An Overview of Homomorphic Encryption

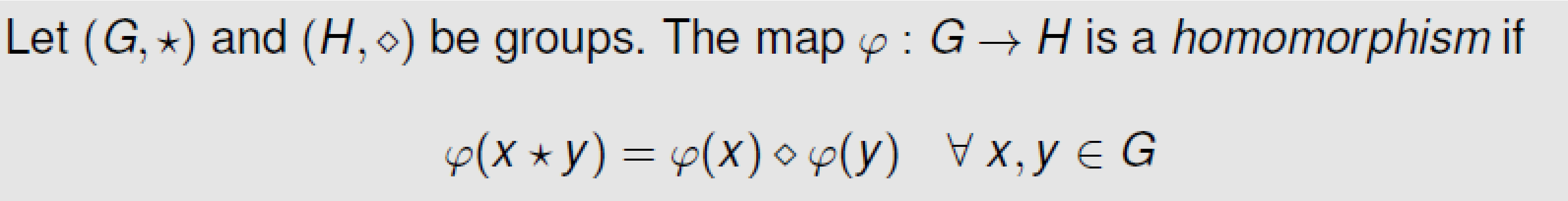
Asst. Prof. Dr. Serap ŞAHİN

Ceng 471 Cryptography

Ref: *Presentation of Alexander Lange, Department of Computer Science Rochester Institute of Technology, 2011*.

Algebraic Homomorphisms

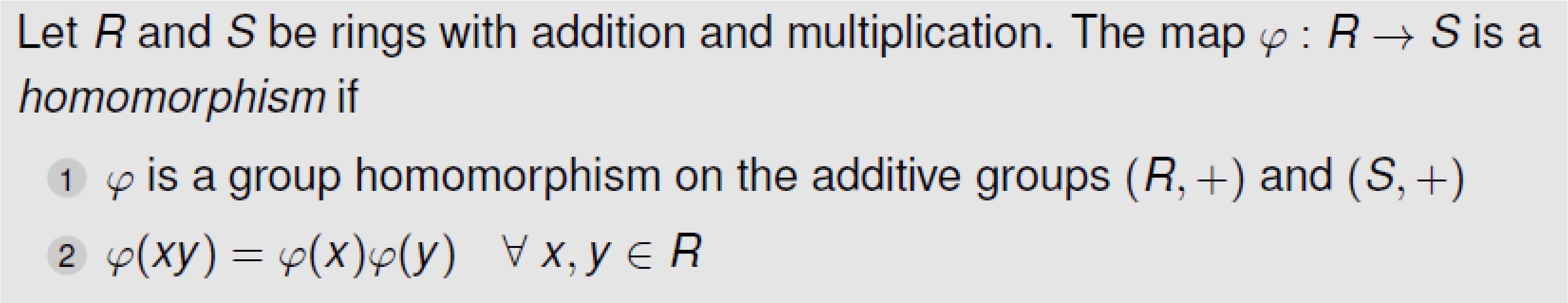
- Definition (Group Homomorphism)



Let $(G, \star)$ and $(H, \diamond)$ be groups. The map $\varphi : G \to H$ is a *homomorphism* if

$$\varphi(x \star y) = \varphi(x) \diamond \varphi(y) \quad \forall\, x, y \in G$$

- Definition (Ring Homomorphism)



Let R and S be rings with addition and multiplication. The map $\varphi : R \to S$ is a *homomorphism* if

1. φ is a group homomorphism on the additive groups $(R, +)$ and $(S, +)$
2. $\varphi(xy) = \varphi(x)\varphi(y) \quad \forall\, x, y \in R$

Application to Cryptography

A homomorphic encryption function allows for the manipulation of encrypted data with out the seemingly inherent loss of the encryption.

