Homomorphic encryption

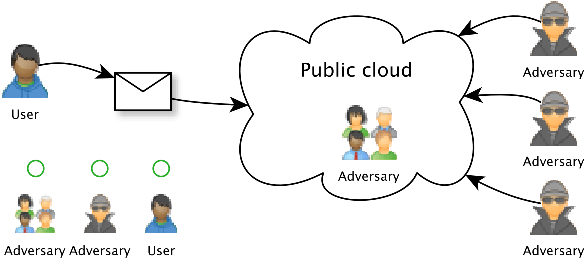
Oleksandr Kazymyrov

University of Bergen
Norway

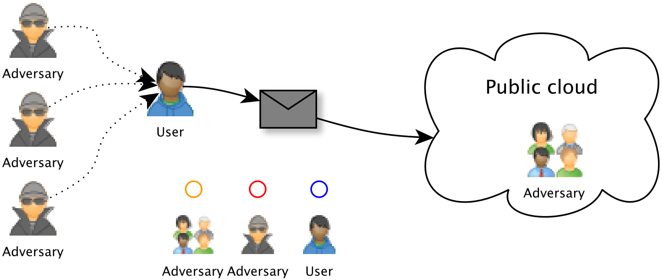
20th of October, 2014

Outline

Cloud computations



Encrypted cloud computations



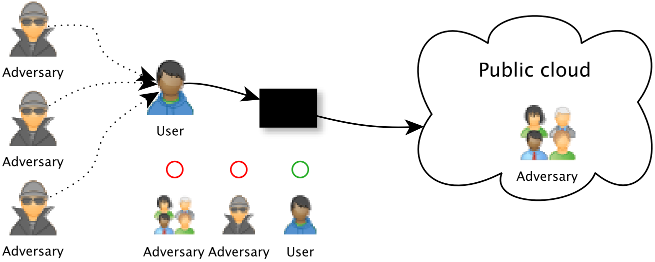
Computing on encrypted data

It would be nice to be able to ...

- encrypt data in the cloud
  - $\hookrightarrow$ while still allowing the cloud to search, sort, edit ...
- keep the data in the cloud in encrypted form
  - $\hookrightarrow$ without need to encrypt/decrypt every time
- encrypt queries to the cloud
  - $\hookrightarrow$ while still allowing the cloud to process them
  - $\hookrightarrow$ the cloud returns encrypted answers

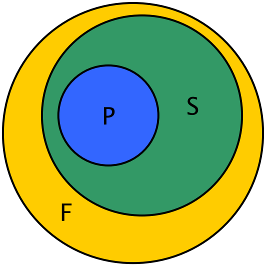
Homomorphic encryption

Cloud computations with homomorphic encryption



What is a homomorphic encryption (HE)?

- An encryption scheme: $(KeyGen, Enc, Dec)$
  - $\star$ $(pk, sk) = KeyGen(rnd),\ c_i = Enc_{pk}(m_i),\ m_i = Dec_{sk}(c_i)$
- A homomorphic encryption scheme: $(KeyGen, Enc, Dec,$ $Eval Eval)$
  - $\star$ $\{c_i'\} = Eval_{pk}(f, \{c_i\})$
  - $\star$ $\{c_i'\} = Eval_{pk}(f, \{c_i\})$, $Dec_{sk}(Eval_{pk}(f, \{Enc_{pk}(x_i)\})) = f(\{x_i\})$



P - partially

S - “somewhat”

F - full

Outline

Partially homomorphic encryption schemes

RSA

KeyGen: $N = p \cdot q$, where p and q large prime numbers

$$gcd(e, \phi(N)) = 1 \Rightarrow d \cdot e \equiv 1 \pmod{\phi(N)}$$

Enc: $c \equiv m^e \pmod{N}$

Dec: $m \equiv c^d \pmod{N}$

$Enc(m_1) \cdot Enc(m_2) = Enc(m_1 \cdot m_2)$:

$$c_1 \equiv m_1^e \pmod{N} \qquad c_2 \equiv m_2^e \pmod{N}$$

$$c_1 \cdot c_2 \equiv m_1^e \cdot m_2^e \equiv (m_1 \cdot m_2)^e \pmod{N}$$

Partially homomorphic encryption schemes

- RSA, ElGamal work for multiplication
- Paillier, Benaloh work for addition
- Goldwasser-Micali works for XOR
- MGH'08 works for degree d polynomials
- ...

$(+, \cdot)$-Homomorphic encryption

It would be really nice to have ...

