Cryptography: Information confidentiality, integrity, authenticity, and person identification

Kserks **Anchient Babilon**
Caesar **Rome empire**
Vernam **1917**

New Directions in Cryptography
Whitfield Diffie (Member, IEEE),
Martin E. Hellman (Member, IEEE)

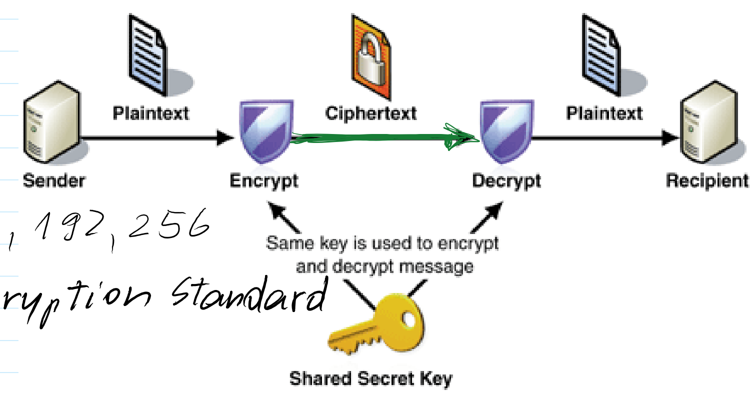
Originally published in IEEE Transactions on Information Theory, Vol. IT-22, No. 6, November 1976

Symmetric cryptography -------------------- Asymmetric cryptography

Symmetric encryption
H-functions, Message digest
HMAC H-Message Authentication Code

Asymmetric encryption
E-signature - Public Key Infrastructure - PKI
E-money, cryptocurrencies
E-voting
Digital Rights Management - DRM
Etc.

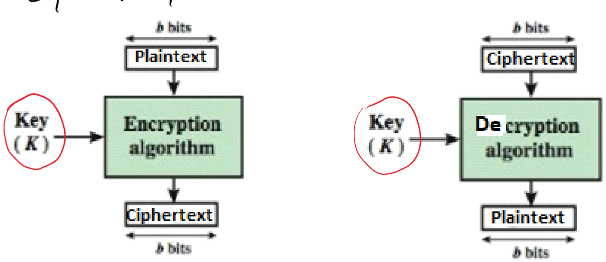
Symmetric - Secret Key Encryption



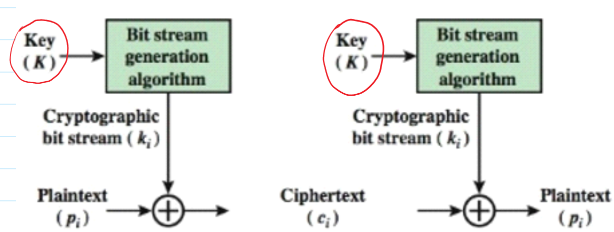
AES - 128, 192, 256
Advanced Encryption Standard
~ 2000 m.

Symmetric ciphers

AES: Block Ciphers
128, 196, 256



Stream Ciphers



bitwise XOR

b bits b bits

bitwise XOR

Vernam Cipher (1917)

A: $m \in \{0,1\}$; $k \leftarrow rand\{0,1\}$. B: $k = 1$.

$c = m \oplus k$ —c→ $m = c - k$; $\oplus$ - is selfinverse

$m = c \oplus k = m \oplus k \oplus k =$

$= m \oplus 0 = m = 1$

m	k	$c = m \oplus k$
0	0	0
0	1	1
1	0	1
1	1	0

Encryption of multiple bits :

Decryption —"—

. . . . . . $k_2 k_1 k_0$

m :	1001 1011 0110
$\oplus$ k :	0101 1001 0011
c :	1100 0010 0101
$\oplus$ k :	0101 1001 0011
m :	1001 1011 0110