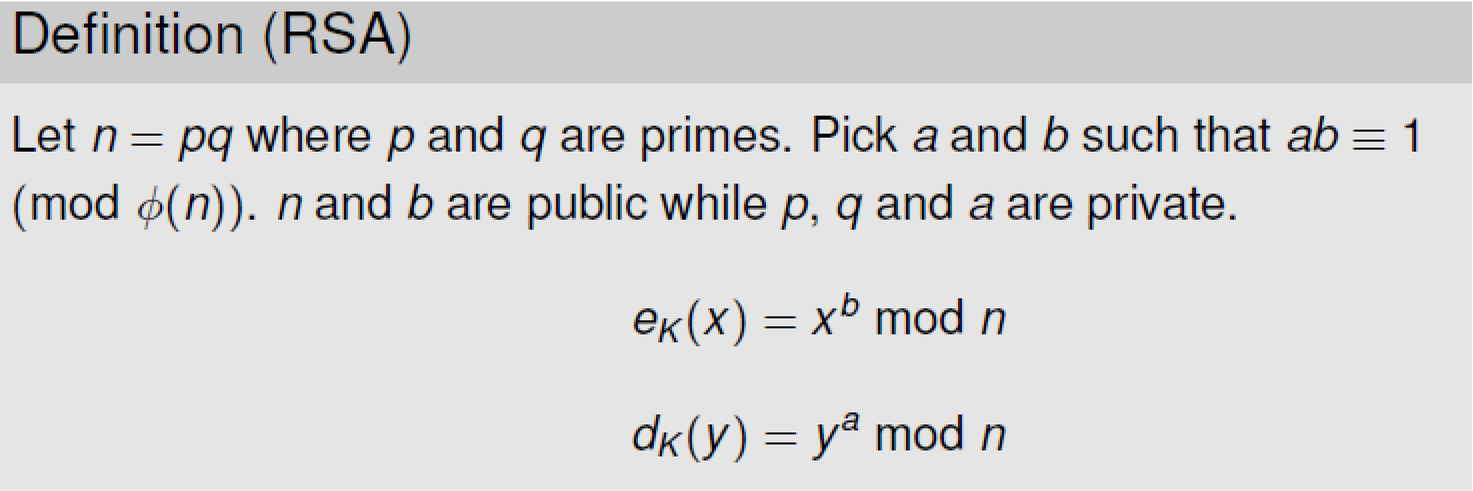
Applications;

- E-Cash
- E-Voting
- Private information retrieval
- Cloud computing

A fully homomorphic encryption function (two operations) has been an open problem in cryptography for more than 30 years. The first ever system was proposed by Craig Gentry in 2009.

However, encryption systems that respect one operation have been utilized for decades.

Example: The RSA Cryptosystem



Definition (RSA)

Let $n = pq$ where p and q are primes. Pick a and b such that $ab \equiv 1 \pmod{\phi(n)}$. n and b are public while p, q and a are private.

$$e_K(x) = x^b \bmod n$$

$$d_K(y) = y^a \bmod n$$

The Homomorphism: Suppose x_1 and x_2 are plaintexts. Then,

$$e_K(x_1)e_K(x_2) = x_1^b\, x_2^b \bmod n = (x_1 x_2)^b \bmod n = e_K(x_1 x_2)$$

History

On Data Banks and Privacy Homomorphisms

- Rivest, Adleman and Dertouzos, 1978
- Introduced idea of “Privacy Homomorphisms”
- “...it appears likely that there exist encryption functions which permit encrypted data to be operated on without preliminary decryption.”
- Introduced four possible encryption functions (RSA was one of them)

History

Blind signatures for untraceable payments

- David Chaum, 1982

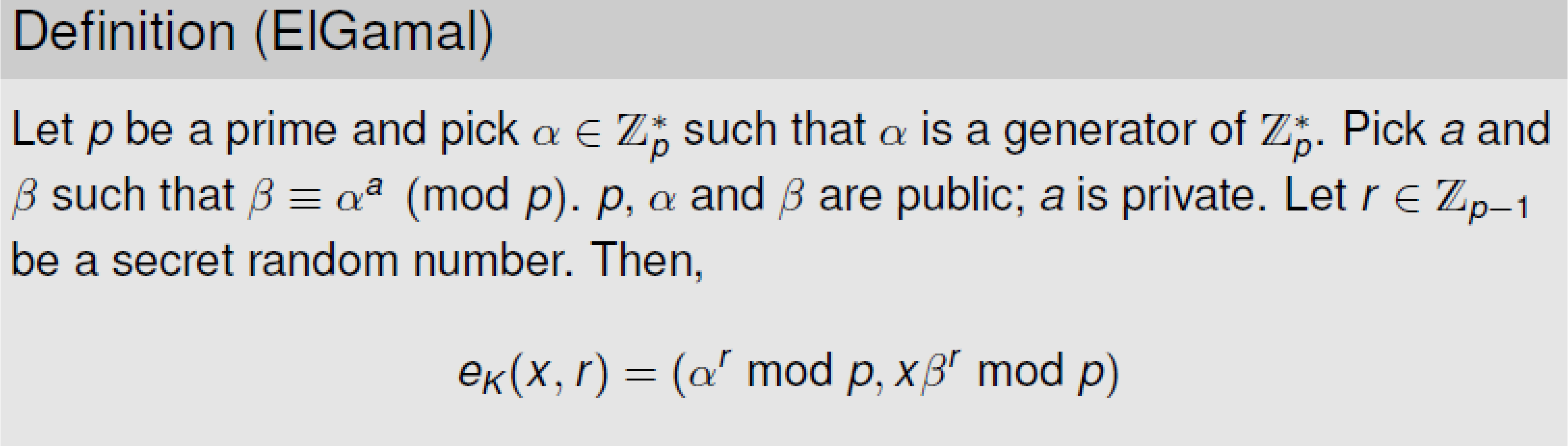
Calls for payment system with:

