Privacy Homomorphism and Applications through Symmetric Key Encryption Algorithms

Devesh C Jinwala,
Professor, SVNIT, Surat and Adjunct Professor, IIT Jammu

Sardar Vallabhbhai National Institute of Technology Surat

1

Asymmetric Key Homomorphic Algorithms

- Deterministic Algorithms
  - RSA Algorithm
- Probabilistic Algorithms
  - The Goldwasser-Micali Algorithm
  - The Paillier Encryption Algorithm
  - The ElGamal Cryptosystem
  - The Okamoto-Uchiyama Cryptosystem

Asymmetric Key Homomorphic Algorithms

- Deterministic Algorithms
  - RSA Algorithm
- Probabilistic Algorithms
  - The Goldwasser-Micali Algorithm
  - The Paillier Encryption Algorithm
  - The ElGamal Cryptosystem
  - The Okamoto-Uchiyama Cryptosystem

RSA – Key Generation

- Select primes: p=17 & q=11
- Compute $n = pq$ =17×11=187
- Compute ø(n)=(p–1)(q-1)=16×10=160
- Select e : gcd(e,160)=1; choose e=7
- Determine d: $d * e = 1 \bmod 160$ and $d < 160$ Value is d=23 since 23×7=161= 10×160+1
- Publish public key P_k={7,187}
- Keep secret private key S_k={23,17,11}

RSA Algorithm

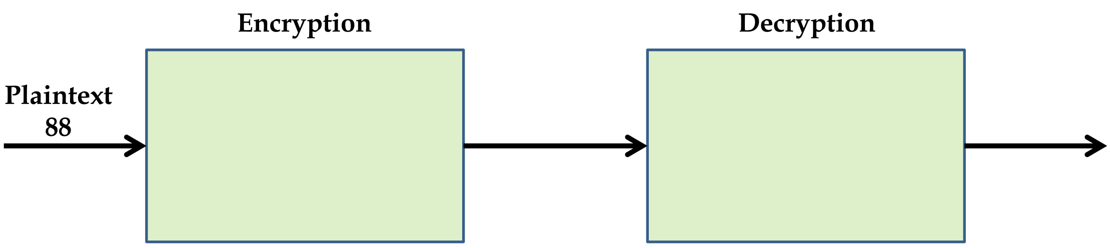
Algorithm RSA ()

Key Generation: Choose two distinct prime numbers p and q.
Compute n=pq.
Compute $\Phi(n) = (p-1)(q-1)$, where Φ is Eulers totient function.
Choose an integer e such that $1 < e < \Phi(n)$ and $gcd(e, \Phi(n)) = 1$,
i.e. e and $\Phi(n)$ are co primes.
Determine $d = e^{-1} mod\, \Phi(n)$;
i.e. d is the multiplicative inverse of $e\ mod\ \Phi(n)$.

Message Encryption: $c = m^e (mod\, n)$

Decryption: $m = c^d (mod\, n)$

RSA – Algorithm



Plaintext m $\in \mathbb{Z}_n$
Here, $m = 88 \in \mathbb{Z}_{187}$

RSA – Algorithm

Encryption

Decryption

Plaintext
88

88^7 *mod* 187 = 11

Ciphertext
11

Public Key $P_k = (\mathrm{e}, \mathrm{n}) = (7, 187)$
Ciphertext $\mathrm{C} = \mathrm{m^e} \bmod \mathrm{n}$

7

RSA – Algorithm

Encryption

Decryption

Plaintext
88

88^7 *mod* 187 = 11

Ciphertext
11

11^{23} *mod* 187 = 88

Plaintext
88

Secret Key $S_k = \ d = 23$
Plaintext $\mathrm{m} = c^{\mathrm{d}} \bmod \mathrm{n}$

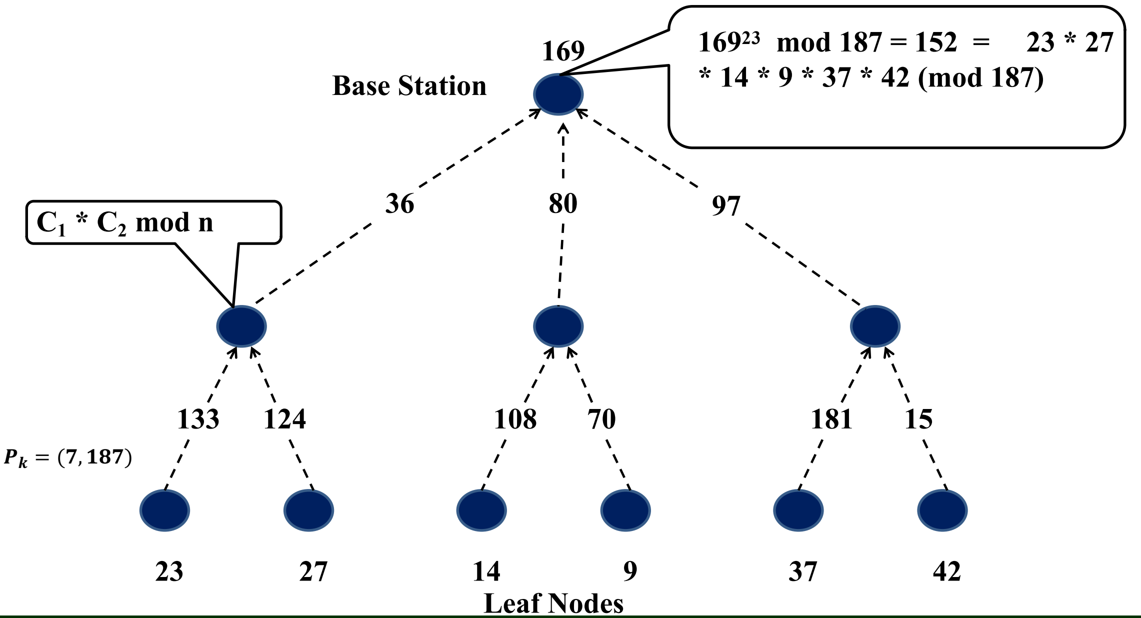
8

RSA – Algorithm – Homomorphic Property

$$C_1 * C_2 \bmod n = E(m_1 * m_2) \bmod n$$

9

RSA – Algorithm – Example



10

Asymmetric Key Homomorphic Algorithms

- Deterministic Algorithms
  - RSA Algorithm
- Probabilistic Algorithms
  - The Goldwasser-Micali Algorithm
  - The Paillier Encryption Algorithm
  - The ElGamal Cryptosystem
  - The Okamoto-Uchiyama Cryptosystem

Goldwasser-Micali – Key Generation

- Select primes: p=23 & q=37, where p ≠ q
- Select some Quadratic non-residue a = 80 ∋ $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$
- …
-
- $\left(\frac{a}{p}\right) = \begin{cases} 1 \text{ if } a \text{ is a quadratic residue modulo } p \text{ and } a \not\equiv 0 \pmod{p} \\ -1 \text{ if } a \text{ is a quadratic non-residue modulo } p \\ 0 \text{ if } a \equiv 0 \pmod{p}. \end{cases}$

security of the scheme is based on the hardness of determining whether a number x is a QR modulo n, when the factoring of n is unknown and the Jacobi symbol $\left(\frac{x}{n}\right)$ is 1

If p is an odd prime and if α is a generator of Z^*_p. Then, a∈ Z^*_p is a QR modulo p iff a = α^i mod p, where i is an even integer.

