Security for Data Scientists

Pascal Lafourcade

LIMOS

UNIVERSITÉ Clermont Auvergne

Mars 2018

Outline

Different Adversaries

Intuition of Computational Security

Cloud Security

Partial and Full Homomorphic Encryption

Secure Matrix Multiplication

SSE

Privacy in DB

Conclusion

Which adversary?

Adversary Model

Qualities of the adversary:

- Clever: Can perform all operations he wants
- Limited time:
  - Do not consider attack in 2^{60}.
  - Otherwise a Brute force by enumeration is always possible.

Model used: **Any Turing Machine**.

- Represents all possible algorithms.
- Probabilistic: adversary can generates keys, random number...

Adversary Models

The adversary is given access to oracles :

→ encryption of all messages of his choice
→ decryption of all messages of his choice

Three classical security levels:

- Chosen-Plain-text Attacks (CPA)

- Non adaptive Chosen-Cipher-text Attacks (CCA1)
  only before the challenge

- Adaptive Chosen-Cipher-text Attacks (CCA2)
  unlimited access to the oracle (except for the challenge)

Chosen-Plain-text Attacks (CPA)

Adversary can obtain all cipher-texts from any plain-texts.
It is always the case with a Public Encryption scheme.

Non adaptive Chosen-Cipher-text Attacks (CCA1)

Adversary knows the public key, has access to a **decryption oracle multiple times before to get the challenge** (cipher-text), also called “Lunchtime Attack” introduced by M. Naor and M. Yung ([NY90]).

Adaptive Chosen-Cipher-text Attacks (CCA2)

Adversary knows the public key, has access to a **decryption oracle multiple times before and AFTER to get the challenge**, but of course cannot decrypt the challenge (cipher-text) introduced by C. Rackoff and D. Simon ([RS92]).

Summary of Adversaries

CCA2: $\mathcal{O}_1 = \mathcal{O}_2 = \{\mathcal{D}\}$ Adaptive Chosen Cipher text Attack

$\Downarrow$

CCA1: $\mathcal{O}_1 = \{\mathcal{D}\}$, $\mathcal{O}_2 = \emptyset$ Non-adaptive Chosen Cipher-text Attack

$\Downarrow$

CPA: $\mathcal{O}_1 = \mathcal{O}_2 = \emptyset$ Chosen Plain text Attack

Outline

Different Adversaries

Intuition of Computational Security

Cloud Security

Partial and Full Homomorphic Encryption

Secure Matrix Multiplication

SSE

Privacy in DB

Conclusion

One-Wayness (OW)

Put your message in a translucent bag, but you cannot read the text.

One-Wayness (OW)

Put your message in a translucent bag, but you cannot read the text.

Without the private key, it is computationally **impossible to recover the plain-text**.

RSA Is it preserving your privacy?

RSA Is it preserving your privacy?

4096 RSA encryption

RSA Is it preserving your privacy?

4096 RSA encryption

Environs 60 températures possibles: 35 ... 41

RSA Is it preserving your privacy?

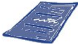
4096 RSA encryption

Environs 60 températures possibles: 35 ... 41

$$\{35\}_{pk}, \{35,1\}_{pk}, ..., \{41\}_{pk}$$

Is it secure ?

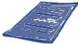
Is it secure ?



Is it secure ?

- you cannot read the text but you can distinguish which one has been encrypted.

Is it secure ?



- you cannot read the text but you can distinguish which one has been encrypted.
- Does not exclude to recover half of the plain-text
- Even worse if one has already partial information of the message:
  - Subject: XXXX
  - From: XXXX

Indistinguishability (IND)

Put your message in a black bag, you can not read anything.

Now a black bag is of course IND and it implies OW.

Indistinguishability (IND)

Put your message in a black bag, you can not read anything.

Now a black bag is of course IND and it implies OW.
The adversary is not able to **guess in polynomial-time even a bit of the plain-text knowing the cipher-text**, notion introduced by S. Goldwasser and S.Micali ([GM84]).

Is it secure?

Is it secure?

Is it secure?

- It is possible to scramble it in order to produce a new cipher. In more you know the relation between the two plain text because you know the moves you have done.

Non Malleability (NM)

Put your message in a black box.

But in a black box you cannot touch the cube (message), hence NM implies IND.

Non Malleability (NM)

Put your message in a black box.

But in a black box you cannot touch the cube (message), hence NM implies IND.
The adversary should **not be able to produce a new cipher-text** such that the plain-texts are meaningfully related, notion introduced by D. Dolev, C. Dwork and M. Naor in 1991 ([DDN91,BDPR98,BS99]).

Summary of Security Notions

Non Malleability

$\Downarrow$

Indistinguishability

$\Downarrow$

One-Wayness

Outline

Different Adversaries

Intuition of Computational Security

Cloud Security

Partial and Full Homomorphic Encryption

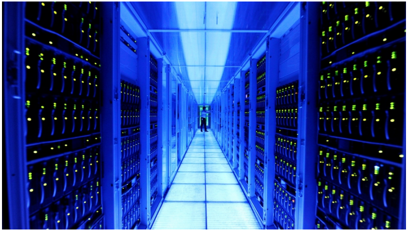
Secure Matrix Multiplication

SSE

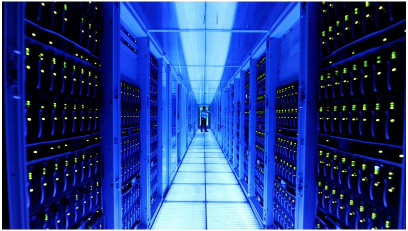
Privacy in DB

Conclusion

Should we trust our remote storage?



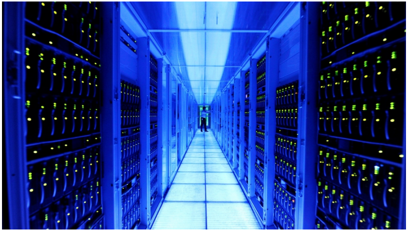
Should we trust our remote storage?



Many reasons not to

- Outsourced backups and storage
- Sysadmins have root access
- Hackers breaking in

Should we trust our remote storage?