- Anonymity of payment
- Proof of payment

Analogy to secure voting

- Place vote in a carbon envelope
- The signer can then sign the envelope, consequently signing the vote without ever knowing what the vote is
- Although no mention of a private homomorphism, the paper helps introduce the need for secure voting as well as the relationship between e-cash and e-voting

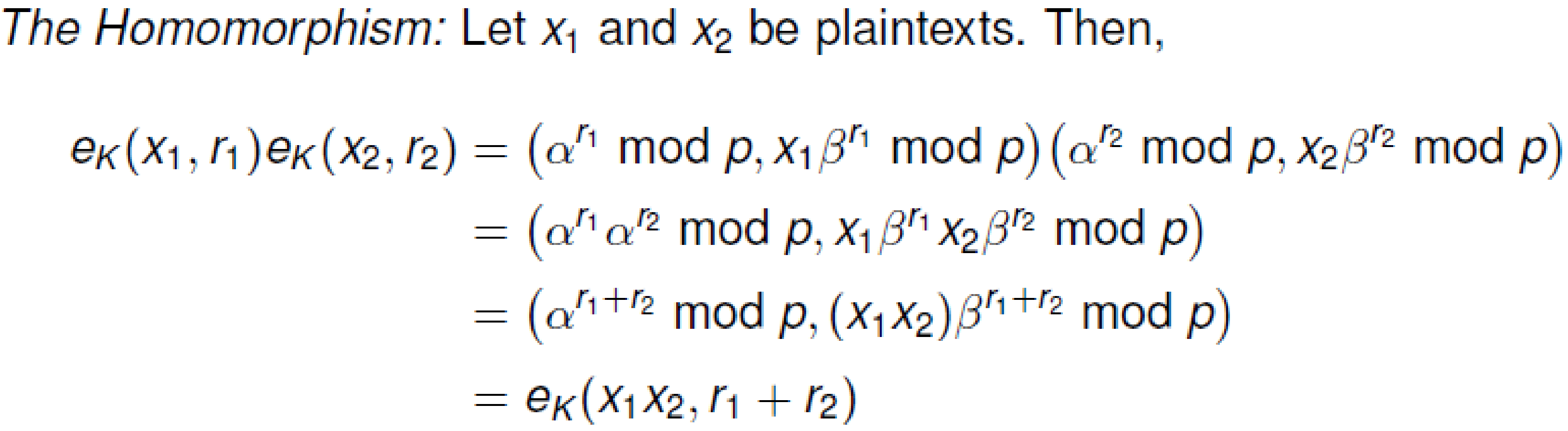
ElGamal Cryptosystem



Definition (ElGamal)

Let p be a prime and pick $\alpha \in \mathbb{Z}_p^*$ such that α is a generator of $\mathbb{Z}_p^*$. Pick a and β such that $\beta \equiv \alpha^a \pmod{p}$. p, α and β are public; a is private. Let $r \in \mathbb{Z}_{p-1}$ be a secret random number. Then,

$$e_K(x, r) = (\alpha^r \bmod p, x\beta^r \bmod p)$$



The Homomorphism: Let x_1 and x_2 be plaintexts. Then,

$$\begin{aligned} e_K(x_1, r_1)e_K(x_2, r_2) &= (\alpha^{r_1} \bmod p, x_1\beta^{r_1} \bmod p)(\alpha^{r_2} \bmod p, x_2\beta^{r_2} \bmod p) \\ &= (\alpha^{r_1}\alpha^{r_2} \bmod p, x_1\beta^{r_1}x_2\beta^{r_2} \bmod p) \\ &= (\alpha^{r_1+r_2} \bmod p, (x_1x_2)\beta^{r_1+r_2} \bmod p) \\ &= e_K(x_1x_2, r_1 + r_2) \end{aligned}$$

ElGamal

The Problem: This homomorphism is multiplicative

- E-cash and e-voting would benefit from an additive homomorphism

One solution: Modify ElGamal

- Put the plaintext in the exponent

If we modify ElGamal so that

$$e_K(x, r) = (\alpha^r \bmod p, \alpha^x \beta^r \bmod p)$$

Then the homomorphism is

$$\begin{aligned} e_K(x_1, r_1) e_K(x_2, r_2) &= (\alpha^{r_1} \bmod p, \alpha^{x_1} \beta^{r_1} \bmod p)(\alpha^{r_2} \bmod p, \alpha^{x_2} \beta^{r_2} \bmod p) \\ &= (\alpha^{r_1 + r_2} \bmod p, \alpha^{x_1 + x_2} \beta^{r_1 + r_2} \bmod p) \\ &= e_K(x_1 + x_2, r_1 + r_2) \end{aligned}$$

The problem with this modification is that $d_K = \alpha^x$, introducing the *discrete logarithm* problem into the decryption. For large enough texts, this becomes impractical.

