The role of cryptography in our information-based society

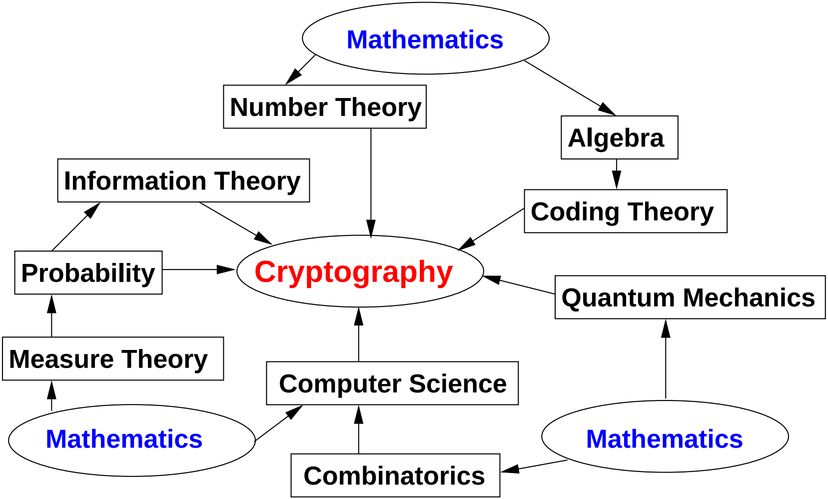
Joseph J. Boutros

Texas A&M University at Qatar
joseph.boutros@qatar.tamu.edu

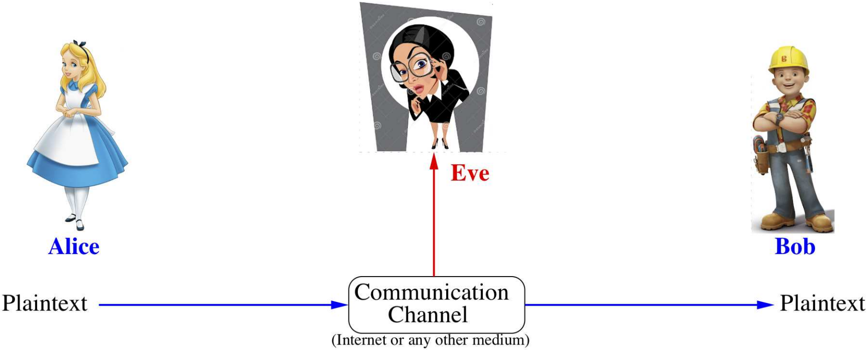
www.josephboutros.org

2 July 2020

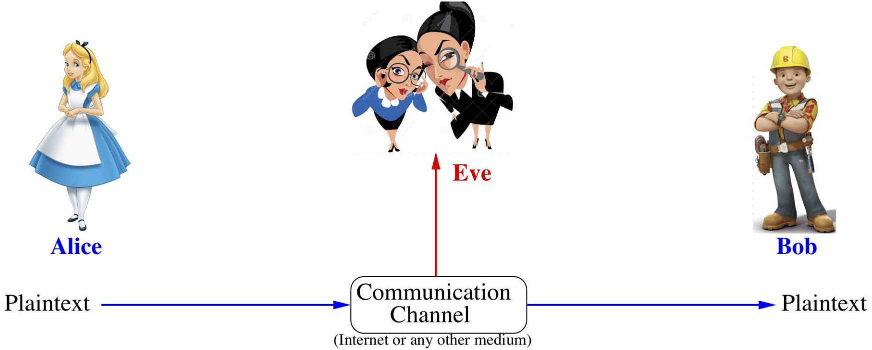
What is Cryptography? (1)



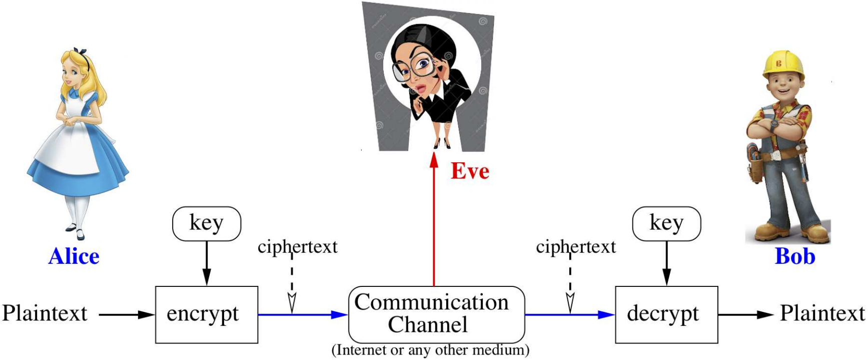
What is Cryptography? (2)



What is Cryptography? (2)



What is Cryptography? (3)



What is Cryptography? (4)

WORD	MEANING
Cryptography	Hidden or secret writing
Encrypt/encode/encipher	Make the writing secret
Plaintext/message/ normal language	The text before encryption
Ciphertext/code/cipher	The text after encryption
Decrypt/decode/decipher	Convert the ciphertext into a plaintext
Cipher/Cypher	Set of algorithms for encryption and decryption
Cryptosystem	Set of three algorithms for key exchange, encryption, and decryption
Cryptanalysis	Algorithms (attacks) used to breach cryptographic security systems and gain access to the contents of encrypted messages, even if the cryptographic key is unknown
Cryptology	The science grouping cryptography and cryptanalysis

What is Cryptography? (5)

Origin of the word "cipher"

Arabic → Medieval Latin → French → Cipher

- "Cipher" means zero in Arabic (Al Sifr).
- It became "chiffre" in French and later "cipher" or encryption/chiffrement.
- How the word "cipher" may have come to mean "encoding/encryption":
  - Encoding involves numbers.
  - Absence of zero in the Roman number system.

-**Ibrahim A. Al-Kadi**, Cryptography and Data Security: Cryptographic Properties of Arabic", proceedings of the 3rd Saudi Eng. Conference, Riyadh, Nov. 1991.

-**Georges Ifrah**, The Universal History of Numbers: From Prehistory to the Invention of the Computer, 2000.

-**Claude E. Shannon**, Communication Theory of Secrecy Systems, 1949.

What is Cryptography? (5)

Origin of the word “cipher”

Arabic → Medieval Latin → French → Cipher

- “Cipher” means zero in Arabic (Al Sifr).
- It became “chiffre” in French and later “cipher” or encryption/chiffrement.
- How the word “cipher” may have come to mean “encoding/encryption”:
  - Encoding involves numbers.
  - Absence of zero in the Roman number system.

-**Ibrahim A. Al-Kadi**, Cryptography and Data Security: Cryptographic Properties of Arabic”, proceedings of the 3rd Saudi Eng. Conference, Riyadh, Nov. 1991.

-**Georges Ifrah**, The Universal History of Numbers: From Prehistory to the Invention of the Computer, 2000.

-**Claude E. Shannon**, Communication Theory of Secrecy Systems, 1949.

