# A Short Introduction to Cryptography

Carsten Rudolph
Monash University
FIT1047

# Introduction to Cryptography

- **What is cryptography?**
- **Why do we need it**

**Look at three types of algorithms:**

- **Symmetric encryption**
- **Public key cryptography (asymmetric)**
- **Hash functions for security**

# Encryption: Protect communication from eavesdroppers

**Alice**

Bob says:
Hello Alice!

Encrypted:
Hello Alice
-AXFf>$g87FOPxc

**Bob**

Message:
Hello Alice!

**Evil Eavesdropper**

Bob says to Alice:
????
Hello Alice!

# Integrity: Protect communication from changes

**Alice**

Bob says:
I don’t like
you Alice!

←

Encrypted:
-AXFf>$g87FOPxc

←

**Bob**

Message:
Hello Alice!

↕

**Evil Interceptor**

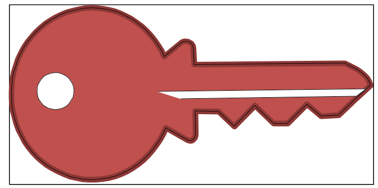
Changed:
XvGc76gFa23.df

# Symmetric Cryptography

A cryptographic key is **shared** between two (or more) principals. Has been used for more than 3000 years.

- **Early example:** Alphabetic substitution (we will try this in the tutorial/lab). CAESAR cipher or Vigenère cipher.
- **Main idea**: Use the shared secret to scramble a message in a way that it cannot be understood without knowledge of the secret.

# Encryption using Symmetric Cryptography



**Alice**

Message:
Hello Alice!

←

←

**Bob**

→

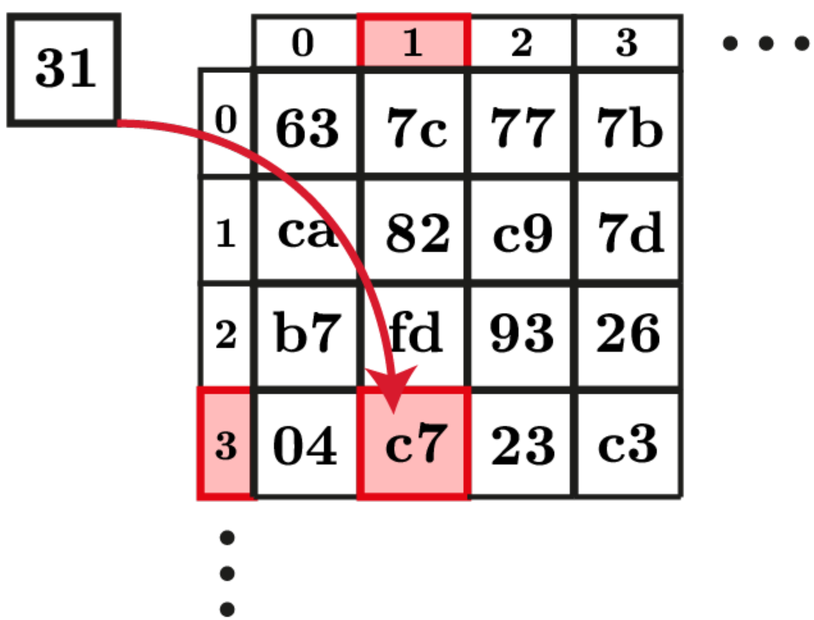
Message:
Hello Alice!