- Plaintext space $\mathbb{Z}_N$
- Ciphertext space $\mathbb{Z}_N$
- Homomorphic $Enc(x)/Eval(x)$ for both "$+$" and "$\cdot$"

$$Enc(m_1) + Enc(m_2) \equiv Enc(m_1 + m_2 \pmod{N})$$
$$Enc(m_1) \cdot Enc(m_2) \equiv Enc(m_1 \cdot m_2 \pmod{N})$$

- Then we can compute many useful functions on ciphertexts

Outline

Breakthrough

- Genrty'09: a bootstrapping technique
  - ⋆ "Somewhat" homomorphic → Fully homomorphic
- Gentry also described a candidate "bootstrappable" scheme
  - ⋆ Based on ideal lattices
- Gentry's scheme was complex
  - ⋆ it used advanced algebraic number theory
- Can it be simpler?
  - ⋆ polynomials, matrices ... integers

(XOR, AND)-Homomorphic encryption

Why (XOR, AND)?

- because XOR and AND gives a Turing-complete system
  - $\hookrightarrow$ if we can compute XOR and AND on encrypted bits
    - $\hookrightarrow$ we can compute ANY function on encrypted inputs

A secret-key homomorphic encryption

Dijk, Gentry, Halevi and Vaikuntanathan [DGHV'10]

Secret key: large odd number p

Encryption steps of a bit m:

Choose at random large q and small r

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

$\hookrightarrow$ $2 \cdot r + m$ much smaller than p

$\hookrightarrow$ ciphertext is close to a multiple of p

$Dec : m \equiv (c \bmod p) \bmod 2$

Parameters: $|r| = n$, $|p| = n^2$ and $|q| = n^5$

Why is this homomorphic?

$$c_1 = p \cdot q_1 + 2 \cdot r_1 + m_1 \qquad c_2 = p \cdot q_2 + 2 \cdot r_2 + m_2$$

Adding (XORing) two encrypted bits

$$c_1 + c_2 = (q_1 + q_2) \cdot p + 2 \cdot (r_1 + r_2) + (m_1 + m_2)$$

$\hookrightarrow$ if $2 \cdot (r_1 + r_2) + (m_1 + m_2)$ much smaller than p

$\hookrightarrow$ $(c_1 + c_2 \mod p) \mod 2 \equiv m_1 + m_2 \pmod 2$

Multiplying (ANDing) two encrypted bits

$$\begin{aligned} c_1 \cdot c_2 = & q_1 q_2 p^2 + 2 q_1 p r_2 + q_1 m_2 p + 2 q_2 p r_1 + 4 r_1 r_2 + 2_r 1 m_2 + \\ & q_2 m_1 p + 2 m_1 r_2 + m_1 m_2 = \\ & (q_1 q_2 p + 2 q_1 r_2 + q_1 m_2 + 2 q_2 r_1 + q_2 m_1) p + \\ & 2(2 r_1 r_2 + r_1 m_2 + m_1 r_2) + m_1 m_2 \end{aligned}$$

Why is this homomorphic?

$$c_1 = p \cdot q_1 + 2 \cdot r_1 + m_1 \qquad c_2 = p \cdot q_2 + 2 \cdot r_2 + m_2$$

Multiplying (ANDing) two encrypted bits (continue)

$$\begin{aligned}
c_1 \cdot c_2 = \ldots = (q_1 q_2 p + 2q_1 r_2 + q_1 m_2 + 2q_2 r_1 + q_2 m_1)p + \\
2(2r_1 r_2 + r_1 m_2 + m_1 r_2) + m_1 m_2 \\
c_1 q_2 =& q_1 q_2 p + 2q_2 r_1 + q_2 m_1 \\
c_2 q_1 =& q_1 q_2 p + 2q_1 r_2 + q_1 m_2 \\
c_1 \cdot c_2 =& (c_1 q_2 + c_2 q_1 - q_1 q_2 p)p + 2(2r_1 r_2 + r_1 m_2 + m_1 r_2) + m_1 m_2
\end{aligned}$$

$\hookrightarrow$ if $2(2r_1 r_2 + \ldots) + m_1 m_2$ much smaller than p

$\hookrightarrow$ $(c_1 \cdot c_2 \bmod p) \bmod 2 \equiv m_1 \cdot m_2 \pmod 2$

Dijk, Gentry, Halevi and Vaikuntanathan [DGHV'10]

Example

Secret key: choose $p = 99$

Encryption of $m_1 = 1$ and $m_2 = 1$:

Choose $q_1 = 37, r_1 = 3$ and $q_2 = 82, r_2 = 2$

$$c_1 = 99 \cdot 37 + 2 \cdot 3 + 1 = 3670 \quad c_2 = 99 \cdot 82 + 2 \cdot 2 + 1 = 8123$$

Evaluate $m_1 \oplus m_2$ and $m_1 \cdot m_2$:

$$c_1 + c_2 = 11793 \qquad c_1 \cdot c_2 = 29811410$$

Decryption results in:

$$m_1 + m_2 \equiv (11793 \bmod 99) \bmod 2 \equiv 12 \bmod 2 \equiv 0 \pmod 2$$
$$m_1 \cdot m_2 \equiv (29811410 \bmod 99) \bmod 2 \equiv 35 \bmod 2 \equiv 1 \pmod 2$$

Two issues

- Ciphertext grows with each operation

  $\hookrightarrow$ if $|c_1| = |c_2| = n$ then $|c_1 + c_2| = n + 1$ and $|c_1 \cdot c_2| = 2 \cdot n$

  $\hookrightarrow$ we can do lots of additions and some multiplications

  ("somewhat" homomorphic encryption)

- Noise grows with each operation

  $\hookrightarrow$ it doubles on addition and squares on multiplication

  $\hookrightarrow$ after some operations ($|\text{noise}| > \frac{p}{2}$) the ciphertext becomes "unrecoverable"

How secure is this?

Trivial attack

- if there was no noise ($r_1 = 0$ and $r_2 = 0$)
  - $\hookrightarrow$ and encrypted two 0 bits, that is $c_1 = q_1 p$, $c_2 = q_2 p$
    - $\hookrightarrow$ then the secret key $p = GCD(q_1 p, q_2 p)$

Other cases

- there is noise
  - $\hookrightarrow$ the GCD attack doesn't work
    - $\hookrightarrow$ this is called the approximate GCD assumption

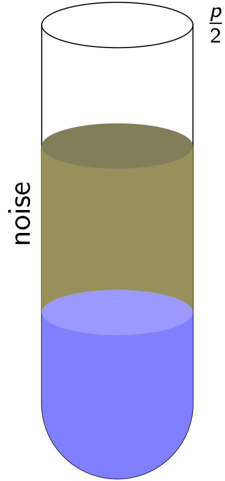
Outline

Gentry's "bootstrapping" theorem

Theorem [Gentry'09]

If an encryption scheme can evaluate its own decryption circuit, then it can evaluate everything.

SHE $\rightarrow$ FHE



- Problem: addition and multiplication increase noise
  - $\hookrightarrow$ addition
  - $\hookrightarrow$ multiplication
- Goal: somehow reduce the noise
- What is the best noise-reduction procedure?
  - $\hookrightarrow$ decryption
    - $\hookrightarrow$ Problem: key is secret
    - $\hookrightarrow$ Goal: reduce noise without publishing secret key

Reduce noise "somewhat" secure

$$Dec_{sk_1}(Eval_{pk_1}(f, \{Enc_{pk_1}(x_i)\})) = f(\{x_i\})$$

$$\begin{aligned}
Dec_{sk_1}(Eval_{pk_1}(Enc_{pk_1}, \{Enc_{pk_1}(x_i)\})) &= Enc_{pk_1}(\{x_i\}) \\
Dec_{sk_1}(Eval_{pk_1}(Dec_{sk_1} \circ Enc_{pk_1}, \{Enc_{pk_1}(x_i)\})) &= \{x_i\} \\
Dec_{sk_1}(Eval_{pk_1}(Enc_{pk_2} \circ Dec_{sk_1} \circ Enc_{pk_1}, \{Enc_{pk_1}(x_i)\})) &= Enc_{pk_2}(\{x_i\})
\end{aligned}$$

$$\begin{aligned}
Enc_{pk_2} \circ Dec_{sk_1} \circ Enc_{pk_1} &= h \\
Dec_{sk_1}(Eval_{pk_1}(h, \{Enc_{pk_1}(x_i)\})) &= Enc_{pk_2}(\{x_i\})
\end{aligned}$$

$$Dec_{sk_2} \circ Enc_{pk_2}(\{x_i\}) = \{x_i\}$$

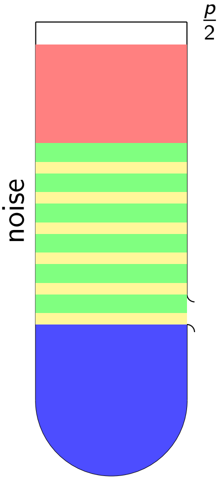
Reduce noise

$$Dec_{sk_1}\left(Eval_{pk_1}\left(f, \{Enc_{pk_1}(x_i)\}\right)\right) = f\left(\{x_i\}\right)$$
$$c_1 = Enc_{pk_1}(m) \quad c_2 = Enc_{pk_1}(pk_2) \quad f = Enc$$