Block cipher AES - 128, 192, 256 --> Encryption --> Decryption

Advanced Encryption Standard ~ 2000

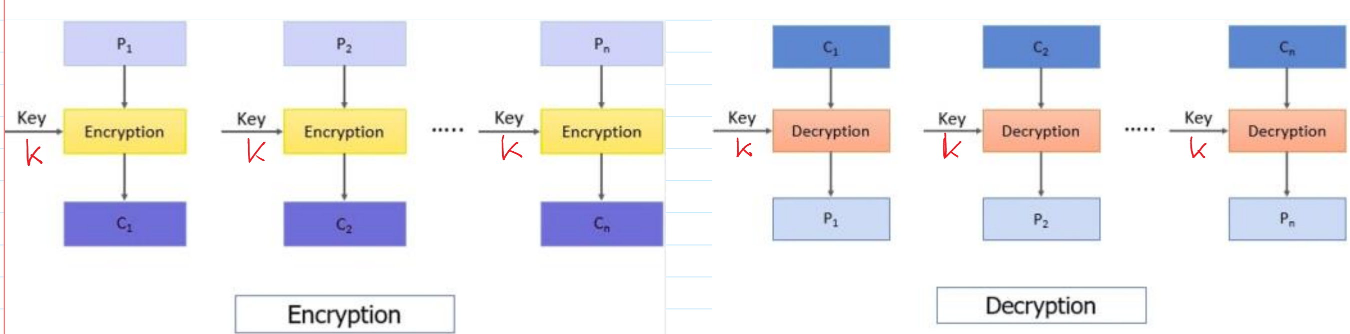
Key length 128, 192, 256, bits : $k \in \{128b, 192b, 256b\}$

Block Cipher: Electronic Code Book -**ECB** mode of encryption : 1 Byte = 8 bits

$|k| = 128$ bits = 16 Bytes

16 Bytes Data to be encripted : message m

B1	B2	B3	- - - -	Bi		Bn



The length of any block Bi should be $|B_i| = 128$ bits
192 bits
256 bits

$|B_i| = |k| = 128$ bits $= 2^7$ bits

$EncAES(k, B1) = C1$)

$c = c1 \| c2 \| \ldots \| c$

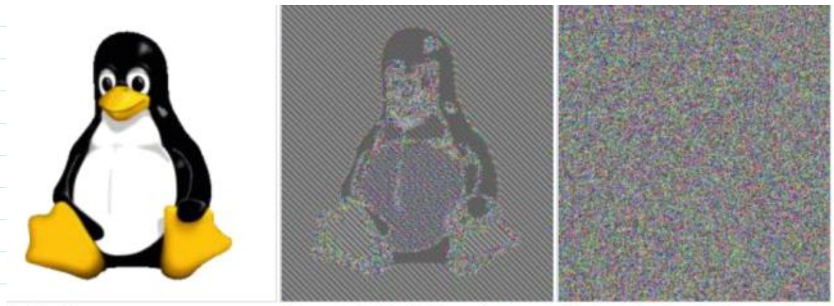
$$EncAES(k, B1) = C1$$
$$EncAES(k, B2) = C2$$
$$\ldots$$
$$EncAES(k, Bn) = Cn$$

$C = C1 \| C2 \| \ldots \| Cn$

Electronic Code Book - ECB encrypt.