We would like another cryptosystem which takes advantage of this additive property of exponentiation, but does so with out extra decryption time.

Solution: the Paillier Cryptosystem

Quadratic Residue and Cryptography

- Suppose p is an odd prime and a is an integer.
- a is defined to be a quadratic residue $modulo\ p$
  - if $a \neq 0\ (mod\ p)$ and congruence $y^2 \equiv a\ (mod\ p)$ has a solution y where nonnegative y is less than n.
- For Example Rabin Cryptosystem:
  - Let $n = pq$, where p and q are primes and $p, q \equiv 3\ (mod\ 4)$
  - The value n is public key, while p and q are the private key.
  - For the plain text $m < n$, we define $c = m^2\ (mod\ n)$
- The Paillier cryptosystem uses composite residue.

Composite Residue

- $n = pq$ where p and q are large primes.
- The set of $n - th$ residues is a multiplicative subgroups of $Z^*_{n^2}$ of order $\Phi(n)$.
- **The Problem of deciding n-th residuosity**, that is, distinguishing n-th residues from non n-th residues will be denoted CR[n]. As for prime residiosity, deciding n-th residiosity, is believed to be computationally hard.
- Let g be some element of $Z^*_{n^2}$ and denote by ε_g the integer-valued function defined by

$$Z_n \times Z^*_n \rightarrow Z^*_{n^2}$$

$$(x, y) \rightarrow g^x y^n \ (mod\ n^2)$$

Here, if the order of g is a nonzero multiple of n then ε_g is bijection.

Definition: Assume that $g \in B$. For $w \in Z_{n^2}^*$, we call n-th residiosity class of w with respect to g the unique integer $x \in Z_n$ for which there exists y $\in Z_{n^2}^*$, such that

$$\varepsilon_g(x, y) = w$$

The class of w is denoted $[[w]]_g$

Lemma: $[[w]]_g = 0$ iff w is an n-th residue modulo n^2. Furtheremore;

$$\forall w_1, w_2 \in Z_{n^2}^* \;\; \left[[w_1 w_2]\right]_g = [[w_1]]_g + [[w_2]]_g (mod\ n)$$

That is, the class function $w \rightarrow [[w]]_g$ is a homomorphism from $(Z_{n^2}^*, \mathrm{x})$ to $(Z_n, +)$ for any $g \in B$.

It can easily be shown that, for any w $\in Z_{n^2}^*$ and $g_1, g_2 \in B$, we have;

$[[w]]_{g_1} = [[w]]_{g_2}[[g_2]]_{g_1}\ (mod\ n)$, which yields $[[g_1]]_{g_2} = [[g_2]]_{g_1}{}^{-1}\ mod\ n$ and thus $[[g_2]]_{g_1}$ is invertible modulo n.

The set $S_n = \{u < \ n^2 \mid \ u = 1\ mod\ n\}$ is a multiplicative subgroup of integers modulo n^2 over which function such that : $\forall u \in S_n \qquad L(u) = \frac{u-1}{n}$

Application of n-th residuosity class problem on Cryptography

- $n = pq$ and randomlu select a base $g \in B$. We review that ε_g be the function defined by;

$$Z_n \times Z_n^* \rightarrow Z_{n^2}^*$$

$$(x, y) \rightarrow \varepsilon_g(x, y) = g^x y^n \ (mod\ n^2)$$

Encryption function ε_g is employed for the plain text x.

$$S_n = \{u < \ n^2 \mid \ u = 1\ mod\ n\} \rightarrow \ Z_n$$

$$u \ \rightarrow \ L(u) = \frac{u-1}{n}$$

For the ciphertext $c = \varepsilon_g(x, y)$, we employ the rate of these two functions $L(c^\lambda)$ and $L(g^\lambda)$ as an decryption function.

Application of n-th residuosity class problem on Cryptography

Theorem: We set $n = pq$ and $\lambda = lcm(p-1, q-1)$. For any $g \in B$, we obtain public-key cryptosystem as public keys (n, g) and private keys (p, q). For a plaintext $m < n$, we select a random $r < n$, and compute the ciphertext by the **encryption function:** $c = g^m r^n \ (mod\ n^2)$

And we compute the plaintext by the **decryption function:**

$$m = \frac{L(c^{\lambda}\ (mod\ n^2))}{L(g^{\lambda}\ (mod\ n^2))}\ (mod\ n)$$

Example

$n = 15$ and $g = 14$. For the plaintext $m = 3$, a random number $r = 4$ is selected, we compute the ciphertext $c = 206$ with $c = g^m r^n \ (mod\ n^2)$.

