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

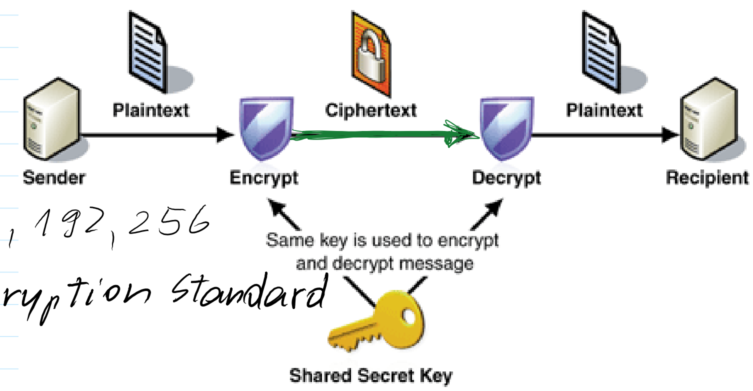
Diffie, Hellman. New trend ...

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

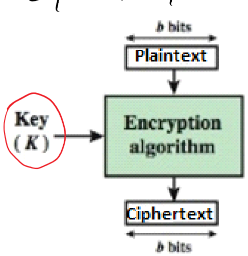


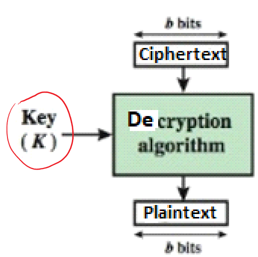
AES - 128, 192, 256
Advanced Encryption Standard
~ 2000 m.

Symmetric ciphers

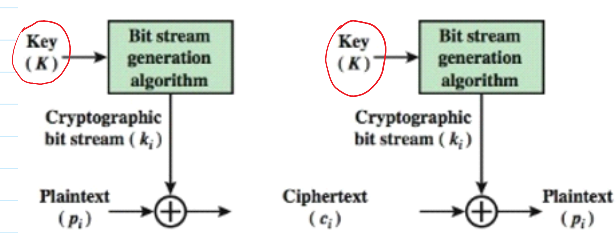
AES: Block Ciphers

128, 196, 256





Stream Ciphers



bitwise XOR

Vernam Cipher (1917)

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

$c = m \oplus k$ —— c ——>

B: $k = 1$.

$m = c - k$

$\oplus$ - is selfinverse

SI: $m \in \{0,1\}, k \leftarrow rand\{0,1\}$.

$c = m \oplus k$ —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 -"-

$\ldots\ldots k_2 k_1 k_0$

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

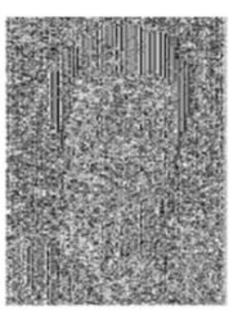
B1	B2	B3	----	Bi		Bn

The length of any block Bi should be $|B_i| = 128$ bits
$|B_i| = |k| = 128$ bits $= 2^7$ bits — 192 bits, 256 bits

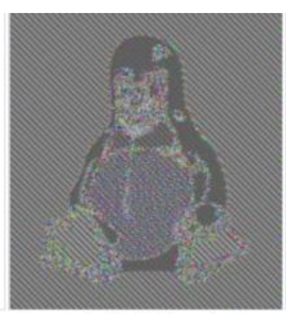
$EncAES(k, B1) = C1$
$EncAES(k, B2) = C2$
- - - - - - - - -
$EncAES(k, Bn) = Cn$

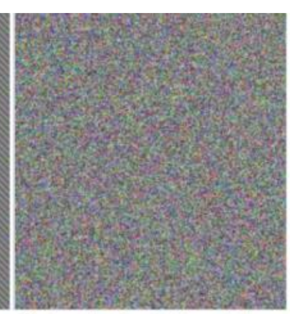
$C = C1 \| C2 \| \ldots \| C_n$
Electronic Code Book - ECB encrypt.