(a) plaintext

(b) plaintext encrypted in ECB mode using AES



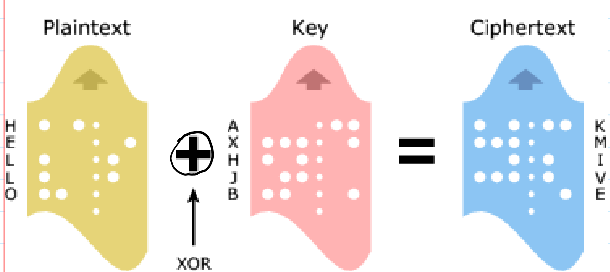
Original image

Encrypted using ECB mode

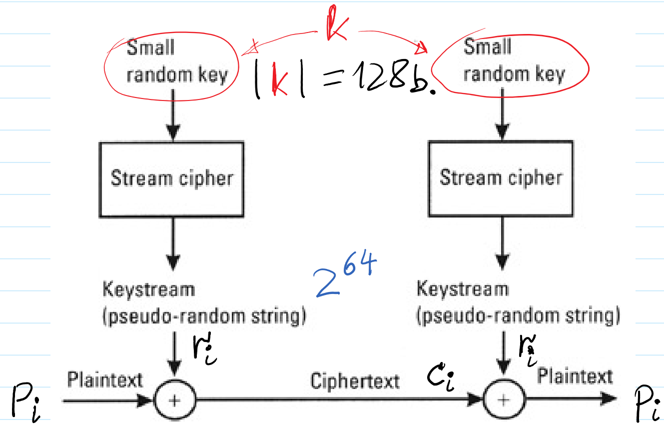
Modes other than ECB result in pseudo-randomness

CBC, CTR modes of encr.

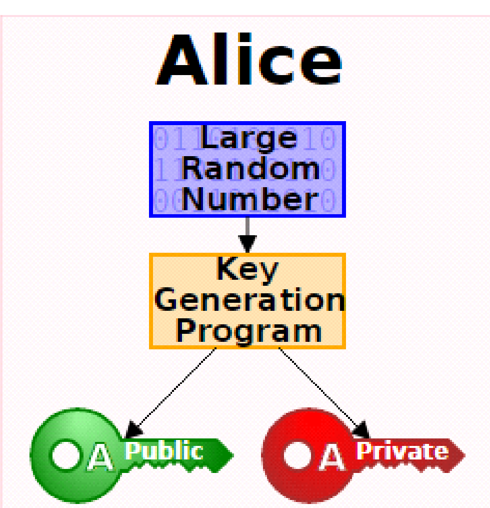
Stream Cipher - Vernam Cipher - One-Time Pad



bitwise



Asymmetric cryptography



PrK and **PuK** are related

PuK = **F**(**PrK**)

F is one-way function - OWF:

It is erasy to compute **PuK** when **F** and **PrK** are given.

Kerchoff principe.

Having **PuK** and **F**, it is infeasible to find **PrK** = $\mathbf{F}^{-1}$(**PuK**).

Public Parameters PP = (p, g) $p \sim 2^{2048} \approx 10^{760}$; $|p| = 2048$ b.
= 760 dec. digits

We will use $|p| = 28$ bits.
To generate PrK and PuK we need to generate PP = (p, g)

PrK = *x* <-- randi ==> PuK = *a* = g^x mod p

Open SSL software
Python
Go

|PrK| = 2048 bits
|PuK| = 2048 bits
$[1, 2^{2048}]$

RSA cryptosystem
Rvest-Shamir-Addleman

Encryption, Signature, Masking: for confidential e-money withdrawing
for confidential e-voting

Multiplication Tab.	Z15													
*	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	2	4	6	8	10	12	14	1	3	5	7	9	11	13
3	3	6	9	12	0	3	6	9	12	0	3	6	9	12
4	4	8	12	1	5	9	13	2	6	10	14	3	7	11
5	5	10	0	5	10	0	5	10	0	5	10	0	5	10
6	6	12	3	9	0	6	12	3	9	0	6	12	3	9
7	7	14	6	13	5	12	4	11	3	10	2	9	1	8
8	8	1	9	2	10	3	11	4	12	5	13	6	14	7
9	9	3	12	6	0	9	3	12	6	0	9	3	12	6
10	10	5	0	10	5	0	10	5	0	10	5	0	10	5
11	11	7	3	14	10	6	2	13	9	5	1	12	8	4
12	12	9	6	3	0	12	9	6	3	0	12	9	6	3
13	13	11	9	7	5	3	1	14	12	10	8	6	4	2
14	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Exponent Tab.	**Z15**														
^	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	4	8	1	2	4	8	1	2	4	8	1	2	4
3	1	3	9	12	6	3	9	12	6	3	9	12	6	3	9
4	1	4	1	4	1	4	1	4	1	4	1	4	1	4	1
5	1	5	10	5	10	5	10	5	10	5	10	5	10	5	10
6	1	6	6	6	6	6	6	6	6	6	6	6	6	6	6