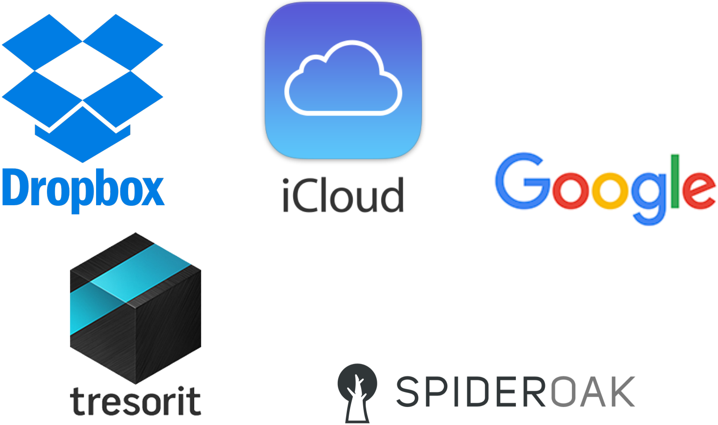
Many reasons not to

- Outsourced backups and storage
- Sysadmins have root access
- Hackers breaking in

Solution: ☁

Clouds

Clouds



Properties

Acces from everywhere

Avaible for everything:

- Store documents, photos, etc
- Share them with colleagues, friends, family
- Process the data
- Ask queries on the data

Current solutions

Cloud provider knows the content and claims to actually

- identify users and apply access rights
- safely store the data
- securely process the data
- protect privacy

Users need more Storage and Privacy guarantees

- confidentiality of the data
- anonymity of the users
- obliviousness of the queries

Broadcast encryption (Fiat-Noar 1994)

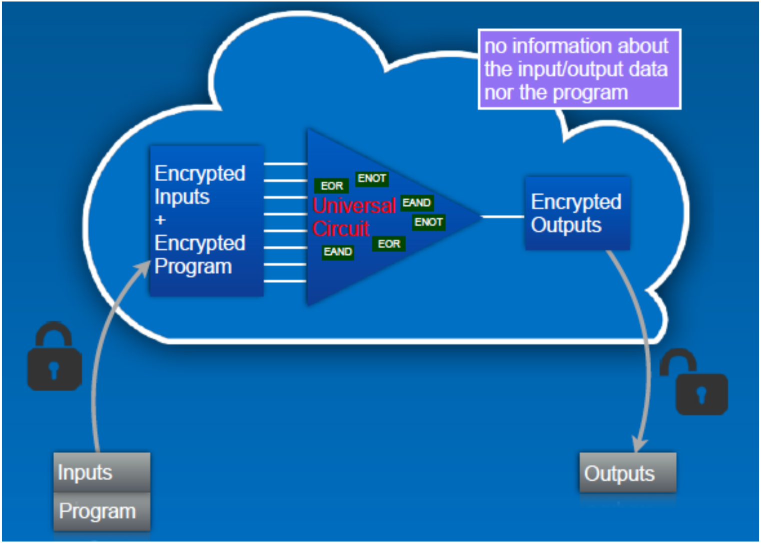
The sender can select the target group of receivers to control who access to the data like in PAYTV

Functional encryption [Boneh-Sahai-Waters 2011]

The user generates sub-keys K_y according to the input y to control the amount of shared data.

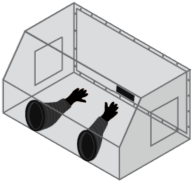
From $C = \mathit{Encrypt}(x)$, then $\mathit{Decrypt}(K_y, C)$, outputs $f(x, y)$

Fully Homomorphic Encryption [Gentry 2009]


no information about the input/output data nor the program
Encrypted Inputs + Encrypted Program
EOR
ENOT
Universal Circuit
EAND
ENOT
EOR
EAND
Encrypted Outputs
Inputs
Program
Outputs

Fully Homomorphic Encryption [Gentry 2009]

FHE: encrypt data, allow manipulation over data.
Symmetric Encryption (secret key) is enough



$$f(\{x_1\}_K, \{x_2\}_K, \ldots, \{x_n\}_K) = \{f(x_1, x_2, \ldots, x_n)\}_K$$

- Allows private storage
- Allows private computations
- Private queries in an encrypted database
- Private search: without leaking the content, queries and answers.

Outline

Different Adversaries

Intuition of Computational Security

Cloud Security

Partial and Full Homomorphic Encryption

Secure Matrix Multiplication

SSE

Privacy in DB

Conclusion

Rivest Adleman Dertouzos 1978

"Going beyond the storage/retrieval of encrypted data by permitting encrypted data to be operated on for interesting operations, in a public fashion?"

Partial Homomorphic Encryption

Definition (additively homomorphic)

$$E(m_1) \otimes E(m_2) \equiv E(m_1 \oplus m_2).$$

Applications

- Electronic voting
- Secure Fonction Evaluation
- Private Multi-Party Trust Computation
- Private Information Retrieval
- Private Searching
- Outsourcing of Computations (e.g., Secure Cloud Computing)
- Private Smart Metering and Smart Billing
- Privacy-Preserving Face Recognition
- . . .

Brief history of partially homomorphic cryptosystems

$$Enc(a, k) * Enc(b, k) = Enc(a * b, k)$$

Year	Name	Security hypothesis	Expansion
1977	RSA	factorization	
1982	Goldwasser - Micali	quadratic residuosity	$\log_2(n)$
1994	Benaloh	higher residuosity	> 2
1998	Naccache - Stern	higher residuosity	> 2
1998	Okamoto - Uchiyama	p-subgroup	3
1999	Paillier	composite residuosity	2
2001	Damgaard - Jurik	composite residuosity	$\frac{d+1}{d}$
2005	Boneh - Goh - Nissim	ECC Log	
2010	Aguilar-Gaborit-Herranz	SIVP integer lattices	

Expansion factor is the ration ciphertext over plaintext.

Scheme Unpadded RSA

If the RSA public key is modulus m and exponent e, then the encryption of a message x is given by

$$\mathcal{E}(x) = x^e \mod m$$

$$\begin{aligned} \mathcal{E}(x_1) \cdot \mathcal{E}(x_2) &= x_1^e x_2^e \mod m \\ &= (x_1 x_2)^e \mod m \\ &= \mathcal{E}(x_1 \cdot x_2) \end{aligned}$$

Scheme ElGamal

In the ElGamal cryptosystem, in a cyclic group G of order q with generator g, if the public key is (G, q, g, h), where $h = g^x$ and x is the secret key, then the encryption of a message m is $\mathcal{E}(m) = (g^r, m \cdot h^r)$, for some random $r \in \{0, \ldots, q-1\}$.

$$\begin{aligned}
\mathcal{E}(m_1) \cdot \mathcal{E}(m_2) &= (g^{r_1}, m_1 \cdot h^{r_1})(g^{r_2}, m_2 \cdot h^{r_2}) \\
&= (g^{r_1+r_2}, (m_1 \cdot m_2)h^{r_1+r_2}) \\
&= \mathcal{E}(m_1 \cdot m_2)
\end{aligned}$$

Fully Homomorphic Encryption

$$Enc(a,k) * Enc(b,k) = Enc(a*b,k)$$
$$Enc(a,k) + Enc(b,k) = Enc(a+b,k)$$
$$f(Enc(a,k), Enc(b,k)) = Enc(f(a,b),k)$$

Fully Homomorphic encryption

- Craig Gentry (STOC 2009) using lattices
- Marten van Dijk; Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan using integer
- Craig Gentry; Shai Halevi. "A Working Implementation of Fully Homomorphic Encryption"
- ...

Simple SHE: SGHV Scheme [vDGHV10]

Public error-free element : $x_0 = q_0 \cdot p$
Secret key $sk = p$

Encryption of $m \in \{0, 1\}$

$$c = q \cdot p + 2 \cdot r + m$$

where q is a large random and r a small random.

Simple SHE: SGHV Scheme [vDGHV10]

Public error-free element : $x_0 = q_0 \cdot p$
Secret key $sk = p$

Encryption of $m \in \{0, 1\}$

$$c = q \cdot p + 2 \cdot r + m$$

where q is a large random and r a small random.