(a) plaintext


(b) plaintext encrypted in ECB mode using AES


Original image

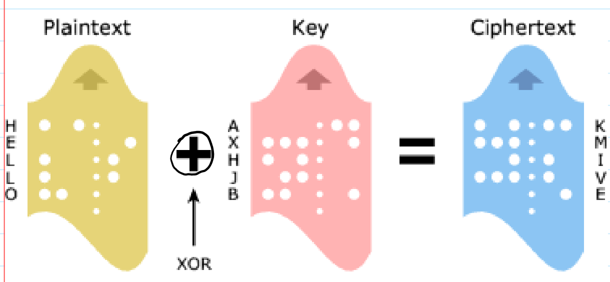

Encrypted using ECB mode

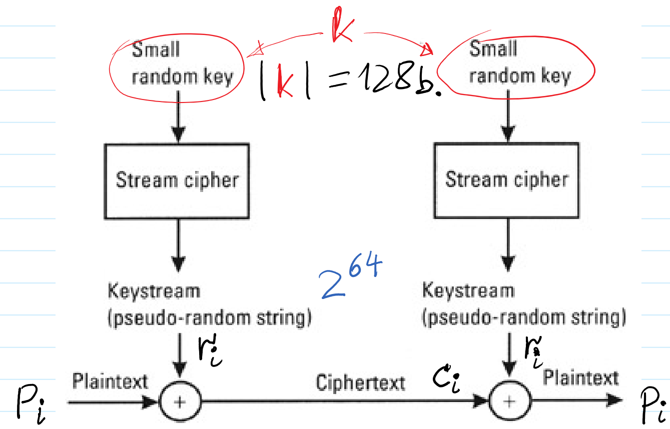

Modes other than ECB result in pseudo-randomness

CBC, CTR

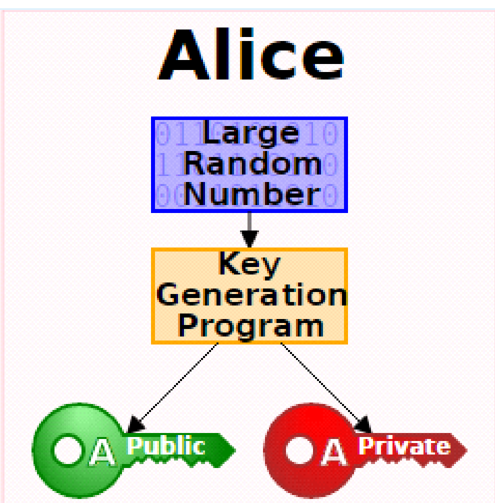
CBC, CTR
modes of encr.

Stream Cipher - Vernam Cipher - One-Time Pad





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 = 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)

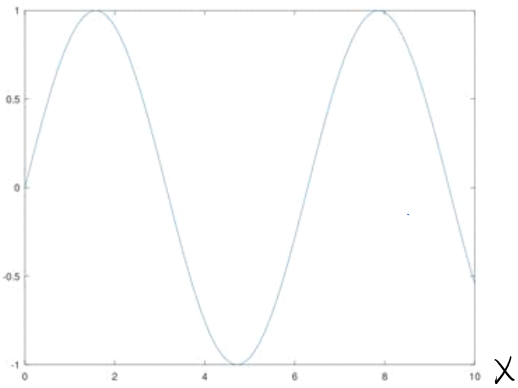
PrK = *x* <-- randi ==> PuK = *a* = g^x mod p

