OPCUG Cryptography 101

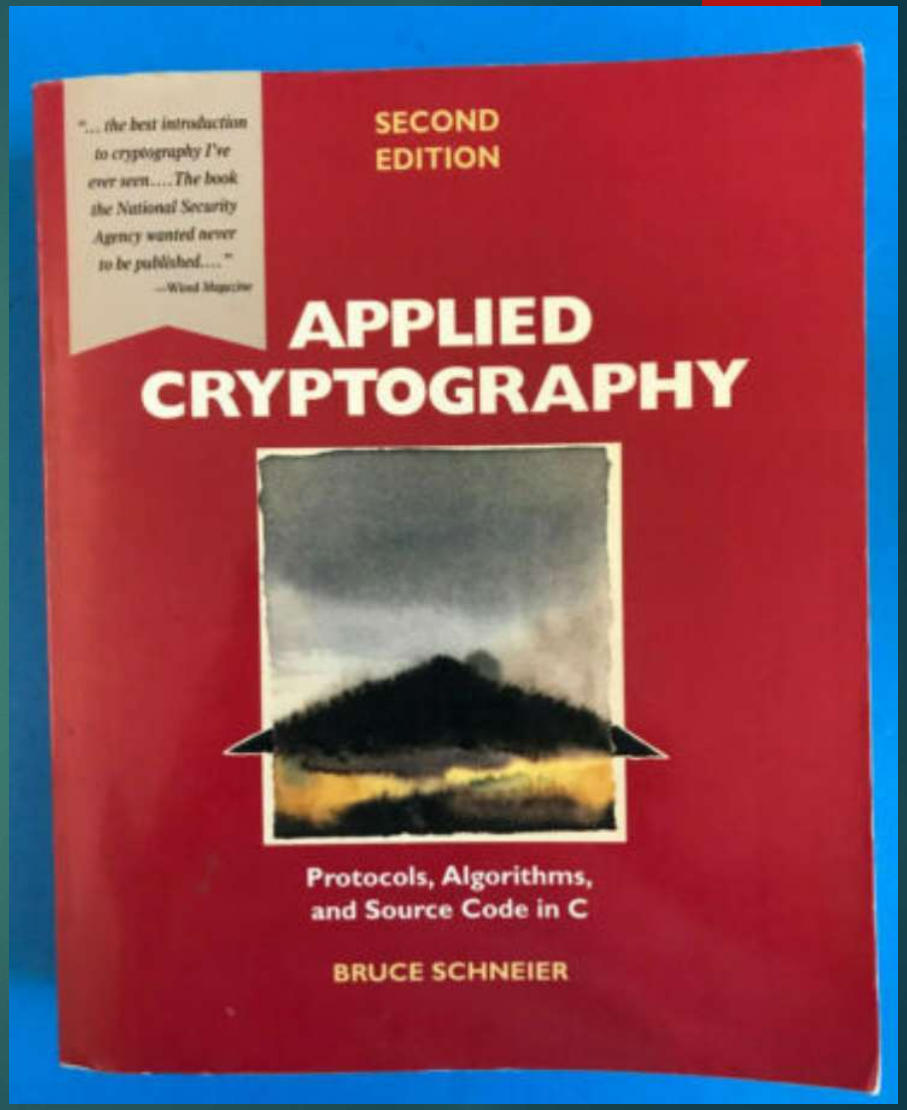
STEPHANE RICHARD

Content

- Sources and references
- Definitions
- Basic cryptology mathematic
- Example of algorithms
- Applications of cryptology
- Quantum Computing and its Impact on Cryptography

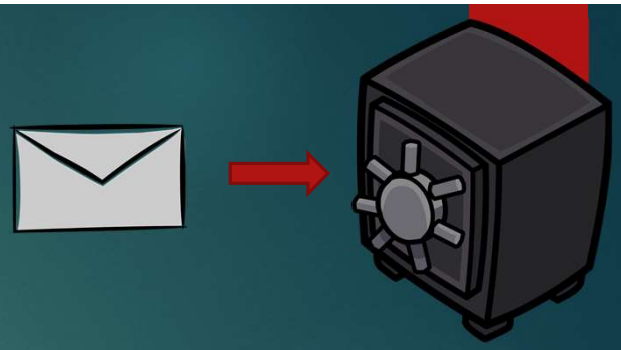
Sources

- Applied Cryptography by Bruce Schneier
- Wikipedia
- Symmetric vs Asymmetric Encryption - 5 Differences Explained by Experts (sectigostore.com)
- Quantum Computing and its Impact on Cryptography (cryptomathic.com)



Definitions

What is security



- Take a letter and lock it in a safe:
  - Hide the safe somewhere - that's not security, it is obscurity
  - Give the safe along with its design specifications and a hundred identical safes with their combinations so that the world's best safecrackers can study the locking mechanism and still can't open the safe and read the letter - that's security
- Same principle can be applied to cryptography:
  - Instead of safe - cryptographic algorithm
  - Instead of combination - key

