CSA2 - Security

Encryption / Hashing / Validation

Topics – today

- **Encryption**
  - Symmetrical Encryption
  - Asymmetrical Encryption
  - Common Problems
- **Hashing**
  - File-Validation
  - Flaws
- **Signing & Validation**
- **PGP - Mail Encryption**

Stenography

- **Hiding information inside other information**

Terminology

- Plaintext / Message (M)
- Ciphertext (C)
- Cipher
- Key (K)
- Cryptanalysis
- Pseudo-randomness

Symmetrical Encryption

- *In cryptography, encryption is the process of encoding a message or information in such a way that only authorized parties can access it.*
- **Shared Key**
- **Block cipher**
- Stream cipher

History of Encryption

- **Substitution cipher**
  - Caesar cipher
- **Rotor cipher machine**
  - Enigma

M= I AM GROOT

C= J BN HSPPU

K=? K=1

Very simple Example

- **XOR – Encryption**

Message: 110101001011
Key: 101010

M: 110101001011		C: 011111100001
K: 101010101010	**XOR**	K: 101010101010
C: 011111100001		M: 110101001011

Popular Encryption Algorithms

- AES (Rijndael)
- Twofish
- ~~DES~~ / ~~Triple-DES~~
- Serpent

How to compare ciphers

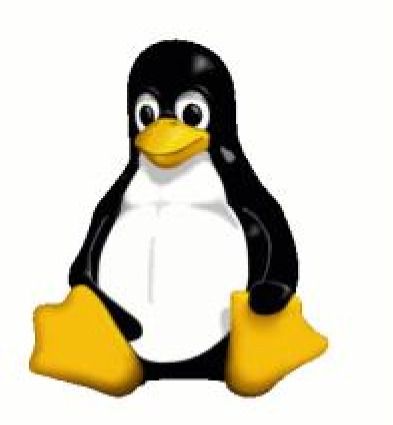
- **Key Size**
- **Block Size**
- **Performance**
  - Rounds
  - Implementation

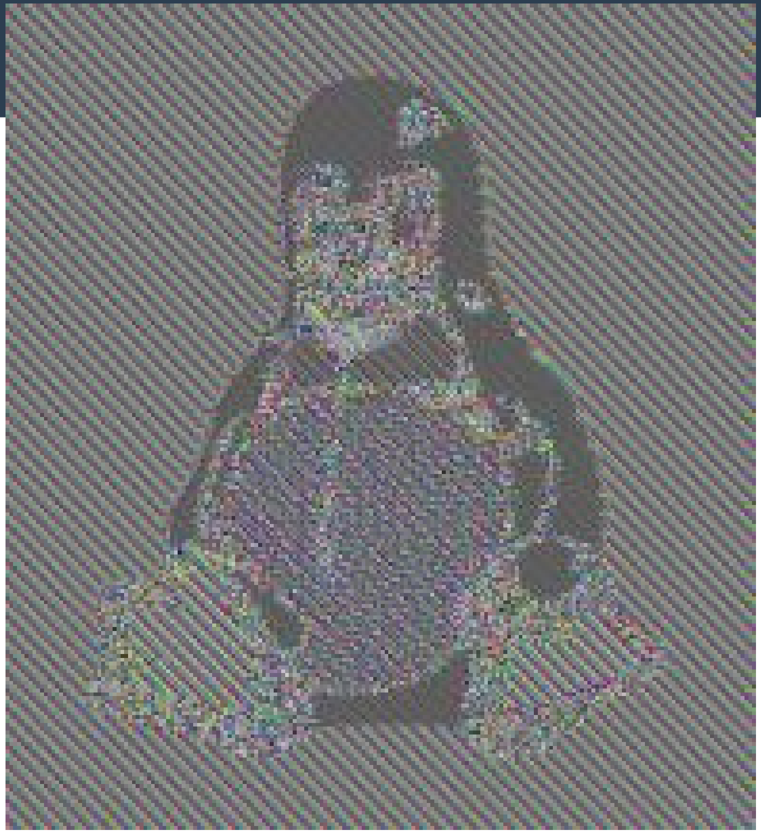
Comparison of different ciphers

	DES	3DES	AES	Twofish	Serpent
Key Size (Bit)	56	168	128 192 256	128 192 256	128 192 256
Block Size (Bit)	64	64	128	128	128
Rounds	16	48	10 12 14	16	32