7	1	7	4	13	1	7	4	13	1	7	4	13	1	7	4
8	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4
9	1	9	6	9	6	9	6	9	6	9	6	9	6	9	6
10	1	10	10	10	10	10	10	10	10	10	10	10	10	10	10
11	1	11	1	11	1	11	1	11	1	11	1	11	1	11	1
12	1	12	9	3	6	12	9	3	6	12	9	3	6	12	9
13	1	13	4	7	1	13	4	7	1	13	4	7	1	13	4
14	1	14	1	14	1	14	1	14	1	14	1	14	1	14	1

1. Parameters generation: $p \leftarrow$ prime; $q \leftarrow$ prime: generated at random

$n = p \cdot q$ – RSA module

$\mathcal{Z}_n = \{0, 1, 2, \ldots, n-1\}$; Let $z \in \mathcal{Z}_n$

If $\gcd(z, n) = 1 \Rightarrow \exists!\ z^{-1}$ such that $z \cdot z^{-1} \bmod n = 1$

E.g. Let $n = 15$; $z = 2 \Rightarrow z^{-1} \bmod 15 = 8$ since $2 \cdot 8 \bmod 15 = 1$

The numbe of numbers having inverse values in $\mathcal{Z}_n$ is defined by Euler Totient function $\phi(n) = \phi \equiv fy$.

If $n = p \cdot q$, when p, q – primes $\Rightarrow \phi(n) = \phi = (p-1) \cdot (q-1)$

2. Generate number e – exponent of RSA, such that $\gcd(e, \phi) = 1 \Rightarrow e$ has its multiplicative inverse element $e^{-1} \bmod \phi$.

According to RSA standard $e = 2^{16} + 1$.

3. Generate $PrK = d \Rightarrow d = e^{-1} \bmod \phi$.

Security considerations: if p, q – are large primes $\Rightarrow$ $n = p \cdot q$ – is large. To find $PrK = d$ it is necessary to factor n. When n – is large it is infeasible with classical computers.

>> factor(n) $\rightarrow$ $p, q \rightarrow \phi = (p-1) \cdot (q-1) \rightarrow$

>> mulinv$(e, \phi) = d$.

Security relies on the complexity of factorization problem.

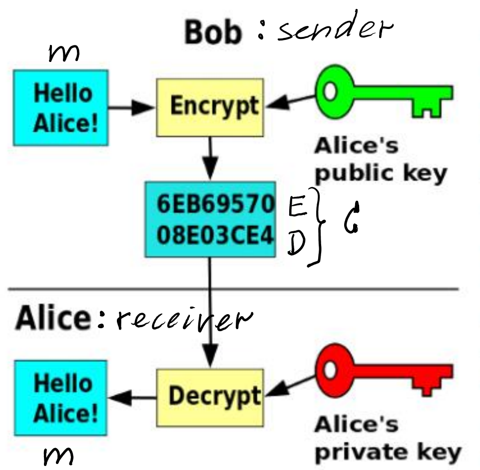
Euler theorem. If gcd(***z***,***n***)=1 then

$$z^{\Phi} = 1 \bmod n$$

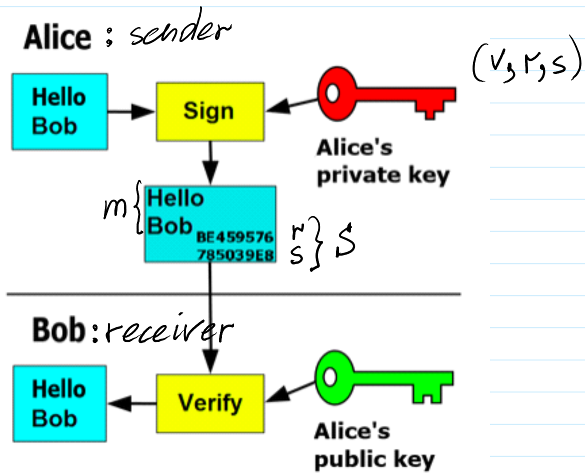
According to Euler theorem exponents are computing mod ϕ.