Multiplicative Group

- A multiplicative group Z^*_n
  - A group whose group operation is identified with multiplication.
  - The multiplication operation on group elements is denoted by a raised dot $\cdot$ i.e. $g \cdot h$.
  - In a multiplicative group, the identity element is denoted 1, and the inverse of the element g is written as g^{-1}, voiced "g inverse."
  - If n is prime, then $Z^*_n = \{a \mid 1 \le a \le n-1\}$

Multiplicative Group

- A multiplicative group Z^*_n & Euler's Totient function
- The order of a multiplicative group Z^*_n - denoted $|Z^*_n|$ is defined as
  - $|Z^*_n|$ i.e. the number of elements in Z^*_n.
- Illustration:
  - Let n = 21. Then, $Z_{21}^* = \{1,2,4,5,8,10,11,13,16,17,19,20\}$
  - Now, $\varnothing(21)$ =
    - $\varnothing(7).\varnothing(3)=6.2=12=|Z^*_{21}|$

Euler's theorem

- Let $n \geq 2$ be an integer. Then if $a \in Z^*_n$, $a^{\varnothing(n)} \equiv 1 \pmod n$
- e.g.
  - a=3;n=10; ø(10)=4; hence 3^4 = 81 ≡ 1 mod 10
  - a=2;n=11; ø(11)=10; hence 2^{10} = 1024 = 1 mod 11

What about a=7
i.e. 7^4 mod 10 ?
And a=5 ?

- If n is a product of distinct primes,
  - and if $r \equiv s \pmod{\varnothing(n)}$, then $a^r \equiv a^s \pmod n$
  - i.e. when working with modulo such as n, exponents can be reduced modulo ø(n)

Order of elements of an MG

- Let $a \in Z^*_n$. Then, the **order of a**, denoted by **ord(a)**,
  - is the **least** positive integer t such that $a^t \equiv 1 \pmod n$
  - e.g. consider again Z_{21}^*= {1,2,4,5,8,10,11,13,16,17,19,20}
  - ø(21)=12=$|Z^*_{21}|$.
  - Now the orders of various elements in Z_{21}^* are:

a	1	2	4	5	8	10	11	13	16	17	19	20
Ord(a)	1	6	3	6	2	6	6	2	3	6	6	2

  - `Ord(a) = mod(power(a,Ai),21)` in Excel sheet

Generator, Cyclic group

- Let $\alpha \in Z^*_n$.
  - if the order of α is $\varnothing(n)$, then α is said to be a generator or a primitive element of Z^*_n.
  - Are there any generators in the group Z^*_{21} ?

a	1	2	3	4	5	6	7	8	9	10
Ord(a)	1	6	-	3	6	-	-	2	-	6
a	11	12	13	14	15	16	17	18	19	20
Ord(a)	6	-	2	-	-	3	6	-	6	2

Generator, Cyclic group

- IF Z^*_n has a generator, then Z^*_n is said to be a cyclic group.
  - In the above example, Z^*_{21} is not a cyclic group, since no generator is equal to $\varnothing(n)$ i.e. 12.

a	1	2	4	5	8	10	11	13	16	17	19	20
Ord(a)	1	6	3	6	2	6	6	2	3	6	6	2

Generator, Cyclic group (contd)

- Consider now a group Z_{25}^*
  - $Z_{25}^* = \{1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19, 21, 22, 23, 24\}$
  - i.e. $\Phi(25) = |Z_{25}^*| = 20$
  - Now the orders of various elements in Z_{25}^* are:

Use the formula Ord(a) = mod(power(a,Ai),25) in Excelsheet												
a	1	2	3	4	6	7	8	9	10	11	12	13
Ord(a)	1	20	20	10	5	5	20	10	–	5	?	?
a	14	15	16	17	18	19	21	23	24			
Ord(a)	?	?	?	?	?	?						

- Thus, Z_{25}^* is indeed a cyclic group because 2,3,8,… are the generators of the group.

Generator, Cyclic group (contd)..

Members of Z*25 are {1,2,3,4,6,7,8,9,11,12,13,14,16,17,18,19,21,22,23,24} and Phi(25)=20

	Z^*_{25} with α=2	Z^*_{25} with α=3	Z^*_{25} with α=4	Z^*_{25} with α=6	Z^*_{25} with α=7	Z^*_{25} with α=8	Z^*_{25} with α=9	Z^*_{25} with α=11	Z^*_{25} with α=12	Z^*_{25} with α=13	Z^*_{25} with α=14	Z^*_{25} with α=16	Z^*_{25} with α=17	Z^*_{25} with α=18	Z^*_{25} with α=19	Z^*_{25} with α=21	Z^*_{25} with α=22	Z^*_{25} with α=23	Z^*_{25} with α=2
1	2	3	4	6	7	8	9	11	12	13	14	16	17	18	19	21	22	23	24
2	4	9	16	11	24	14	6	21	19	19	21	6	14	24	11	16	9	4	1
3	8	2	14	16	18	12	4	6	3	22	19	21	13	7	9	11	23	17	24
4	16	6	6	21	1	21	11	16	11	11	16	11	21	1	21	6	6	16	1
5	7	18	24	1	7	18	24	1	7	18	24	1	7	18	24	1	7	18	24
6	14	4	21	6	24	19	16	11	9	9	11	16	19	24	6	21	4	14	1
7	3	12	9	11	18	2	19	21	8	17	4	6	23	7	14	16	13	22	24
8	6	11	11	16	1	16	21	6	21	21	6	21	16	1	16	11	11	6	1
9	12	8	19	21	7	3	14	16	2	23	9	11	22	18	4	6	17	13	24
10	24	24	1	1	24	24	1	1	24	24	1	1	24	24	1	1	24	#NUM!	#NUM!
11	23	22	4	6	18	17	9	11	13	12	14	16	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!
12	21	16	16	11	1	11	6	21	6	6	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	
13	17	23	14	16	7	13	4	#NUM!	#NUM!	#NUM!	#NUM!								
14	9	19	6	21	24	4	11	#NUM!	#NUM!	#NUM!	#NUM!								
15	18	7	24	1	18	7	#NUM!	#NUM!	#NUM!	#NUM!									
16	11	21	21	6	#NUM!	8		#NUM!											
17	22	13	9	11	#NUM!	23													
18	19	14	11	#NUM!	#NUM!	9													
19	13	17	19	#NUM!	#NUM!	22													
20	1	1	1	#NUM!	#NUM!	1													
21	2	3	4			8													
22	4	9	16																
23	8	2	#NUM!																

Snapshot of Z^*_{25} computation from the Excel sheet

Generator, Cyclic group (contd)

- Consider now a multiplicative group Z_{13}^*
  - $Z_{13}^* = \{1, 2, 3, 4, 6, 7, 8, 9, 11, 12\}$
  - i.e. $\Phi(13) = |Z_{13}^*| = 12$
  - Compute the orders of various elements in Z_{13}^*:

α	0	1	2	3	4	5	6	7	8	9	10	11
α^i mod 13	1	6	12	3	7	4	12	12	4	3	6	12

- Thus,
  - α = 2, 6, 7, 11 are the generators of the group.
  - Note the case of 5^t mod 13 with t=4,12.