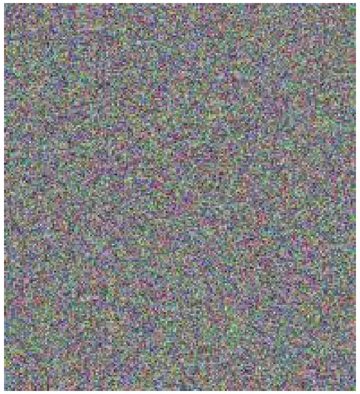
Block cipher operation mode

- Electronic Codebook (ECB) Dangerous!
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)





- **Differences:**
  - Performance
  - Impact of bit errors
  - Complication of cryptanalysis



Problems?

Key Exchange

- Key exchange over the internet
- Safe channel
- Man-in-the-middle

How can we exchange a shared key using an unsafe network while keeping the key secret?

Diffie-Hellman Method

- 1976
- Generate a shared key using mathematical functions
- The key is not sent directly onto the network.
- Used in HTTPS

Asymmetrical Encryption

- **Exponential Encryption**
- **Key consists of two parts**
  - Public Key
  - Private Key
- **One key for encryption**
- **Other key for decryption**

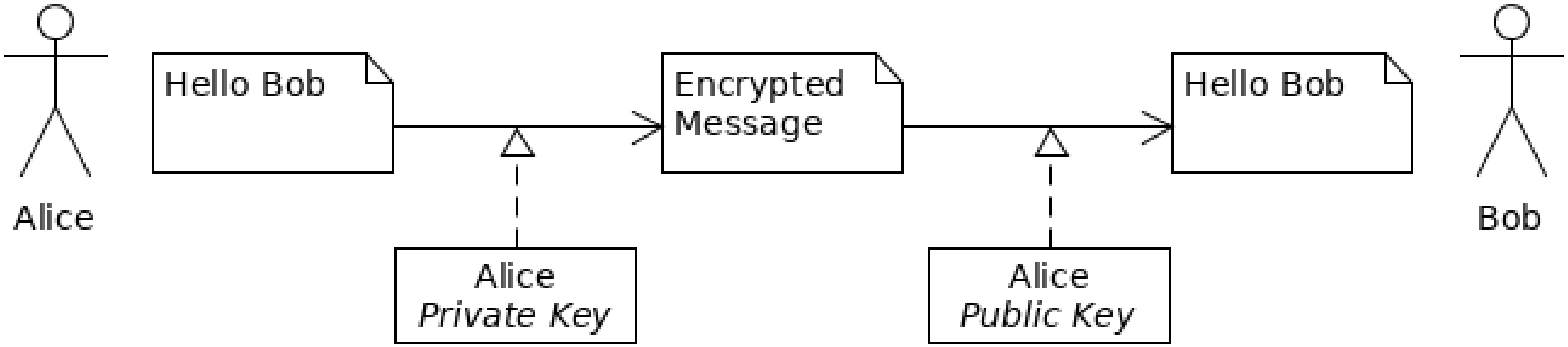
A Message encrypted with one key, can only be decrypted with the other.

RSA - Algorithm

- 1977
- Most common algorithm for asymmetrical encryption.
- Security based on factorization of a product in its prime factors.
- Completely random numbers needed

