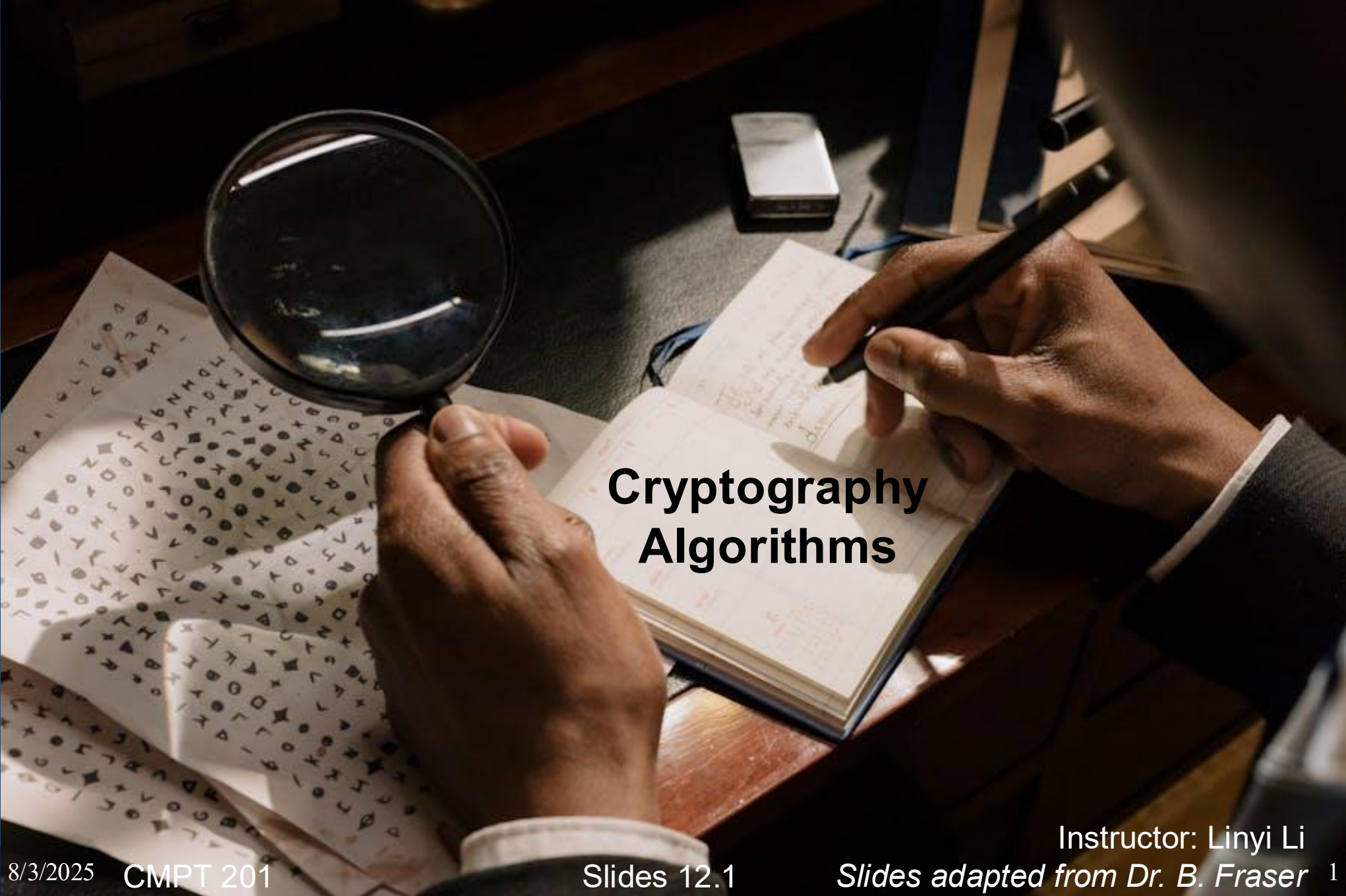


Cryptography Algorithms

Instructor: Linyi Li

Topics

- What is cryptography?
- What are the basics of cryptographic algorithms?
  - What are cryptographic hashes?
  - What is a secret key encryption?
  - What is public-key encryption?