What is Cryptography? (5)

Origin of the word "cipher"

Arabic → Medieval Latin → French → Cipher

- "Cipher" means zero in Arabic (Al Sifr).
- It became "chiffre" in French and later "cipher" or encryption/chiffrement.
- How the word "cipher" may have come to mean "encoding/encryption":
  - Encoding involves numbers.
  - Absence of zero in the Roman number system.

-**Ibrahim A. Al-Kadi**, Cryptography and Data Security: Cryptographic Properties of Arabic", proceedings of the 3rd Saudi Eng. Conference, Riyadh, Nov. 1991.

-**Georges Ifrah**, The Universal History of Numbers: From Prehistory to the Invention of the Computer, 2000.

-**Claude E. Shannon**, Communication Theory of Secrecy Systems, 1949.

Early examples of cryptography

- Classic cryptography: from ancient times to the Internet.
- It is a *weak* cryptography.
- Use unknown symbols, transposition of letters, substitution of letters.

-Seminal book: **David Kahn, The Codebreakers**: The Comprehensive History of Secret Communication from Ancient Times to the Internet, 1996.

-Recent reference: **Bruce Schneier, Secrets and Lies**: Digital Security in a Networked World, 2015.

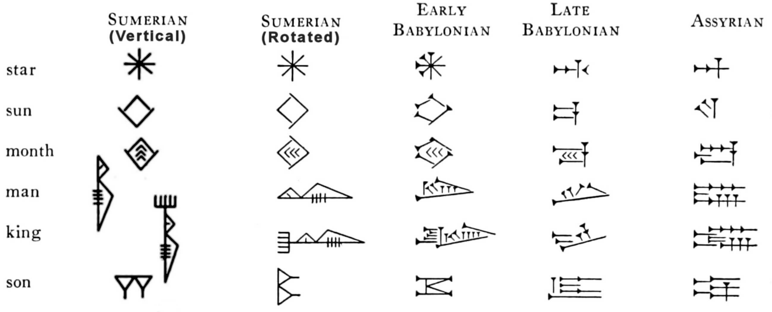
Early examples of cryptography - Ancient Egypt (1)

- The first documented use of cryptography, around 1900 BC in Egypt.
- During the reign of pharaohs Amenemhat II and Senusret/Sésostris II of the 12th Dynasty, Middle Kingdom.
- A scribe used non-standard hieroglyphs in an inscription on the tomb of the great chief Khnumhotep II at Beni Hasan, Egypt. Some references cite archaeologists who supposedly have found basic examples of encrypted hieroglyphs dating back to the Old Kingdom (2686-2181 BC).

Early examples of cryptography - Ancient Egypt (2)

- Most of the people were illiterate and only the elite could read any written language.
- Some references assume that these non-standard hieroglyphs were not made to protect critical information, but rather to provide enjoyment for the intellectual members of the community.
- Left: Sphinx of Amenemhat II, Louvre Museum, Paris. Right: Khnumhotep II depicted while hunting birds, Beni Hasan tomb 3, Egypt.

Early examples of cryptography - Mesopotamia (Iraq)



- Some clay tablets from Mesopotamia are meant to protect information-one dated near 1500 BC was found to encrypt a craftsman's recipe for pottery glaze, presumably commercially valuable.
- Tablets were written in Cuneiform (this writing preceded the Egyptian hieroglyphs). Encryption was made by substitution of cuneiform signs.

Early examples of cryptography - Atbash (Hebrew abjad)

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

- Hebrew scholars made use of simple monoalphabetic substitution ciphers (such as the Atbash cipher) in the period 600-500 BC.
- The Atbash cipher, also known as the *mirror code*, is formed by taking the alphabet and mapping it to its reverse, so that the first letter becomes the last letter, the second letter becomes the second to last letter, and so on.
- In the book of Jeremiah (around 600 BC), biblical verses encrypted Babylon as Sheshach, and Chaldeans was encrypted as Lev-kamai. A compact table for Latin-Atbash encryption/decryption is shown below.

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

Early examples of cryptography - Atbash (Hebrew abjad)

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

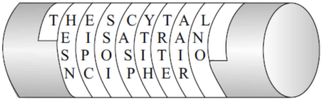
- Hebrew scholars made use of simple monoalphabetic substitution ciphers (such as the Atbash cipher) in the period 600-500 BC.
- The Atbash cipher, also known as the *mirror code*, is formed by taking the alphabet and mapping it to its reverse, so that the first letter becomes the last letter, the second letter becomes the second to last letter, and so on.
- In the book of Jeremiah (around 600 BC), biblical verses encrypted Babylon as Sheshach, and Chaldeans was encrypted as Lev-kamai. A compact table for Latin-Atbash encryption/decryption is shown below.

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

Early examples of cryptography - Ancient Greece (1)

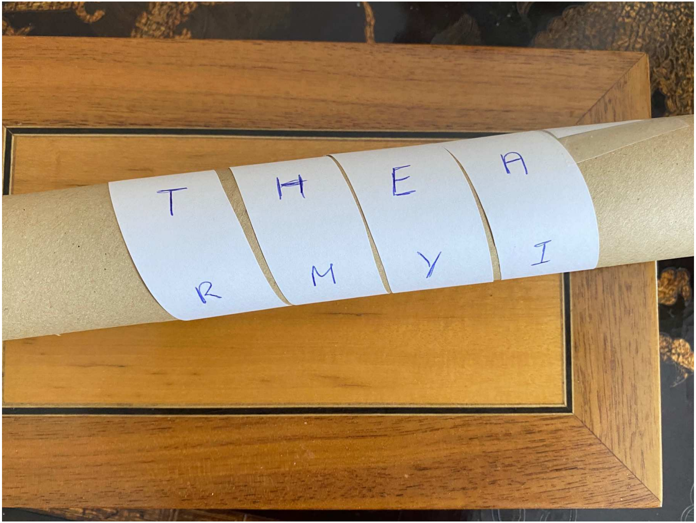
The Scytale, Le bâton de Plutarque.

- The scytale transposition cipher was used by the Greek/Spartan military.
- The Greek philosopher Plutarch documented the use of the scytale by Lysander of Sparta around 400 BC.
- It consists of a cylinder with a leather strip around it on which is written a message. The key (the secret or the password) is the rod/cylinder diameter.