Open SSL software
Python
Go

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

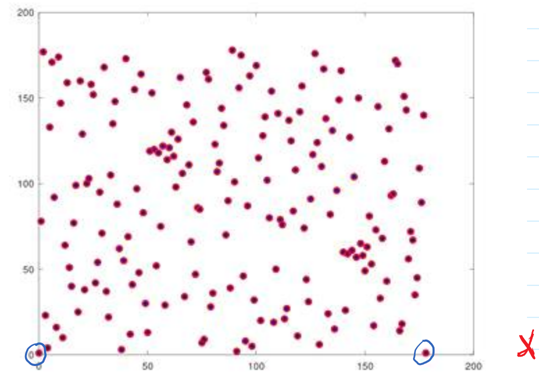
```
>> p=genstrongprime(28)
```

$y = \sin x$

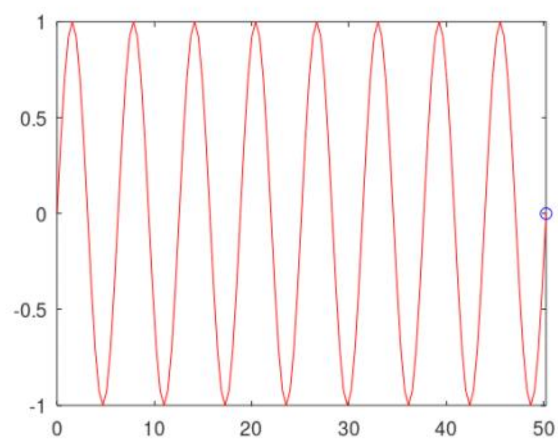


$a = g^x \bmod p$

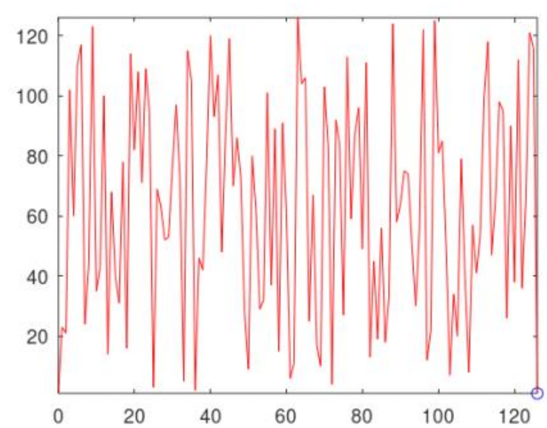
$p = 179$




```
>> pi
ans = 3.1416
>> xrange=16*pi
xrange = 50.265
>> step=xrange/128
step = 0.3927
>> x=0:step:xrange;
>> y=sin(x);
>> comet(x,y)
```




```
>> p=127
p = 127
>> g = 23
g = 23
>> x=0:p-1;
>> a=mod_expv(g,x,p)
>> comet(x,a)
```



Public Parameters $PP = (p, g)$

p – strong prime number: $p = 2 \cdot q + 1$, where p – is prime and q – is prime.

$q = 3$: $p = 2 \cdot 3 + 1 = 7$ – is strong prime

$q = 5$: $p = 2 \cdot 5 + 1 = 11$ – is strong prime

$q = 7$: $p = 2 \cdot 7 + 1 = 15$ – ~~is strong prime~~ not prime.

$q = 5$: $p = 2 \cdot 5 + 1 = 11$ – is strong prime

$q = 7$: $p = 2 \cdot 7 + 1 = 15$ – ~~is strong prime~~ not prime.

```
>> p=genstrongprime(28)
p = 204105323
>> q=(p-1)/2
q = 102052661
>> isprime(q)
ans = 1
>> isprime(p)
ans = 1
>> p=2*q+1
p = 204105323

>> mod(17,11)
ans = 6
```

Cryptographic functions are defined in the set $\mathbb{Z}_p^* = \{1, 2, 3, \dots, p-1\}$

$*\bmod p$ & $:\bmod p$

E.g. if $p = 11 \Rightarrow 17 \bmod 11 = 6$

$$\begin{array}{r|l} 17 & 11 \\ 11 & 1 \\ \hline 6 & \end{array}$$

$\mathbb{Z}_{11}^* = \{1, 2, 3, \dots, 10\}$

$*\bmod 11$ & $:\bmod 11$

Multiplication Tab. Z_{11}*

*	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	1	3	5	7	9
3	3	6	9	1	4	7	10	2	5	8
4	4	8	1	5	9	2	6	10	3	7
5	5	10	4	9	3	8	2	7	1	6
6	6	1	7	2	8	3	9	4	10	5
7	7	3	10	6	2	9	5	1	8	4
8	8	5	2	10	7	4	1	9	6	3
9	9	7	5	3	1	10	8	6	4	2
10	10	9	8	7	6	5	4	3	2	1