Cryptography:
The absolute basics

Context

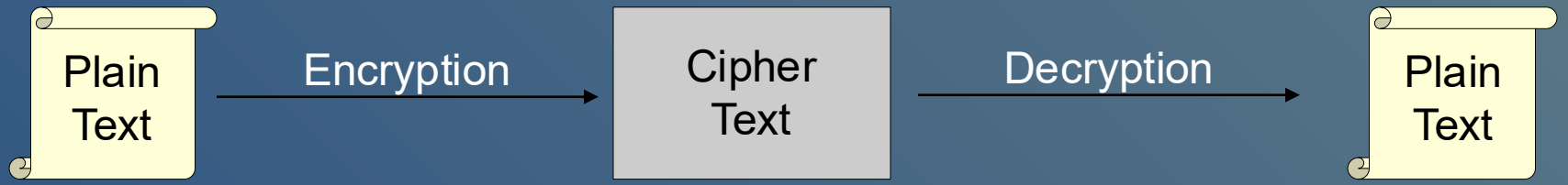
- Cryptography
  - A very broad area.
  - We'll focus on how to *use* cryptography.
  - We just touch on the basics!
- Reference Material
  - Coursera course: Cryptography I
    https://www.coursera.org/learn/crypto

The CIA Model

- CIA model: the classic security model.
  - Confidentiality:
    - information is only disclosed to those authorized to know it.
  - Integrity:
    - only modify information in allowed ways, and only by authorized parties.
  - Availability:
    - those authorized for access are not prevented from it.

- Threat examples
  - Against confidentiality: classified information leak
  - Against integrity: fake images/videos
  - Against availability: Denial-of-Service (DoS) attacks

General Cryptography Process



- Cryptographers invented secret codes to hide messages from unauthorized observers.
- Challenges:
  - How can you hide a message from everyone but the intended recipient?
  - How can the recipient know the message is authentic?

ABCD: Traditional Cryptography

- Traditional Cryptography:
  - Secret codes, which are secret algorithms.
  - E.g., Caesar Cipher: .. shift each letter a certain number of letters down the alphabet.

  For +1 'A' becomes 'B'.

- ABCD: Which of the following is the cipher text from using a 3-letter shift Caesar Cipher on the plain text "Hello world"?

(a)EBIIL TLOIA

(b)KHOOR ZRUOG

(c)IFMMP XPSME

(d)LOWOR LDHEL

- What is the problem with a secret algorithm?
  - When your algorithm (or code book) is compromised,

  .. your code is broken (useless).

Modern Encryption

- Algorithms are Public
  - Keys are secret which provide the security.
  - May be symmetric (secret key) or asymmetric (public key).
- Why is this better?
  - If algorithm or code is secret,
    then if it falls into the wrong hands it means code is useless.
  - If only key is private,
    then if it falls into the wrong hands then
    - it is easy to replace with new key.

Crypto Algorithm Goals

- Choose an encryption algorithm such that:
  - Given a key, it should be
    - relatively easy to encrypt or decrypt a message.
  - Without a key, it should be
    - hard to compute (invert) or decrypt a message.
- Strength of security often based on length of key:

Longer key is more difficult to guess (by brute-force).

Window of Validity

- Window of Validity
  - ..The minimum time to compromise a cryptographic algorithm.
  - –Must only use algorithm that have not been compromised.
- Problem:

  Window of validity of your crypto function
  .. may be shorter than the lifetime of your system.

  –Design systems so you can replace the crypto function easily.
- Example Windows of Validity
  - –1993: SHA-0 was published.
  - –1995: Possible weakness was found in the SHA-0 algorithm; replaced with SHA-1.
  - –2004: Published way to compromise SHA-0
  - –2017: Published way to compromise SHA-1
  - –????: Published way to compromise SHA-256?

Three Types

- Types of cryptography algorithms based on their keys:
  - Zero keys: .. Cryptographic hash functions
  - One key: .. Secret-key functions (symmetric encryption)
  - Two keys: .. Public-key functions (asymmetric encryption)

Cryptographic Hash Functions
(Zero Keys)

Cryptographic Hash Functions

- Suppose we have a cryptographic hash function *h()*
  - It takes a message *m* of arbitrary length as input and
  - .. produces a smaller (short) number *h(m)*
- Toy example:

***h*(*m*) = (*m*^{2}) % 4321**

m	m in hex	h(m)
AAAA	(0x41414141) -->	2242
BBBB	(0x42424242) -->	893
CCCC	(0x43434343) -->	2558
DDDD	(0x44444444) -->	2916
EEEE	(0x45454545) -->	1967
FFFF	(0x46464646) -->	4032
GGGG	(0x47474747) -->	469
HHHH	(0x48484848) -->	4241
IIII	(0x49494949) -->	2385
JJJJ	(0x4A4A4A4A) -->	3543
KKKK	(0x4B4B4B4B) -->	3394
LLLL	(0x4C4C4C4C) -->	1938

Hash Function Properties

- ..Easy to compute:
  - It should be easy to compute *h(m)*
- ..One-way function:
  - Given *h(x)*, it should be difficult to find *x*.
  - i.e., the reverse of *h()* should be difficult to compute.
- ..Weak collision resistance:
  - Given x, it should be difficult to find *x'* where *h(x') == h(x)*
  - i.e., Given a value and a hash function, it should be difficult to find another value that produces the same hash.
- ..Strong collision resistance:
  - It should be difficult to find two messages *x* and *x'* where *h(x) == h(x')*
  - i.e., given a hash function, it should be difficult to find two values that produce the same hash.

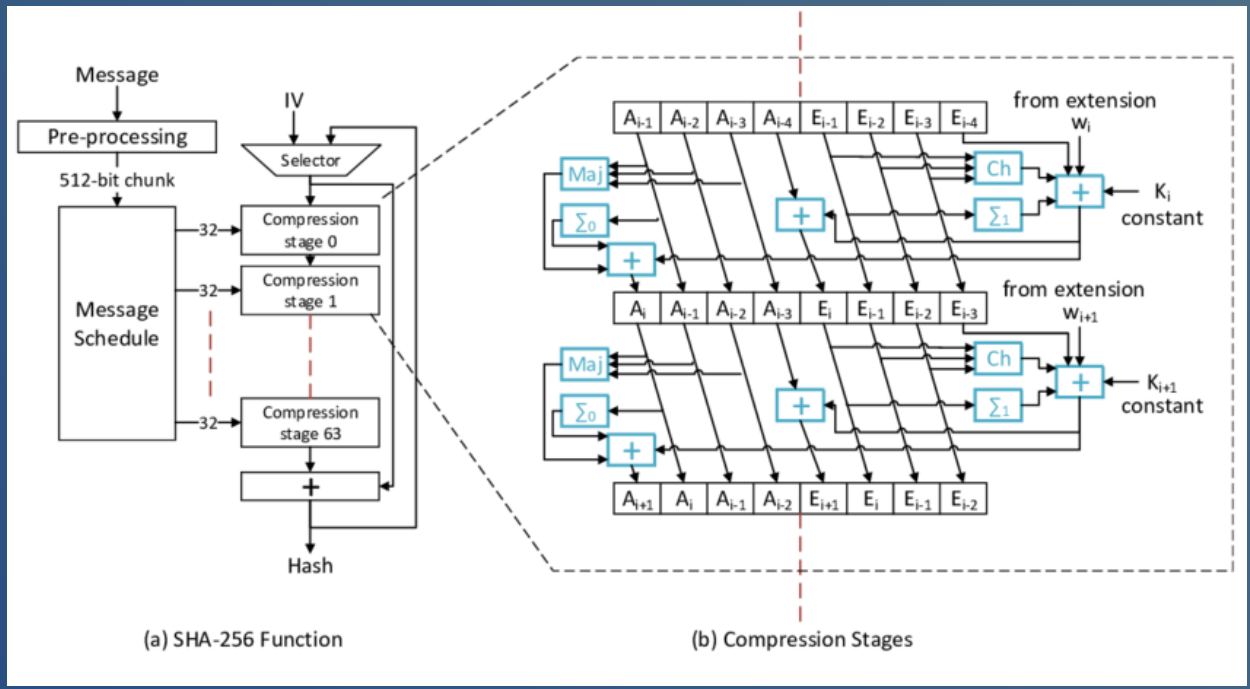
Ideal Hash

- Ideally, we want all these properties

for a strong cryptographic hash function.

–However, not all hash functions provide all these properties.