→

Encrypted:
-AXFf>$g87FOPxc

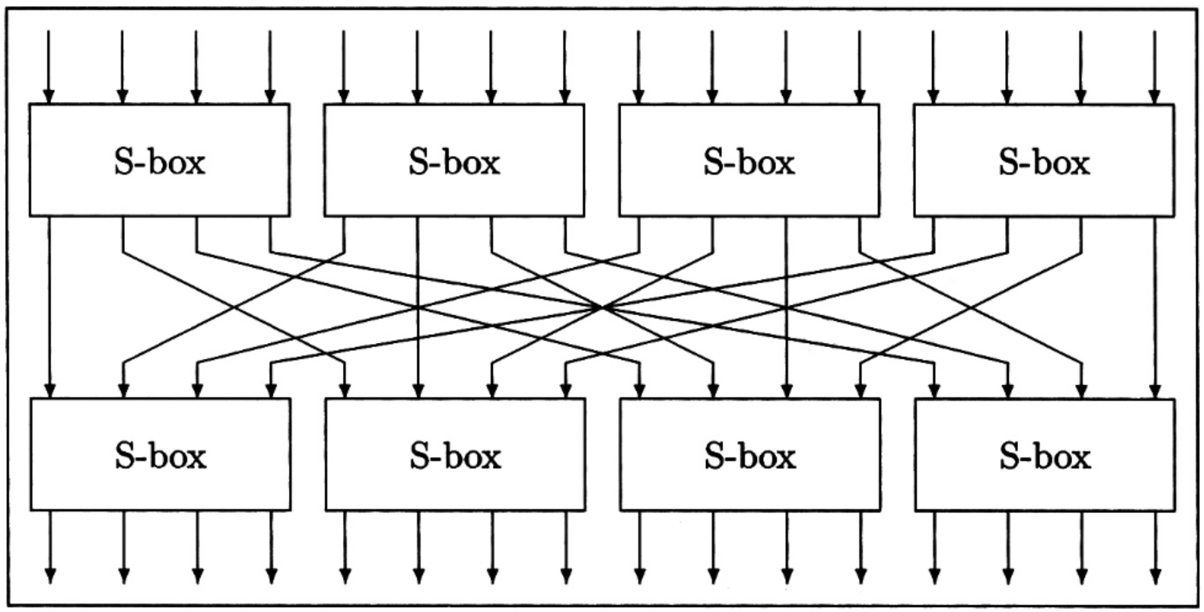
# S-Boxes

Symmetric cryptography often is based on so-called S-Boxes (Substitution Boxes). They work like a look-up table for a part of the message block.