Generators.....

- How many Generators can be there of a group if Z^*_n is a cyclic group ?
  - if Z_n^* is cyclic, then the number of generators is $\Phi(\Phi(n))$.
    - e.g. Z_{21}^* is not cyclic – doesn't have a generator because n does not satisfy any of the conditions above in first

- Are $Z^*_{11}, Z^*_7, Z^*_{13}, Z^*_{17}, Z^*_{19}$ cyclic ?
- Is Z_{30}^* cyclic ? $\Phi(30)$ is $\Phi(6)^* \Phi(5)$ = 2*4=8.

How to test for a given number to be a Generator ?

- Consider a MG $Z^*_{p,}$ where p is a prime.
- Then, it is easy to test whether a given element is its generator or not. How ?
  - As p is a prime, $\Phi(p) = p-1$, and
  - the number of generators in it is $\Phi(p-1)$,
  - now, if $p_1, p_2, p_3\ldots.p_k$ are the distinct prime factors of p-1, then,
    - g is a generator of Z^*_p if and only if

$$g^{(p-1)/pi} \neq 1 \bmod p \text{ for all } p_i \; 1 \leq i \leq k$$

How to test for a given number to be a Generator ?

- e.g. consider Z^*_{13}. Check whether 7 is a generator or not.
- Now,
  - $\Phi(13) = p-1 = 12$, and
  - the number of generators in it is $\Phi(p-1) = \Phi((12) = 4$.
  - Also, the distinct prime factors of p-1 i.e. 12 are 2,3. Hence, $p_1=2$, $p_2=3$.
  - Then,
    - $g^{(p-1)/p_1} = 7^{12/2} = 7^6 \bmod 13 = 12 \bmod 13 \neq 1 \; mod\; 13, and$
    - $g^{(p-1)/p_2} = 7^{12/3} = 7^4 \bmod 13 = 9 \bmod 13 \neq 1 \; mod \; 13$
- Hence, 7 is indeed a generator of Z^*_{13}

$$g^{(p-1)/pi} \neq 1 \bmod p \text{ for all } p_i \; 1 \leq i \leq k$$

How to test for a given number to be a Generator ?

- e.g. consider Z^*_{13}. Now, check whether 8 is a generator or not.
- Now,
  - $\Phi(13) = p-1 = 12$, and
  - the number of generators in it is $\Phi(p-1) = \Phi((12) = 4$.
  - Also, the distinct prime factors of p-1 i.e. 12 are 2, 3. Hence, $p_1=2$, $p_2=3$.
  - Then,
    - $g^{(p-1)/p_1} = 8^{12/2} = 8^6 \bmod 13 = 12 \bmod 13 \neq 1 \; mod\; 13, and$
    - $g^{(p-1)/p_2} = 8^{12/3} = 8^4 \bmod 13 = 1 \bmod 13$
- Hence, 8 is NOT a generator of Z^*_{13}

$g^{(p-1)/pi} \neq 1 \; mod\; p$ for all p_i $1 \leq i \leq k$

Quadratic Residues – an illustration

- e.g. for Z^*_{13}, one of its generator is 6 (since $6^{\Phi(13)} \bmod 13 = 1 \bmod 13$)…
- Hence,

$6^2 \bmod 13 = 10$	$6^4 \bmod 13 = 9$	$6^6 \bmod 13 = 12$
$6^8 \bmod 13 = 3$	$6^{10} \bmod 13 = 4$	$6^{12} \bmod 13 = 1$
$6^{14} \bmod 13 = 10$	$6^{16} \bmod 13 = 9$	$6^{18} \bmod 13 = \ldots$

- Therefore,
  - the Quadratic Residues set is $Q_{13} = \{1,3,4,9,10,12\}$ and
  - the Quadratic non-Residues set $\overline{Q_{13}}$ is $= \{2,5,6,7,8,11\}$

Goldwasser-Micali – Key Generation

- Select primes: p=23 & q=37, where p $\neq$ q
- Select some a such that (i.e. $\ni$) $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$. i.e. a is quadratic non-residue modulo p and is quadratic non-residue modulo q
- Choose a=80.
- Compute N = p * q = 851
- Public Key P_k = (a, N) = (80, 851), Secret Key S_k= (p, q) = (23, 37)

How is 80 a Q non-residue ?

$$\left(\frac{a}{p}\right) = \begin{cases} 1 \text{ if } a \text{ is a quadratic residue modulo } p \text{ and } a \not\equiv 0 \pmod{p} \\ -1 \text{ if } a \text{ is a quadratic non-residue modulo } p \\ 0 \text{ if } a \equiv 0 \pmod{p}. \end{cases}$$

If p is an odd prime and if α is a generator of Z^*_p. Then, a∈ Z^*_p is a QR modulo p iff a = α^i mod p, where i is an even integer.

Calculating Lagrange's number

Definition 3.1.6. *An integer a is said to be a quadratic residue modulo n if there exists $0 < x < n$ such that*

$$x^2 \equiv a \mod n.$$

Otherwise, a is said to be a non-quadratic residue modulo n.

If n is an odd prime, then determining whether or not an integer a is a quadratic residue modulo p is equivalent to calculating the Legendre symbol

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } a \equiv 0 \mod p \\ 1 & \text{if } a \not\equiv 0 \text{ and there exists } x \in \mathbb{Z} \text{ such that } a \equiv x^2 \mod p \\ -1 & \text{if no such } x \text{ exists} \end{cases}$$

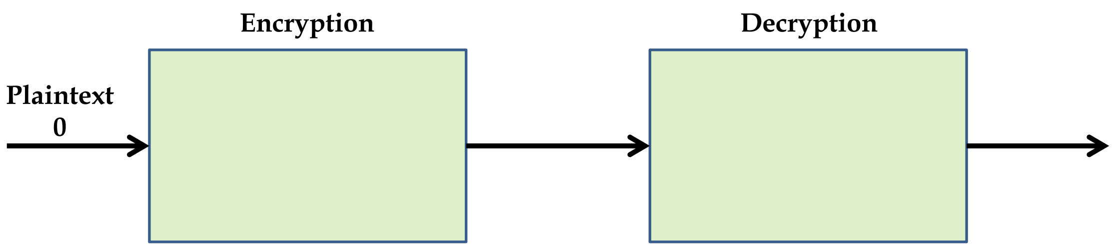
which can be efficiently calculated by the formula

$$\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \mod p.$$

Aids to the calculations

- Power Mod calculator: https://www.mtholyoke.edu/courses/quenell/s2003/ma139/js/powermod.html
- Quadratic residues calculator: https://asecuritysite.com/encryption/modsq?aval=44&pval=83
- Primitive roots calculator:
  http://www.bluetulip.org/2014/programs/primitive.html

Goldwasser-Micali – Algorithm



Plaintext m $\in \{0, 1\}$

Here, $m = 0 \in \{0, 1\}, N = 851,$

Goldwasser-Micali – Algorithm

Encryption

Decryption

Plaintext
0

$32^2 \; mod \; 851 = 173$

Ciphertext
173

Choose random r = 32 where 1 < r < n

Public Key $P_k = (\mathrm{a}, \mathrm{n}) = (80,\ 851)$

Ciphertext $\mathrm{C} = \begin{cases} r^2 \text{ mod n if m = 0} \\ ar^2 \text{ mod n if m = 1} \end{cases}$