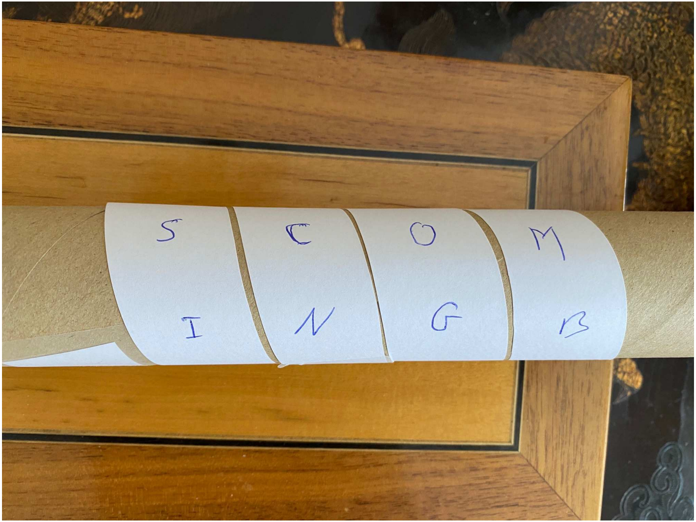


Early examples of cryptography - Ancient Greece (2)

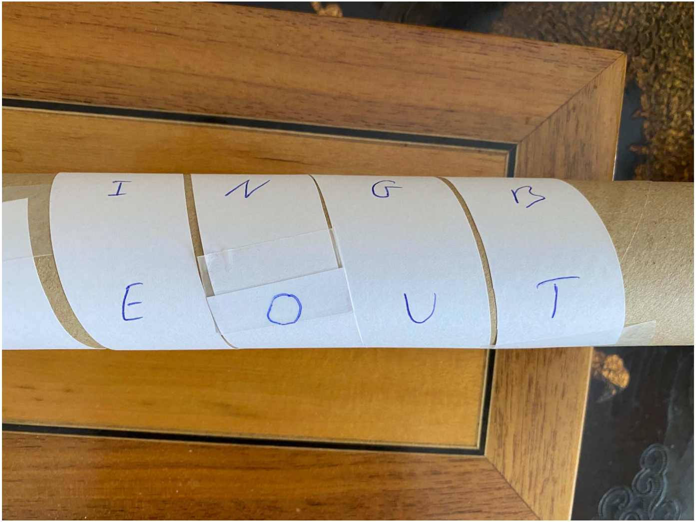
Early examples of cryptography - Ancient Greece (2)



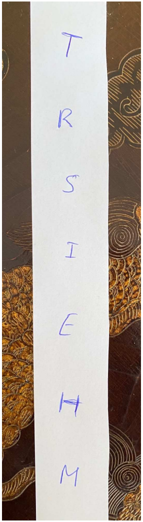
Early examples of cryptography - Ancient Greece (2)

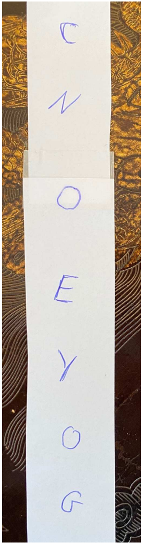


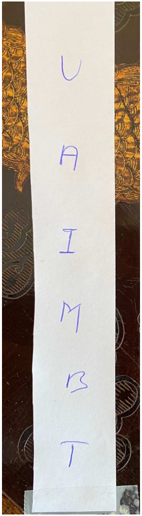
Early examples of cryptography - Ancient Greece (2)



Early examples of cryptography - Ancient Greece (2)







Other main developments before Modern Cryptography

Due to the lack of time and space, we just give this brief list:

1. Caesar cipher (100-44 BC), shifting the alphabet by 3 positions to the left.
2. The frequency analysis by Abu Yusuf Al-Kindi (801-873 AD).
3. The Vernam (1917) polyalphabet substitution cipher inspired from Vigenere cipher (1523-1596).
4. The Enigma/Lorentz German machines of WWII.
5. The Data Encryption Standard, the DES (1975), ancestor of the AES.
6. The weak ROT13 cipher (used in games and newsgroups since 1980), similar to Caesar cipher.

We focus next on public-key cryptography before showing secret-key cryptography with the AES encryption.

Asymmetric Cryptography - Public Key

- Public key algorithms are fundamental security ingredients in cryptosystems, applications, and protocols. Public key cryptography is based on prime numbers and elliptic curves.
- Main functions: Encryption, key distribution, and digital signature.
- 1976: The Diffie-Hellman protocol for key exchange. By three American cryptographers: Whitfield Diffie, Martin Hellman, and Ralph Merkle.
- 1977: The RSA algorithm designed at MIT, by Ron Rivest (US), Adi Shamir (Israel), and Leonard Adleman (US). Keys of lengths from 1024 to 4096 bits are used in RSA.
- 1985: ElGamal encryption derived from Diffie-Hellman, by Taher ElGamal (Egypt+US).
- ECC: Elliptical Curve Cryptography. First proposed by Neal Koblitz (Univ. of Wash.) and Victor Miller (IBM) in 1985. It yields smaller keys, e.g. 164 bits.

Asymmetric Cryptography - Public Key

- Public key algorithms are fundamental security ingredients in cryptosystems, applications, and protocols. Public key cryptography is based on prime numbers and elliptic curves.
- Main functions: Encryption, key distribution, and digital signature.
- 1976: The Diffie-Hellman protocol for key exchange. By three American cryptographers: Whitfield Diffie, Martin Hellman, and Ralph Merkle.
- 1977: The RSA algorithm designed at MIT, by Ron Rivest (US), Adi Shamir (Israel), and Leonard Adleman (US). Keys of lengths from 1024 to 4096 bits are used in RSA.
- 1985: ElGamal encryption derived from Diffie-Hellman, by Taher ElGamal (Egypt+US).
- ECC: Elliptical Curve Cryptography. First proposed by Neal Koblitz (Univ. of Wash.) and Victor Miller (IBM) in 1985. It yields smaller keys, e.g. 164 bits.

Prime Numbers for Cryptography

Prime Number

Let $p \geq 2$ be an integer. The integer p is prime if it is only divisible by 1 and itself.

- Examples of small prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, ...
- These numbers are not prime: 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, ...

Existence of Large Prime Numbers

Let n be an integer, $n > 1$. Bertrand's postulate (now a theorem, originally conjectured by Joseph Bertrand 1822-1900) states that there exists at least one prime number p such that $n < p \leq 2n$.

- Examples of large prime numbers: 40099, 76693691, 12612466877, 1518068879230479685717, 599970664556404984568165167066519.

Prime Numbers for Cryptography

Prime Number

Let $p \geq 2$ be an integer. The integer p is prime if it is only divisible by 1 and itself.

- Examples of small prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, ...
- These numbers are not prime: 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, ...

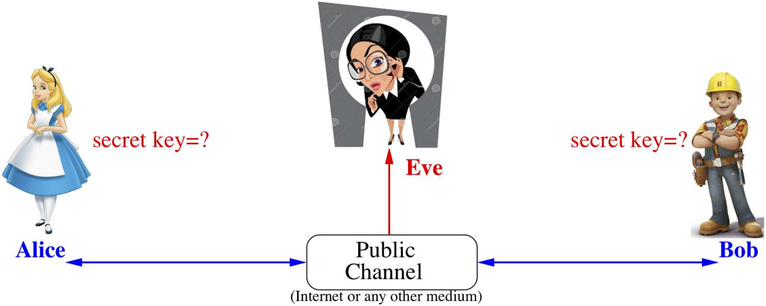
Existence of Large Prime Numbers

Let n be an integer, $n > 1$. Bertrand's postulate (now a theorem, originally conjectured by Joseph Bertrand 1822-1900) states that there exists at least one prime number p such that $n < p \leq 2n$.

