Cryptography Basics

bdNOG7

19-22 November 2017

Dhaka, Bangladesh

Cryptography

- All about hiding information in plain sight!

Cryptography Basics

- At its core is the aim to change ordered data into a seemingly random string
  - Using a secret key

$$C = F(P, k)$$

P – plain text
C – cipher text
k – cryptographic key

Terminology

- Encryption/Decryption
  - the method of transforming data (plain text) into an unreadable format (cipher text) & vice-versa
- Plaintext
  - Unencrypted data
- Cipher text
  - the scrambled data after encryption
- Key
  - Information/value used to encrypt or decrypt a message
- Algorithm
  - Rules describing how to encrypt and decrypt messages

Key is the key

- key length is a measure in bits
- key space is the number of possibilities that can be generated by a specific key length

- Example :
  - 2^2 key = a keyspace of **4**
  - 2^4 key = a keyspace of **16**
  - 2^{40} key = a keyspace of **1,099,511,627,776**

Key is the key

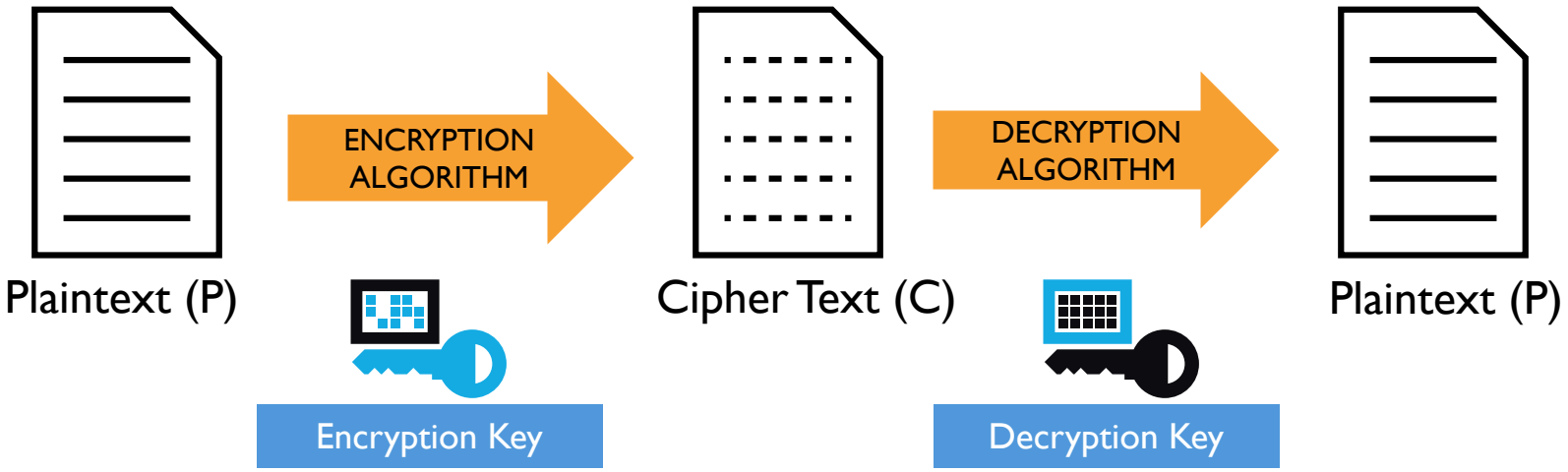
- Assume everyone knows your encryption/decryption algorithm
  - Security of encryption lies in the secrecy of the keys, not the algorithm!
    - Kerckhoff's Principle (1883)

- How do we keep them safe and secure?

Work Factor

- The amount of processing power and time to break a crypto system
  - No system is unbreakable!
- The idea is to make it “expensive” to break

Encryption and Decryption



Symmetric & Asymmetric keys

- Two categories of cryptographic methods
  - Symmetric and Asymmetric key encryption