Decryption of c

$$m = (c \mod p) \mod 2$$

Limitations

- Efficiency: HEtest: A Homomorphic Encryption Testing Framework (2015)

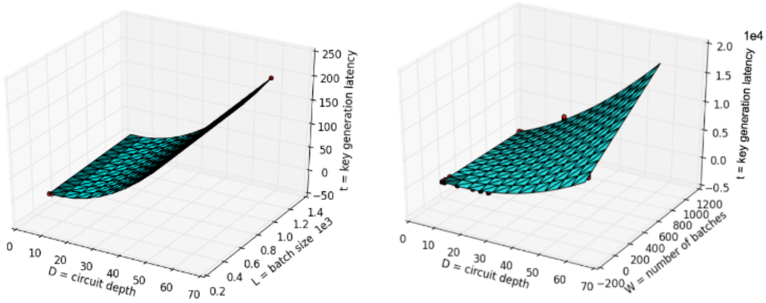


Fig. 9. Key generation time (left) and homomorphic evaluation time (right), in seconds

Outline

Different Adversaries

Intuition of Computational Security

Cloud Security

Partial and Full Homomorphic Encryption

Secure Matrix Multiplication

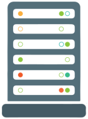
SSE

Privacy in DB

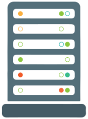
Conclusion

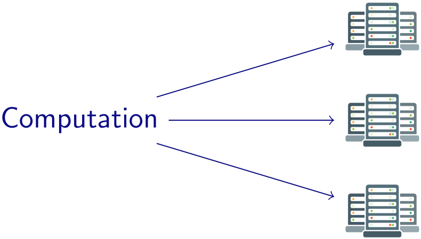
Context: The Big Data

Problem: How to process this amount of data?

“Amount of data” $>$ 

Problem: How to process this amount of data?

“Amount of data” > 



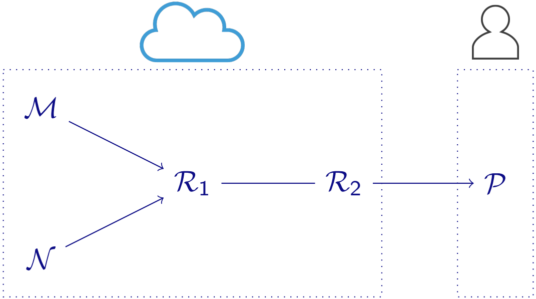
Matrix Multiplication with MapReduce

MapReduce Paradigm

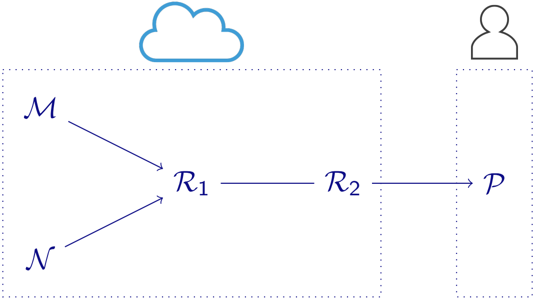
Google created *MapReduce* in 2004 to perform computation in the PageRank algorithm.

How it works?

Matrix Multiplication with MapReduce

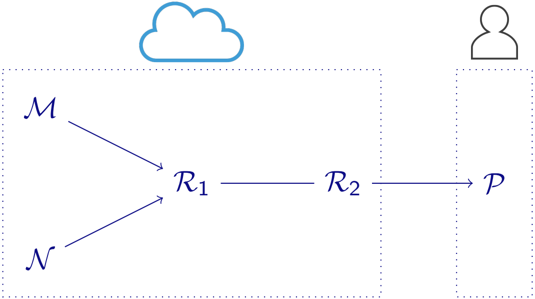


Matrix Multiplication with MapReduce



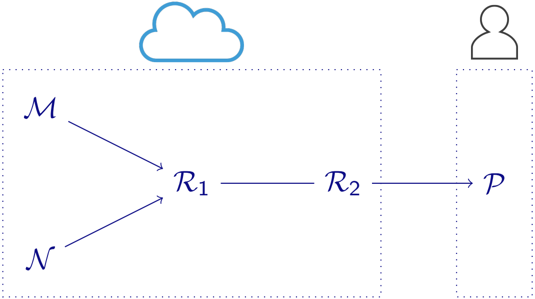
- $\mathcal{M}$ emits to $\mathcal{R}_1$: $\{(j,(M,i,m_{ij}))\}_{1\leq i\leq a,1\leq j\leq b}$

Matrix Multiplication with MapReduce



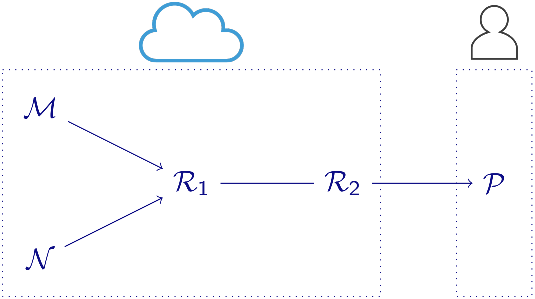
- $\mathcal{M}$ emits to $\mathcal{R}_1$: $\{(j,(M,i,m_{ij}))\}_{1\leq i\leq a, 1\leq j\leq b}$
- $\mathcal{N}$ emits to $\mathcal{R}_1$: $\{(j,(N,k,n_{jk}))\}_{1\leq j\leq b, 1\leq k\leq c}$

Matrix Multiplication with MapReduce



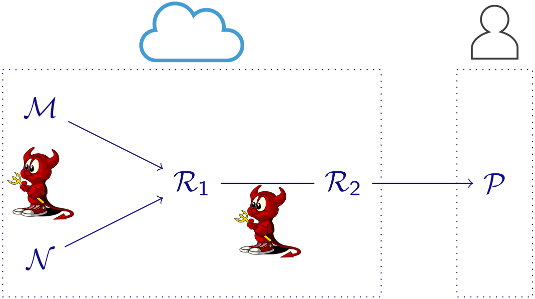
- $\mathcal{M}$ emits to $\mathcal{R}_1$: $\{(j,(M,i,m_{ij}))\}_{1\leq i\leq a, 1\leq j\leq b}$
- $\mathcal{N}$ emits to $\mathcal{R}_1$: $\{(j,(N,k,n_{jk}))\}_{1\leq j\leq b, 1\leq k\leq c}$
- $\mathcal{R}_1$ emits to $\mathcal{R}_2$: $\{((i,k),m_{ij}\cdot n_{jk})\}_{1\leq i\leq a, 1\leq j\leq b, 1\leq k\leq c}$