- Example good crypto hash function: SHA-256.



https://www.researchgate.net/figure/SHA-256-algorithm-block-diagram-a-SHA-256-execution-flow-including-the-preprocessing_fig3_371457507

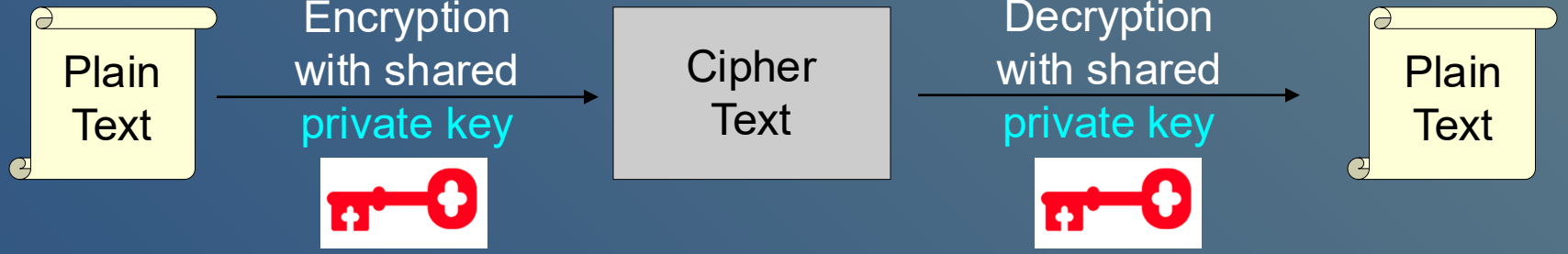
Private Key Cryptography

or

Symmetric Key Cryptography

(One key)

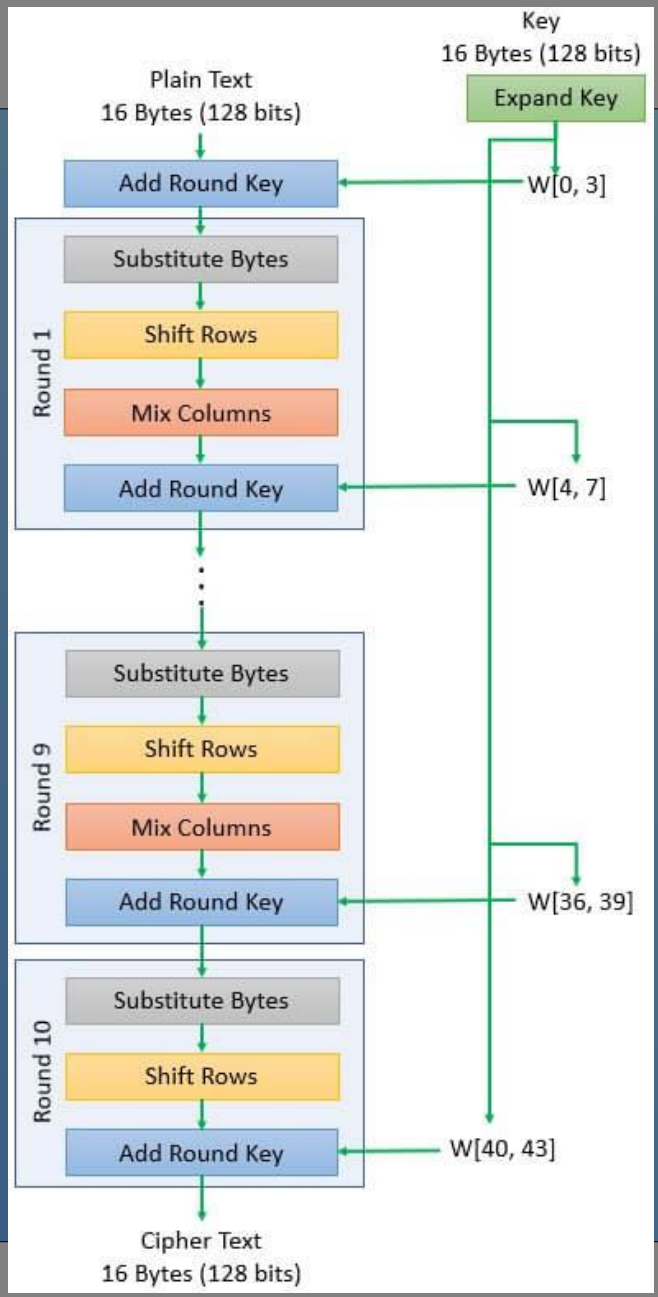
Private (Symmetric) Key Crypto



- One key:
  ..used for both encryption and decryption.
  - .. Requires a secure way to share the secret key!
  - This was the only type of encryption prior to invention of public-key in 1970's.