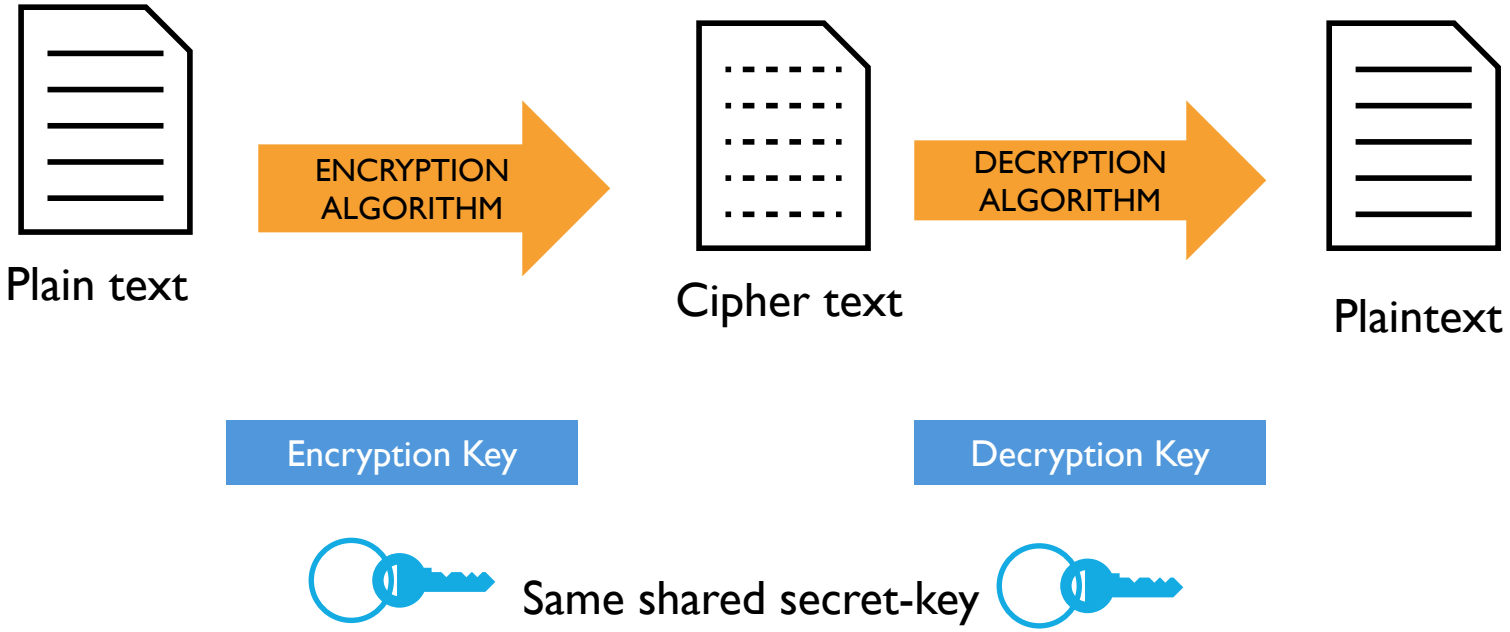
Symmetric Encryption

- **Same key** is used to encrypt and decrypt
  - Both sender and receiver needs to know the key
    - Also called **shared secret-key cryptography**
  - The key must be kept a "secret" to maintain security

- Follows the more traditional form of cryptography (pre 1970)
  - key lengths ranging from 40 to 256 bits

- Widely used examples:
  - DES/3DES, AES, RC4/6

Symmetric Encryption



Symmetric Encryption

- Advantages
  - fast computation since the algorithms require small number of operations

- Disadvantages:
  - The sender and receiver needs to know the shared secret key before any encrypted conversation starts
    - How do we securely distribute the shared secret-key between the sender and receiver?
  - What if you want to communicate with multiple people, and each communication needs to be confidential?
    - How many keys do we have to manage? A key for each!
    - Key EXPLOSION!

Symmetric Key Algorithms

Symmetric Algorithm	Key Size
DES	56-bit keys (8 bits parity)
Triple DES (3DES)	112-bit and 168-bit keys
AES	128, 192, and 256-bit keys
Software Encryption (SEAL)	160-bit keys
RC2	40 and 64-bit keys
RC4	1 to 256-bit keys
RC5	0 to 2040-bit keys
RC6	128, 192, and 256-bit keys
Blowfish	32 to 448-bit keys