Matrix Multiplication with MapReduce

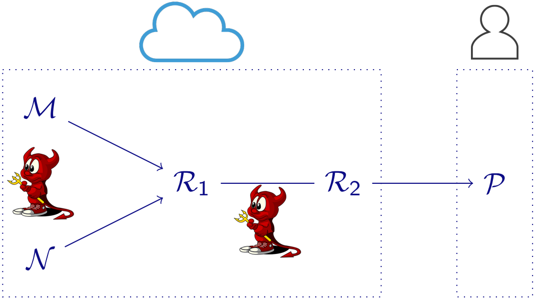


- $\mathcal{M}$ emits to $\mathcal{R}_1$: $\{(j,(M,i,m_{ij}))\}_{1\leq i\leq a,1\leq j\leq b}$
- $\mathcal{N}$ emits to $\mathcal{R}_1$: $\{(j,(N,k,n_{jk}))\}_{1\leq j\leq b,1\leq k\leq c}$
- $\mathcal{R}_1$ emits to $\mathcal{R}_2$: $\{((i,k),m_{ij}\cdot n_{jk})\}_{1\leq i\leq a,1\leq j\leq b,1\leq k\leq c}$
- $\mathcal{R}_2$ emits to $\mathcal{P}$: $\left\{p_{ik}=\sum_{j=1}^{b} m_{ij}\cdot n_{jk}\right\}_{1\leq i\leq a,1\leq k\leq c}$

Privacy Properties



Privacy Properties



P1 $\mathcal{M}$ cannot learn any information about matrices N and P

P2 $\mathcal{N}$ cannot learn any information about matrices M and P

P3 $\mathcal{R}_1$ and $\mathcal{R}_2$ cannot learn any information about matrices M, N, and P

Contributions

Two cryptographic approaches:

- *Secure-Private* (SP); assumes that nodes do not collude.
- *Collision-Resistant-Secure-Private* (CRSP).

Paillier's Cryptosystem

Key Generation

- **Public Key** $pk = (n, g)$
  - $n = p \cdot q$
  - $g \in \mathbb{Z}^*_{n^2}$
- **Secret Key** $sk = (\lambda, \mu)$
  - $\lambda = \mathrm{lcm(p-1, q-1)}$
  - $\mu = (L(g^\lambda \mod n^2))^{-1} \mod n$ where $L(x) = (x-1)/n$

Encryption $\mathcal{E}_{pk}(\cdot)$

$$\mathcal{E}_{pk}(m) = g^m \cdot r^n \mod n^2$$

Decryption $\mathcal{D}_{sk}(\cdot)$

$$\mathcal{D}_{sk}(c) = L(c^\lambda \mod n^2) \cdot \mu \mod n$$

Homomorphic Properties of the Paillier's Cryptosystem

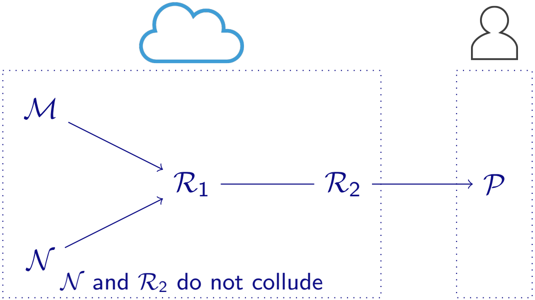
Homomorphic Addition of Plaintexts

$$\mathcal{E}_{pk}(m_1) \cdot \mathcal{E}_{pk}(m_2) = \mathcal{E}_{pk}(m_1 + m_2)$$

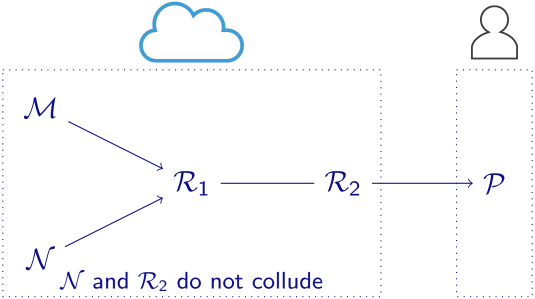
Specific Homomorphic Multiplication of Plaintexts

$$\mathcal{E}_{pk}(m_1)^{m_2} = \mathcal{E}_{pk}(m_1 \cdot m_2 \mod n)$$

Secure-Private Approach

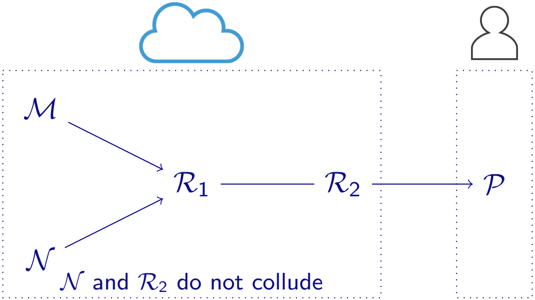


Secure-Private Approach



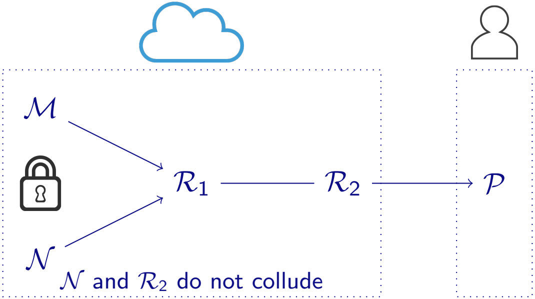
- $\mathcal{M}$ emits to $\mathcal{R}_1$: $(j, (M, i, \mathcal{E}_{pk}(m_{ij})))_{1 \leq i \leq a, 1 \leq j \leq b}$

Secure-Private Approach



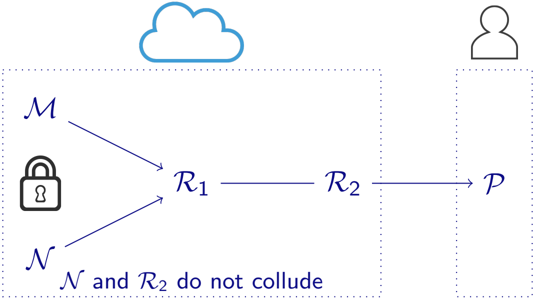
- $\mathcal{M}$ emits to $\mathcal{R}_1$: $(j, (M, i, \mathcal{E}_{pk}(m_{ij})))_{1\leq i\leq a, 1\leq j\leq b}$
- $\mathcal{N}$ emits to $\mathcal{R}_1$: $\left(j, (N, k, n_{jk} + \tau_{jk}, \mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk}))\right)_{1\leq j\leq b, 1\leq k\leq c}$

Secure-Private Approach



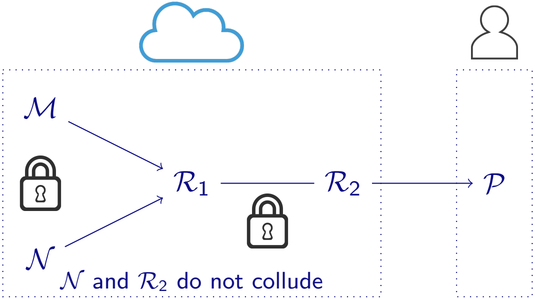
- $\mathcal{M}$ emits to $\mathcal{R}_1$: $(j, (M, i, \mathcal{E}_{pk}(m_{ij})))_{1 \leq i \leq a, 1 \leq j \leq b}$
- $\mathcal{N}$ emits to $\mathcal{R}_1$: $\left(j, (N, k, n_{jk} + \tau_{jk}, \mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk}))\right)_{1 \leq j \leq b, 1 \leq k \leq c}$