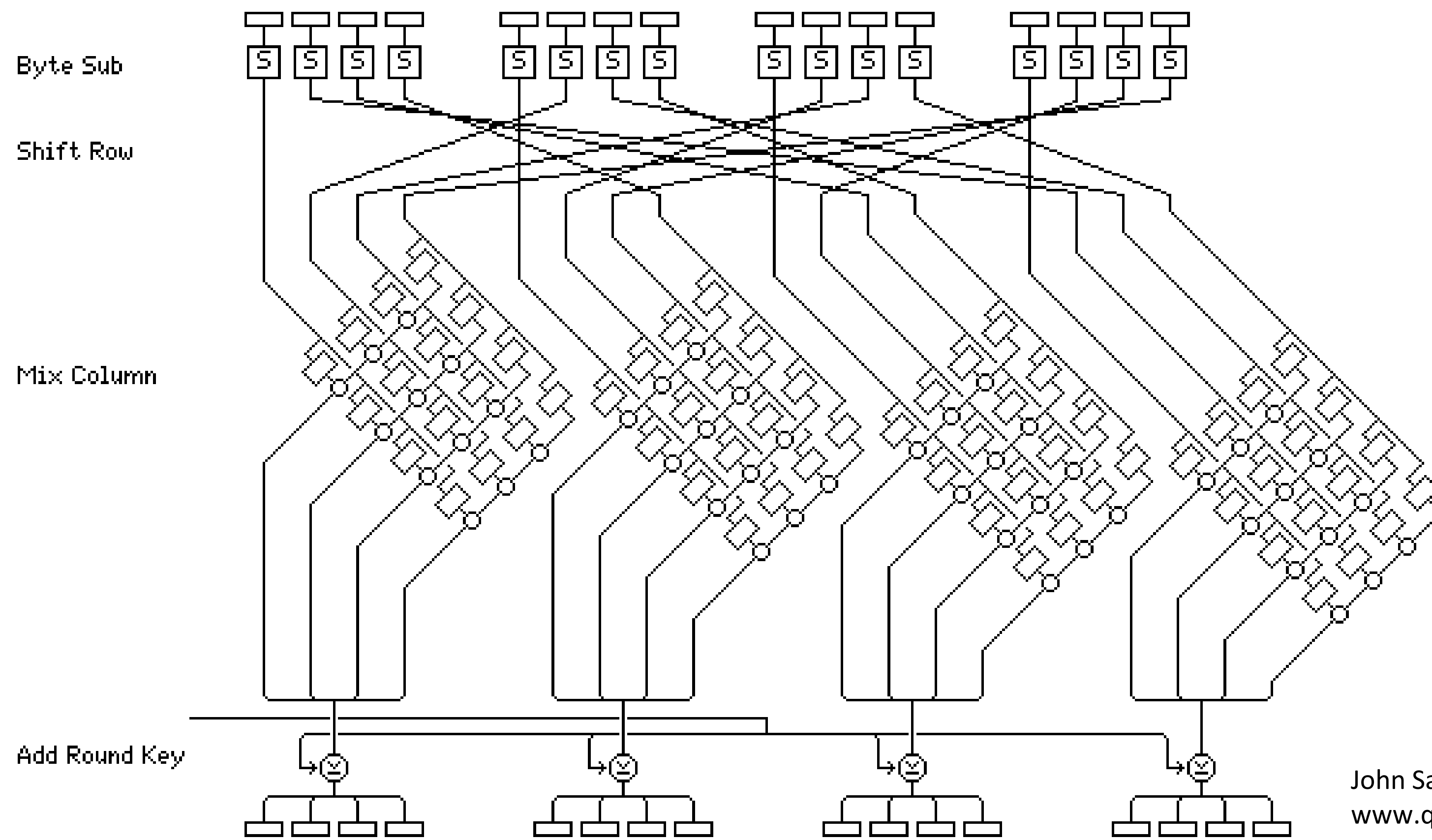
# Permutations

In addition to substitutions (S-Boxes), the order of message parts is changed.



# AES – Advanced Encryption Standard

AES uses 14 cycles in the 256-bit version and each round looks like this (picture shows 128-bit):




John Savard,
www.quadiblog.com

# A modern Algorithm for Symmetric Cryptography: AES

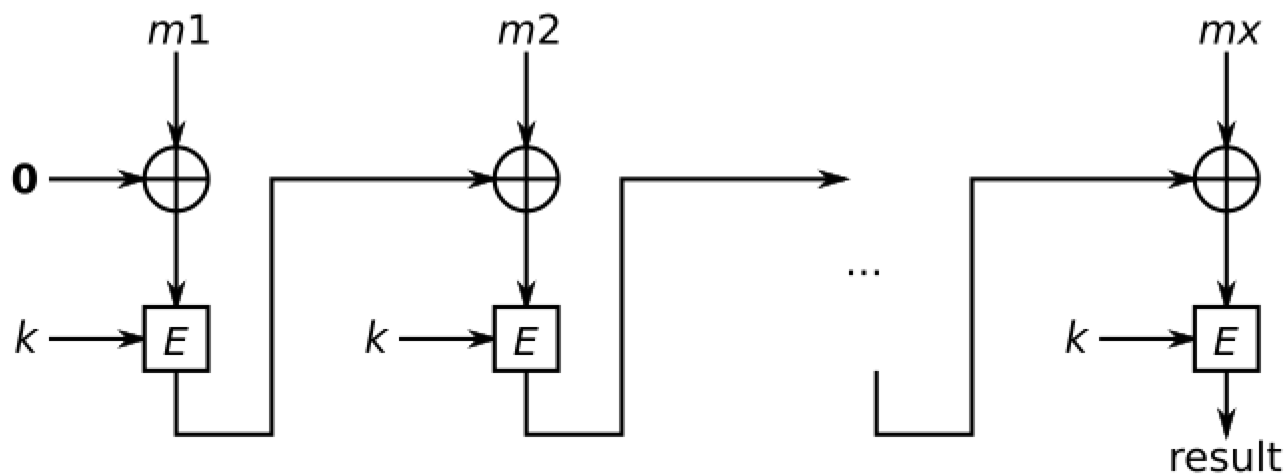
- Open selection process by NIST (National Institute for Standards and Technology, U.S.)
- 15 designs were submitted
- Winner was announced in October 2000
- *Rijndael* developed by two Belgian cryptographers (Joan Daemen and Vincent Rijmen) was chosen to become AES

# Security Properties of Symmetric Encryption (AES)

- AES works on message blocks. It provides confidentiality. Integrity is not straightforward (e.g. change order of blocks, change bits etc.)
- Different types of ***block chaining***
- Start with an initialization vector and then combine each encrypted block with the next block. Thus, blocks in wrong order cannot be decrypted and a changed block will disable decryption of next bock.