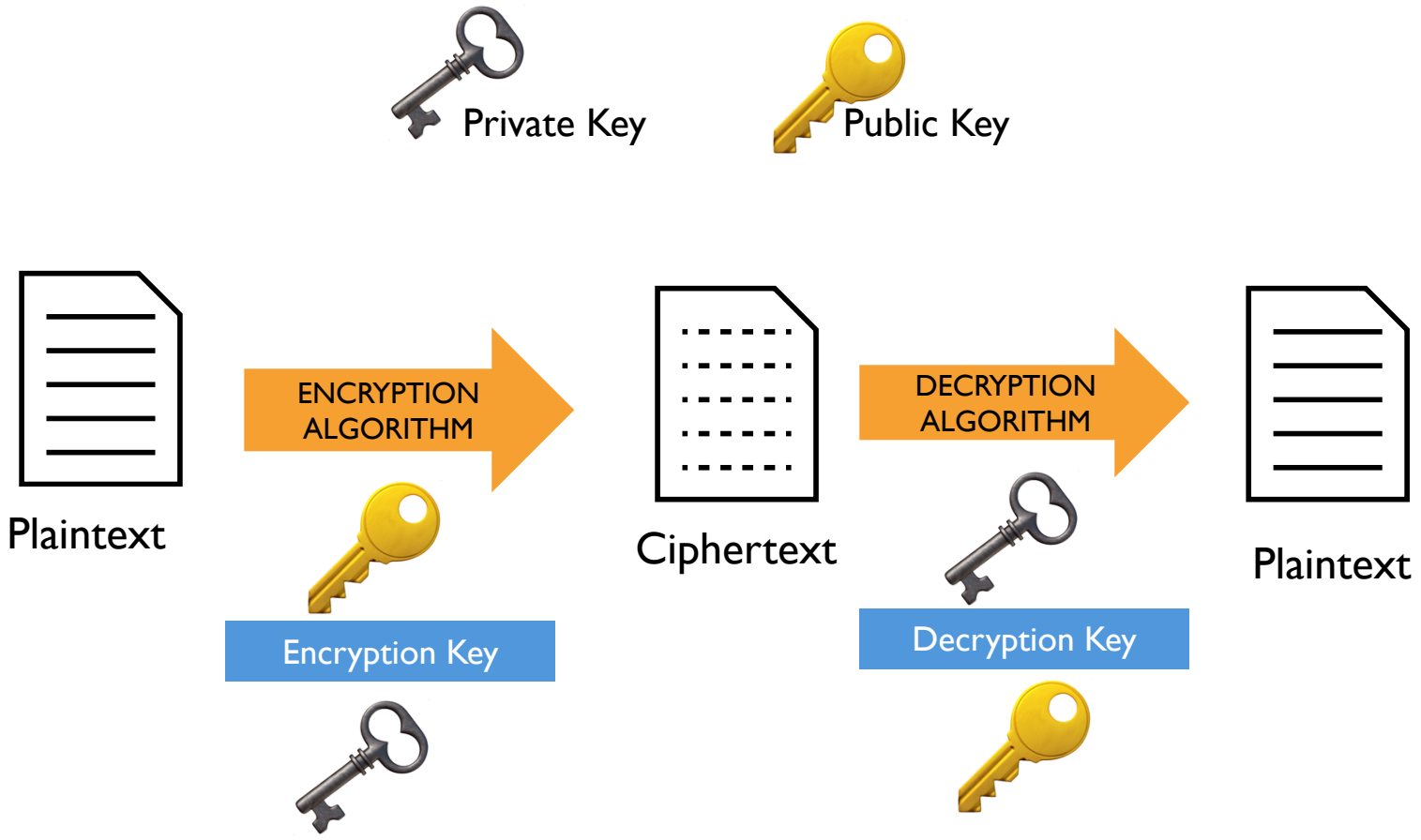
Note:

- Longer keys are more difficult to crack, but more computationally expensive.

Asymmetric Encryption

- Also called **public-key cryptography**
- Use of **Public-Private** key pair
  - The key pairs are mathematically linked
  - Messages encrypted with one key can only be decrypted by the other key of the key pair
- The decryption key cannot, at least in a reasonable amount of time, be calculated from the encryption key and vice-versa

Asymmetric Encryption



Asymmetric Encryption

- Advantages:
  - Solves the key explosion and distribution problem
  - No exchange of confidential information before communication
    - Public key is **published** (everyone knows)
    - Private key is kept ***secret*** (only the owner knows)

- Disadvantages
  - Much slower than symmetric algorithms

Asymmetric Key Algorithms

Algorithm	Key Size (bits)	Description
RSA	512-2048	- Rivest-Shamir-Adlemen - Based on factoring 100 to 200 digit prime numbers - Base on the assumption that while it is easy to compute products of two large numbers, it is very difficult to factor a large number to be a product of two primes
DSA	512-1024	- Digital signature algorithm - Provides capability for authenticating messages
DH	512, 1024, 2048	- Diffie-Hellman - Allows two parties to agree on a key to encrypt messages (used for secret key exchange) - Security based on the assumption that while it is easy to raise a number to a certain power, it is difficult to find out which power was used
ElGamal	512-1024	- Based on DH key agreement - Used in GPG/PGP - Encrypted message becomes twice the size of the original (hence used only for sharing secret keys)
Elliptical curve	160	- Keys are much smaller - Can adapt many algorithms – DH or ElGamal