Secure-Private Approach



- $\mathcal{M}$ emits to $\mathcal{R}_1$: $(j,(M,i,\mathcal{E}_{pk}(m_{ij})))_{1\leq i\leq a,1\leq j\leq b}$
- $\mathcal{N}$ emits to $\mathcal{R}_1$: $\Big(j,(N,k,n_{jk}+\tau_{jk},\mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk}))\Big)_{1\leq j\leq b,1\leq k\leq c}$
- $\mathcal{R}_1$ emits to $\mathcal{R}_2$: $\Big((i,k),\mathcal{E}_{pk}(m_{ij})^{n_{jk}+\tau_{jk}},\mathcal{E}_{pk}(m_{ij}),\mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk})\Big)_{1\leq i\leq a,1\leq j\leq b,1\leq k\leq c}$

Secure-Private Approach



- $\mathcal{M}$ emits to $\mathcal{R}_1$: $(j, (M, i, \mathcal{E}_{pk}(m_{ij})))_{1\leq i\leq a, 1\leq j\leq b}$
- $\mathcal{N}$ emits to $\mathcal{R}_1$: $\Big(j, (N, k, n_{jk} + \tau_{jk}, \mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk}))\Big)_{1\leq j\leq b, 1\leq k\leq c}$
- $\mathcal{R}_1$ emits to $\mathcal{R}_2$: $\Big((i,k), \mathcal{E}_{pk}(m_{ij})^{n_{jk}+\tau_{jk}}, \mathcal{E}_{pk}(m_{ij}), \mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk})\Big)_{1\leq i\leq a, 1\leq j\leq b, 1\leq k\leq c}$

Secure-Private Approach

How the user retrieves the matrix P?

1. $\mathcal{R}_2$ receives:
$\Big((i,k), \mathcal{E}_{pk}(m_{ij})^{n_{jk}+\tau_{jk}}, \mathcal{E}_{pk}(m_{ij}), \mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk})\Big)_{1\leq i\leq a, 1\leq j\leq b, 1\leq k\leq c}$

Secure-Private Approach

How the user retrieves the matrix P?

1. $\mathcal{R}_2$ receives:
   $\Big((i,k), \mathcal{E}_{pk}(m_{ij})^{n_{jk}+\tau_{jk}}, \mathcal{E}_{pk}(m_{ij}), \mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk})\Big)_{1\leq i\leq a, 1\leq j\leq b, 1\leq k\leq c}$
2. For each (i,k), $\mathcal{R}_2$ computes:

$$\begin{aligned}\prod_{j=1}^{b} \frac{\mathcal{E}_{pk}(m_{ij})^{n_{jk}+\tau_{jk}}}{\mathcal{E}_{pk}(m_{ij})^{\mathcal{D}_{sk_{\mathcal{R}_2}}(\mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk}))}} &= \prod_{j=1}^{b} \mathcal{E}_{pk}(m_{ij} \cdot n_{jk}) \\ &= \mathcal{E}_{pk}(p_{ik})\end{aligned}$$

Secure-Private Approach

How the user retrieves the matrix P?

1. $\mathcal{R}_2$ receives:
   $\Big((i,k), \mathcal{E}_{pk}(m_{ij})^{n_{jk}+\tau_{jk}}, \mathcal{E}_{pk}(m_{ij}), \mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk})\Big)_{1\leq i\leq a, 1\leq j\leq b, 1\leq k\leq c}$
2. For each (i,k), $\mathcal{R}_2$ computes:

$$\begin{aligned}\prod_{j=1}^{b} \frac{\mathcal{E}_{pk}(m_{ij})^{n_{jk}+\tau_{jk}}}{\mathcal{E}_{pk}(m_{ij})^{\mathcal{D}_{sk_{\mathcal{R}_2}}(\mathcal{E}_{pk_{\mathcal{R}_2}}(\tau_{jk}))}} &= \prod_{j=1}^{b} \mathcal{E}_{pk}(m_{ij} \cdot n_{jk}) \\ &= \mathcal{E}_{pk}(p_{ik})\end{aligned}$$

3. Then, $\mathcal{R}_2$ sends to $\mathcal{P}$ all $\mathcal{E}_{pk}(p_{ik})$.

Interactive Homomorphic Multiplication of Ciphertexts [Cramer et al.]

How to compute $\mathcal{E}_{pk}(m_1 \cdot m_2)$?

Interactive Homomorphic Multiplication of Ciphertexts [Cramer et al.]

How to compute $\mathcal{E}_{pk}(m_1 \cdot m_2)$?

(sk, pk)

$c_1 = \mathcal{E}_{pk}(m_1)$ and $c_2 = \mathcal{E}_{pk}(m_2)$

Interactive Homomorphic Multiplication of Ciphertexts [Cramer et al.]

How to compute $\mathcal{E}_{pk}(m_1 \cdot m_2)$?

(sk, pk)

$c_1 = \mathcal{E}_{pk}(m_1)$ and $c_2 = \mathcal{E}_{pk}(m_2)$

Picks two randoms $\delta_1, \delta_2 \in \mathbb{Z}_n$

Interactive Homomorphic Multiplication of Ciphertexts [Cramer et al.]

How to compute $\mathcal{E}_{pk}(m_1 \cdot m_2)$?

(sk, pk)

$c_1 = \mathcal{E}_{pk}(m_1)$ and $c_2 = \mathcal{E}_{pk}(m_2)$

Picks two randoms $\delta_1, \delta_2 \in \mathbb{Z}_n$

$\alpha_1 = c_1 \cdot \mathcal{E}_{pk}(\delta_1)$

$\alpha_2 = c_2 \cdot \mathcal{E}_{pk}(\delta_2)$

Interactive Homomorphic Multiplication of Ciphertexts [Cramer et al.]

How to compute $\mathcal{E}_{pk}(m_1 \cdot m_2)$?

(sk, pk)

$c_1 = \mathcal{E}_{pk}(m_1)$ and $c_2 = \mathcal{E}_{pk}(m_2)$

Picks two randoms $\delta_1, \delta_2 \in \mathbb{Z}_n$

$$\alpha_1 = c_1 \cdot \mathcal{E}_{pk}(\delta_1)$$
$$\alpha_2 = c_2 \cdot \mathcal{E}_{pk}(\delta_2)$$

$$\xleftarrow{\alpha_1,\alpha_2}$$

$$\mathcal{D}_{sk}(\alpha_1) = m_1 + \delta_1$$
$$\mathcal{D}_{sk}(\alpha_2) = m_2 + \delta_2$$

Interactive Homomorphic Multiplication of Ciphertexts [Cramer et al.]

How to compute $\mathcal{E}_{pk}(m_1 \cdot m_2)$?