RSA: $PuK=(n, e)$; $PrK=d$.

Asymmetric Encryption - Decryption
c=Enc(PuK_A, m)
m=Dec(PrK_A, c)



Asymmetric Signing - Verification
S=Sign(PrK_A, m)
V=Ver(PuK_A, m, s), V∈{True, False} ≡ {1, 0}



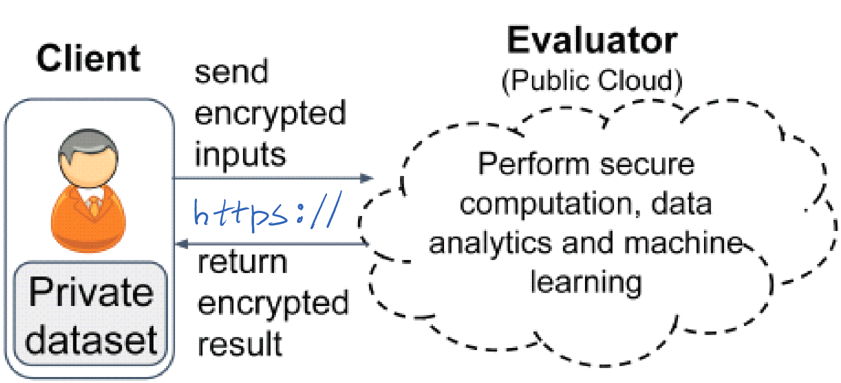
Encryption: $c = m^e \bmod n$

Decryption: $c^d \bmod n =$

$= (m^e)^d \bmod n = m^{ed \bmod \phi} \bmod n =$

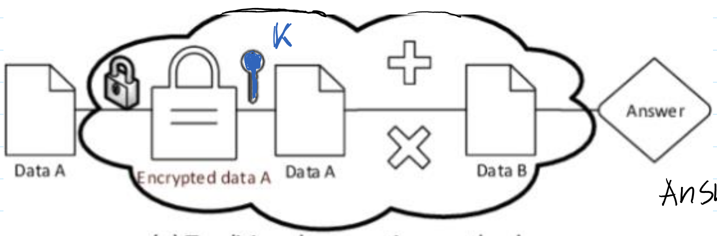
$= m^1 \bmod n \overset{m<n}{=\!=\!=} m$