Power Tab. Z_{11}*

^	0	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	4	8	5	10	9	7	3	6	1
3	1	3	9	5	4	1	3	9	5	4	1
4	1	4	5	9	3	1	4	5	9	3	1
5	1	5	3	4	9	1	5	3	4	9	1
6	1	6	3	7	9	10	5	8	4	2	1
7	1	7	5	2	3	10	4	6	9	8	1
8	1	8	9	6	4	10	3	2	5	7	1
9	1	9	4	3	5	1	9	4	3	5	1
10	1	10	1	10	1	10	1	10	1	10	1

If p is prime $\Rightarrow z^{p-1} = 1 \bmod p$

$\Gamma = \{2\ 6\ 7\ 8\}$; $|\Gamma| = 4$

Probability to find a generator in $\mathbb{Z}_p^* = \{1, 2, 3, \dots, p-1\}$ is about 0,4 ~ 40%.

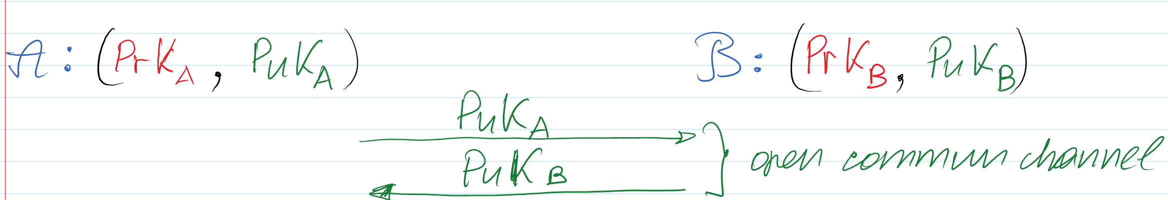
Till this plaace

C.5.3 Finding generators.

We have to look inside Z_P* and find a generator. How?
Even if we have a candidate, how do we test it?
The condition is that <g> = G which would take |G| steps to check: p~2^{2048} --> |G|~2^{2048}.
In fact, finding a generator given p is in general a hard problem.

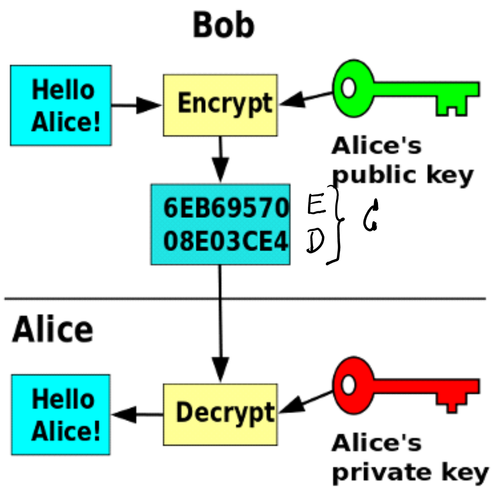
We can exploit the particular prime numbers names as **strong primes**.
If p is prime and p=2q+1 with q prime then p is a **strong prime**.
Note that the order of the group Z_P* is p-1=2q, i.e. |Z_P*|=2q.
Fact C.23. Say p=2q+1 is **strong prime** where q = (p-1)/2 is prime, then g in Z_P* is a generator of Z_P*
iff
$g^2 \neq 1 \bmod p$ and $g^q \neq 1 \bmod p$.
Testing whether g is a generator is easy given strong prime p.
Now, given p=2q+1, the generator can be found by randomly generation numbers g<p and verifying two relations.The probability to find a generator is ~0.4.

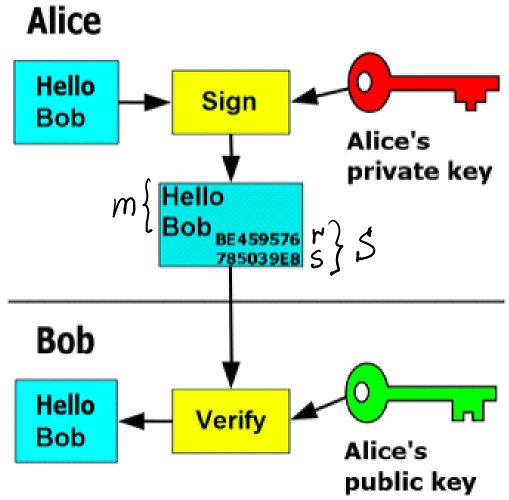
How to fing more generators when g one is found?
Fact C.24. If g is a generator and i is not divisible by q and **2** then g^i is a generator as well, i.e.
If g is a generator and gcd(i,q)=1 and gcd(i,2)=1, then g^i is a generator as well.