Hash Functions

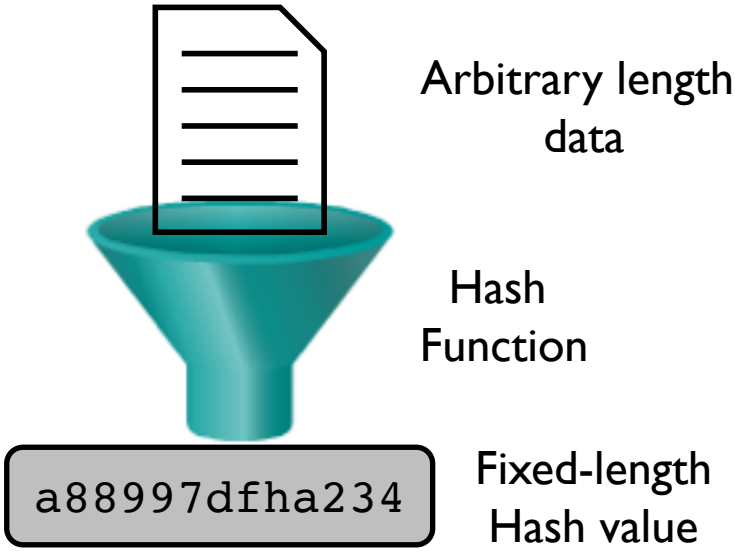
- Takes a message of arbitrary length and outputs a small fixed-length code
  - called the **hash** or **message digest**, or **digital fingerprint**

- One-way mathematical function
  - Easy to compute, difficult to reverse
    - Single bit change in input => large indeterminate change in output

- Uses:
  - Verifying integrity
  - Digitally signing documents
  - Authentication (Hashing passwords)

Hash Functions

- A form of signature that uniquely represents a data



Well-known Hash Functions

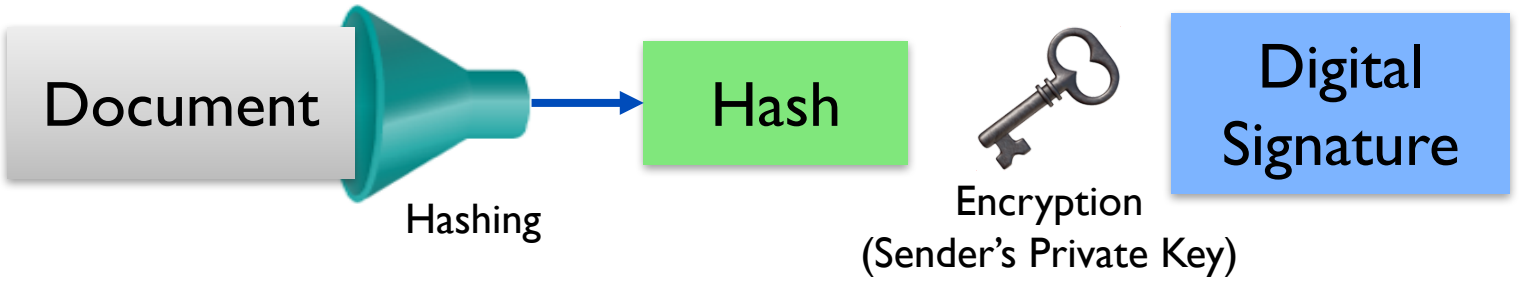
- Message Digest (MD) Algorithm
  - Outputs a 128-bit fingerprint of an arbitrary-length input
  - MD5 is widely-used

- Secure Hash Algorithm (SHA)
  - SHA-1 produces a 160-bit message digest similar to MD5
    - Widely-used on security applications (TLS, SSL, PGP, SSH, S/MIME, IPsec) ☹
  - SHA-256, SHA-384, SHA-512 produce longer hash values

Digital Signature

- Electronic documents can be signed
  - to prove the identity of the sender, and
  - the integrity of the message

- Encrypted hash of the message
  - Hash the data
  - Encrypt the hash with the sender's private key