For this cyphertext we compute plaintext:

$$m = \frac{L(206^4\ (mod\ n^2))}{L(14^4\ (mod\ n^2))} = \frac{L(46)}{L(166)}\ (mod\ n)$$

Here we compute:

$$L(46) = \frac{46-1}{15} = 3\ (mod\ n)$$

$$L(166) = \frac{166-1}{15} = 11\ (mod\ n)$$

Then;

$$m = \frac{L(46)}{L(166)} = \frac{3}{11} = 3\ (mod\ n)$$

- For $n = pq$, we obtain the public key cryptosystem based on the n-th residuosity class problem.

Paillier Cryptosystem

Introduced by Pascal Paillier in Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, 1999

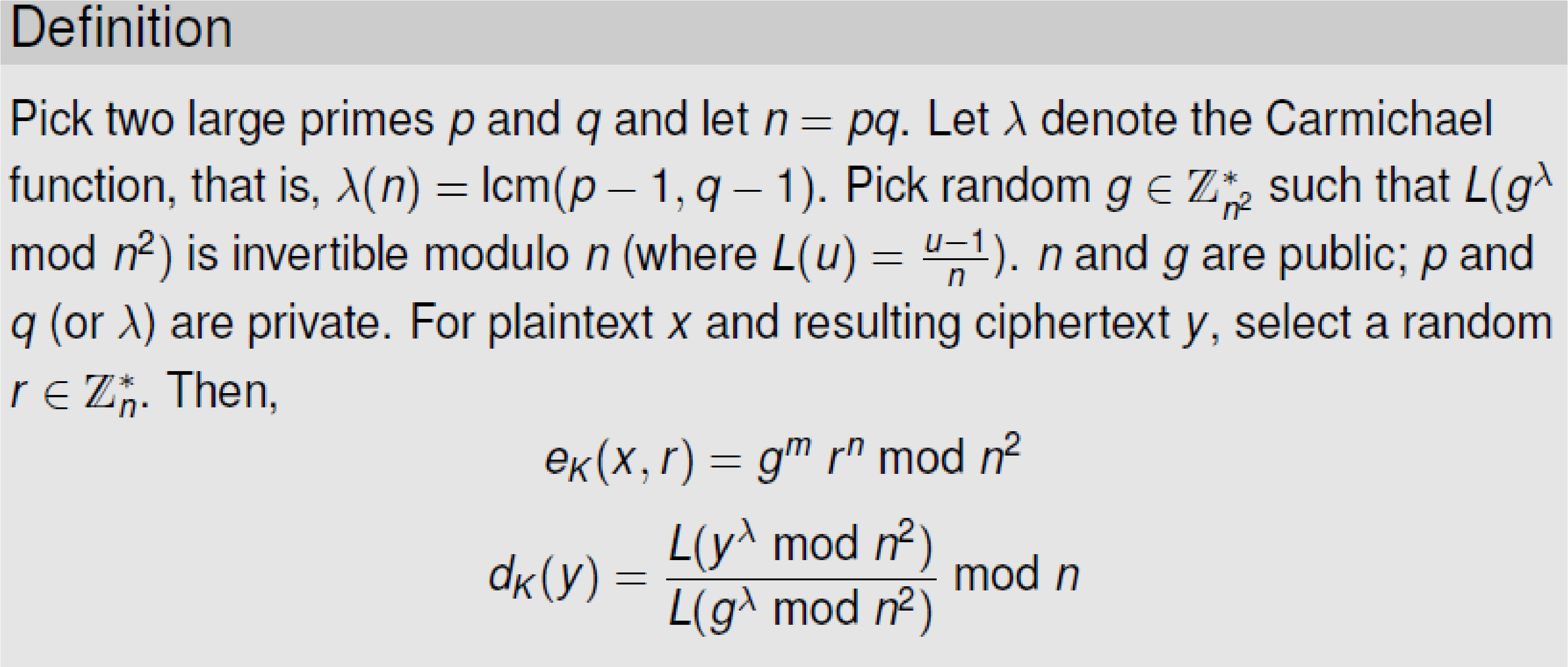
- Probabilistic, asymmetric algorithm
- Decisional composite residuosity assumption
- Given composite n and integer z, it is hard to determine if y exists such that