Asymmetric Encryption - Decryption
C=Enc(PuK$_A$, m)
m=Dec(PrK$_A$, c)

Asymmetric Signing - Verification
S=Sig(PrK$_A$, m)
V=Ver(PuK$_A$, m, s), V$\in$ {True, False} $\equiv$ {1, 0}





(v, r, s)

Micro SD - Software Development Kit (SDK)

PrK — Crypto software

Decription data: ciphertex c
Signing data: message m

} Taiwan GoTrust

IoT
Smart contracts
Initial coin offer (ICO)

} Blockchain

Dan Boneh
Zether
↓
DRM

Zero Knowledg Proof (ZKP) — China — Zcoin — 2020 m.

Proof-of-work (PoW) → Proof-of-stake (PoS)

Zether: Towards Privacy in a Smart Contract World

Benedikt Bünz[1], Shashank Agrawal[2], Mahdi Zamani[3], and Dan Boneh[4]

[1]Stanford University, benedikt@cs.stanford.edu
[2]Visa Research, shaagraw@visa.com
[3]Visa Research, mzamani@visa.com
[4]Stanford University, dabo@cs.stanford.edu

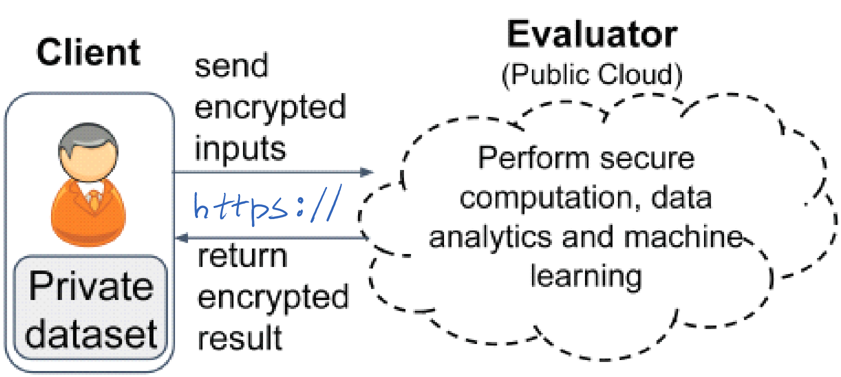
Zether: Towards Privacy in a Smart Contract World

Benedikt Bunz1, Shashank Agrawal2, Mahdi Zamani3, and **Dan Boneh**4

1Stanford University, benedikt@cs.stanford.edu
2Visa Research, shaagraw@visa.com
3Visa Research, mzamani@visa.com
4**Stanford University**, dabo@cs.stanford.edu

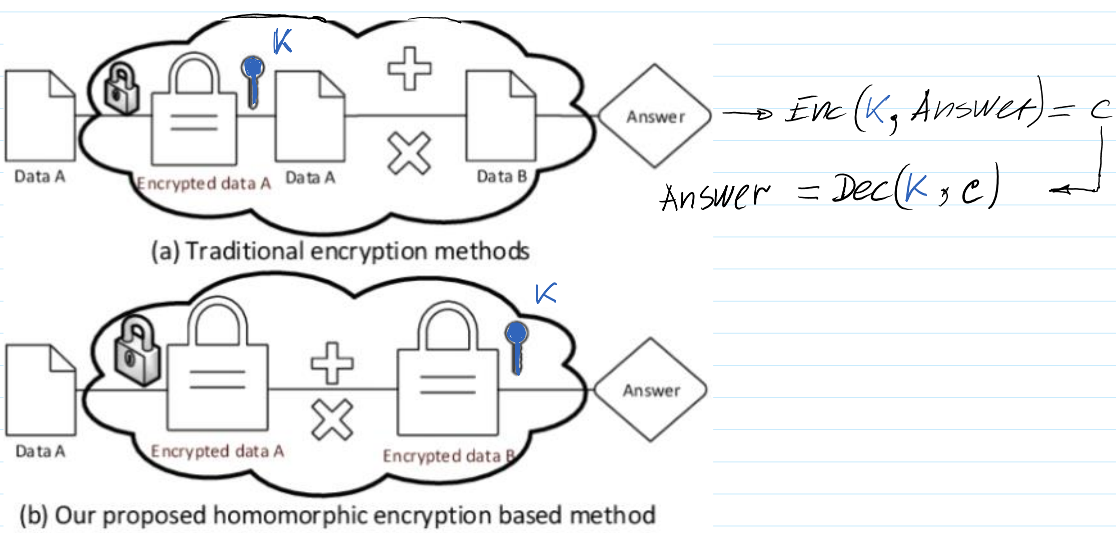
Ctrl/F --> ElGamal --> Exact mathes **21**