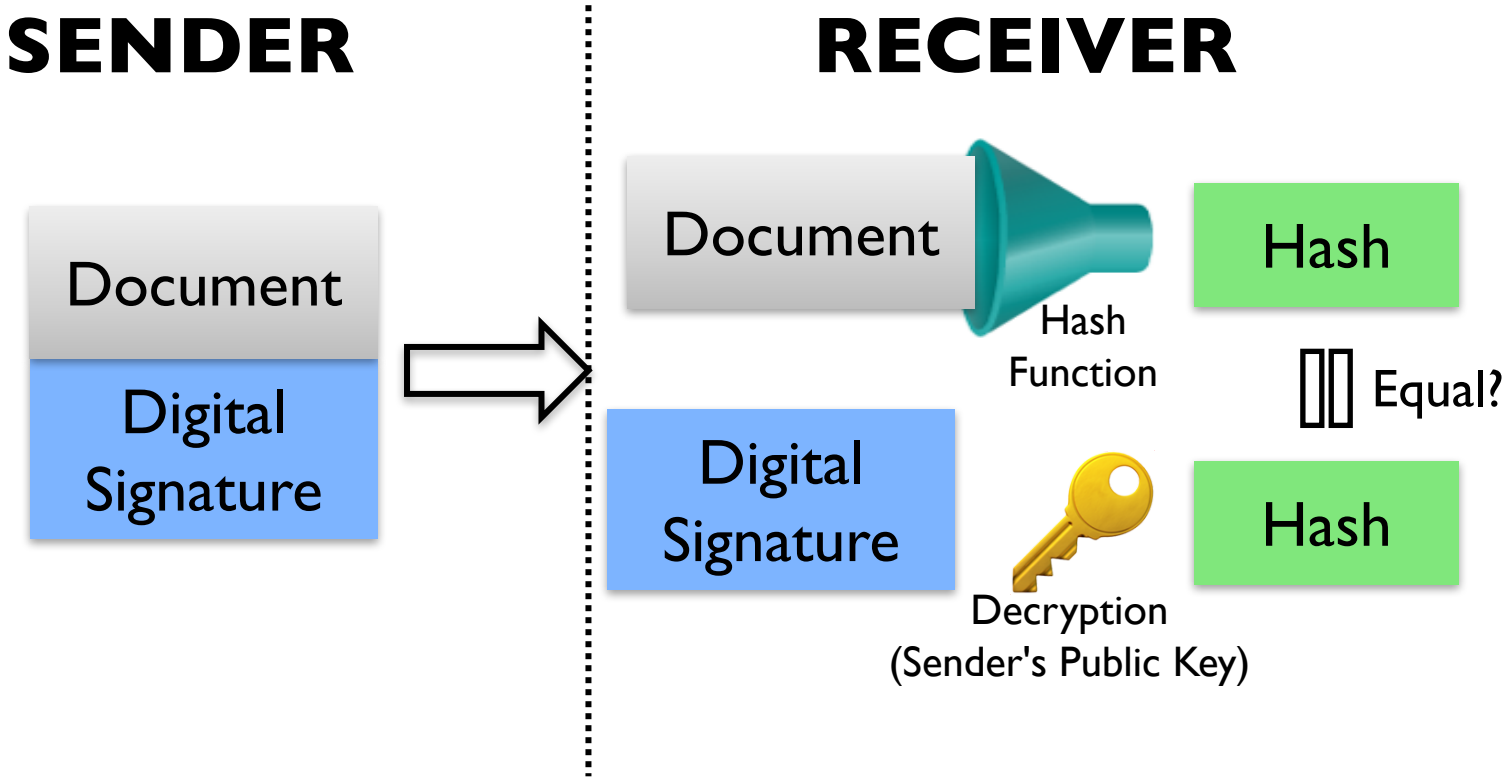


Digital Signature Validation

- Sender
  - Appends the signature to the original document
  - Sends to receiver

- Receiver
  - Computes the hash of the received data
    - Using same hash function
  - Decrypts the encrypted hash (signature) using sender's public key
    - Authentication
  - Compares the hashes
    - If match, the data was not modified (integrity) and signed by the sender

Digital Signature Validation



Example

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

hello
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iQCVAwUBVsDzeDoo7EGKtqONAQKLFQQApJs+32OcZFZUdzQAO6GT8px6+F20CmO/
hInDACZnM2mvKP34J+fdslYyZWaivIhcaYeQsel+yvVjIO5NOLkdpjyOHw0ce99a
kAOP5cvSzw+fxGLkegrM3IhVCHlinKzLswpRCGOWP4xyAi1qaNPoykUzuInvnZ3u
3dVssbaXSN0=
=za1/
-----END PGP SIGNATURE-----
```

https://www.gpg4win.org (MSWIN)
https://www.gpgtools.org (OS X)

PKI Recap

- 🔑 Public Key
- 🗝 Private Key
- 📝 Message

- 📝+🔑 = 🔒✉️ Encrypted
- 🔒✉️+🗝 = 🔓📝 Decrypted
- 📝+🗝 = 🖋✉️ Signed
- 🖋✉️ + 🔑 = 👤 Authenticated


Questions