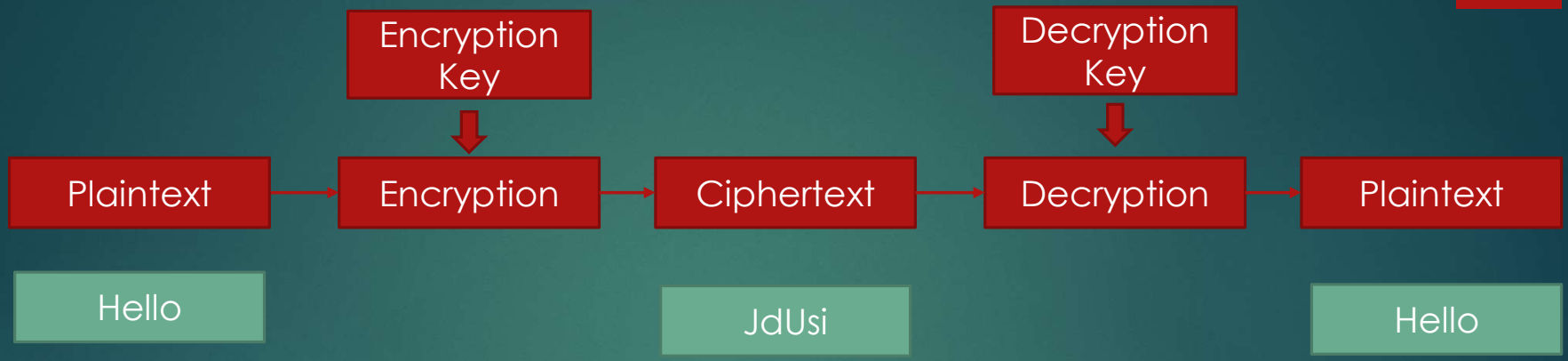
Basic definitions

- **Plaintext**: the message
- **Encryption**: the process of disguising a message in such a way as to hide its substance
- **Ciphertext**: encrypted message
- **Decryption**: the process of turning ciphertext back into plaintext
- **Cryptography**: the art and science of keeping messages secure
- **Cryptanalysis**: the art and science of breaking ciphertext; that is, seeing through the disguise
- **Cryptology**: The branch of mathematics encompassing both cryptography and cryptanalysis

Why cryptography

- **Confidentiality**: The state of keeping or being kept secret or private
- **Authentication**: It should be possible for the receiver of a message to ascertain its origin. An intruder should not be able to masquerade as someone else
- **Integrity:** It should be possible for the receiver of a message to verify that it has not been modified in transit. An intruder should not be able to substitute a false message for a legitimate one
- **Nonrepudiation**: A sender should not be able to falsely deny later that he sent a message

Process



- Encryption and decryption are done via a cryptographic algorithm or cipher
- Encryption/Decryption Key or cryptovariable:
  - Controls the operation or behavior of the cryptographic algorithm
  - Can be private (secret) or public

Two types of key-based algorithms

- Symmetric algorithms:
  - Also called conventional algorithms
- Asymmetric algorithms:
  - Also called Public-key algorithms

Symmetric algorithms

- In most symmetric algorithms, the encryption key and the decryption key are the same or encryption key can be calculated from the decryption key and vice versa
- Require that the sender and receiver agree on an algorithm and a key before they can communicate securely
- **The security of a symmetric algorithm rests in the key:**
  - Divulging the key means that anyone could encrypt and decrypt messages
  - As long as the communication needs to remain secret, the key must remain secret
- Two categories:
  - **Stream algorithms** or **Stream ciphers**: operate on the plaintext a single bit (or sometimes byte) at a time
  - **Block algorithms** or **Block ciphers**: operate on the plaintext in groups of bits. The groups of bits are called blocks

Communications Using Symmetric Cryptography

- Protocol for two parties to communicate securely:
  - Sender and Receiver agree on a cryptographic algorithm
  - Sender and Receiver agree on a key
  - Sender takes the plaintext message and encrypts it using the encryption algorithm and the key. This creates a ciphertext message
  - Sender sends the ciphertext message to Receiver
  - Receiver decrypts the ciphertext message with the same algorithm and key and reads it
- If Bad Guys, sitting between Sender and Receiver, have access to the ciphertext, they must try to cryptanalyze the ciphertext. There are algorithms that are resistant (as far as we know) to whatever computing power Bad Guys could realistically bring to bear on the problem (more on than later)

Communications Using Symmetric Cryptography (Cont'd)

- Bad Guys aren't stupid, though. They also want to listen in on first two steps. Then, they would know the algorithm and the key - just as well as Receiver. When the message comes across the communications channel, all they have to do is decrypt the message
- **A good cryptosystem is one in which all the security is inherent in knowledge of the key and none is inherent in knowledge of the algorithm. This is why key management is so important in cryptography**
- With a symmetric algorithm, Sender and Receiver can select the algorithm in public, but the selection of a key must be done in secret