Private Key Crypto: AES

- AES is an example private key crypto algorithm
  - Need the same key to encrypt and decrypt.



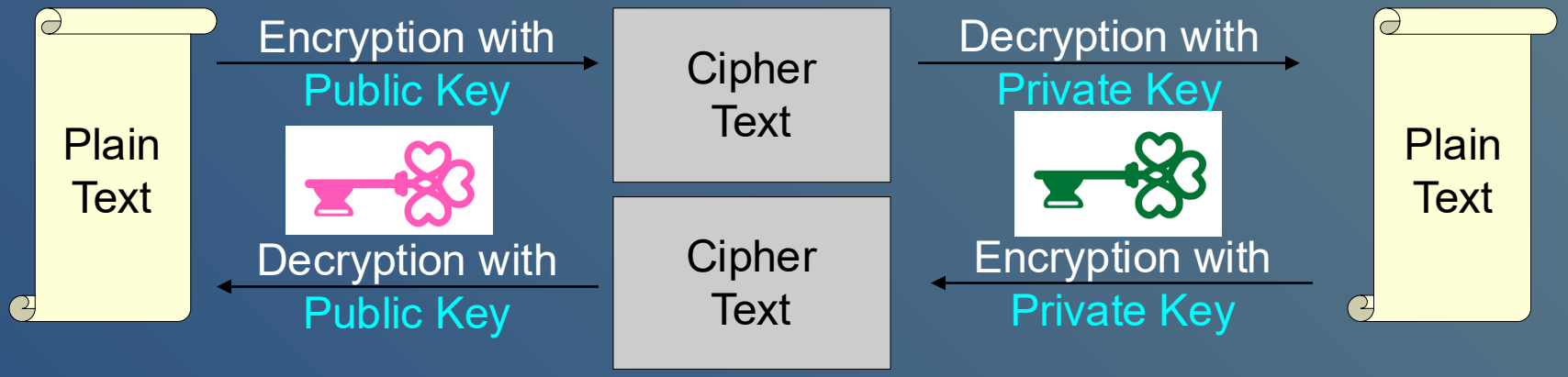
Public Key Crypto

or

Asymmetric Crypto

(Two keys)

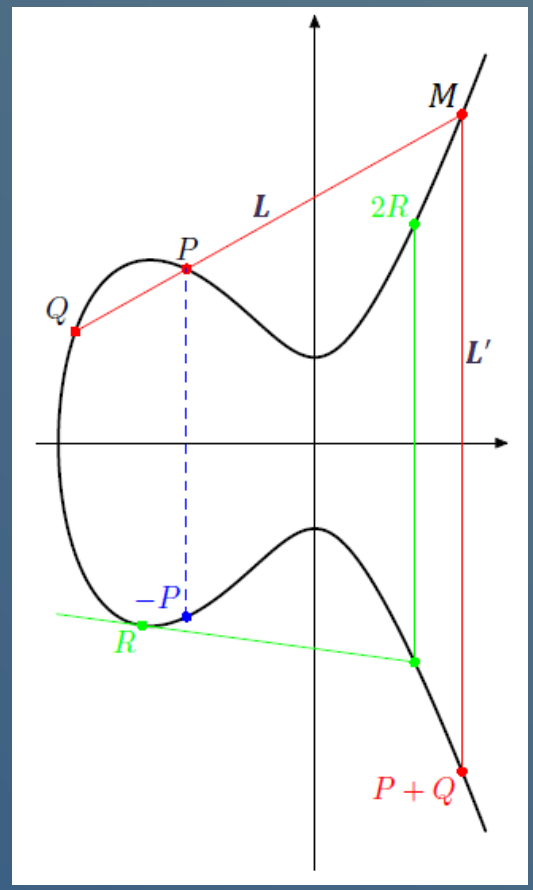
Public Key Crypto (Asymmetric)



- There are two keys:
  - Public key: can be known to anybody
    - Used to encrypt and verify signatures (more below).
  - Private key: .. should be known only to the owner of the key
    - Used to decrypt and sign signatures (more below).
- Fundamental property of public key encryption:
  - .. When encrypted with one key, only the other key can decrypt it.