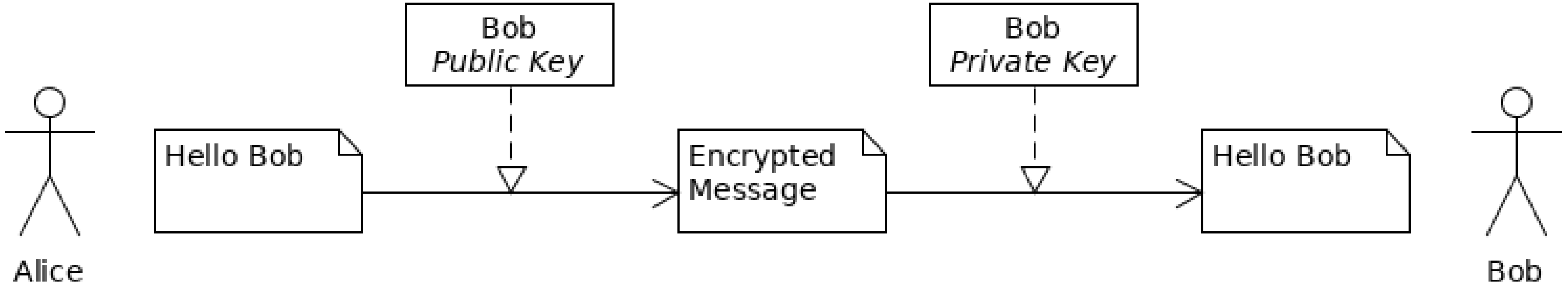
Sign a message

- **How to ensure a message´s author?**



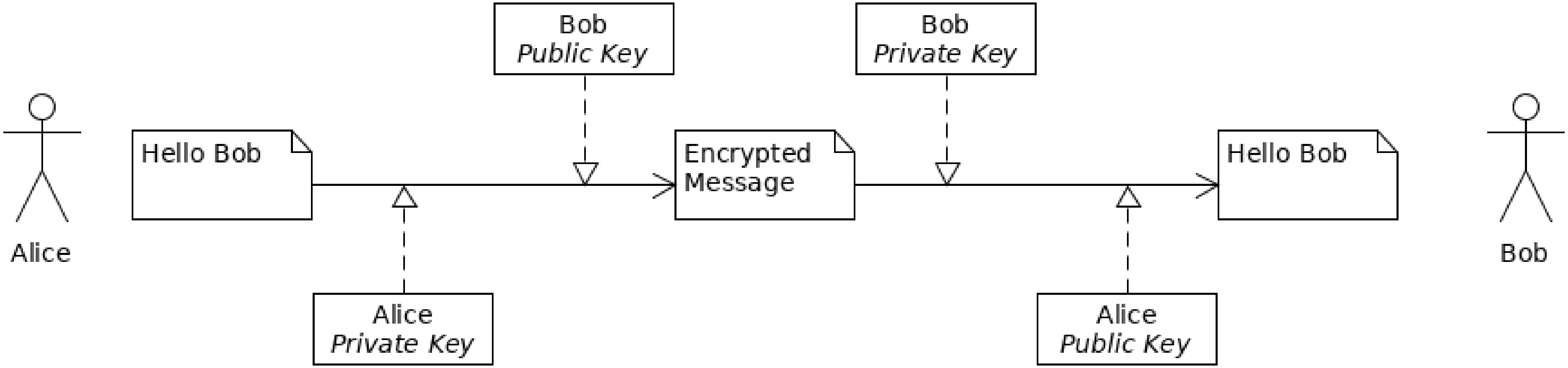
Secure a message

- **How to ensure a message can only be read by the authorized person?**



Sign and Secure a message

- **How can we ensure a message´s author and confidentiality?**



Problems?

Problems!

- Compromised (private) key
- Bad implementations
- Sabotaged algorithm
- Bad random numbers

Random numbers

- **Logical machine cannot produces purely random numbers!**
- **Special chips are using photons**
- **Random number generator can be influenced.**
  - NSA developed random number generator
- **Random number generator can be badly implemented**
  - Happened 2008 to OpenSSL library
- **Cryptographically secure pseudorandom number generator**