# WPA2/CCMP uses AES and a CBC-MAC

- CBC – Cipher Block Chaining
- MAC – Message Authentication Code
- Result is one block that can be used to check integrity of the complete message: *m1 m2 ….mx*



# Disadvantages of Symmetric Cryptography

Symmetric cryptography is very efficient, but has a number of disadvantages:

- **Key distribution**: somehow, one needs to establish a shared secret. An alternative secure channel for key distribution is necessary.
- **Scalability**: Each pair of sender and receiver needs a unique secret key.
  The number of keys grows exponentially with the number of participants (12 participants need 66 keys, 1000 need 499,500 keys and a million participants need an unrealistic 499,999,500,000 keys)
- **Non-repudiation** is not possible.

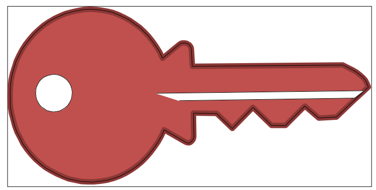
# Public Key Cryptography

- In the early 1970s cryptographers developed the idea of “non-secret encryption”.
- First (public) practically usable schemes were developed in 1976 by Diffie and Hellman (influenced by Merkle)  (known as *Diffie-Hellman Key Exchange*) and in 1978 by Rivest, Shamir and Adleman (known as RSA).
- General idea: Based on a “hard“ mathematical problem and a large random number, a key-pair is generated, such that the private key cannot be derived from the public key without solving the underlying mathematical problem. Every principal owns a unique pair of keys.

# Encryption using Public Key Cryptography

**Alice**

Message:
Hello Alice!