Generating Keys

- Generating keys:
  - The public and private keys are
  - .. generated together as part of a solution to some very hard to reverse computation.
- Example approaches to generating keys
  - Factoring very large prime numbers,
  - Solving "Twisted Edwards curves" (ed25519)



Keeping Secrets

- Example: Keeping Secrets
  - Alice wants to send a secret message to Bob
  - .. Alice encrypts the plain-text message using Bob's public key
  - Bob decrypts the cipher-text using his private key
- Analysis
  - Since only Bob knows Bob's private key, only Bob can decrypt the cipher-text.
  - Hence Alice and Bob can securely share the message.

Verifying Sender

- Example: Verifying Sender
  - Bob wants Alice to know that he sent a messages and it has not been altered.
  - .. Bob encrypts the plain-text with his private key.
  - Alice decrypts the cipher-text using Bob's public key.
- Analysis
  - Since only Bob knows Bob's private key,
  - .. Bob is the only one who can encrypt a message that can be decrypted with Bob's public key.
  - Alice knows it was Bob who created the message.

Secret and Verified

- Example: Secret and Verified

Combine previous two examples.

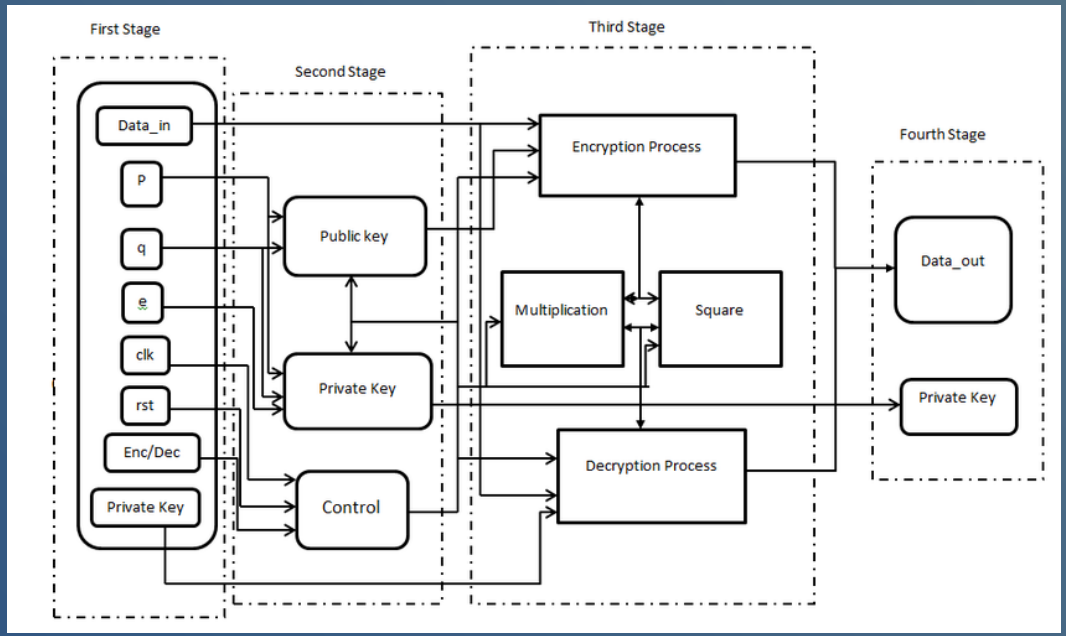
- Alice wants to send a verified, secret message.
- .. Alice encodes a message with her private key.
  - Anyone can decrypt it with her public key.
  - But only she can encrypt with it; so we know she sent it!
- .. Alice encodes the result with Bob's public key
  - Only Bob can decrypt it with his private key.

- Analysis

- Only Bob can decrypt the message (using his private key), and he'll know that only Alice can create it (using her private key).

Public Key

- Benefit:
  - This does not require having
  - .. a secure key distribution mechanism.
  - Lots of other use cases beyond encryption / decryption
- Example algorithm: RSA.



https://www.researchgate.net/figure/RSA-Algorithm-Process_fig3_282249995

Summary

- Cryptography
  - From plain text, create cipher text that others cannot read or change.
- Types of algorithms
  - 0 Keys: Hash function
  - 1 Key: Symmetric encryption (private-key)
    - Both sides know the same secret key.
  - 2 Keys: Asymmetric encryption (public-key)
    - You share a public key with the world.
    - Anyone can encrypt messages for you using this key.
    - Only you can decrypt messages using your secret private key which matches the public key.