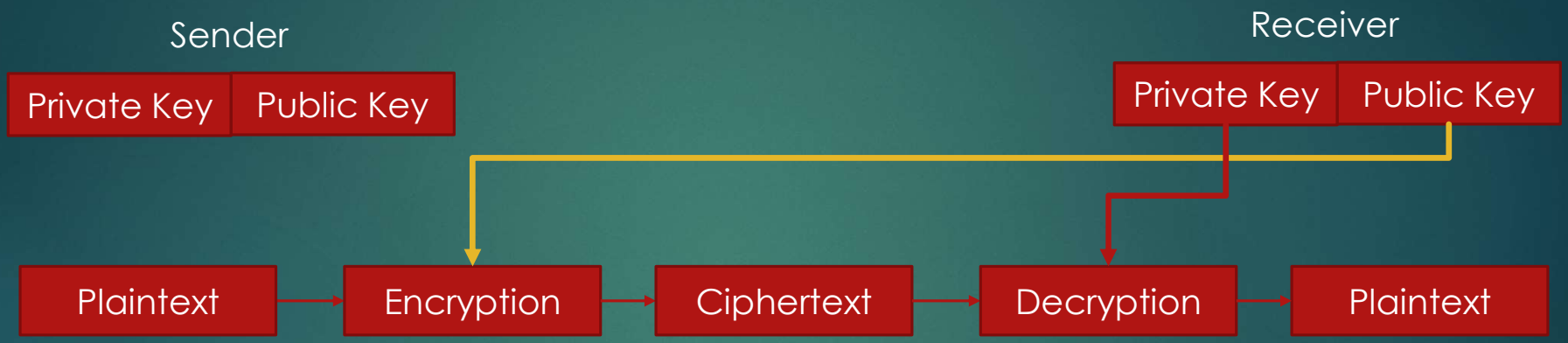
Communications Using Symmetric Cryptography (Cont'd)

- In summary, symmetric cryptosystems have the following problems:
  - Keys must be distributed in secret. They are as valuable as all the messages they encrypt, since knowledge of the key gives knowledge of all the messages. For encryption systems that span the world, this can be a daunting task. Often couriers hand-carry keys to their destinations
  - If a key is compromised (e.g. stolen, guessed, extorted, bribed, etc.), then Bad Guys can decrypt all message traffic encrypted with that key. They can also pretend to be one of the parties and produce false messages to fool the other party
  - Assuming a separate key is used for each pair of users in a network, the total number of keys increases rapidly as the number of users increases. A network of n users requires n(n -1)/2 keys. For example, 10 users require 45 different keys to talk with one another

Asymmetric algorithms

- Key used for encryption is different from the key used for decryption
- The key has two elements generated at the same time:
  - A Public Key which is made public, normally via a Public Key Infrastructure (PKI)
  - A Private Key which is not made public
- Private key cannot (at least in any reasonable amount of time) be calculated from the Public key
- A PKI is a system for the creation, storage, and distribution of **digital certificates** which are used to verify that a particular public key belongs to a certain entity
- A complete stranger can use the Public key to encrypt a message, but only a specific person with the corresponding Private key can decrypt the message

Communications Using Asymmetric (aka Public-Key) Cryptography



- Protocol for two parties to communicate securely:
  - Sender and Receiver agree on a public-key cryptosystem
  - Receiver sends Sender his public key or Sender obtains it from the PKI server
  - Sender encrypts his message using Receiver's public key and sends the ciphertext to Receiver
  - Receiver decrypts Sender's message using his private key

Communications Using Asymmetric (aka Public-Key) Cryptography (Cont'd)

- Notice how public-key cryptography solves the key-management problem with symmetric cryptosystems
- Bad Guys, listening in on the entire exchange, have Receiver's public key and a message encrypted in that key, but cannot recover either Receiver's private key or the message

Symmetric Encryption vs Asymmetric Encryption

Differentiator	Symmetric Key Encryption	Asymmetric Key Encryption
1. Symmetric Key vs Asymmetric key	Only one key (symmetric key) is used, and the same key is used to encrypt and decrypt the message.	Two different cryptographic keys (asymmetric keys), called the public and the private keys, are used for encryption and decryption.
2. Complexity and Speed of Execution	It's a simple technique, and because of this, the encryption process can be carried out quickly.	It's a much more complicated process than symmetric key encryption, and the process is slower.
3. Length of Keys	The length of the keys used is typically 128 or 256 bits, based on the security requirement.	The length of the keys is much larger, e.g., the recommended RSA key size is 2048 bits or higher.
4. Usage	It's mostly used when large chunks of data need to be transferred.	It's used in smaller transactions, primarily to authenticate and establish a secure communication channel prior to the actual data transfer.
5. Security	The secret key is shared. Consequently, the risk of compromise is higher.	The private key is not shared, and the overall process is more secure as compared to symmetric encryption.
Examples of Algorithms	Examples include DES, AES, etc.	Examples include RSA, ECC, etc.

What is a Brute-Force Attack?