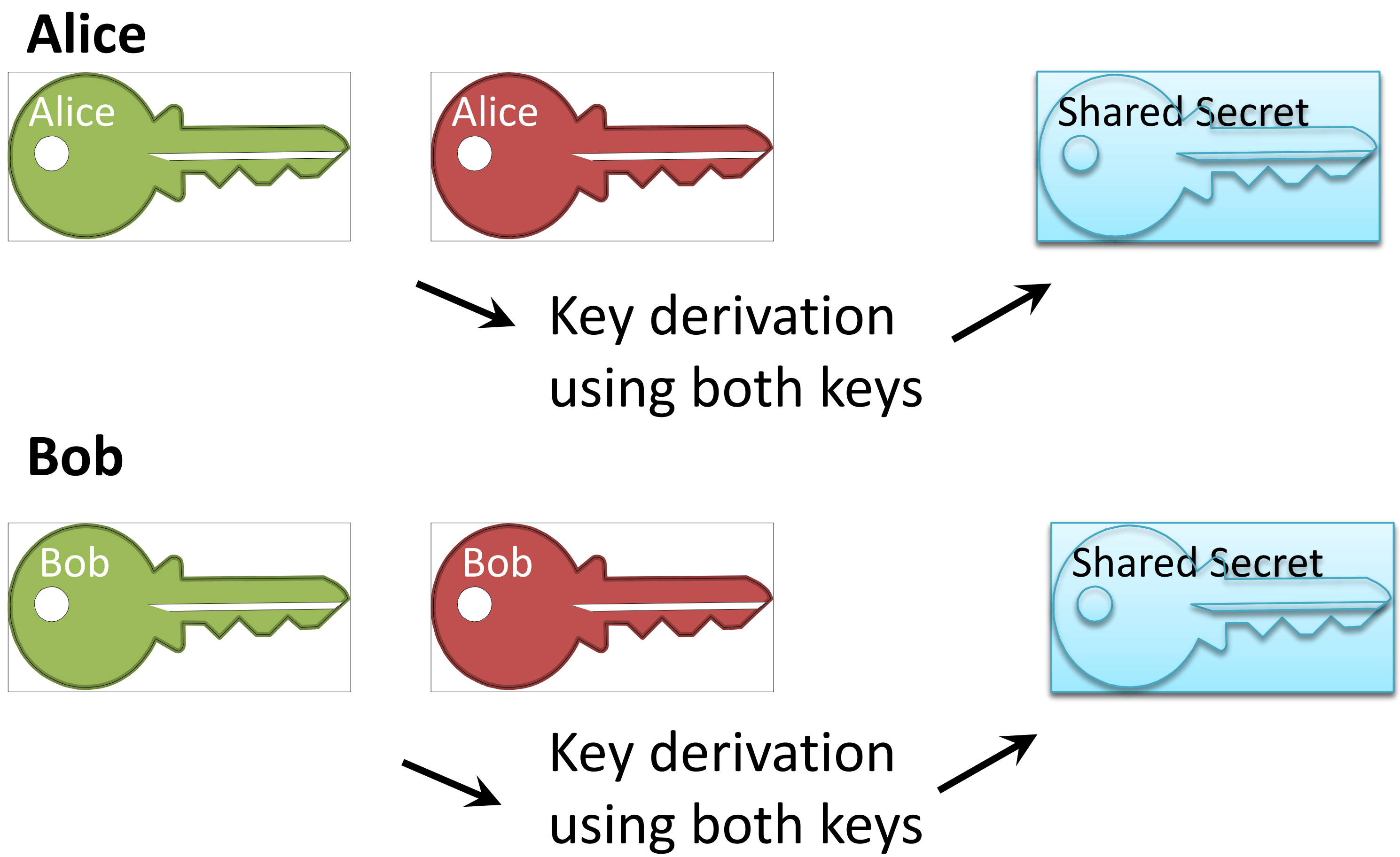


**Bob**

Message:
Hello Alice!

Encrypted:
356jK$^klmGxL1dF

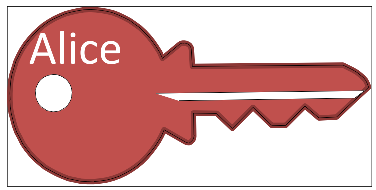
# Key Establishment using Public Key Cryptography



# Digital Signatures / Authenticity

**Alice**


Message: **I agree!**




Message: **I agree!**
Signature: **147JkX78GhC**

Sign message
using private key

**Bob**

Message: **I agree!**
Alice's signature is verified.

Verify signature
using public key

# Other Uses of Public Key Cryptography

Based on the basic mechanisms, many cryptographic protocols and security applications have been developed.

Some examples:

- electronic cash
- non-repudiation protocols
- fair exchange protocols
- electronic voting
- multi-party key agreement

# Example for asymmetric cryptography: RSA

- Developed by Ron Rivest, Adi Shamir and Leonard Adleman.
- First published in 1977.
- Private key *d* public key *e, n.*

Encryption: $cipher = (message)^e \, mod \, n$

Decryption: $message = (cipher)^d \, mod \, n$

- $x \, mod \, n$ means the remainder of $x$ divided by $n$

# Random numbers

- All types of cryptography need random numbers for
  - Key generation
  - Use in protocols to mark messages as new
  - Initialisation vectors
- Many attacks on cryptography have been based on bad random numbers.

# Cryptographic hash functions

- A hash function maps input of arbitrary length to a fixed length output.
- Cryptographic hash functions are infeasible to invert.
- Used in digital signatures, for storing and comparing passwords, in message authentication codes, etc.

# Ideal cryptographic hash functions

Need to have the following properties:

- Computing a hash value for a message needs to be fast and use low resources.
- Given just a hash, it is infeasible to find the original message (except by trying all possible messages)
- Hashes for similar messages should not be correlated (small change in message -> large change in hash)
- Infeasible to find collisions (i.e. two messages with the same hash).

# Some Hash functions

- MD5 was widely used, but is not secure. Sometimes it is still used for integrity protection.
- SHA1 is better, but attacking it is much easier than brute-force. Attacks get more efficient. Is no longer recommended for digital signatures.
- Current recommendations are SHA-256, SHA-384 and SHA-512

# What would you use if you want to protect the integrity of a message?

A. RSA
B. MAC
C. SHA-256
D. SHA-1

Feed RFKIY8

# In symmetric key cryptography, how many keys are used and who knows them?

A. Two keys, one for each side.
B. One key known to both sides.
C. Depends on the actual algorithm.
D. One key pair each, on both sides.

Feed: RFKIY8

# Can public key cryptography keep a message secret?

A. No, that is why it is called *public.*
B. Only if it is protected by an additional MAC.
C. No, public key cryptography can only be used for digital signatures.
D. Yes. Encrypt with the public key. This message can only be decrypted with the correct private key.

Feed: RFKIY8

# Recommended key lengths

- AES (symmetric): Currently, 128 bit is considered secure. Long term recommendations (after 2030) go towards 256 bit.
- RSA (public key): Currently, 2048 bits is considered secure. Some agencies/government bodies recommend 3072 bits after 2020, others after 2030.

- Recommendations from NIST, NSA and the German BSI differ in details.