- Examples of large prime numbers: 40099, 76693691, 12612466877, 1518068879230479685717, 599970664556404984568165167066519.

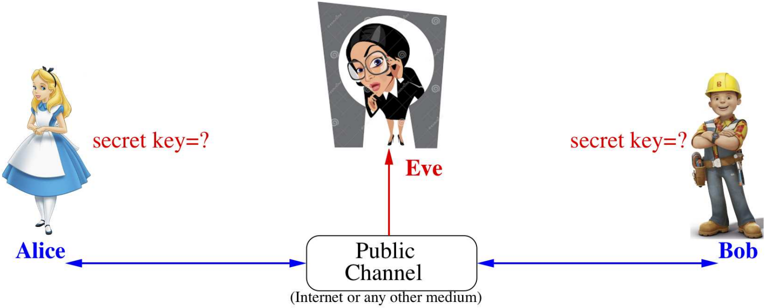
The Diffie-Hellman Protocol (1)

How to exchange a secret key?



The Diffie-Hellman Protocol (2)

How to exchange a secret key?

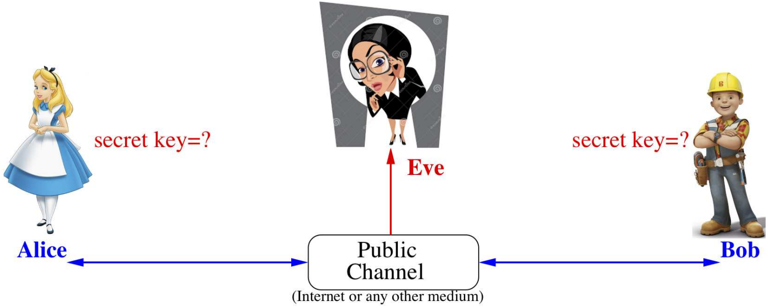


Alice = Your Machine, desktop, laptop, tablet, or smartphone.

Bob = Your bank server, your GMail account server, or a WhatsApp server.

The Diffie-Hellman Protocol (3)

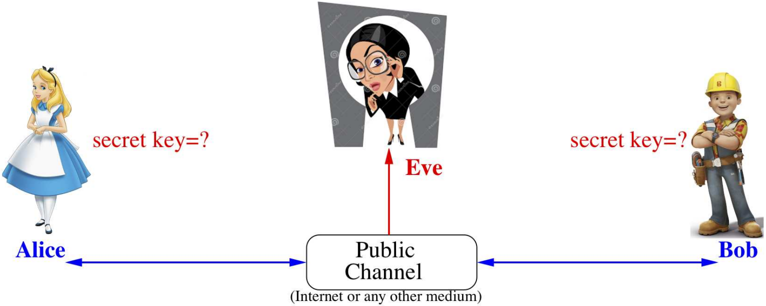
How to exchange a secret key?



Select a public prime number p and public number α:
p=24021135745533513541866782302279999450211367669841346251544850744
34466320980337897577273486061438683139481546653325618644861885569289
59685282412522321725339795687780734031491136494570984416579578581222
25936879877190600478225060176787220574430652371647297523641705903430
0702122577342770982520968473778353129761.
and $\alpha = 41$. Notice $p \approx 10^{308} \approx 2^{1024}$ (1024 bits).

The Diffie-Hellman Protocol (4)

How to exchange a secret key?



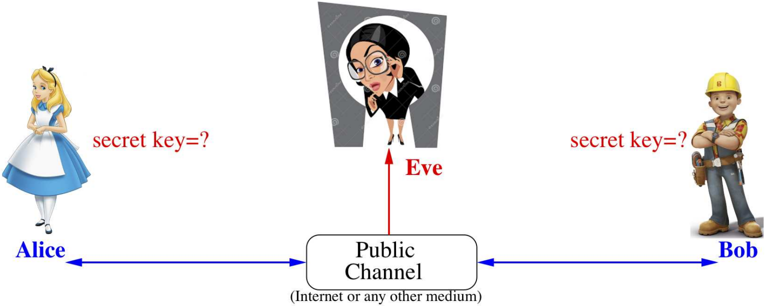
Alice picks up a random integer number a (secret):

a=18535002323881753764573617876936692752827468607526196772384332819
57654935279430802782177356138300496831255493643308633464159688127005
78569056266557476456668133551612469926032645616048086633759718072163
65619113321442865391480341547257835520325127465107154999309047505314
444503015817573931596358076845878870665 8.

Only Alice knows a.

The Diffie-Hellman Protocol (5)

How to exchange a secret key?



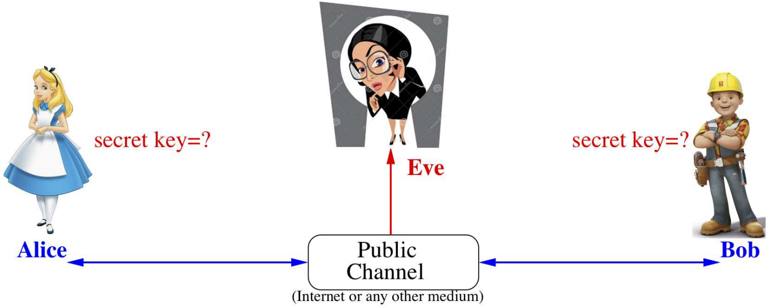
Alice sends $A = \alpha^a$ to Bob (public):

A=102687934056460263294103966687856065410159035054796125378392718545322530619357594032014480118077555262387866799525629494302114976579708902449052729327290027792199031751465143158560351881484274076087147539556144264388525258340742652595056476335295946587283724631993247011882276209534962043845442137491613286382.

All operations are modulo p. Everyone knows A.

The Diffie-Hellman Protocol (6)

How to exchange a secret key?



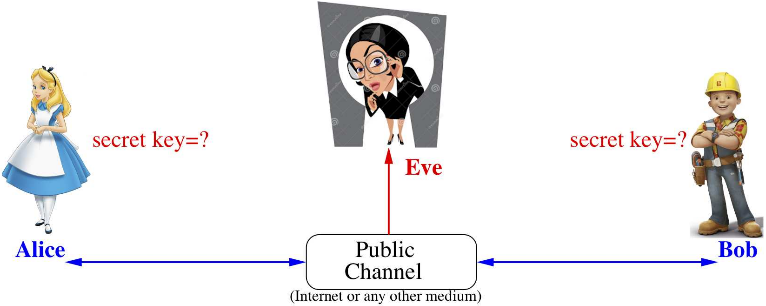
Bob picks up a random integer number b (secret):

b=46067498278252895820456844693840410829833274736620995083059379137
11277055000799065055754920365818240031802894284084348787197396081370
43594945109651050677165032391157879832888786881640710954154561181573
82809812097793187056170746094700343446610354229899811932204069074676
2867083833951436040785453340226694056 93.