- An algorithm is **unconditionally secure** if, no matter how much ciphertext a cryptanalyst has, there is not enough information to recover the plaintext:
  - Only a one-time pad (covered later in the presentation) is unbreakable given infinite resources
- Brute-Force attack:
  - All other cryptosystems are breakable in a ciphertext only attack, simply by trying every possible key one by one and checking whether the resulting plaintext is meaningful
  - Obviously, to do a Brute-Force attack, you need to know which cryptographic algorithm is used
  - There is a 50% chance of finding the key after half of the attempts

Key length in bits – Number of Keys

Key length in bits	Number of Keys - $2^{(\text{Key length in bits})}$	
40	1,099,511,627,776	$1.09951X10^{12}$
56	72,057,594,037,927,936	$7.20576X10^{16}$
64	18,446,744,073,709,551,616	$1.84467X10^{19}$
80	1,208,925,819,614,629,174,706,176	$1.20893X10^{24}$
112	5,192,296,858,534,827,628,530,496,329,220,096	$5.19230X10^{33}$
128	340,282,366,920,938,463,463,374,607,431,768,211,456	$3.40282X10^{38}$

Doubling the key length does not double the number of keys, it squares the number of keys! Example 40 bits (1 Trillion keys) versus 80 bits in the table above.

Average Time Estimates for a Hardware Brute-Force Attack in 1995

		Length of Key in Bits					
Cost (US)	2020 Est.	40	56	64	80	112	128
$100 T	$3 T	2 sec	35 hr	1 yr	70,000 yr	10^{14} yr	10^{19} yr
$1 M	$33 T	.2 sec	3.5 hr	37 d	7000 yr	10^{13} yr	10^{18} yr
$10 M	$333 T	.02 sec	21 min	4 d	700 yr	10^{12} yr	10^{17} yr
$100 M	$3 M	2 millisec	2 min	9 hr	70 yr	10^{11} yr	10^{16} yr
$1 B	$33 M	.2 millisec	13 sec	1 hr	7 yr	10^{10} yr	10^{15} yr
$10 B	$333 M	.02 millisec	1 sec	5.4 min	245 d	10^{9} yr	10^{14} yr
$100 B	$3 B	2 microsec	.1 sec	32 sec	24 d	10^{8} yr	10^{13} yr
$1 T	$33 B	.2 microsec	.01 sec	3 sec	2.4 d	10^{7} yr	10^{12} yr
$10 T	$333 B	.02 microsec	1 millisec	.3 sec	6 hr	10^{6} yr	10^{11} yr

Abb.	Value	Meaning
T	10^{3}	Thousand
M	10^{6}	Million
B	10^{9}	Billion
T	10^{12}	Trillion

Note: Time until our sun goes nova is 10^{9} years

Computer Algorithms

- There are many cryptographic algorithms
- These are the most common:
  - Data Encryption Standard (DES) was the most popular computer encryption algorithm. DES is a U.S. and international standard. It is a symmetric algorithm; the same key is used for encryption and decryption
  - Advanced Encryption Standard (AES) supersedes DES
  - RSA (named for its creators—Rivest, Shamir, and Adleman) is the most popular public-key algorithm. It can be used for both encryption and digital signatures
  - DSA (Digital Signature Algorithm, used as part of the Digital Signature Standard) is another public-key algorithm. It cannot be used for encryption, but only for digital signatures

Basic cryptology mathematic

Modulo operation aka mod

- In computing, the modulo operation returns the remainder of a division, called the modulus of the operation
- Given two positive numbers A and N, A modulo N (abbreviated as A mod N) is the remainder of the division of A by N
- Example:
  - 5 mod 2 would evaluate to 1, because 5 divided by 2 has a quotient of 2 and a remainder of 1
  - 9 mod 3 would evaluate to 0, because the division of 9 by 3 has a quotient of 3 and a remainder of 0
- The range of numbers for a modulo operation of N is 0 to N − 1 inclusive

Exclusive OR (XOR)

OR Truth Table

0 = False 1 = True

Input A	Input B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

XOR Truth Table

Input A	Input B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

OR properties:

A OR 0 = A
A OR A = A
(A OR B) OR B = A OR B

XOR properties:

A XOR 0 = A
A XOR A = 0
(A XOR B) XOR B = A

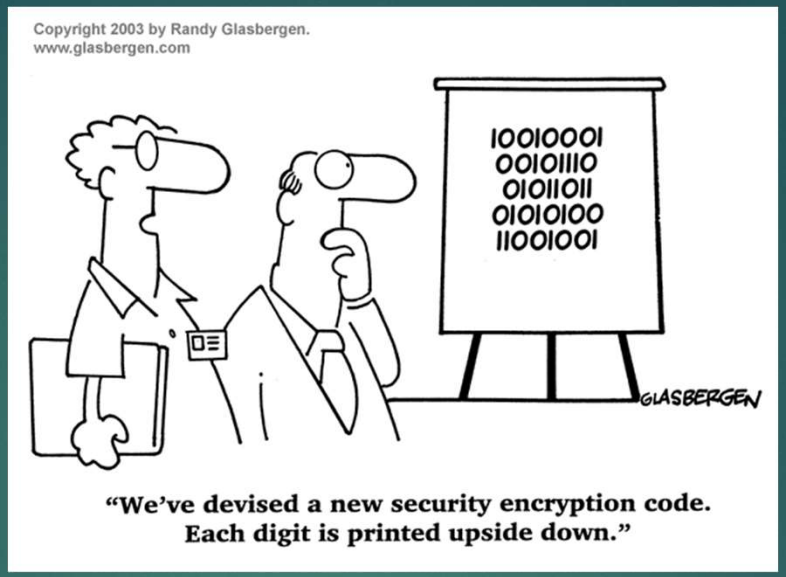
This property is used a lot in cryptography