(sk, pk)

$c_1 = \mathcal{E}_{pk}(m_1)$ and $c_2 = \mathcal{E}_{pk}(m_2)$

Picks two randoms $\delta_1, \delta_2 \in \mathbb{Z}_n$

$\alpha_1 = c_1 \cdot \mathcal{E}_{pk}(\delta_1)$

$\alpha_2 = c_2 \cdot \mathcal{E}_{pk}(\delta_2)$

$\xleftarrow{\alpha_1, \alpha_2}$

$\mathcal{D}_{sk}(\alpha_1) = m_1 + \delta_1$

$\mathcal{D}_{sk}(\alpha_2) = m_2 + \delta_2$

$\beta = \mathcal{E}_{pk}((m_1 + \delta_1) \cdot (m_2 + \delta_2))$

Interactive Homomorphic Multiplication of Ciphertexts [Cramer et al.]

How to compute $\mathcal{E}_{pk}(m_1 \cdot m_2)$?

(sk, pk)

$c_1 = \mathcal{E}_{pk}(m_1)$ and $c_2 = \mathcal{E}_{pk}(m_2)$

Picks two randoms $\delta_1, \delta_2 \in \mathbb{Z}_n$

$\alpha_1 = c_1 \cdot \mathcal{E}_{pk}(\delta_1)$

$\alpha_2 = c_2 \cdot \mathcal{E}_{pk}(\delta_2)$

$\xleftarrow{\alpha_1, \alpha_2}$

$\mathcal{D}_{sk}(\alpha_1) = m_1 + \delta_1$

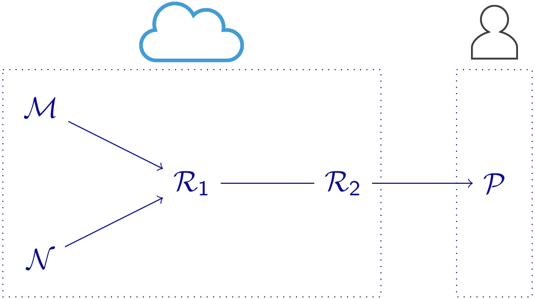
$\mathcal{D}_{sk}(\alpha_2) = m_2 + \delta_2$

$\beta = \mathcal{E}_{pk}((m_1 + \delta_1) \cdot (m_2 + \delta_2))$ $\xrightarrow{\beta}$ Computes $\frac{\beta}{\mathcal{E}_{pk}(\delta_1 \cdot \delta_2) \cdot c_1^{\delta_2} \cdot c_2^{\delta_1}}$

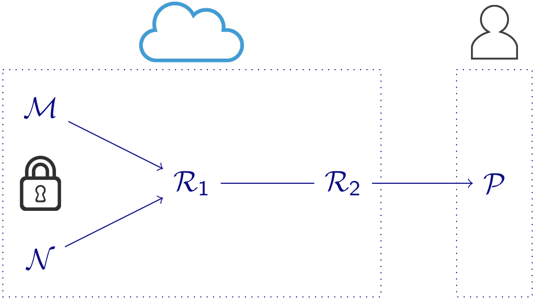
Verification of the Computation

$$\begin{aligned}
\frac{\beta}{\mathcal{E}_{pk}(\delta_1 \cdot \delta_2) \cdot c_1^{\delta_1} \cdot c_2^{\delta_2}} &= \frac{\mathcal{E}_{pk}((m_1 + \delta_1) \cdot (m_2 + \delta_2))}{\mathcal{E}_{pk}(\delta_1 \cdot \delta_2) \cdot \mathcal{E}_{pk}(m_1)^{\delta_2} \cdot \mathcal{E}_{pk}(m_2)^{\delta_1}} \\
&= \frac{\mathcal{E}_{pk}(m_1 \cdot m_2 + m_1 \cdot \delta_2 + m_2 \cdot \delta_1 + \delta_1 \cdot \delta_2)}{\mathcal{E}_{pk}(\delta_1 \cdot \delta_2) \cdot \mathcal{E}_{pk}(m_1)^{\delta_2} \cdot \mathcal{E}_{pk}(m_2)^{\delta_1}} \\
&= \frac{\mathcal{E}_{pk}(m_1 \cdot m_2) \cdot \mathcal{E}_{pk}(m_1 \cdot \delta_2) \cdot \mathcal{E}_{pk}(m_2 \cdot \delta_1) \cdot \mathcal{E}_{pk}(\delta_1 \cdot \delta_2)}{\mathcal{E}_{pk}(\delta_1 \cdot \delta_2) \cdot \mathcal{E}_{pk}(m_1 \cdot \delta_2) \cdot \mathcal{E}_{pk}(m_2 \cdot \delta_1)} \\
&= \mathcal{E}_{pk}(m_1 \cdot m_2)
\end{aligned}$$

Secure Matrix Multiplication with MapReduce

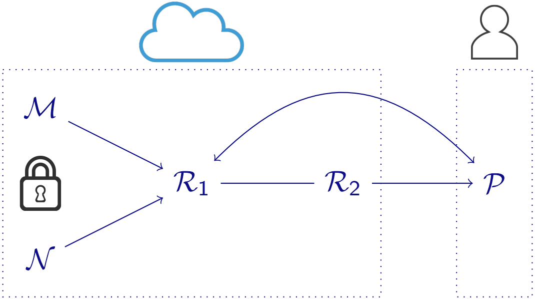


Secure Matrix Multiplication with MapReduce



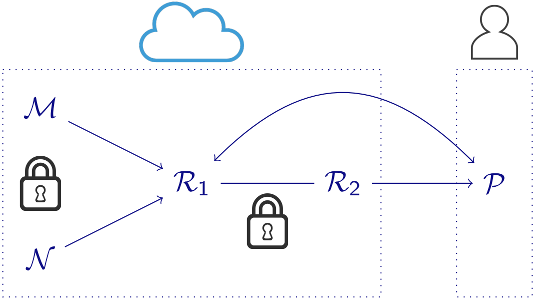
- $\mathcal{M}$ emits to $\mathcal{R}_1$: $(j,(M,i,\mathcal{E}_{pk}(m_{ij})))_{1\le i\le a, 1\le j\le b}$
- $\mathcal{N}$ emits to $\mathcal{R}_1$: $(j,(N,k,\mathcal{E}_{pk}(n_{jk})))_{1\le j\le b, 1\le k\le c}$

Secure Matrix Multiplication with MapReduce



- $\mathcal{M}$ emits to $\mathcal{R}_1$: $(j,(M,i,\mathcal{E}_{pk}(m_{ij})))_{1\leq i\leq a,1\leq j\leq b}$
- $\mathcal{N}$ emits to $\mathcal{R}_1$: $(j,(N,k,\mathcal{E}_{pk}(n_{jk})))_{1\leq j\leq b,1\leq k\leq c}$
- $\mathcal{R}_1$ emits to $\mathcal{R}_2$: $\mathcal{E}_{pk}(m_{ij}\cdot n_{jk})_{1\leq i\leq a,1\leq j\leq b,1\leq k\leq c}$ (computed via *Interactive Homomorphic Multiplication*)