Goldwasser-Micali – Algorithm

Encryption

Decryption

Plaintext
0

$32^2 \; mod \; 851 = 173$

Ciphertext
173

$\left(\frac{173}{23}\right) = 173^{11} \; mod \; 23 = 1$

Plaintext
0

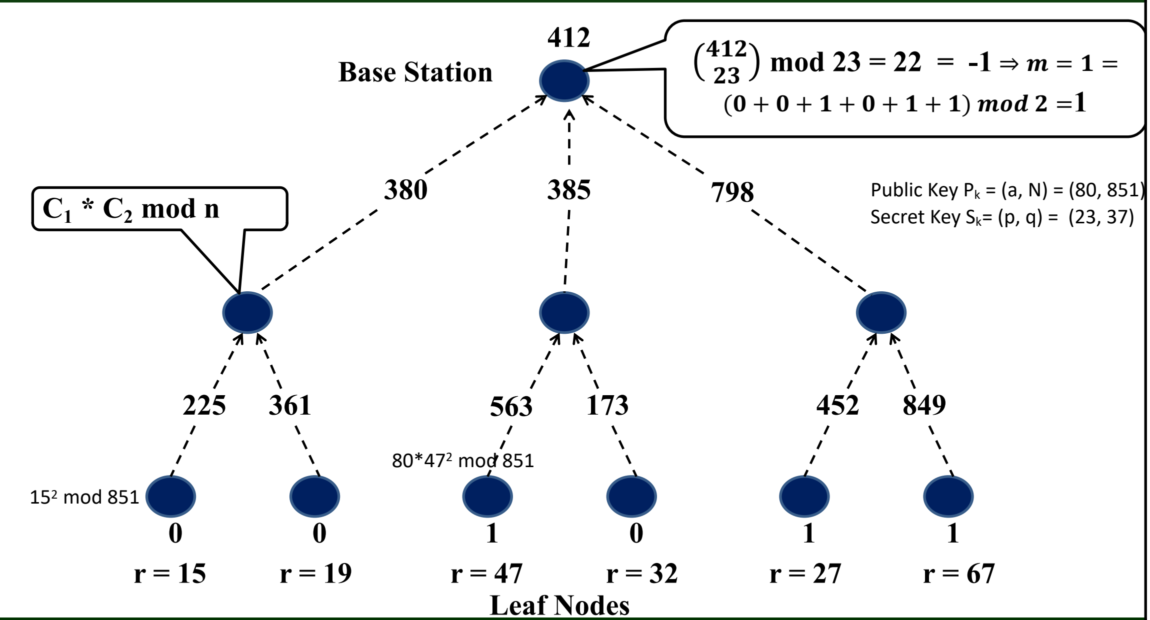
Compute, $\left(\frac{c}{p}\right) = C^{\frac{p-1}{2}} \; mod \; \mathrm{p}$

Ciphertext $\mathrm{m} = \begin{cases} \mathrm{m = 0 \; if} \left(\frac{c}{p}\right) = 1 \\ \mathrm{m = 1 \; if} \left(\frac{c}{p}\right) = -1 \end{cases}$

Goldwasser-Micali – Algorithm – Homomorphic Property

$$C_1 * C_2 \bmod n = E(m_1 + m_2) \bmod 2$$

Goldwasser-Micali – Algorithm – Example



Asymmetric Key Homomorphic Algorithms

- Deterministic Algorithms
  - RSA Algorithm
- Probabilistic Algorithms
  - The Goldwasser-Micali Algorithm
  - The Paillier Encryption Algorithm
  - The ElGamal Cryptosystem
  - The Okamoto-Uchiyama Cryptosystem

What are the primitive roots ?

- def: Primitive root: A primitive root of a prime p is an integer g such that g (mod p) has multiplicative order p−1.
  - Let $\alpha \in Z^*n$, the multiplicative order of α is ø(n).
  - So, what do we mean by saying that g is a primitive root if g (mod p) has multiplicative order p−1 ?
  - So, we can test for an element to a primitive as we did before.
  - Find ø(p) and find its distint prime factors and test whether each mod p is not 1 mod p.

What are the primitive roots in this example ?

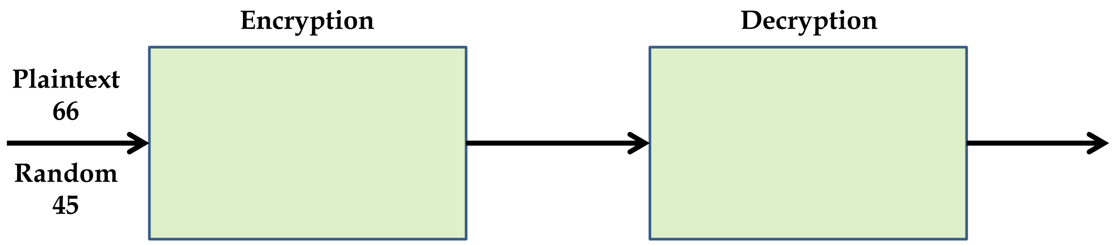
- Finding primitive roots of GF(107)
  - Find ø(p) and find its distinct prime factors and test whether each mod p is not 1 mod p.
  - Here, ø(p)= ø(107) =106.
  - And prime factors of 106 are 53 and 2.
  - Let us start from 2,
  - $2^{106/53}$ mod 107 = 2^2 mod 107 = 4 mod 107 $\not\equiv$ 1 mod 107
  - $2^{106/2}$ mod 107 = 2^{53} mod 107 $\not\equiv$ 1 mod 107.
  - Therefore, 2 is indeed primitive root of GF(107).

ElGamal – Key Generation

- Prime p = 107 and primitive root $\alpha = 2$
- Private key is chosen at random from {1..p-1} i.e. S_k= a = 67
- $\beta = \alpha^a \, mod \, p = 2^{67} mod \, 107 = 94$
- Public Key is $\{p, \alpha, \beta\}$ = {107, 2, 94}

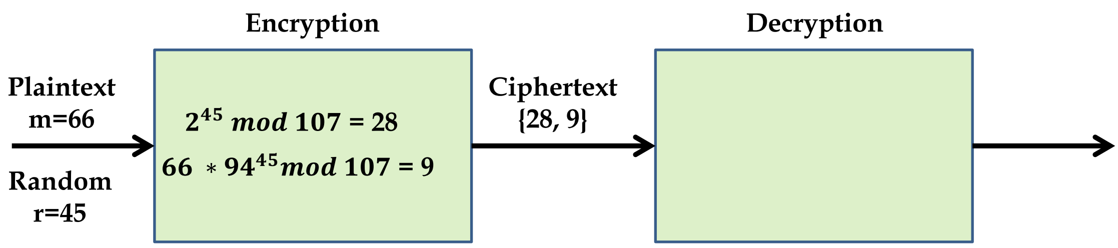
$\alpha \in$ GF(q) is called a primitive element of GF(q) if all the non-zero elements of GF(q) can be written as α^i for some (positive) integer *i*.