One-way hash function

- A one-way hash function has many names: compression function, contraction function, message digest, fingerprint, cryptographic checksum, Message Integrity Check (MIC), and Manipulation Detection Code (MDC)
- One-way hash functions are another building block for many cryptographic protocols
- A hash function is a function, mathematical or otherwise, that takes a variable-length input string, called a pre-image, and converts it to a fixed-length (generally smaller) output string called a hash value
- A one-way hash function is a hash function that works in one direction: It is easy to compute a hash value from pre-image, but it is hard to generate a pre-image that hashes to a particular value

One-way hash function (Cont'd)

- A good one-way hash function is also collision-free: It is hard to generate two pre-images with the same hash value
- A single bit change in the pre-image changes, on the average, half of the bits in the hash value
- Think of it as a way of fingerprinting files. If you want to verify that someone has a particular file (that you also have), but you don't want him to send it to you, then ask him for the hash value. If he sends you the correct hash value, then it is almost certain that he has that file. Also used in software distribution:



Example of algorithms

Perfect encryption scheme

- There is one: the one-time pad
- Invented in 1917 by Major Joseph Mauborgne and AT&T's Gilbert Vernam
- One-time pad is nothing more than a large nonrepeating set of truly random key letters, written on sheets of paper, and glued together in a pad
- The sender uses key letter in turn on the pad to encrypt exactly one plaintext character. Encryption is the addition modulo 26 (i.e., number of letters in the alphabet) of the plaintext character and the one-time pad key character
- Each key letter is used exactly once, for only one message

Perfect encryption scheme (cont'd)

- The sender encrypts the message and then destroys the used pages of the pad
- The receiver has an identical pad and uses each key letter on the pad, in turn, to decrypt each letter of the ciphertext and then destroys the same pad pages
- A random key sequence added to a non-random plaintext message produces a completely random ciphertext message and no amount of computing power can change that
- The key letters have to be generated randomly and never be used again, ever

Perfect encryption scheme Examples

Input	Key Letter	Output	Notes
ONETIMEPAD	TBFRGFARFM	HOJKOREGFP	(O(14) + T(19)) mod 26 = 33 mod 26 = H (7) (N(13) + B(1)) mod 26 = 14 mod 26 = O(14) etc.
HOJKOREGFP	TBFRGFARFM	ONETIMEPAD	Using the same key letter for decoding, by subtracting instead adding before the mod operation, yields the original text: (H(7) – T(19)) mod 26 = -12 mod 26 = 14 mod 26 = O(14) (O(14) – B(1)) mod 26 = 13 mod 26 = N(13) etc.
HOJKOREGFP	(key guesses) POYYAEAAZX BXFGBMTMXM ZGYHDUJISK	SALMONEGGS GREENFLUID IILDLXVYNF	Since every plaintext message is equally possible, even with a brute force attack, there is no way for the cryptanalyst to determine which plaintext message is the correct one. The number of possible keys is $26^{\text{(message length)}}$, 10 characters so 141,167,095,653,376 keys

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Perfect encryption scheme (cont'd)

- The idea of a one-time pad can be easily extended to binary data:
  - Instead of a one-time pad consisting of letters, use a one-time pad of bits
  - Instead of adding the plaintext to the one-time pad, use XOR
  - To decrypt, XOR the ciphertext with the same one-time pad
  - Everything else remains the same and the security is just as perfect
- This all sounds good, but there are a few problems:
  - Since the key bits must be random and can never be used again, the length of the key sequence must be equal to the length of the message
  - Need to be able to destroy the bits already used

Algorithm building block - substitution-box (S-box)

- A substitution-box (S-box) is a basic component of symmetric key algorithms which performs substitution
- In general, an S-box takes some number of input bits, m, and transforms them into some number of output bits, n, where n is not necessarily equal to m
- A S-box can be implemented as a lookup table
- Fixed tables are normally used, as in the Data Encryption Standard (DES), but in some ciphers the tables are generated dynamically from the key

Example DES S-Box Table Round 5 - mapping 6-bit input into a 4-bit output

Input 011011
Output 1001

S_5		Middle 4 bits of input															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