Till this place



Database Encryption

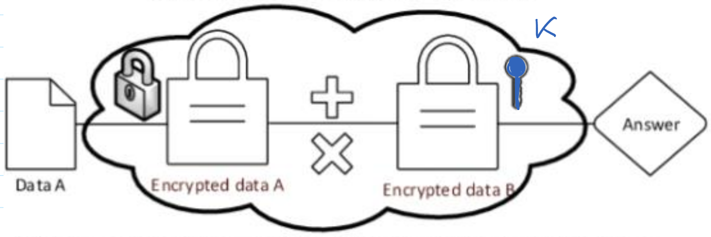
Fully Homomorphic Encryption



(a) Traditional encryption methods

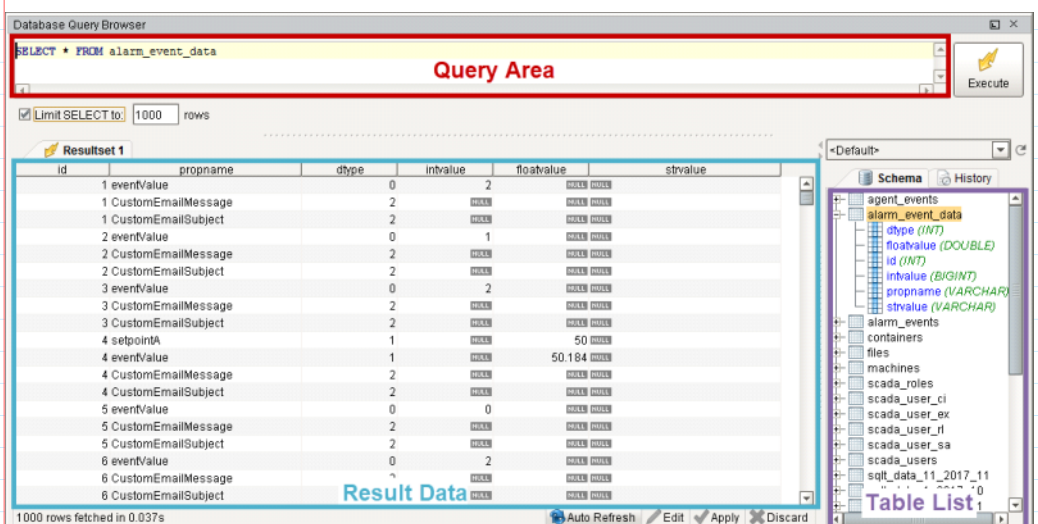
$\to Enc(K, Answer) = C$

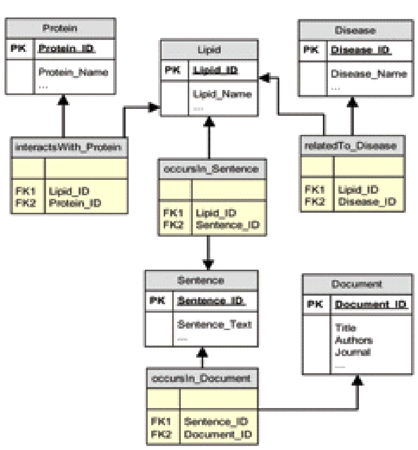
$Answer = Dec(K, C)$



(b) Our proposed homomorphic encryption based method

Database Query






```
SELECT * FROM
        Lipid L,
        Protein P,
        Disease D,
        Sentence S,
        Document DOC,
        interactsWith_Protein I,
        occursIn_Sentence O,
        relatedTo_Disease R,
        occursIn_Document OD
WHERE
        L.LipidID = O.Lipid_ID AND
        L.LipidID = I.Lipid_ID AND
        L.LipidID = R.Lipid_ID AND
        P.Protein_ID = I.Protein_ID AND
        D.Disease_ID = R.Disease_ID AND
        S.Sentence_ID = O.Sentence_ID AND
        S.Sentence_ID = OD.Sentence_ID AND
        DOC.Document_ID = OD.DocumentID;
```

Primary Key

Search in Database is performed in the fields which are ordered.

Id		Id	
3, 15, 7, 1, 4, 8, ...	ordering → $O(\log_2 n)$, $\log_2 2^{32} = 32$	1, 2, 3, 4, ..., 7, 8, ...	← search $O\log_2(n)$ 32-steps

n - records

$n \sim 2^{32}$

Order-Revealing Encryption - OREnc 2020 MDPI Symmetry 2.6...

Database encryption has received increased attention recently due to the enormous amount of sensitive data stored in outsourcing cloud databases. One of promising solutions to protect the confidentiality of sensitive data is to use encryption and **performing query evaluation over encrypted data**.

Order-Preserving Encryption. Property-preserving encryption which preserves some property of plaintexts enables performing query evaluation on ciphertexts. Among them, order-preserving encryption (OPEnc) whose ciphertexts preserve the numerical ordering of their underlying plaintexts has received a lot of attention since it can support efficient query operation on encrypted data such as sorting and range queries using the ordering information. In 2004, Agrawal et al. first proposed the concept of OPEnc. Later, Boldyreva et al. provided the security notions of OPEnc formally and also showed that any immutable OPEnc schemes with ideal security must have the ciphertext length which grows exponentially in the plaintext length. Recently, some ideally-secure OPEnc schemes whose ciphertexts reveal no additional information beyond the order of the underlying plaintexts have been proposed. However, these schemes require large communication and storage complexities.

A new ideally-secure OREncS scheme with shorter ciphertexts is proposed in 2020. Combining it with the domain-extension scheme the new OREncL scheme with shorter ciphertexts under the same security level is obtained ...