Only Bob knows b.

The Diffie-Hellman Protocol (7)

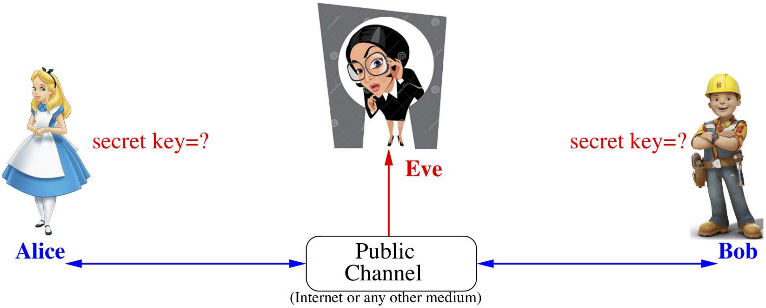
How to exchange a secret key?



Bob sends $B = \alpha^b$ to Alice (public):
B=171532336960837157390969221766002378260097916416050276706493702116697740463542788044973615016009207311844239851077675628649520076886479135133143633153895268637846720910287484164981973779113857336644373747225081090088070069687292304518895964446352529259913911140610043906135408732960335035621162085042981822726.
All operations are modulo p. Everyone knows B.

The Diffie-Hellman Protocol (8)

How to exchange a secret key?



Alice computes $s = B^a = \alpha^{ab} = s$. Bob computes $A^b = \alpha^{ab} = s$.
s=65963369188673414771026985734489154951425143596399624471179005220
49356036626737520884988249171493910211721260943146193232755545907449
57014377426276999336200533629625993953121556987800138558887636464577
68397031851062234488919620296305239708153536130629972387020537679935
4751986205510058468152108460893646570 31.

Besides Alice and Bob, no one knows the secret s.

The Diffie-Hellman Protocol (9)

Summary of the key-exchange algorithm.

- Alice selects a secret key a. Alice sends $A = \alpha^a$ to Bob on a public channel.
- Bob selects a secret key b. Bob sends $B = \alpha^b$ to Alice on a public channel.
- A spy listening to the public channel will get A and B, but neither a nor b.
- Alice computes $B^a = (\alpha^b)^a = \alpha^{ab} = s$.
- Bob computes $A^b = (\alpha^a)^b = \alpha^{ab} = s$.
- Now Alice and Bob both have s as a shared secret key. A spy cannot find s.

All operations are made modulo a large prime number p (public).
The number α is also public.

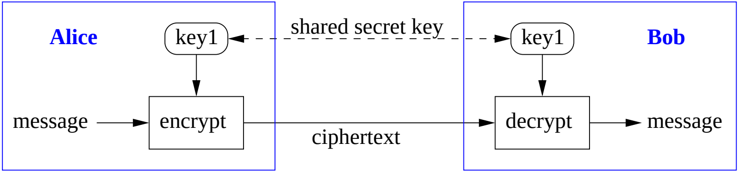
Top 5 Super Computers

The size of p in the previous example (1024 bits) does not allow current technology, whether based on supercomputers or distributed computing, to break the Diffie-Hellman key exchange (same for RSA).

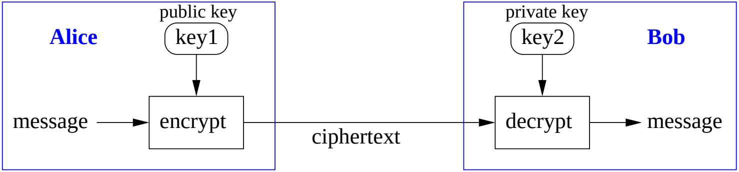
Top 5 most powerful supercomputers (June 2020)					
Rank	**Name**	**Country**	**Cores**	**R_{max} (TFlop/s)**	**Power (kW)**
1	Fugaku	Japan	7,299,072	415,530	28,335
2	Summit (IBM)	United States	2,414,592	148,600	10,096
3	Sierra (IBM)	United States	1,572,480	94,640	7,438
4	Sunway TaihuLight	China	10,649,600	93,014	15,371
5	Tianhe-2A	China	4,981,760	61,444	18,482

Symmetric versus Asymmetric Encryption (1)

Symmetric Key System

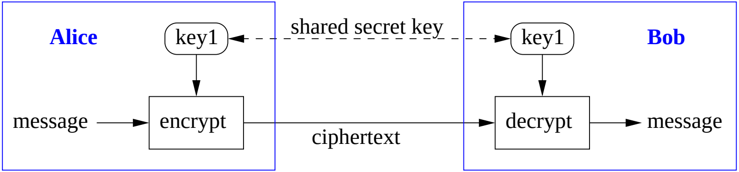


Asymmetric Key System

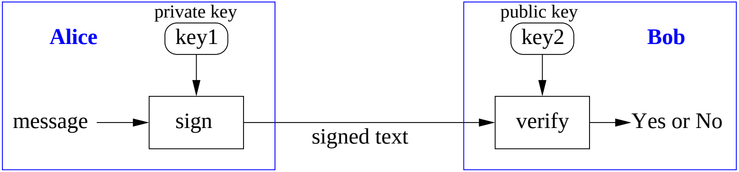


Symmetric versus Asymmetric Encryption (2)

Symmetric Key System



Asymmetric Key System



RSA Public Key Encryption (1)

RSA Key Generation.

- Choose two large (and distinct) prime numbers p and q.
  p and q are kept private.
- Compute $n = pq$. All operations will be made modulo n. n is public.
- Compute $\lambda = \text{lcm}(p-1, q-1)$. λ is kept private.
- Choose an integer e such that $1 < e < \lambda$ and $\gcd(e, \lambda) = 1$. e is usually small to make efficient encryption. e is public.
- Determine d such that $d \cdot e = 1 \mod \lambda$. d is private.
- Public key: n and e. Private key: p, q, λ, and d.

RSA Public Key Encryption (1)

RSA Key Generation.

- Choose two large (and distinct) prime numbers p and q.
  p and q are kept private.
- Compute $n = pq$. All operations will be made modulo n. n is public.
- Compute $\lambda = \text{lcm}(p-1, q-1)$. λ is kept private.
- Choose an integer e such that $1 < e < \lambda$ and $\gcd(e, \lambda) = 1$. e is usually small to make efficient encryption. e is public.
- Determine d such that $d \cdot e = 1 \mod \lambda$. d is private.
- Public key: n and e. Private key: p, q, λ, and d.

RSA Public Key Encryption (1)

RSA Key Generation.

- Choose two large (and distinct) prime numbers p and q.
  p and q are kept private.
- Compute $n = pq$. All operations will be made modulo n. n is public.
- Compute $\lambda = \text{lcm}(p-1, q-1)$. λ is kept private.
- Choose an integer e such that $1 < e < \lambda$ and $\gcd(e, \lambda) = 1$. e is usually small to make efficient encryption. e is public.
- Determine d such that $d \cdot e = 1 \mod \lambda$. d is private.
- Public key: n and e. Private key: p, q, λ, and d.