Iki čia

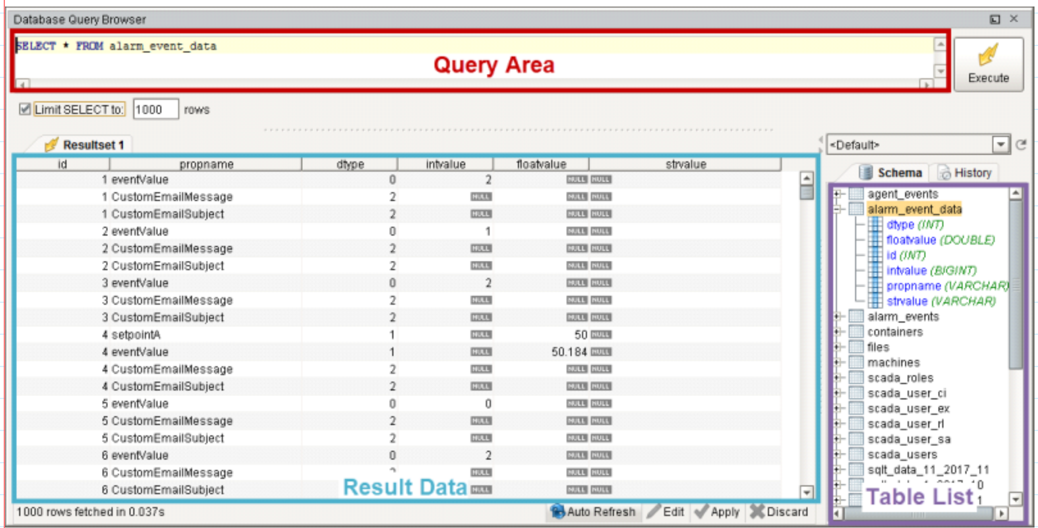


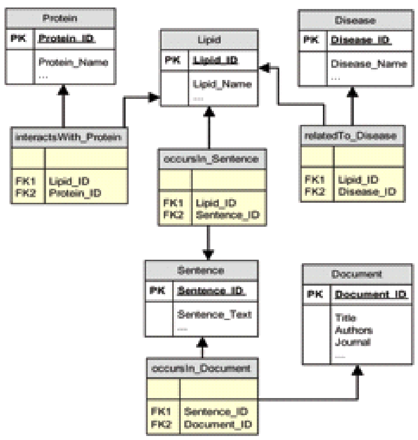
Database Encryption

Fully Homomorphic Encryption



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

n - records

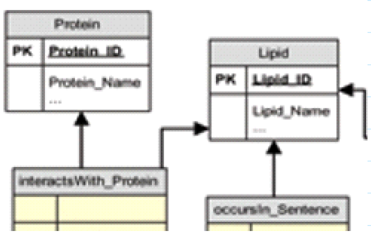
$n \sim 2^{32}$

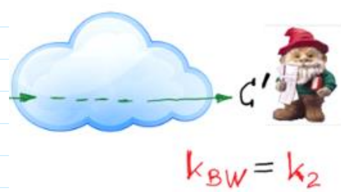
Order-Revealing Encryption - OREnc   2020   MDPI Symmetry 2.6...

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 ...





$k_{BW} = k_2$