$$z \equiv y^n \pmod{n^2}$$

- Homomorphic and self-blinding
- Extended by Damgard and Jurik in 2001

$$\text{modulo } n^2 \Longrightarrow \text{modulo } n^{s+1}$$

Paillier Cryptosystem



Definition

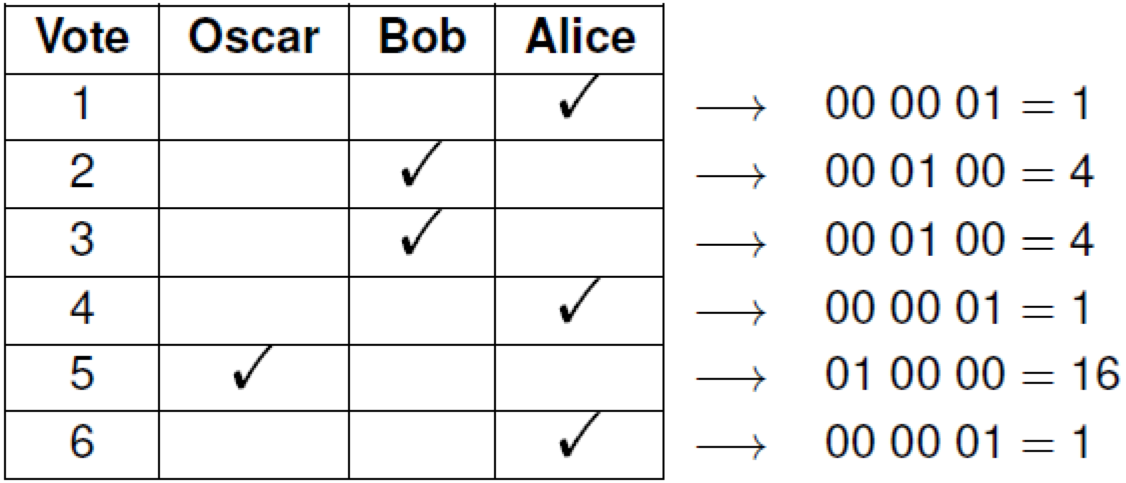
Pick two large primes p and q and let $n = pq$. Let λ denote the Carmichael function, that is, $\lambda(n) = \text{lcm}(p-1, q-1)$. Pick random $g \in \mathbb{Z}^*_{n^2}$ such that $L(g^\lambda \bmod n^2)$ is invertible modulo n (where $L(u) = \frac{u-1}{n}$). n and g are public; p and q (or λ) are private. For plaintext x and resulting ciphertext y, select a random $r \in \mathbb{Z}^*_n$. Then,

$$e_K(x, r) = g^m\ r^n \bmod n^2$$

$$d_K(y) = \frac{L(y^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$$

Paillier Example: E-Voting

- Suppose Alice, Bob and Oscar are running in an election. Only 6 people
- voted in the election, and the results are tabulated below.



Vote	Oscar	Bob	Alice		
1			✓	$\longrightarrow$	$00\ 00\ 01 = 1$
2		✓		$\longrightarrow$	$00\ 01\ 00 = 4$
3		✓		$\longrightarrow$	$00\ 01\ 00 = 4$
4			✓	$\longrightarrow$	$00\ 00\ 01 = 1$
5	✓			$\longrightarrow$	$01\ 00\ 00 = 16$
6			✓	$\longrightarrow$	$00\ 00\ 01 = 1$

Paillier Example: E-Voting

Let $p = 5$ and $q = 7$. Then $n = 35$, $n^2 = 1225$ and $\lambda = 12$. g is chosen to be 141. For the first vote $x_1 = 1$, r is randomly chosen as 4. Then,

$$e_K(x_1, r_1) = e_K(1, 4) = 141^1 \cdot 4^{35} = 141 \cdot 324 = 359 \bmod 1225$$

All votes, r values and resulting encryptions are shown below

Vote	Oscar	Bob	Alice		
1			✓	$\longrightarrow$	00 00 01 = 1
2		✓		$\longrightarrow$	00 01 00 = 4
3		✓		$\longrightarrow$	00 01 00 = 4
4			✓	$\longrightarrow$	00 00 01 = 1
5	✓			$\longrightarrow$	01 00 00 = 16
6			✓	$\longrightarrow$	00 00 01 = 1

x	r	$e_K(x, r)$
1	4	359
4	17	173
4	26	486
1	12	1088
16	11	541
1	32	163

Paillier Example: E-Voting

In order to sum the votes, we *multiply* the encrypted data modulo n^2:

$$359 \cdot 173 \cdot 486 \cdot 1088 \cdot 541 \cdot 163 \bmod 1225 = 983$$

We then decrypt:

$$\begin{aligned}
L(y^\lambda \bmod n^2) &= L(983^{12} \bmod 1225) = \frac{36-1}{35} = 1 \\
L(g^\lambda \bmod n^2) &= L(141^{12} \bmod 1225) = \frac{456-1}{35} = 13 \\
d_K(y) &= \left(L(y^\lambda \bmod n^2)\right)\left(L(g^\lambda \bmod n^2)\right)^{-1} \bmod n \\
&= 1 \cdot 13^{-1} \bmod 35 \\
&= 27
\end{aligned}$$

We convert 27 to (01 02 03) for the final results.