RSA Public Key Encryption (1)

RSA Key Generation.

- Choose two large (and distinct) prime numbers p and q.
  p and q are kept private.
- Compute $n = pq$. All operations will be made modulo n. n is public.
- Compute $\lambda = \text{lcm}(p-1, q-1)$. λ is kept private.
- Choose an integer e such that $1 < e < \lambda$ and $\gcd(e, \lambda) = 1$. e is usually small to make efficient encryption. e is public.
- Determine d such that $d \cdot e = 1$ modulo λ. d is private.
- Public key: n and e. Private key: p, q, λ, and d.

RSA Public Key Encryption (1)

RSA Key Generation.

- Choose two large (and distinct) prime numbers p and q.
  p and q are kept private.
- Compute $n = pq$. All operations will be made modulo n. n is public.
- Compute $\lambda = \text{lcm}(p-1, q-1)$. λ is kept private.
- Choose an integer e such that $1 < e < \lambda$ and $\gcd(e, \lambda) = 1$. e is usually small to make efficient encryption. e is public.
- Determine d such that $d \cdot e = 1$ modulo λ. d is private.
- Public key: n and e. Private key: p, q, λ, and d.

RSA Public Key Encryption (1)

RSA Key Generation.

- Choose two large (and distinct) prime numbers p and q.
  p and q are kept private.
- Compute $n = pq$. All operations will be made modulo n. n is public.
- Compute $\lambda = \mathrm{lcm}(p-1, q-1)$. λ is kept private.
- Choose an integer e such that $1 < e < \lambda$ and $\gcd(e, \lambda) = 1$. e is usually small to make efficient encryption. e is public.
- Determine d such that $d \cdot e = 1$ modulo λ. d is private.
- Public key: n and e. Private key: p, q, λ, and d.

RSA Public Key Encryption (2)

RSA Key Distribution, Encryption, and Decryption.

- Key Distribution: Bob sends his public key (n, e) to Alice.
- Encryption: Alice would like to send a message m to Bob, $0 < m < n^{\dagger}$.
- Alice computes the ciphertext $c = m^e \mod n$ and transmits c to Bob.
- Decryption: Bob recovers m from c using his private key d by computing $c^d = (m^e)^d =^{\ddagger} m \mod n$.

RSA Signature: Now d is the private key of Alice. She sends signature s by $c = s^d$ and Bob checks it by $c^e = s$.

† The message m should also satisfy $\gcd(m, n) = 1$, i.e. m different from p and q.

‡ The proof is based on Fermat's little theorem and the Chinese remainder theorem.

RSA Public Key Encryption (2)

RSA Key Distribution, Encryption, and Decryption.

- Key Distribution: Bob sends his public key (n, e) to Alice.
- Encryption: Alice would like to send a message m to Bob, $0 < m < n^{\dagger}$.
- Alice computes the ciphertext $c = m^e \mod n$ and transmits c to Bob.
- Decryption: Bob recovers m from c using his private key d by computing $c^d = (m^e)^d =^{\ddagger} m \mod n$.

RSA Signature: Now d is the private key of Alice. She sends signature s by $c = s^d$ and Bob checks it by $c^e = s$.

† The message m should also satisfy $\gcd(m, n) = 1$, i.e. m different from p and q.

‡ The proof is based on Fermat's little theorem and the Chinese remainder theorem.

RSA Public Key Encryption (2)

RSA Key Distribution, Encryption, and Decryption.

- Key Distribution: Bob sends his public key (n, e) to Alice.
- Encryption: Alice would like to send a message m to Bob, $0 < m < n$[†].
- Alice computes the ciphertext $c = m^e \mod n$ and transmits c to Bob.
- Decryption: Bob recovers m from c using his private key d by computing $c^d = (m^e)^d =^{\ddagger} m \mod n$.

RSA Signature: Now d is the private key of Alice. She sends signature s by $c = s^d$ and Bob checks it by $c^e = s$.

[†] The message m should also satisfy $\gcd(m, n) = 1$, i.e. m different from p and q.

[‡] The proof is based on Fermat's little theorem and the Chinese remainder theorem.

RSA Public Key Encryption (2)

RSA Key Distribution, Encryption, and Decryption.

- Key Distribution: Bob sends his public key (n, e) to Alice.
- Encryption: Alice would like to send a message m to Bob, $0 < m < n^{\dagger}$.
- Alice computes the ciphertext $c = m^e \mod n$ and transmits c to Bob.
- Decryption: Bob recovers m from c using his private key d by computing $c^d = (m^e)^d =^{\ddagger} m \mod n$.

RSA Signature: Now d is the private key of Alice. She sends signature s by $c = s^d$ and Bob checks it by $c^e = s$.

† The message m should also satisfy $\gcd(m, n) = 1$, i.e. m different from p and q.

‡ The proof is based on Fermat's little theorem and the Chinese remainder theorem.

ElGamal Encryption System (modified Diffie-Hellman)

ElGamal Encryption.

- Key Distribution: Bob sends his public key $B = \alpha^b$ to Alice.
- Encryption: Alice would like to send a message m to Bob, $0 < m < p$.
- Alice computes the ciphertext $c = m \cdot \alpha^{ab}$ modulo p, and transmits the pair (c, α^a) to Bob.
- Decryption: Bob recovers the message m from (c, α^a) using his private key b by computing $c \cdot \alpha^{-ab} = m$.

Hardness of Problems used in Public Key Cryptography

- Computing discrete logarithms and factoring integers (for Diffie-Hellman and RSA) are distinct problems, but both problems are difficult.
- For both problems, no efficient algorithms are known for non-quantum computers.
- For both problems, efficient algorithms on quantum computers are known.
- Algorithms for one problem are often adapted to the other.
- The difficulty of both problems has been used to construct various cryptographic systems.

AES - Advanced Encryption Standard (1)

- **AES**: Advanced Encryption Standard, original name Rijndael, published in 1998 and standardized in 2001.
- Designed by Joan Daemen and Vincent Rijmen, two Belgian cryptographers (from KUL, Leuven).
- Low memory requirement. Fast enough on hardware and software: 10MB/s up to 1 GB/s. Some implementations run at 10 GB/s.
- Some of the major applications:
  - Point-to-point secure web connections (SSL/TLS).
  - End-to-end WhatsApp encryption.
  - End-to-end Facebook Messenger encryption.
  - IPSec for virtual private networks (VPNs).
  - x86-64 (Intel and AMD) and ARM (e.g Apple) processors instructions set.
  - IEEE 802.11i (WiFi).
- It encrypts data in blocks of 128 bits. It replaced the DES.
- Three versions with 3 key lengths: AES-128, AES-192, AES-256.
- As of today, all possible attacks on the full AES did not succeed.