$$c_3 = Eval_{pk_1}\left(Enc, \{c_1, c_2\}\right)$$
$$Dec_{sk_1}(c_3) = Enc_{pk_2}(m)$$

$$c_4 = Enc_{pk_1}(sk_2) \quad f = Dec$$
$$c_5 = Eval_{pk_1}\left(Dec, \{c_3, c_4\}\right)$$
$$Dec_{sk_1}(c_5) = m$$

SHE $\to$ FHE



- Enc_{pk_u} and Dec_{sk_u}
- $(XOR, AND)^t$
- Enc_{pk_i}
- Dec_{sk_i}
- Repeat

Outline

A public-key homomorphic encryption

Dijk, Gentry, Halevi and Vaikuntanathan [DGHV'10]

Secret key: large odd number p

Public key: a set X of large numbers $\{x_1, x_2, .., x_i\}$ such that $x_j \equiv 2 \cdot r_j \pmod{p}$ $x_j \equiv 2 \cdot r_j \pmod{p} \Rightarrow x_j = p \cdot q_j + 2 \cdot r_j$

Encryption steps of a bit m:

Choose at random r and for $Y \subset X$ calculate $b = \sum Y$

$Enc_{pk}(m) : c = b + 2 \cdot r + m$

$Dec_{sk}(c) : m \equiv (c \bmod p) \bmod 2$

$Eval_{pk}(f, \{c_i\})$: as before

Homomorphic properties

$$c_1 = p \cdot \sum_k q_k + 2 \cdot \sum_k r_k + m_1 \qquad c_2 = p \cdot \sum_h q_h + 2 \cdot \sum_h r_h + m_2$$

Addition of two encrypted bits

$$c_1 + c_2 = (\sum_k q_k + \sum_h q_h) \cdot p + 2 \cdot (\sum_k r_k + \sum_h r_h) + (m_1 + m_2)$$

$$m_1 + m_2 \equiv (c_1 + c_2 \mod p) \mod 2$$

Multiplication of two encrypted bits

Assume $Q_1 = \sum_k q_k$, $R_1 = \sum_k r_k$, $Q_2 = \sum_h q_h$, $R_2 = \sum_h r_h$

$$c_1 \cdot c_2 = (c_1 Q_2 + c_2 Q_1 - Q_1 Q_2 p)p + 2(2R_1 R_2 + R_1 m_2 + m_1 R_2) + m_1 m_2$$

$$m_1 \cdot m_2 \equiv (c_1 \cdot c_2 \mod p) \mod 2$$

Example

Secret key: $p = 233$

Public key: $X = \{31955, 36362, 36627, 40098, 45718\}$

Encryption of $m_1 = 0$ and $m_2 = 1$:

Choose $r_1 = 18$, $r_2 = 5$ and $Y = \{31955, 40098\}$

$c_1 = 72053{+}2{\cdot}18{+}0 = 72089$ $c_2 = 72053{+}2{\cdot}5{+}1 = 72064$

Evaluate $m_1 \oplus m_2$ and $m_1 \cdot m_2$:

$$c_1 + c_2 = 144153 \qquad c_1 \cdot c_2 = 5195021696$$

Decryption results in:

$$\begin{aligned} m_1 &\equiv (72089 \bmod 233) \bmod 2 \equiv 92 \bmod 2 \equiv 0 \pmod 2 \\ m_2 &\equiv (72064 \bmod 233) \bmod 2 \equiv 67 \bmod 2 \equiv 1 \pmod 2 \\ m_1 + m_2 &\equiv (144153 \bmod 233) \bmod 2 \equiv 159 \bmod 2 \equiv 1 \pmod 2 \\ m_1 \cdot m_2 &\equiv (5195021696 \bmod 233) \bmod 2 \equiv 106 \bmod 2 \equiv 0 \pmod 2 \end{aligned}$$

Conclusions

- Gentry gave the first feasible result

  $\hookrightarrow$ showing that it can be done "in principle"
- Bootstrapping technique allows transform SHE to FHE

  $\hookrightarrow$ reduce performance dramatically
- New FHEs without bootstrapping

  $\hookrightarrow$ potentially leads to practical implementations
- Lack of independent security checks

  $\hookrightarrow$ public key encryption with homomorphic properties