Another Application: Private Information Retrieval

Idea first introduced by Chor, Goldreich, Kushilevitz and Sudan in 1997

The problem:

- How can the user access an item from a database without the database knowing which item it is? (Private Information Retrieval)
- How can the user do this without knowing about any other item of the database? (Symmetric Private Information Retrieval)
- The additive homomorphic properties of Paillier allow for the indexing and filtering of an encrypted database

Some PIR Protocols

- **Stern Protocol** - Uses a simple homomorphic scheme and a linear indexing technique
- **Chang Protocol** - Expands on Stern by allowing the indexing to take place on a hyper cube. Uses the Paillier Cryptosystem
- **Lipmaa Protocol** - Expands on Chang by using Damgard-Jurik system - removes the limit set on the plaintext due to Paillier

Fully Homomorphic Encryption

Up until now, the homomorphic systems described have been partially homomorphic

- – They preserve the structures of multiplication or division, but cannot do both

- If a fully homomorphic encryption was implemented, then any arbitrary computation could be performed on a ciphertext, preserving the encryption as if the computation was performed on the plaintext
  - – The additive and multiplicative preservation of a Ring Homomorphism modulo 2 directly correspond to the XOR and AND operations of a circuit

Applications:

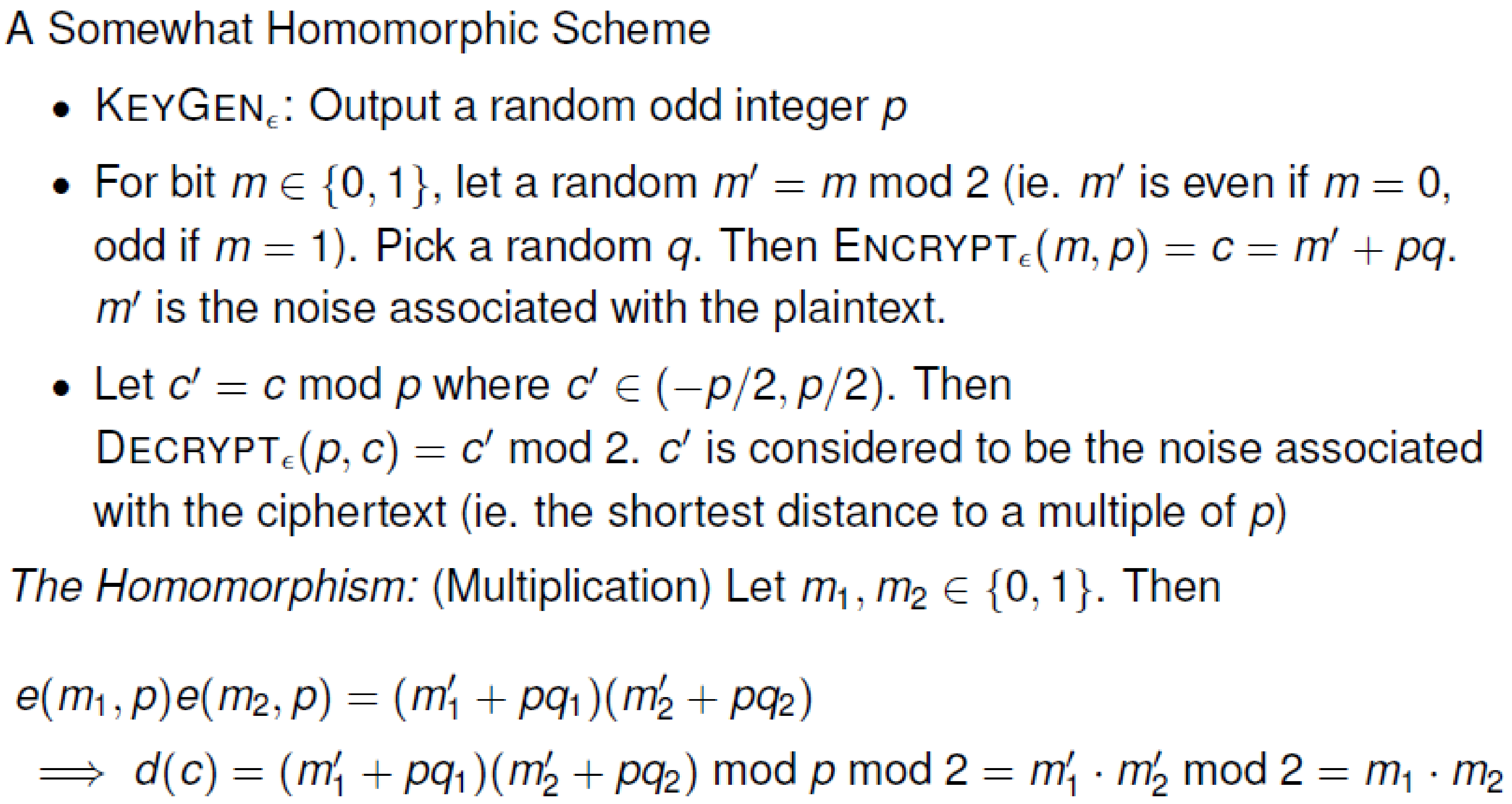
- Private queries on search engines - The search engine would be able to return encrypted data without every decrypting the query
- Cloud Computing - Storing encrypted data on the cloud is seemingly useless; no manipulation of the data can be obtained without allowing the cloud access and/or decrypting the data off the cloud