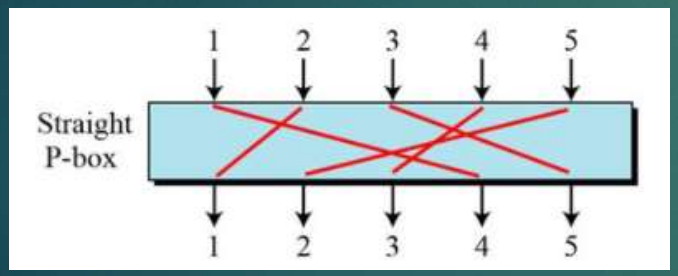
Example of Substitution- Caesar cipher

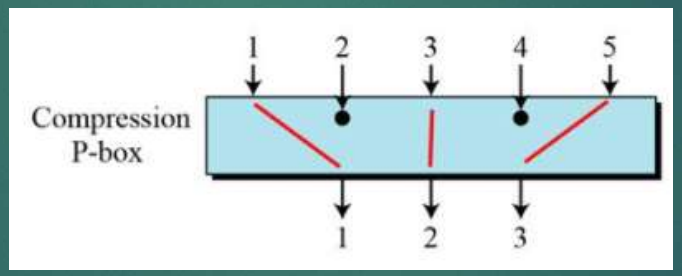
- The method is named after Julius Caesar, who used it in his private correspondence
- Caesar cipher is one of the simplest and most widely known encryption techniques
- It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet
- The Caesar cipher is easily broken and in modern practice offers essentially no communications security
- Caesar cipher using a left rotation of three places, equivalent to a right shift of 23
- The shift parameter can be used as the key for a more generic cipher

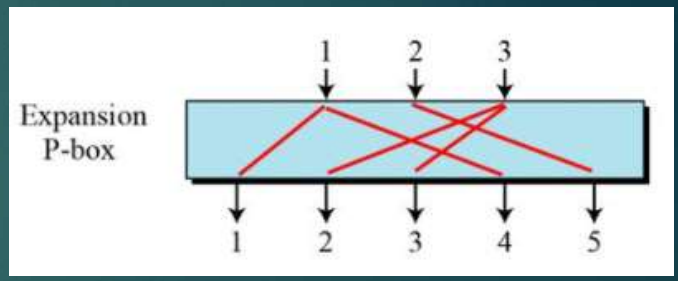
Plain	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipher	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W

Algorithm building block - Permutation-box (P-box)

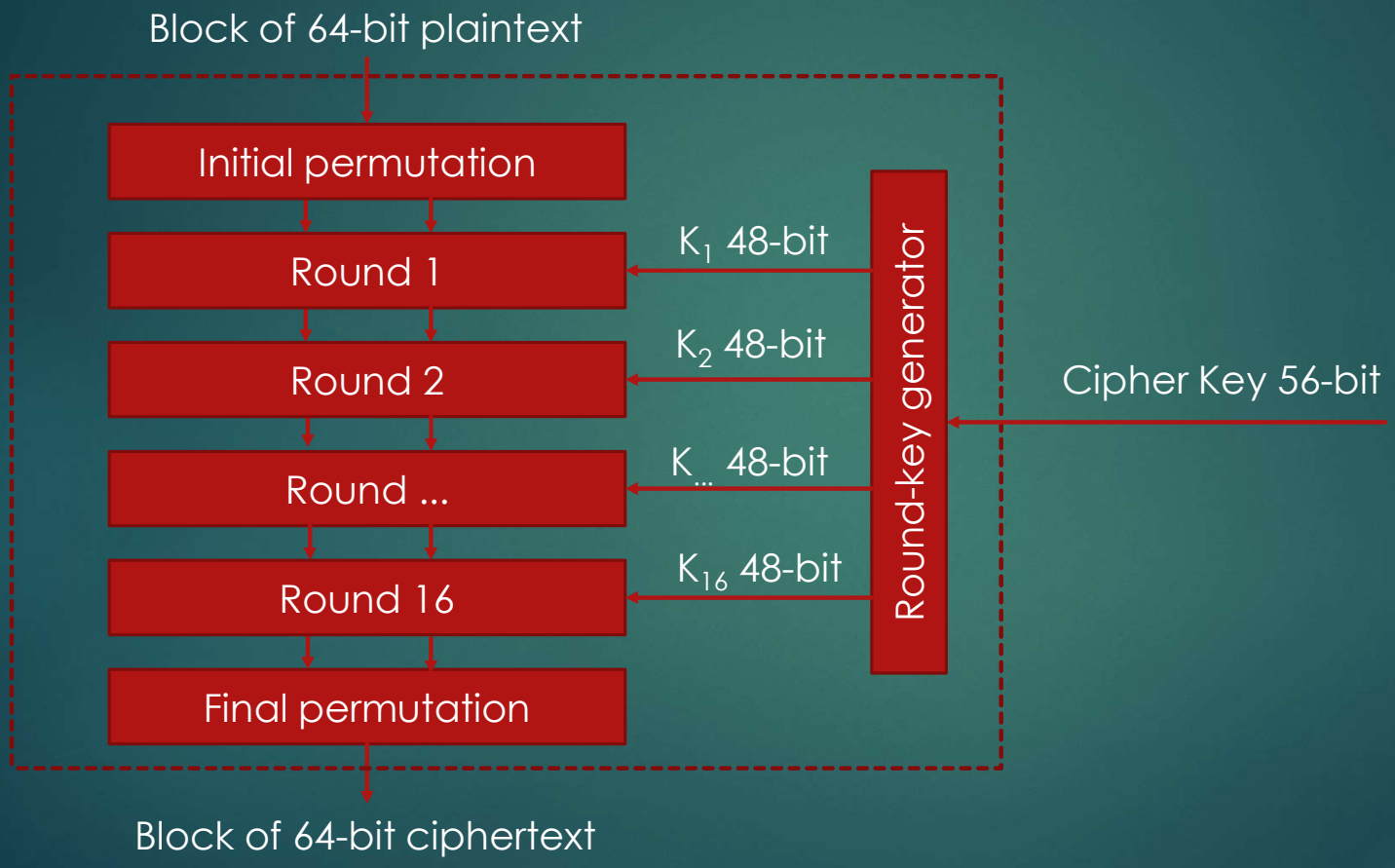
- A permutation box (P-box) is a method of bit-shuffling used to permute or transpose bits across S-boxes inputs
- 3 types of permutation boxes:







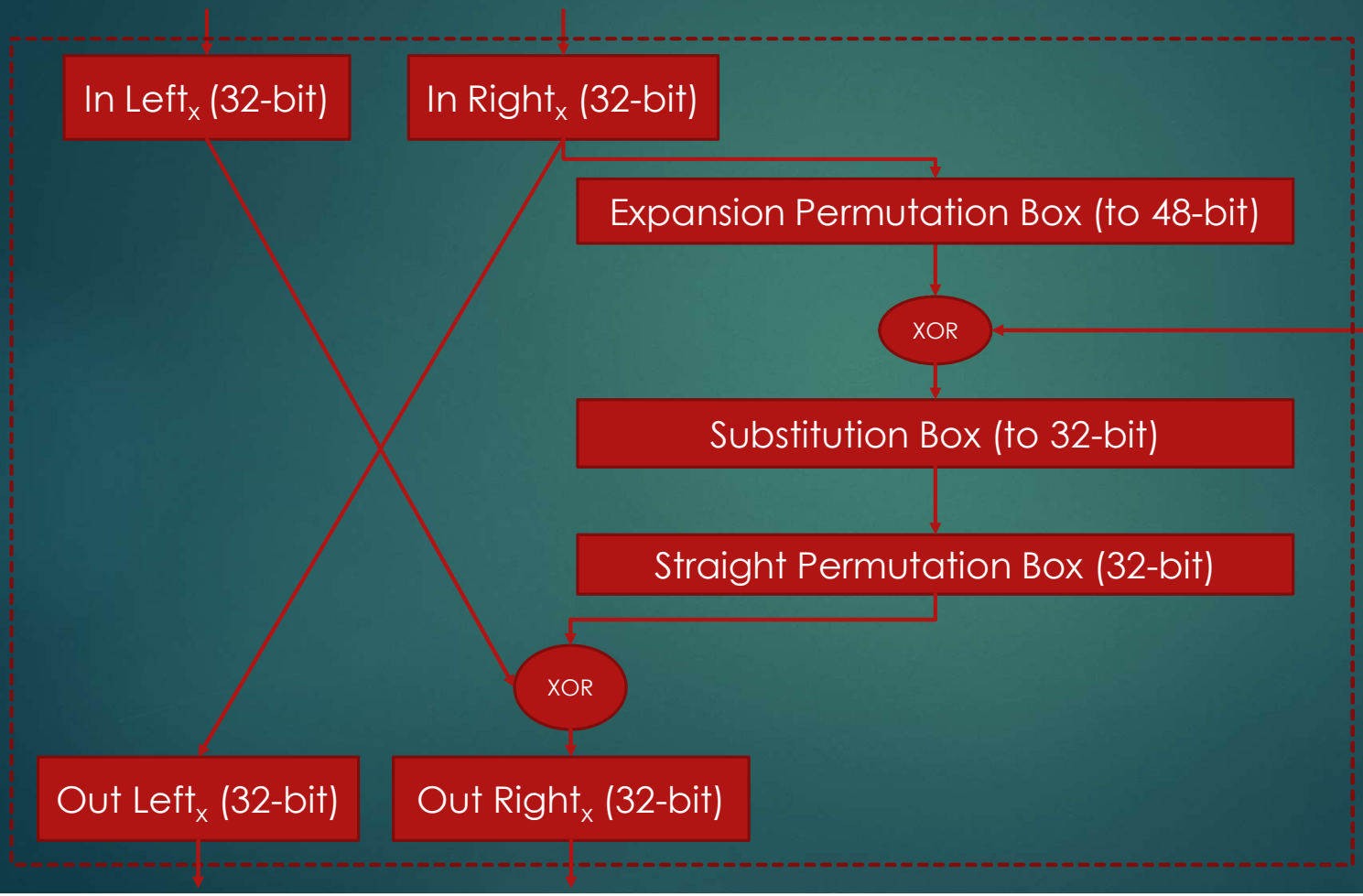
Data Encryption Standard (DES)



2020 Est.	56 bits
$3 T	35 hr
$33 T	3.5 hr
$333 T	21 min
$3 M	2 min
$33 M	13 sec
$333 M	1 sec
$3 B	.1 sec
$33 B	.01 sec
$333 B	1 millisec

DES is not longer seen as secure due to key length. Also, a number of weak keys were found.