AES - Advanced Encryption Standard (1)

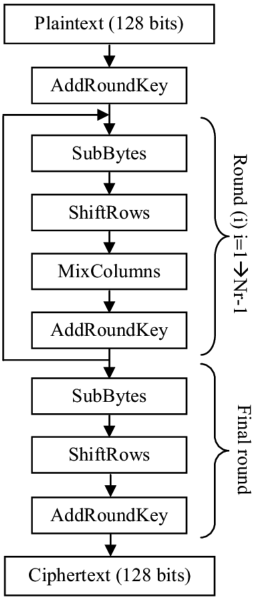
- **AES**: Advanced Encryption Standard, original name Rijndael, published in 1998 and standardized in 2001.
- Designed by Joan Daemen and Vincent Rijmen, two Belgian cryptographers (from KUL, Leuven).
- Low memory requirement. Fast enough on hardware and software: 10MB/s up to 1 GB/s. Some implementations run at 10 GB/s.
- Some of the major applications:
  - Point-to-point secure web connections (SSL/TLS).
  - End-to-end WhatsApp encryption.
  - End-to-end Facebook Messenger encryption.
  - IPSec for virtual private networks (VPNs).
  - x86-64 (Intel and AMD) and ARM (e.g Apple) processors instructions set.
  - IEEE 802.11i (WiFi).
- It encrypts data in blocks of 128 bits. It replaced the DES.
- Three versions with 3 key lengths: AES-128, AES-192, AES-256.
- As of today, all possible attacks on the full AES did not succeed.

AES - Advanced Encryption Standard (1)

- **AES**: Advanced Encryption Standard, original name Rijndael, published in 1998 and standardized in 2001.
- Designed by Joan Daemen and Vincent Rijmen, two Belgian cryptographers (from KUL, Leuven).
- Low memory requirement. Fast enough on hardware and software: 10MB/s up to 1 GB/s. Some implementations run at 10 GB/s.
- Some of the major applications:
  - Point-to-point secure web connections (SSL/TLS).
  - End-to-end WhatsApp encryption.
  - End-to-end Facebook Messenger encryption.
  - IPSec for virtual private networks (VPNs).
  - x86-64 (Intel and AMD) and ARM (e.g Apple) processors instructions set.
  - IEEE 802.11i (WiFi).
- It encrypts data in blocks of 128 bits. It replaced the DES.
- Three versions with 3 key lengths: AES-128, AES-192, AES-256.
- As of today, all possible attacks on the full AES did not succeed.

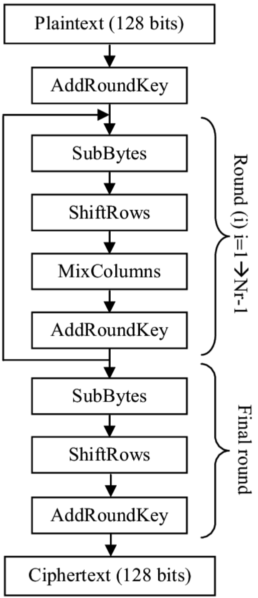
AES - Advanced Encryption Standard (2)

- AES is a network performing many rounds of substitution and permutation, after expanding the keys. It applies the diffusion and the confusion concepts.
- Kerckhoffs' Principle (Auguste Kerckhoffs 1883): A cryptosystem should be secure even if everything about the system, except the key, is public knowledge. Reformulated by Shannon as *the enemy knows the system*.



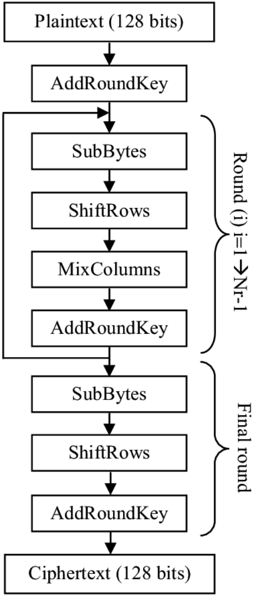
AES - Advanced Encryption Standard (3)

- The Confusion Property (Claude Shannon 1949): Each digit of the ciphertext should depend on several parts of the key.
- The Diffusion Property (Claude Shannon 1949): If we change a single digit of the plaintext, then (statistically) half of the digits in the ciphertext should change.



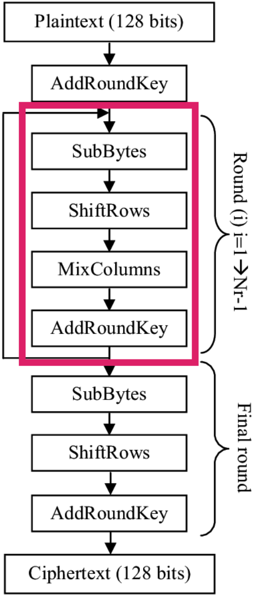
AES - Advanced Encryption Standard (4)

- AES is a block cipher (it belongs to symmetric ciphers).
- Plaintext size is 128 bits. Ciphertext size is 128 bits.
- The 128 bits are written in a 4×4 matrix of 16 bytes, called the *state*.
- Each byte is considered as an element of $\mathbb{F}_{256} = \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1)$ (The Rijndael finite field).



AES - Advanced Encryption Standard (5)

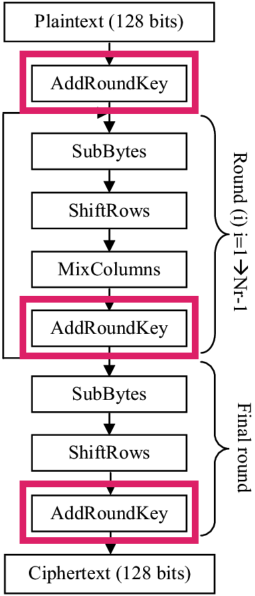
- The AES encryption/decryption key size can be 128, 192, or 256 bits.
- The AES applies 10, 12, or 14 rounds depending on the key size.
- It iterates the function (**one round**) that does substitution and permutation.



AES - Advanced Encryption Standard (6)

- **AES Key Schedule** (key expansion for confusion). Out of the encryption key, a new key is created for each round: the initial round, the 13 main rounds (AES-256), and the final round.

- At each round, a new round key is generated from the previous key by: 1) Cyclic rotation of the 4 bytes in the 4th column, 2) Substitution (S-Box) applied to each byte, 3) Adding a 32-bit constant to the 4th column, and 4) the resulting 4th column is XOR-ed with the 1st column in the previous key. Other columns are also XOR-ed with the column at the next position in the previous key.



AES - Advanced Encryption Standard (7)

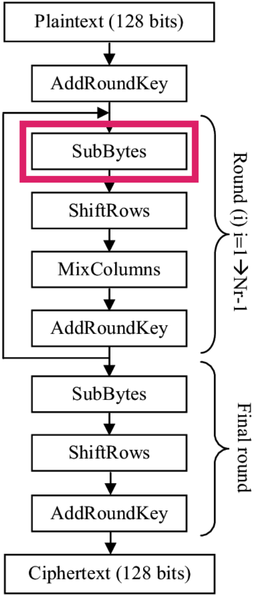
- The **Rijndael S-box** used in AES is a one-to-one mapping (substitution) of elements of $\mathbb{F}_{256}$ (bytes).