ElGamal– Algorithm



Plaintext, m $\in \mathbb{Z}_p$, Random Number, r
Here, m = 66 and r = 45

ElGamal – Algorithm



Public Key p_k = $\{p, \alpha, \beta\} = \{107, 2, 94\}$
$C_1 = \alpha^r \bmod p$
$C_2 = m * \beta^r \bmod p$

ElGamal – Algorithm

Encryption

Plaintext 66

Random 45

2^{45} mod $107 = 28$

$66 * 94^{45} mod$ $107 = 9$

Ciphertext {28, 9}

Decryption

$9 * (28^{-1} \ mod \ 107)^{67} mod \ 107$

$= 9 * 65^{67} \ mod \ 107 = 66$

Plaintext 66

Secret Key $S_k = a = 67$

$d_1 = C_2 * C_1^{-a} \ mod \ p$

ElGamal – Key Generation

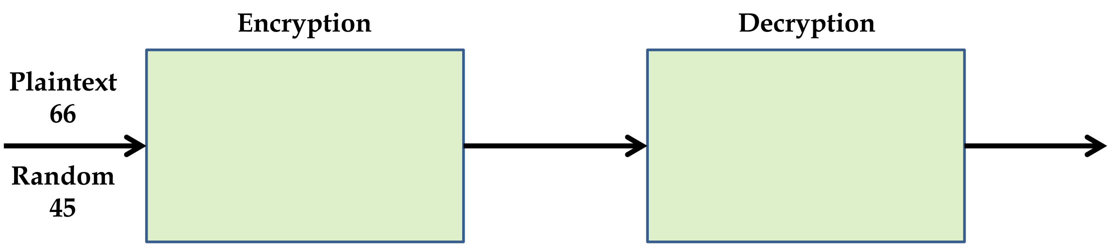
- Key setup with some other element as a primtive root …..
- Let GF be GF(107).
- Is 3 a primtive root of GF(107) ?
- Are 4, 8, 16 primitive roots of GF(107) ?
- Is 5 a primitive root ?

ElGamal – Key Generation

- Prime p = 107 and primitive root $\alpha = 5$
- Private key is chosen at random from {1..p-1} i.e. S_k= a = 67
- $\beta = \alpha^a \bmod p = 5^{67} \bmod 107 = 96$
- Public Key is $\{p, \alpha, \beta\}$ = {107, 5, 96}

$\alpha \in$ GF(q) is called a primitive element of GF(q) if all the non-zero elements of GF(q) can be written as α^i for some (positive) integer *i*.

ElGamal– Algorithm



Plaintext, m $\in \mathbb{Z}_p$, Random Number, r
Here, m = 66 and r = 45

ElGamal – Algorithm

Encryption

Plaintext
m=66

Random
r=45

5^{45} mod 107 = 97

66 $*$ $96^{45}$$mod$ 107 = 85

Ciphertext
{97, 85}

Decryption

Public Key p_k = $\{p, \alpha, \beta\} = \{107, 5, 96\}$

$$C_1 = \alpha^r \; mod \; p$$

$$C_2 = m * \beta^r mod \; p$$

ElGamal – Algorithm

Encryption

Plaintext
66

Random
45

5^{45} mod 107 = 97

66 $*$ $96^{45}$$mod$ 107 = 85

Ciphertext
{97, 85}

Decryption

85 $*$
$(97^{-1} \; mod \; 107)^{67} mod \; 107 =$
85 $*$ 32^{67} mod 107 = 66

Plaintext
66

Secret Key S_k = a = 67

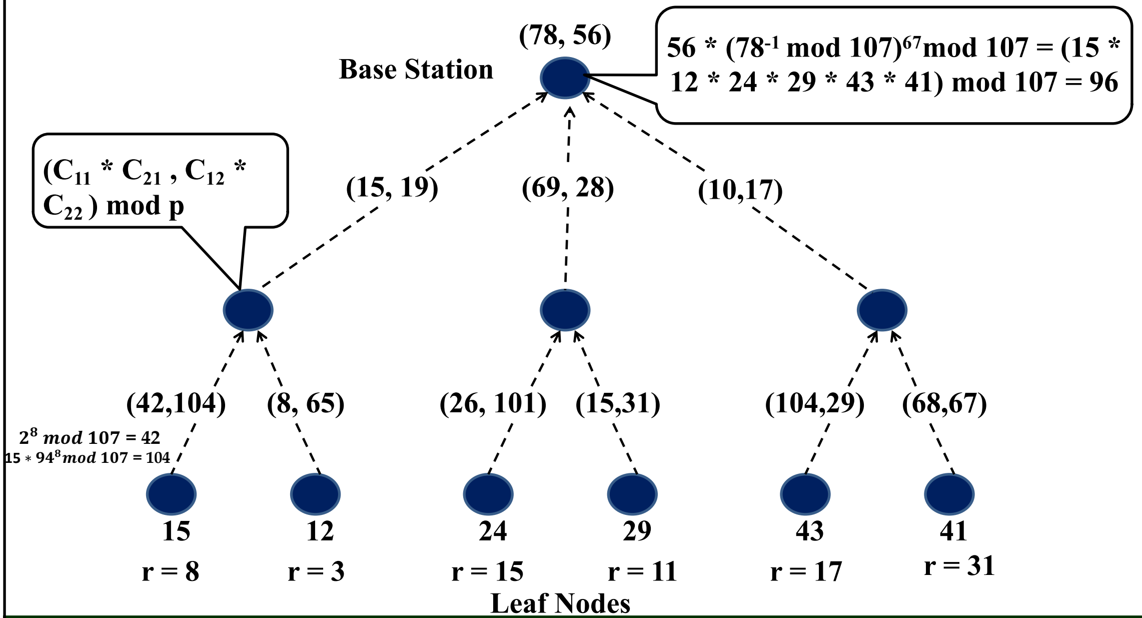
$$d_1 = C_2 * C_1^{-a} \; mod \; p$$

ElGamal Algorithm – Homomorphic Property

$$(C_{11} * C_{21}, C_{12} C_{22}) \bmod p = E(m_1 * m_2) \bmod p$$

47

ElGamal Algorithm – Example



48

Asymmetric Key Homomorphic Algorithms

- Deterministic Algorithms
  - RSA Algorithm
- Probabilistic Algorithms
  - The Goldwasser-Micali Algorithm
  - The Paillier Encryption Algorithm
  - The ElGamal Cryptosystem
  - The Okamoto-Uchiyama Cryptosystem

Paillier Algorithm

Algorithm Paillier ()

Key Generation:

1. Choose two large prime numbers p and q randomly and independently of each other such that gcd(pq,(p-1)(q-1))=1.
2. This property is assured if both primes are of equivalent length, i.e. $p, q \in 1||\{0,1\}^{\{s-1\}}$ for security parameter s.
3. Compute n=pq and $\lambda = lcm(p-1, q-1)$.
4. Select random integer g where $g \in Z^*_{n^2}$.
   Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse:
   $\mu = (L(g^{\lambda} mod\, n^2))^{-1} mod\, n,$
   where function L is defined as, L(u) = (u-1)/n
5. The public (encryption) key is (n,g).
6. The private (decryption) key is (λ, μ).