Secure Matrix Multiplication with MapReduce



- $\mathcal{M}$ emits to $\mathcal{R}_1$: $(j,(M,i,\mathcal{E}_{pk}(m_{ij})))_{1\leq i\leq a, 1\leq j\leq b}$
- $\mathcal{N}$ emits to $\mathcal{R}_1$: $(j,(N,k,\mathcal{E}_{pk}(n_{jk})))_{1\leq j\leq b, 1\leq k\leq c}$
- $\mathcal{R}_1$ emits to $\mathcal{R}_2$: $\mathcal{E}_{pk}(m_{ij}\cdot n_{jk})_{1\leq i\leq a, 1\leq j\leq b, 1\leq k\leq c}$ (computed via *Interactive Homomorphic Multiplication*)
- $\mathcal{R}_2$ emits to $\mathcal{P}$: $\left\{\mathcal{E}_{pk}(p_{ik}) = \prod_{j=1}^{b} \mathcal{E}_{pk}(m_{ij}\cdot n_{jk})\right\}_{1\leq i\leq a, 1\leq k\leq c}$

Computation/Communication Costs and Privacy

Algorithm	*Computation cost (big-O)*	*Comm. cost (big-O)*
Standard	$2n^2 + (C_\times + C_+)n^3$	$n^3 + 3n^2$
SP	$(C_+ + 2C_\mathcal{E})n^2 + (2C_\times + 2C_{\text{exp}} + C_\mathcal{D})n^3$	$3n^3 + 4n^2$
CRSP	$2C_\mathcal{E} n^2 + (4C_\mathcal{E} + 7C_\times + 2C_\mathcal{D} + 2C_{\text{exp}})n^3$	$4n^3 + 3n^2$

- $n = \max(a, b, c)$
- $C_+, C_\times, C_\mathcal{E}, C_\mathcal{D}, C_{\text{exp}}$ are the costs of the associated operations

Conclusion

Two secure approaches for matrix multiplication with MapReduce:

1. Secure-Private (SP): Assumes that nodes $\mathcal{N}$ and $\mathcal{R}_2$ do not collude.
2. Collision-Resistant-Secure-Private (CRSP): Resists to collisions but uses interaction between node $\mathcal{R}_1$ and $\mathcal{R}_2$.

Outline

Different Adversaries

Intuition of Computational Security

Cloud Security

Partial and Full Homomorphic Encryption

Secure Matrix Multiplication

SSE

Privacy in DB

Conclusion

Symmetric Searchable Encryption

Store data externally

- encrypted
- want to search data easily
- avoid downloading everything then decrypt
- allow others to search data without having access to plaintext

Context

Symmetric Searchable Encryption (*SSE*)

- Outsource a set of *encrypted data*.
- Basic functionnality: *single keyword query*.

(Client) ⟶ (Server)

Symmetric Searchable Encryption

When searching, what must be protected?

- retrieved data
- search query
- search query outcome (was anything found?)

Scenario

- single query vs multiple queries
- non-adaptive: series of queries, each independent of the others
- adaptive: form next query based on previous results

Number of participants

- single user (owner of data) can query data
- multiple users can query the data, possibly with access rights defined by the owner

SSE by Song, Wagner, Perrig 2000

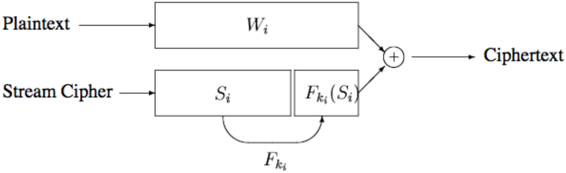


Figure 1. The Basic Scheme

Basic Scheme I

$$C_i = W_i \oplus < S_i, F_{k_i}(S_i) >$$

where S_i are randomly generated and $F_k(x)$ is a MAC with key k.

Basic Scheme

$$C_i = W_i \oplus < S_i, F_{k_i}(S_i) >$$

To search W :

- Alice reveals $\{k_i$, where W may occur$\}$
- Bob checks if $W \oplus C_i$ is of the form $< s, F_{k_i}(s) >$.

For unknown k_i, Bob knows nothing

Basic Scheme

$$C_i = W_i \oplus < S_i, F_{k_i}(S_i) >$$

To search W :

- Alice reveals $\{k_i$, where W may occur$\}$
- Bob checks if $W \oplus C_i$ is of the form $< s, F_{k_i}(s) >$.

For unknown k_i, Bob knows nothing

Problems for Alice !

- she reveals all k_i,
- or she has to know where W may occur !

Scheme II: Controlled Searching

Modifications

$$C_i = W_i \oplus < S_i, F_{k_i}(S_i) >$$

where S_i randoms, $F_k(x)$ is a MAC with key k; $k_i = f_{k'}(W_i)$

To search W :

- Alice only reveals $k = f_{k'}(W)$ and W.
- Bob checks if $W \oplus C_i$ is of the form $< s, F_k(s) >$

$+$ For unknown k_i, Bob knows nothing

$+$ Nothing is revealed about location of W.

Problem

- Still does not support hidden search (Alice reveals W)

Scheme III: Support for Hidden Searches

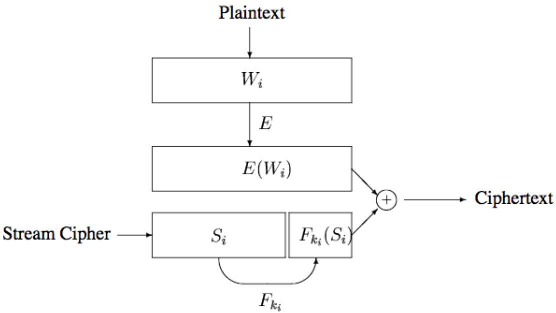


Figure 2. The Scheme for Hidden Search

Scheme III : Hidden Searches

$$C_i = E_{k''}(W_i) \oplus < S_i, F_{k_i}(S_i) >$$

S_i randoms and $F_k(x)$ is a MAC with k and $k_i = f_{k'}(E_{k''}(W_i))$

Scheme III: Support for Hidden Searches

$$C_i = E_{k''}(W_i) \oplus < S_i, F_{k_i}(S_i) >, where\ k_i = f_{k'}(E_{k''}(W_i))$$

To search W :

- Alice gives $X = E_{k''}(W)$ and $k = f_{k'}(X)$.
- Bob checks if $X \oplus C_i$ is of the form $< s, F_k(s) >$

Bob returns to Alice C_i

Scheme III: Support for Hidden Searches

$$C_i = E_{k''}(W_i) \oplus < S_i, F_{k_i}(S_i) >, where\ k_i = f_{k'}(E_{k''}(W_i))$$

To search W :

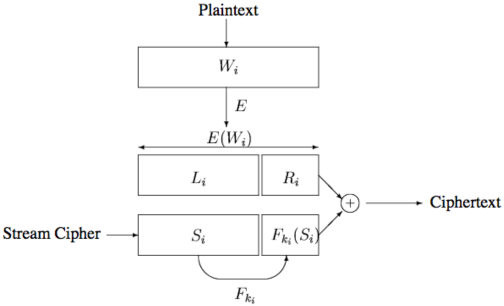
- Alice gives $X = E_{k''}(W)$ and $k = f_{k'}(X)$.
- Bob checks if $X \oplus C_i$ is of the form $< s, F_k(s) >$

Bob returns to Alice C_i

But Alice cannot recover the plaintext

She can recover S_i with X but not $F_{k_i}(S_i)$ because to compute $k_i = f_{k'}(E_{k''}(W_i))$ she needs to have $E_{k''}(Wi)$.
In this case, why do you need search ?