Typical DES round



Each DES Round is different, but similar with different path, permutation and substitution

Applications of cryptology

Hypertext Transfer Protocol Secure aka HTTPS

- When we surf the net using the insecure HTTP protocol, data travels in an unencrypted format that can easily be intercepted and stolen by anyone listening in on the network
- Every time we connect to a website over HTTPS, an encrypted communication channel is established between our client browser and the server hosting the site:
  - Connection Negotiation: determine the supported cipher suites
  - Key Exchange: using asymmetric algorithms, they securely exchange a pre-master secret key. The symmetric key is calculated separately by both the client and the server based on the value of the pre-master secret key
  - Change Cipher Spec: After calculating the symmetric key, both the server and the client send a change cipher spec message to each other. This indicates that the remaining communication involving any bulk data transfer will be done using symmetric keys (by applying encryption standards such as AES) over a secure encrypted channel

Digital signature

- In practical implementations, public-key algorithms are often too inefficient to sign long documents. To save time, digital signature protocols are often implemented with one-way hash functions. Instead of signing a document, the sender signs the hash of the document
- The protocol is:
  - Sender produces a one-way hash of a document
  - Sender encrypts the hash with his private key, thereby signing the document
  - Sender sends the document and the signed hash to Receiver
  - Receiver produces a one-way hash of the document that Sender sent. Receiver decrypts the signed hash with Sender's public key. If the signed hash matches the hash he generated, the signature is valid

Digital signature (Cont'd)

- This protocol also satisfies the characteristics of a signature (in fact better than a real signature):
  - The signature is authentic; when Receiver verifies the message with Sender's public key, he knows that Sender signed it
  - The signature is unforgeable; only Sender knows his private key
  - The signature is not reusable; the signature is a function of the document and cannot be transferred to any other document
  - The signed document is unalterable; if there is any alteration to the document, the signature can no longer be verified with Sender's public key
  - The signature cannot be repudiated. Receiver doesn't need Sender's help to verify his signature

Quantum Computing and its Impact on Cryptography

BASED ON AN ARTICLE FROM ROB STUBBS ON 29 APRIL 2018

Quantum computing

- "Quantum computing" is computation performed using a computing device based on the strange, counter-intuitive physical properties of matter at very small scale, known as quantum mechanics
- Unlike a classical computer based on transistors that encodes data in binary digits (or "bits") that can only be a "1" or a "0" (think "on" or "off"), a quantum computer uses quantum bits (qubits) where a single qubit has two states simultaneously
- Quantum computing should not be confused with "quantum cryptography", which is the science of exploiting quantum mechanical properties to perform cryptographic tasks. A prime example of this is "quantum key distribution", which enables a secret cryptographic key to be shared between two remote parties such that any interception can be reliably detected

Do Quantum Computers Exist

- Yes – simple, small-scale quantum computers have been built and successfully demonstrated. Currently these are laboratory instruments that are large, expensive and complex to use, and have very limited capabilities. But they do prove the underlying physical principles are sound
- The challenge is to build one that is big enough (in terms of qubit capacity) to perform useful tasks better than classical computers

Why are Quantum Computers Useful

- It is possible to create algorithms that run significantly faster on a quantum computer than a classical computer, due to the unique properties of qubits
- These algorithms could be used for a number of different scientific and business applications, and will bring many benefits
- Some of these algorithms have already been tested and proven on prototype quantum computers, but will not be practically useful or economical until larger quantum computers have been built

How are Quantum Computers Relevant to IT Security

- It is possible to create unique algorithms for quantum computers (e.g. "Shor's algorithm") that dramatically reduce the time it takes to break cryptologic algorithms
- Symmetric algorithms used for encryption, like AES, are still thought to be safe with sufficient key length – e.g. AES-256 or larger
- However, current asymmetric algorithms like RSA and ECDSA will be rendered essentially useless once quantum computers reach a certain scale
- **This will break nearly every practical application of cryptography in use today, making e-commerce and many other digital applications that we rely on in our daily lives totally insecure**

What is Preventing Large Quantum Computers Today

- Working at the limits of physics is challenging! Whilst much of the theory is well understood, turning theory into practice at such small scales is a significant scientific and engineering challenge that is taxing many of the world's best scientists
- There are numerous fundamental problems yet to be overcome before large-scale quantum computers become feasible. In particular, qubits are highly susceptible to almost undetectable amounts of thermal and electromagnetic interference that are difficult to eliminate

So when will Large Quantum Computers Become Available

- The short answer is that no-one knows
- It depends on a number of scientific and engineering break-throughs being made, which could come in the next 5-10 years, or 20-30 years, or maybe never
- It may take many more years before such computers are generally affordable outside of large government agencies
- This uncertainty is the biggest worry facing governments and business alike

What will Happen Then

- Fortunately, many mathematicians within academia and government are working on a number of candidate "quantum-resistant" algorithms that cannot be broken using quantum computers
- However, it takes time to gain confidence that these algorithms don't have other weaknesses – it typically takes many years to gain confidence in the safety of any new algorithm
- New standards will have to be written and adopted, many of these being national or industry-specific; applications will have to be adapted to make use of the new algorithms, which can be a real challenge in some industries (such as banking) where there is a huge amount of legacy infrastructure that cannot be easily upgraded, if at all

Conclusion

- This presentation has been an introduction to cryptography and provided the key definitions, basic cryptography mathematics, basic algorithms, applications and impact of quantum computing