Hashing

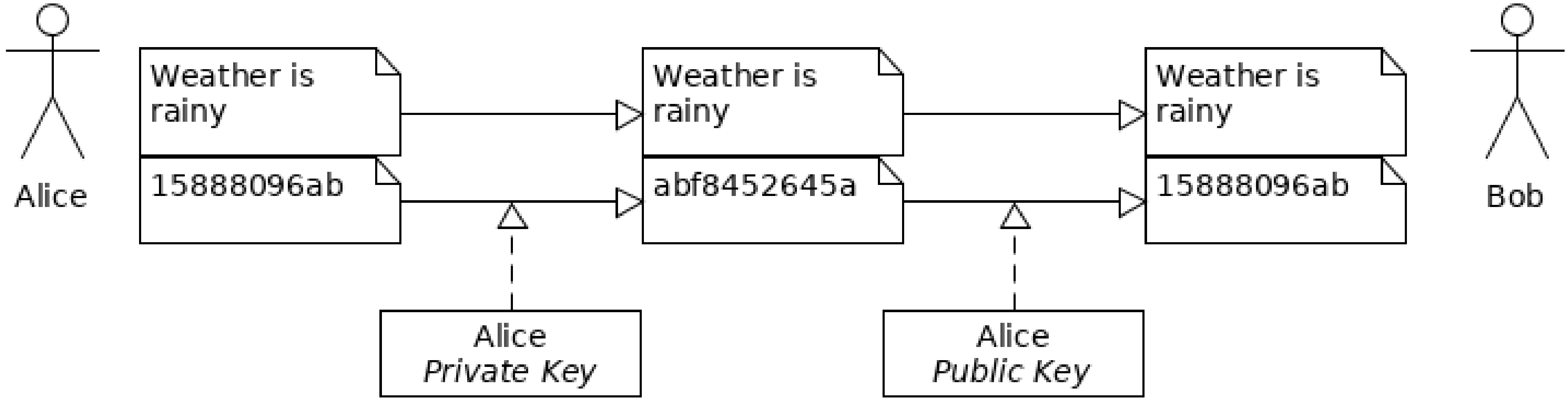
- **A hash function produces from an input of any length a fixed length output.**
- **Characteristics**
  - Speed
  - A small change should end up in a completely different hash
  - Free of collisions

Common Hash Functions

- **~~MD4~~ (1990) → 128 Bits**
- **~~MD5~~ (1991) → 128 Bits**
- **~~SHA-1~~ (1995) → 160 Bits**
- **SHA-2 (2001) → 224/256/384/512 Bits**
- **SHA-3 (2015) → 224/256/384/512 Bits**