Message Encryption: Let m be a message to be encrypted where $m \in Z_n$.
Select a random r where r $\in Z_{n^*}$
Compute ciphertext as: $c = g^m.r^n\ mod\, n^2$

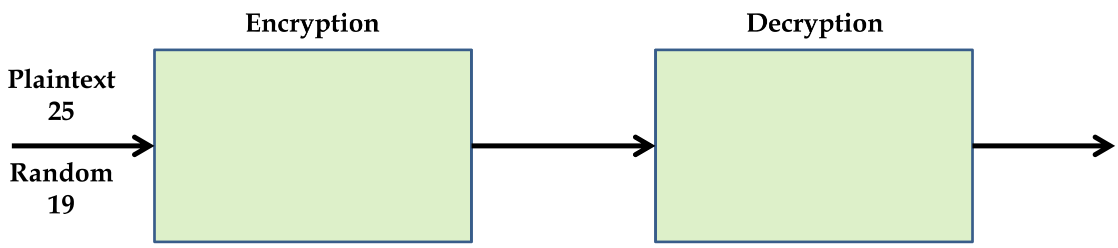
Decryption: Ciphertext c $\in Z^*_{n^2}$
Compute message: $m = L(c^{\lambda}\ mod\, n^2).\mu\ mod\, n$

Paillier – Key Generation

- Select Prime p = 7 and q = 11
- Compute n = p * q = 77……….n^2 = 5929
- Choose at random a number g = 5652 ∈ $\mathbb{Z}^*_{n^2}$
- Compute Carmichael's function $\lambda(n) = lcm[(p-1)(q-1)]$
- Compute $\mu = \left(L\left(g^{\lambda}\ mod\ n^2\right)\right)^{-1}\ mod\ n\ \ldots Here, L(u) = (u-1)/n$

- The security is based on the decisional composite residuosity assumption (DCRA). The DCRA states that given a composite n and an integer z, it is hard to decide whether z is a n-residue modulo n^2 or not, i.e., whether there exists y such that $z \cong y^n \bmod n^2$

Paillier – Algorithm



Plaintext, m ∈ $\mathbb{Z}_n$, Random Number, r ∈ $\mathbb{Z}^*_n$
Here, $m = 25$ and r = 19

Paillier – Algorithm

Encryption

Decryption

Plaintext
m=25

Random
19

$5652^{25}19^{77}\ mod\ 77^2 =$
3390

Ciphertext
{3390}

Public Key p_k = $(n, g) = (77, 5652)$

$C = g^m r^n\ mod\ n^2$

Paillier – Algorithm

Encryption

d = $L(C^{\lambda}\ mod\ n^2)\ \mu$ mod n
Decryption

Plaintext
25

Random
19

$5652^{25}19^{77}\ mod\ 77^2 =$
3390

Ciphertext
{3390}

$L(3390^{30}\ mod\ 77^2)$
$74\ mod\ 77$
= 43 * 74 mod 77 = 25

Plaintext
25

λ = lcm[(p-1),(q-1)] = lcm(10, 6) = 30.

$\mu = \left(L(g^{\lambda}\ \mathrm{mod}\ n^2)\right)^{-1}$ mod n , i.e. $L(u) = (u-1)/n$

μ = $(((5652^{30}\ mod\ 772) - 1)/77)^{-1}\ mod\ 77 = 74$

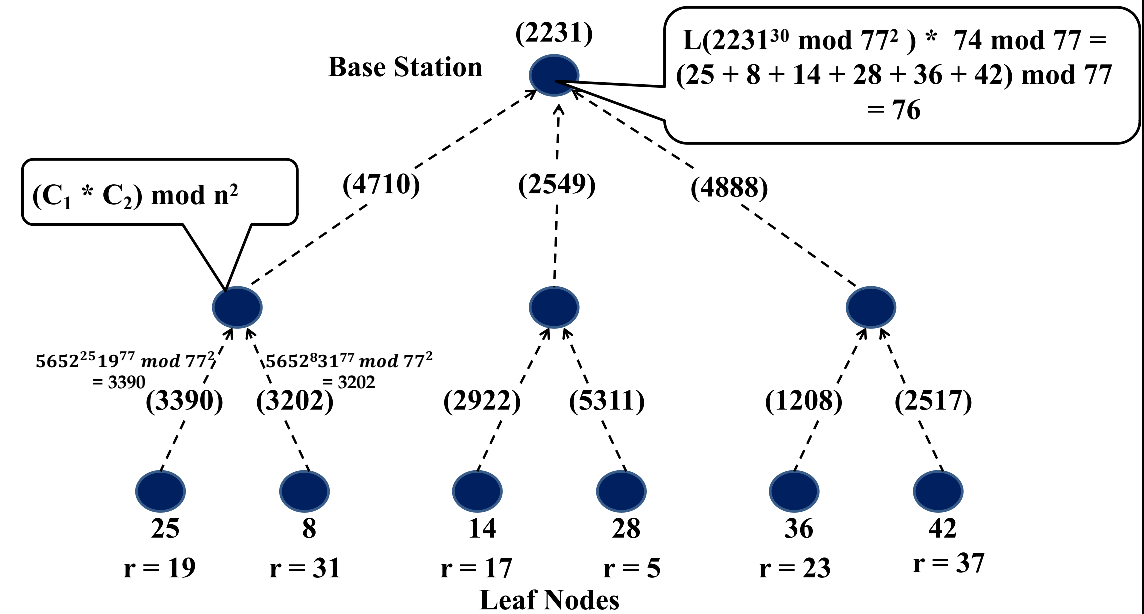
Secret Key S_k = (λ, μ) = (30, 74),

d = $L(C^{\lambda}\ mod\ n^2)\ \mu$ mod n

Paillier Algorithm – Homomorphic Property

$$(C_1 * C_2) \bmod n^2 = E(m_1 + m_2) \bmod n$$

Paillier Algorithm – Example



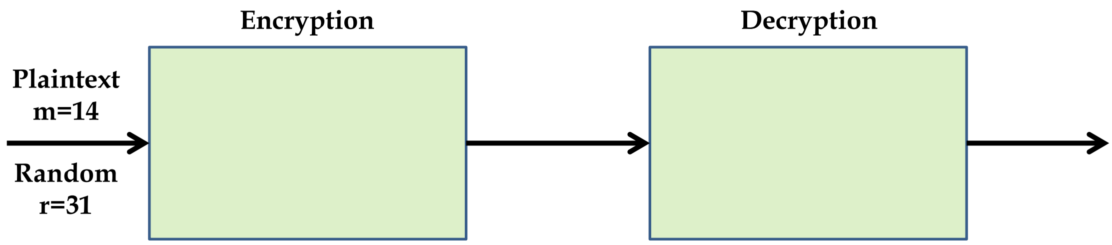
Asymmetric Key Homomorphic Algorithms

- Deterministic Algorithms
  - RSA Algorithm
- Probabilistic Algorithms
  - The Goldwasser-Micali Algorithm
  - The Paillier Encryption Algorithm
  - The ElGamal Cryptosystem
  - The Okamoto-Uchiyama Cryptosystem