Craig Gentry

In 2009, Craig Gentry proposed the first fully homomorphic encryption scheme in his PhD thesis A Fully Homomorphic Encryption Scheme

- Centers around a function which introduces a certain level of noise into the encryption
  - Each operation on the ciphertext results in compounding noise
  - Resolved with the bootstrapability of the encryption
    - Each re-encryption cuts down the noise
    - Analogy to Alice’s jewelry shop
- Involves operations on Ideal Lattices
  - Allows for less complex circuit implementation
  - Correspond to the structure of Rings

A Simple Example Over the Integers



A Somewhat Homomorphic Scheme

- KeyGen_ϵ: Output a random odd integer p
- For bit $m \in \{0, 1\}$, let a random $m' = m \bmod 2$ (ie. m' is even if $m = 0$, odd if $m = 1$). Pick a random q. Then $\text{Encrypt}_\epsilon(m, p) = c = m' + pq$. m' is the noise associated with the plaintext.
- Let $c' = c \bmod p$ where $c' \in (-p/2, p/2)$. Then $\text{Decrypt}_\epsilon(p, c) = c' \bmod 2$. c' is considered to be the noise associated with the ciphertext (ie. the shortest distance to a multiple of p)

The Homomorphism: (Multiplication) Let $m_1, m_2 \in \{0, 1\}$. Then

$$e(m_1, p)e(m_2, p) = (m_1' + pq_1)(m_2' + pq_2)$$

$$\implies d(c) = (m_1' + pq_1)(m_2' + pq_2) \bmod p \bmod 2 = m_1' \cdot m_2' \bmod 2 = m_1 \cdot m_2$$

The Problem

- The compounding noise ($m'_1 \cdot m'_2$ in the example) results in loss of homomorphic property after a certain number of operations
  - The bootstrapping of the algorithm allows for this noise to be reduced, allowing for no limit in operations
- However, the combination of the noise production followed by the noise reduction makes the scheme completely impractical
  - Complexity grows as more and more operations are performed (inherent limitation of the algorithm)
  - Gentry has stated that in order to perform one search on Google using this encryption, the amount of computations needed would increase by a trillion
  - More schemes have been introduced to try and decrease this complexity, but all rely on the same
- Despite this impracticality, Gentry’s discovery is an amazing breakthrough in cryptography and proves that (at least theoretical) fully homomorphic encryption schemes exist

Sources

- R. L. Rivest, L. Adleman, and M. L. Dertouzos. ”On data banks and privacy homomorphisms.” Foundations of Secure Computation, 1978.
- Craig Gentry. ”Computing Arbitrary Functions of Encrypted Data.” Association for Computing Machinery, 2010.
- Pascal Paillier. ”Public-Key Cryptosystems Based on Composite Degree Residuosity Classes.” EUROCRYPT 1999.
- Laura Lincoln. ”Symmetric Private Information Retrieval via Additive Homomorphic Probabilistic Encryption”. RIT, Department of Computer Science, 2006.
- David Chaum. ”Blind signatures for untraceable payments.” Advances in Cryptology - Crypto ’82, Springer-Verlag 1983
- Paillier Cryptosystem Interactive Simulator. Andreas Steffen. HSR Hochschule fur Technik Rapperswil. 2009.
- Steve Weis. ”Verifying Elections with Cryptography”. Google Tech Talks: Theory and Practive of Cyptography. December 2007. Youtube.