Message Signing

- **How to check if a message has been altered by someone else?**



File Validation

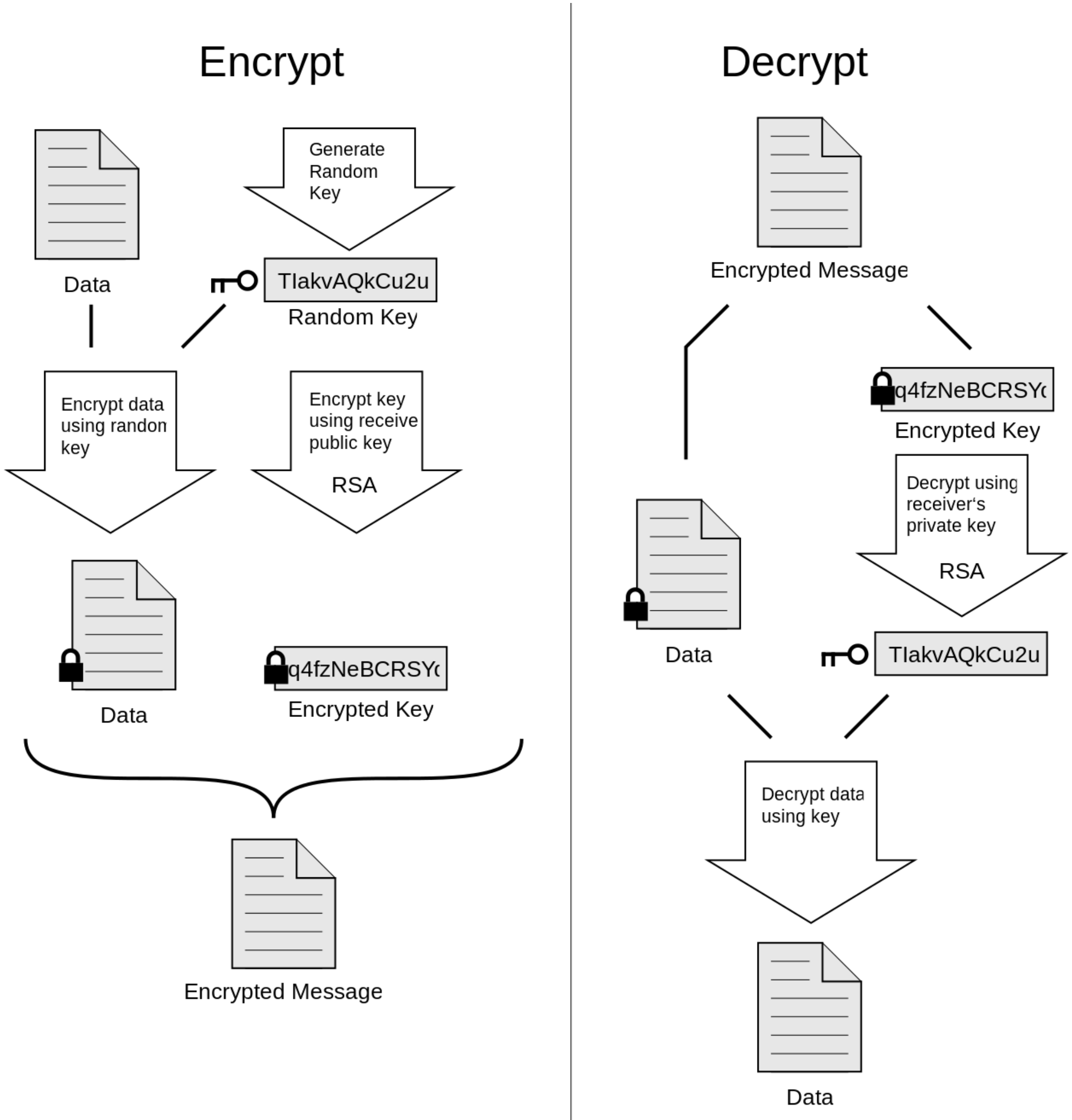
- **How to detect transmission error?**
  - Calculate file hash value and send along with the file.
  - Compare downloaded hash value with source hash
  - Can mostly be downloaded along with the file

http://de.releases.ubuntu.com/17.04/

Flaws of hashing

- **Collision**
  - Attacker can create infected files with valid hash values.
- **Needs safe channel**
  - Hash can be manipulated as well as source file

PGP – Mail Encryption



Encrypt

Data

Generate Random Key

TlakvAQkCu2u

Random Key

Encrypt data using randon key

Encrypt key using receive public key

RSA

Data

q4fzNeBCRSY(

Encrypted Key

Encrypted Message

Decrypt

Encrypted Message

q4fzNeBCRSY(

Encrypted Key

Decrypt using receiver's private key

RSA

Data

TlakvAQkCu2u

Decrypt data using key

Data

PGP – Mail Encryption

- Encrypt Message using random key
- Encrypt key with receivers public key
- Send both via email

- How to you get the receiver´s public key?
- How do you know it is the right one?

How to NOT store passwords

- **As plain text**
  - Obviously very stupid
- **Symmetrical encrypted**
  - Will be encrypted in memory
  - Same password leads to same cipher text
- **Hashed**
  - Insecure hash functions
  - Rainbow tables
  - Same passwords = same hashes

How to store passwords

Don’t do it at all!

Salting

- Add a random string to the password
- Store random string along with the password
  - Random string can even be public
- New random string per password!

- **Increases password length**
- **Equal passwords yield in different hashes**
- **Prevents rainbow tables**
- **Only chance: brute force**

Recommendations on Encryption

- Never implement encryption on your own
  - *Mathematician can do it better*
- Use trusted open-source implementation
  - *Many people checked and use it*
- Use modern and safe algorithm
  - Nothing is unbreakable but modern is usually better
- Stick to standards
  - They became standard by passing a lot of audits
- **It is surprisingly easy to do it wrong, so handle it with care.**

Questions?