Okamoto-Uchiyama Algorithm- Key Generation

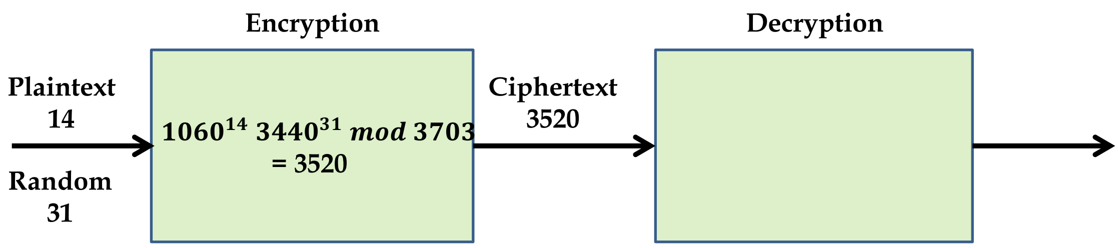
- Choose two large primes p and q – say p = 23, q = 7
- Let $n = p^2 * q$ = 529 * 7= 3703
- Choose $g \in \mathbb{Z}_n^* \ni g^{p(p-1)} \equiv 1 \; mod \; p^2 \; and \; g^{p-1} \neq 1 \; mod \; p^2$
  - say g = 1060…….then,
  - 1060^{23*22} mod p^2 = ???? and
  - 1060^{23*22} mod p^2 = ???
- $h = g^n \; mod \; n = 10603703 \; mod \; 3703 = \; 3440$
- Public Key is (n, g, h) = (3703, 1060, 3440)
- Private Key is (p, q) = (23, 7)

Okamoto-Uchiyama Algorithm



Plaintext, m $\in \mathbb{Z}_p$, Random Number, r $\in \mathbb{Z}_n$
Here, m = 14 and r = 31

Okamoto-Uchiyama Algorithm



Public Key p_k = $\{n, g, h\} = \{3703{,}1060{,}3440\}$
$c = g^m\ h^r\ mod\ n$

Okamoto-Uchiyama Algorithm

Encryption

Plaintext 14

Random 31

$$1060^{14}\ 3440^{31}\ mod\ 3703 = 3520$$

Ciphertext 3520

Decryption

$$\frac{L(3520^{22}\ mod\ 23^2)}{L(1060^{22}\ mod\ 23^2)} mod\ 23 = \frac{8}{17}\ mod\ 23 = 14$$

Plaintext 14

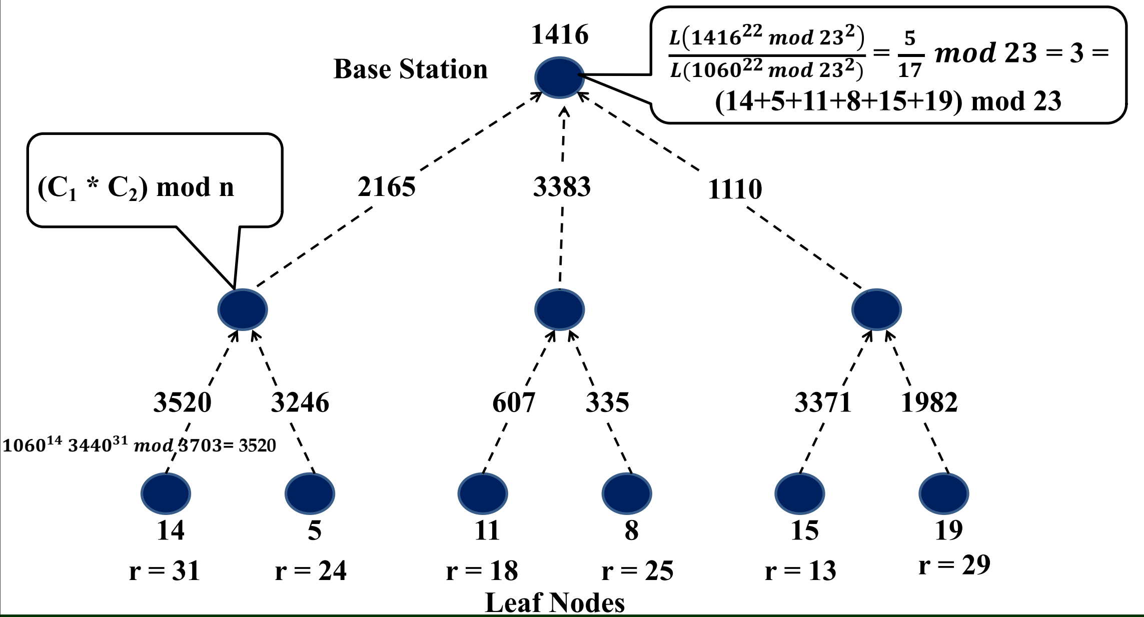
Ciphertext c and Secret key (p, q)

$$Plaintext, m = \frac{L(c^{p-1}\ mod\ p^2)}{L(g^{p-1}\ mod\ p^2)}\ mod\ p, \qquad Here\ L(x) = \frac{(x-1)}{p}$$

Okamoto-Uchiyama Algorithm – Homomorphic Property

$$(C_1 * C_2)\ \mathrm{mod\ n} = \mathrm{E}(\mathrm{m}_1 + \mathrm{m}_2)\ \mathrm{mod\ n}$$

Okamoto-Uchiyama Algorithm – Example



Symmetric/Asymmetric Key SDA Algorithms

Cryptosystem	SKC/ PKC	Security Assumption	Homomorphic Operations	Message Expansion
Castelluccia	SKC	-----	$\oplus$	1
Domingo-Ferrer	SKC	-----	$\oplus$ $\ominus$ $\otimes$ $\otimes_C$	d > 2
Stefeen Peter	SKC	-----	$\oplus$ $\ominus$ $\otimes$ $\otimes_C$	d > 2
RSA	PKC	RSA Problem	$\otimes$	1
Goldwasser-Micali	PKC	Quadratic Residuosity Problem	X-OR	N
Paillier	PKC	Composite Residuosity Problem	$\oplus$ $\ominus$ $\otimes_C$	2
ElGamal	PKC	Discrete Logarithms and Diffie-Hellman Problem	$\otimes$	2
Okamoto-Uchiyama	PKC	Integer Factorization and p-subgroup Problem	$\oplus$ $\ominus$ $\otimes_C$	3

Limitations

- An inherent drawback of homomorphic cryptosystems is
  - that attacks on these systems might possibly exploit their additional structural information.
  - Inherent malleability
- For instance, using plain RSA for signing,
  - the multiplication of two signatures yields a valid signature of the product of the two corresponding messages
- There are ways to avoid such attacks, for instance,
  - by application of hash functions, the use of redundancy or probabilistic schemes,

Contents

More Application Scenarios

- Protection of the mobile agents
- Cloud based secure processing
- Multiparty computation
- Secret sharing scheme
- Threshold schemes
- Zero-knowledge proofs
- Election schemes
- Watermarking and fingerprinting schemes
- Oblivious transfer
- Commitment schemes
- Lottery protocols
- Mix-nets

Protection of the mobile agents

- One of the most interesting applications of homomorphic encryption is its use in protection of mobile agents.
- All conventional computer architectures are based on binary strings and only require multiplication and addition,
  - homomorphic cryptosystems would offer the possibility to encrypt a whole program so that it is still executable.
- Hence, it could be used to protect mobile agents against malicious hosts by encrypting them.
- Two scenarios are possible here
  - computing with encrypted functions and
  - computing with encrypted data.