Final Scheme



Scheme IV : Final

$$C_i = X_i \oplus < S_i, F_{k_i}(S_i) >$$

where S_i randoms and $F_k(x)$ is a MAC with key k,
$X_i = E_{k''}(W_i) =< L_i, R_i >$ and $k_i = f_{k'}(L_i)$

Final Scheme (Ultimate TRICK !)

$$C_i = X_i \oplus < S_i, F_{k_i}(S_i) >$$

To search W :

- Alice gives $X = E_{k''}(W) =< L, R >$ and $k = f_{k'}(L)$
- Bob checks if $X \oplus C_i$ is of the form $< s, F_k(s) >$

Bob returns to Alice C_i

Alice recovers S_i and then $L_i = C_i \oplus S_i$. Then she computes $k_i = f_{k'}(L_i)$ and then $X = C_i \oplus < s, F_k(s) >$ and by decrypting with k'' to obtain W_i.

Alice only needs to remember k'' and k'.

Outline

Different Adversaries

Intuition of Computational Security

Cloud Security

Partial and Full Homomorphic Encryption

Secure Matrix Multiplication

SSE

Privacy in DB

Conclusion

Privacy vs. Confidentiality

Confidentiality

Prevent disclosure of information to unauthorized users

Privacy

- Prevent disclosure of personal information to unauthorized users
- Control of how personal information is collected and used

Data Privacy and Security Measures

Access control

Restrict access to the (subset or view of) data to authorized users

Inference control

Restrict inference from accessible data to additional data

Flow control

Prevent information flow from authorized use to unauthorized use

Encryption

Use cryptography to protect information from unauthorized disclosure while in transmit and in storage

2 kinds of data

- Personal data
- Anonymous data

CNIL:

"Dès lors qu'elles concernent des personnes physiques identifiées directement ou indirectement."

French Law:

"Pour déterminer si une personne est identifiable, il convient de considérer l'ensemble des moyens en vue de permettre son identification dont dispose ou auxquels peut avoir accès le responsable du traitement ou toute autre personne."

How to evaluate the security?

Three criteria of robustness:

- is it still possible to single out an individual ?
  Singling out (Individualisation): the possibility to isolate some or all records which identify an individual in the dataset
- is it still possible to link records relating to an individual ?
  Linkability (Correlation): ability to link, at least, two records concerning the same data subject or a group of data subjects.
- can information be inferred concerning an individual?
  Inference (Deduction): deduce, with significant probability, the value of an attribute from the values of a set of other attributes

Example

ID	Age	CP	Sex	Pathology
Paul Sésame	75	75000	F	Cancer
Pierre Richard	55	78000	F	Cancer
Henri Poincarré	40	71000	M	Influe

Randomization

Alter veracity of the DB to remove the link

- **Noise addition:** modifying attributes in the dataset such that they are less accurate whilst retaining the overall distribution
- **Permutation:** shuffling the values of attributes in a table so that some of them are artificially linked to different data subjects,
- **Differential Privacy:** requires the outcome to be formally indistinguishable when run with and without any particular record in the data set.

Example

Q = select count() where Age = [20,30] and Diagnosis=B
Answer to Q on D1 and D2 should be indistinguishable, if Bob in D1 or Bob out D2.

Differential Privacy

C. Dwork : "Differential Privacy", International Colloquium on Automata, Languages and Programming , 2006.

Definition

Let ϵ be a positive real number and $\mathcal{A}$ be a randomized algorithm that takes a dataset as input (representing the actions of the trusted party holding the data). The algorithm $\mathcal{A}$ is ϵ-differentially private if for all datasets D_1 and D_2 that differ on a single element (i.e., the data of one person), and all subsets S of $\mathrm{im}\mathcal{A}$,

$$\mathsf{Pr}[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \mathsf{Pr}[\mathcal{A}(D_2) \in S]$$

where the probability is taken over the randomness used by the algorithm.

Pseudonymisation

ID	Age	CP	Sex	Pathology
1	75	75000	F	Cancer
2	55	78000	F	Cancer
3	40	71000	M	Influe

Replace identifier field by a new one called pseudonym.
Using Hash function
It does not ensure anonymity. Using several fields you can recover name like it has benn done by Sweeney in 2001.

Example

Sex + birthday date + Zip code are unique for 80 % of USA citizens. (record linkage attack)

k-Anonymity

- Identify the possible fields that can be used to recover data (generalisation).
- Modify them in order to have at least k different lines having the same identifiers.

It reduce the probabolity to guess something to $1/k$
Advantage: Analysis of data still give the same information that the orginal data base.

Example: k-Anonymity

Activity	Age	Pathology
M2	[22,23]	Cancer
M2	[22,23]	Blind
M2	[22,23]	VIH
PhD	[24,27]	Cancer
PhD	[24,27]	Allergies
PhD	[24,27]	Allergies
L	[20,21]	Cancer
L	[20,21]	Cancer
L	[20,21]	Cancer

3-Anonymity
Activity for student can be Master licence or PhD instead of name and activty, age can be ranged.

Disadvantages: k-Anonymity

- It leaks negative information. For instance you are not in all the other catergories.
- If all personn have the same value then the value is leaked.
- Main problem is to determine the right generalisation (it is difficult and expensive).

Minimum Cost 3-Anonymity is NP-Hard for $|\Sigma| = 2$ (Dondi et al. 2007)

l-diversity

Aims at avoiding that all person have the same values once they have been generalized.
l values souhld be inside each field after generalisation. It allows to recover information by mixing information with some probability

Activity	Age	Pathology
M2	[22,23]	Cancer
M2	[22,23]	Allergies
M2	[22,23]	VIH
PhD	[24,27]	Cancer
PhD	[24,27]	VIH
PhD	[24,27]	Allergies
L	[20,21]	VIH
L	[20,21]	Allergies
L	[20,21]	Cancer

3-diversity, each category has 3 different values

t-closeness

Knowledge of global distribution of sensitive data of a class of equivalence.
It tries to reduce the weaknesses introduced by the l-diversity.
t is the factor that says how we are far from a global distribution.

- How to split data into partion to obtain all the same distribution.
- If all class of equivalence have the same number of data, what is the utility of any analysis of the data basis ?

Summary

Is Risky	Singling out	Linkability	Inference
Pseudonymisation	Yes	Yes	Yes
Noise addition	Yes	May not	May not
Substitution	Yes	Yes	May not
Aggregation or K-anonymity	No	Yes	Yes
L-diversity	No	Yes	May not
Differential privacy	May not	May not	May not

Outline

Different Adversaries

Intuition of Computational Security

Cloud Security

Partial and Full Homomorphic Encryption

Secure Matrix Multiplication

SSE

Privacy in DB

Conclusion

Things to bring home

- Date Security is cruciual
- Security should be done by experts!
- Security should be taken from the design and not after!

Protocol + Properties + Intruder = Security

Thank you for your attention.

Questions ?