- The S-box multiplies by the inverse in the finite field than applies an affine transformation.

- If byte b is the inverse in $\mathbb{F}_{256} \setminus \{0\}$ of the S-box input byte, then the output is

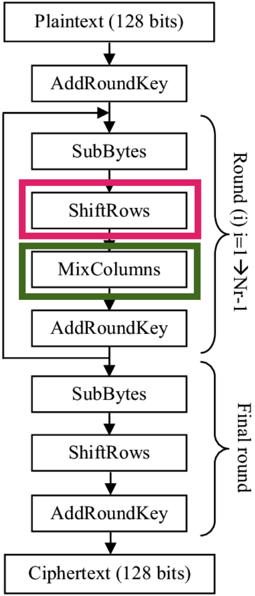
$$\bigoplus_{i=0}^{4} (b << i) \oplus 0x63.$$

- Non-linear properties of Rijndael: resistant to linear and differential attacks.



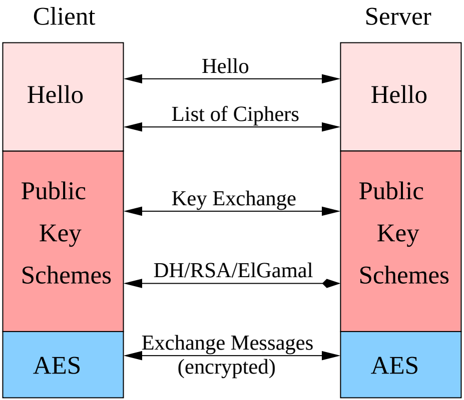
AES - Advanced Encryption Standard (8)

- **ShiftRows step**: The 4 rows of the state are shifted to the left, by 0, 1, 2, and 3 bytes respectively.
- ShiftRows avoid the columns being encrypted independently.
- **MixColumns step**: A 4-byte column in the state is written as a polynomial of $\mathbb{F}_{256}[x]$. Then it is multiplied by $3x^3 + x^2 + x + 2$ modulo $x^4 + 1$. This step can be represented by a 4×4-matrix transformation (in $\mathbb{F}_{256}$).
- ShiftRows and MixColumns provide diffusion in the AES cipher.



Combination of Public Key and Private Key Ciphers

The majority of encrypted communications, HTTPS/TLS, VPN, SSH, proceed in three steps as shown in the simplified model below:



AES (block ciphers) and stream ciphers are much faster than asymmetric (public-key) ciphers.

What is missing in this talk?

Due to the lack of space and time, we did not cover:

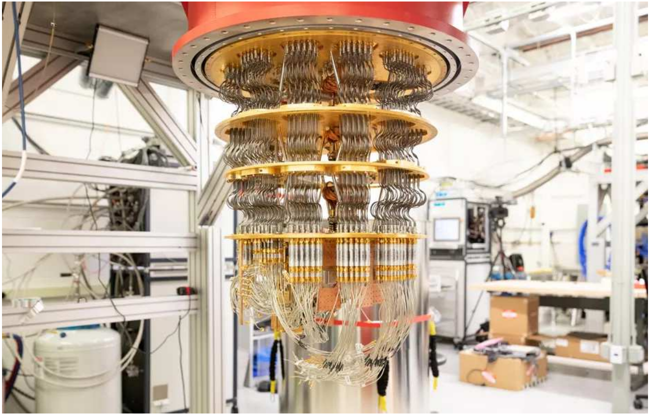
- Stream ciphers like the one-time pad, Enigma, RC4, A5/1, Salsa20, and Chacha20.
- Hash functions used for fingerprinting passwords and for authenticating data.
- The Merkle hash tree and how blockchains are built.
- However, we prepared a nice comparison in the next slide!

Comparison of the security of major protocols

Protocol	Description	Encryption	Key Exchange
TLS 1.3	Secure Web Access, 2018	AES, Chacha20	DHE, ECDHE
IPSec	Virtual Private Network	3DES, AES, Chacha20	DH, ECDH
WireGuard	Virtual Private Network	Chacha20	ECDH
Signal	WhatsApp, Facebook, Skype	AES	ECDH

- Only TLS 1.3 implements ephemeral key exchange (forward secrecy) according to our investigations.
- We did not list OpenVPN because it is based on TLS.
- There is a controversy between AES and Chacha20, mainly about the speed performance when running on software or hardware, on mobile devices or desktops.

Post-Quantum Cryptography (1)



Quantum computers use quantum-mechanical phenomena such as superposition and entanglement to perform computation. Quantum computers are able to solve certain computational problems faster than classical computers.

Post-Quantum Cryptography (2)

Post-quantum cryptography: Cryptosystems that cannot be broken by quantum computers. We list below some major post-quantum methods.

1. Code-based cryptography (McEliece), based on Goppa codes (Augot 2015) and quasi-cylic MDPC codes (Misoczki, Tillich, Sendrier, Barreto, 2013).
2. Hash-based cryptography, related to the security reduction of Merkle Hash Tree to the underlying hash function (Garcia 2005).
3. Lattice-based cryptography:
   - Hoffstein, Silverman, Pipher, 1996: Nth deg. trunc. polyn. ring (NTRU).
   - Goldreich, Goldwasser, Halevi, 1997: GGH (weak initial parameters).
   - Regev, 2009: Learning with errors (LWE).
   - Lyubashevsky, Peikert, Regev, 2010: Ring learning with errors (Ring LWE).
   - Stehlé, Steinfeld, 2013: Provably secure NTRU.

Lattice Theory and Practice is a current research topic by Dr. Joseph J. Boutros, research funds are needed!

Conclusions

- Public-key (asymmetric) cryptography provides key exchange and digital signature.
- Symmetric cryptography provides fast and secure encryption.
- Both are needed in almost all systems nowadays.
- Military, Governmental Institutions, and the Industry have at their disposal excellent cryptography tools in this century to protect data.
- The 21st century also offers individuals a variety of tools to guarantee their privacy and the confidentiality of their data under a mass surveillance by governments and a large number of attacks by cyber hackers.

Conclusions

- Public-key (asymmetric) cryptography provides key exchange and digital signature.
- Symmetric cryptography provides fast and secure encryption.
- Both are needed in almost all systems nowadays.
- Military, Governmental Institutions, and the Industry have at their disposal excellent cryptography tools in this century to protect data.
- The 21st century also offers individuals a variety of tools to guarantee their privacy and the confidentiality of their data under a mass surveillance by governments and a large number of attacks by cyber hackers.

Your questions: All Questions are Welcome

THANK YOU

References for study:

An Introduction to Mathematical Cryptography, by J. Hoffstein, J. Pipher, and .H. Silverman, 2nd edition, Springer, 2014.

Introduction to Modern Cryptography, by J. Katz and Y. Lindell, 2nd edition, CRC, 2014 (3rd edition, 2021).