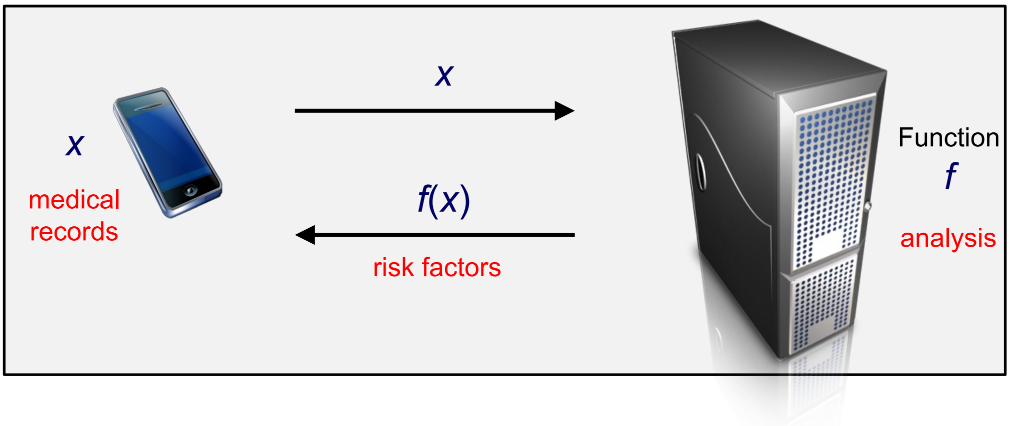
Source: Homomorphic Encryption — Theory and Application. By Jaydip Sen, DOI: 10.5772/56687, Intech Publishers

Protection of the mobile agents

- Computation with encrypted functions
  - a special case of protection of mobile agents.
  - a secret function is publicly evaluated in such a way that the function remains secret.
  - using homomorphic cryptosystems, the encrypted function can be evaluated which guarantees its privacy.

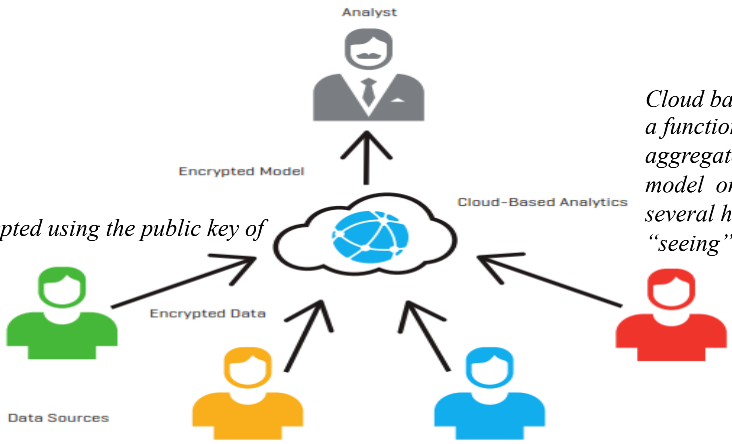
https://www.govinfo.gov/content/pkg/GOVPUB-C13-5e600e94dd9588c3cd717d5201830fdb/pdf/GOVPUB-C13-5e600e94dd9588c3cd717d5201830fdb.pdf

Cloud based secure processing…



Cloud based secure processing…

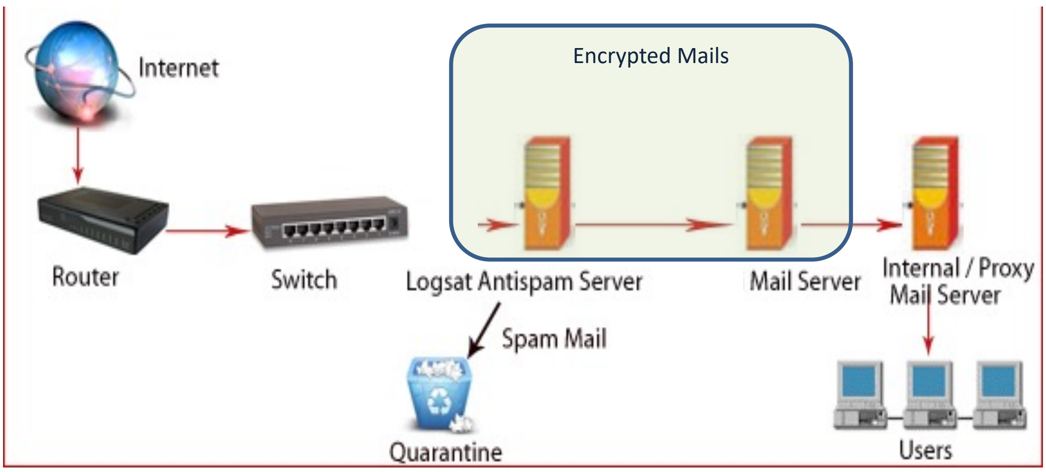
Data encrypted using the public key of the analyst



Cloud based analytics: To execute a function that computes aggregate statistics or develop a model on data collected from several hospitals without cloud "seeing" the data

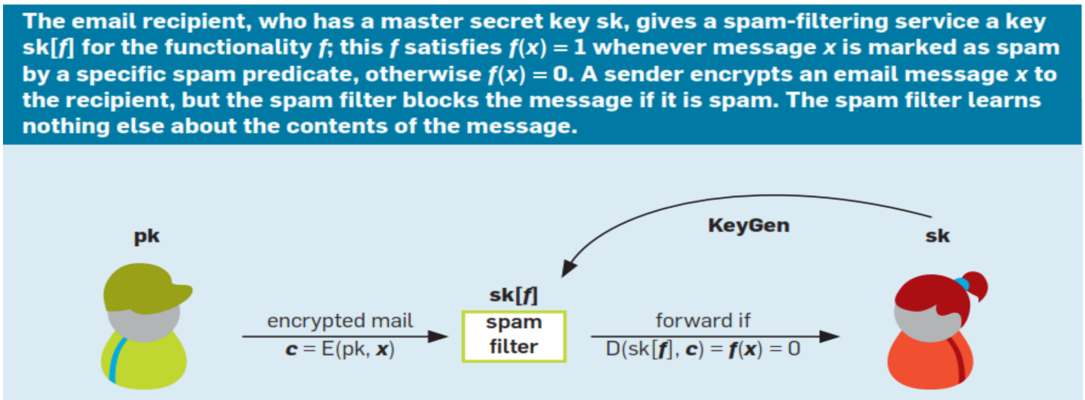
In this example, the goal is to learn a model or compute a function on data collected by several parties. For concreteness, suppose a health specialist is tracking the spread of an epidemic and is requesting patient information from different hospitals. Using FHE, each of the hospitals can submit their records encrypted under the analyst's public key to a cloud-based computing platform. Using the homomorphic properties of the encryption scheme, the cloud then computes some aggregate statistics or develops a model on the input data. The encrypted result of the computation is then provided to the health specialist or analyst. As long as the cloud and the analyst do not collude, the analyst only obtains the model and does not see the individual patient records.

Spam filterring e-mail server with privacy



Brent Waters, CACM 2012

Spam filterring e-mail server with privacy…



Brent Waters, CACM 2012

Protection of the mobile agents

- Computation with encrypted data
  - homomorphic schemes also work on encrypted data
  - the aim is to compute publicly while maintaining the privacy of the secret data.
  - this can be done encrypting the data in advance and then exploiting the homomorphic property to compute with encrypted data.

Cloud based secure processing