

Cryptographic Protocols

Chapter 4

Homomorphic Encryption

4.1 Grand Goal

Computation with encrypted data. No interaction between user and cloud. User holds sk (private key).
Intriguing problem, first posed in 1978

4.2 Single-Homomorphic Encryption

Encryption scheme supports an operation \oplus such that:

$$\text{ENC}(x) \oplus \text{ENC}(y) = \text{ENC}(x + y)$$

OR an operation \otimes such that:

$$\text{ENC}(x) \otimes \text{ENC}(y) = \text{ENC}(x \cdot y)$$

where $x, y \in \mathbb{GF}(2)$ or $\mathbb{GF}(p)$

→ such schemes exist and are efficient

→ Additively kommutative ElGamal; Paillier scheme

4.3 Fully-Homomorphic Encryption

Encryption scheme supports an operation \oplus such that:

$$\text{ENC}(x) \oplus \text{ENC}(y) = \text{ENC}(x + y)$$

AND an operation \otimes such that:

$$\text{ENC}(x) \otimes \text{ENC}(y) = \text{ENC}(x \cdot y)$$

where $x, y \in \mathbb{GF}(2)$ or $\mathbb{GF}(p)$

→ exist since 2009, Gentry

→ not very practical

4.4 Examples

4.4.1 ElGamal-based Single-Homomorphic Encryption

Recall textbook ElGamal in $\mathbb{G} = \langle g \rangle$, $|\mathbb{G}| = q$:

KEYGEN()	ENC(m, y)	DEC($x, (R, C)$)
$x \leftarrow \mathbb{Z}_q$	$r \leftarrow \mathbb{Z}_q$	$\hat{m} \leftarrow c/R^x$
$y \leftarrow g^x$	$R \leftarrow g^r$	return \hat{m}
return (y, x)	$C \leftarrow m \cdot y^r$	
	return (R, C)	

- already multiplication homomorphic
- but mapping of integers to and from elements of \mathbb{G} can be hard
- but addition is preferred

4.4.2 Additively homomorphic ElGamal

- use textbook ElGamal
 - works only for small plaintexts, numbers e.g. from 0 to max
- | | | |
|-----------------------------|---------------------|---|
| KEYGEN() | AH-ENC(y, a) | AH-DEC($x, (R, C)$) |
| $x \leftarrow \mathbb{Z}_q$ | return ENC y, g^a | $h \leftarrow \text{DEC}(x, (R, C))$ |
| $y \leftarrow g^x$ | | // brute-force search for $i = 0, 1, \dots, \text{max}$ |
| return (y, x) | | if $g^i = h$ then: |
| | | return i |
| | | return ERROR |

$$\begin{aligned}
 \text{AH-ENC}(y, a) \oplus \text{AH-ENC}(y, b) &= (R_a, C_a) \oplus (R_b, C_b) \\
 &= (R_a \cdot R_b, C_a \cdot C_b) \\
 &= \text{AH-ENC}(y, (a + b) \bmod q)
 \end{aligned}$$

... $q \approx 2^{256}$

... $\text{max} \approx 10^6$

... $a, b \in [-\text{max}/2, \text{max}/2]$ using $a' = q + a$ for $a < 0$

4.5 Voting protocol using Additional Homomorphic Encryption

1. Parties P_1, \dots, P_n
 2. Each party P_i votes $v_i \in \{-1, 1\}$
 3. One authority \mathbb{A}
1. \mathbb{A} generates
 $(pk, sk) \leftarrow \text{KEYGEN}()$
 \mathbb{A} sends pk to P_1, \dots, P_n

2. \mathbb{A} computes
 $c_0 \leftarrow \text{ENC}(pk, 0)$
and sends c_0 to P_1
for $i = 1, \dots, n$ do
 P_i receives c_{i-1}
 P_i computes
 $c_i \leftarrow c_{i-1} \oplus \text{ENC}(y, v_i)$
 P_i sends c_i to P_{i+1} // P_n sends to \mathbb{A}
3. \mathbb{A} receives c_n from P_n computes
 $z \leftarrow \text{DEC}(x, c_n)$ // $z = \sum_{i=1}^n v_i$
 \mathbb{A} publishes z

Remarks:

- sending must use secure channels
- not robust (against malicious \tilde{P}_i)
 - a) \tilde{P}_i encrypt $+n$
 - b) \tilde{P}_i refuse to send c_i
 - c) \mathbb{A} refuses to decrypt
- defenses exist for all these:
 - a) use zero-knowledge proofs
 - b) use public bulletin board for communication
 - c) distributed implementation of \mathbb{A} using a group of admins
- Helios implements most of this (heliosvoting.org)

4.6 Zero-Knowledge Proofs

- How to prove a statement is true without any more information
- How to prove knowledge of "password" without giving information about it?
- Many application in cryptographic protocols
- Goldwasser & Micali received Turing Award for it

4.6.1 Two kinds of proofs

- Proofs for statements
 - Given a Boolean formula Ψ in n variables, there exists an assignment s.t. $\Psi \equiv \text{TRUE}$
 - Given two graphs \mathbb{G}_0 and \mathbb{G}_1 , they are isomorphic
 - Given a graph \mathbb{G} , there exists a Hamiltonian circuit in \mathbb{G}
 - Given a graph \mathbb{G} , there is a 3-coloring of \mathbb{G}
- Proofs of knowledge
 - Given y (a public key DH \sim) "I know" x such that $y = g^x$
 - Given $h \in \{0, 1\}^k$, "I know" an x s.t. $\mathbb{H}(x) = h$ for some one-way function \mathbb{H}
 - Given N , "I know" P, Q s.t. $N = P \cdot Q$

4.7 Model (Proofs of Statement)

Requirements

1. **Completeness:** If statement S holds, prover \mathbb{P} correct and \mathbb{V} correct, then \mathbb{V} accepts.
2. **Soundness:** If S is FALSE, then (honest) \mathbb{V} will reject with at least a constant probability (no matter what the malicious \mathbb{P} does)
3. **Zero-Knowledge:** \mathbb{V} learns only that S holds (and not more).
(\mathbb{V} could also have simulated the whole protocol with itself)

4.8 ZKP for Graph Isomorphism

- \mathbb{P} and \mathbb{V} are given two graphs $G_0 = (V, E_0)$ and $G_1 = (V, E_1)$.
- \mathbb{P} knows an isomorphism between G_0 and G_1 , i.e. bijective function:

$$f : V \rightarrow V$$

$$\text{s.t. } \forall v, w \in V : (v, w) \in E_0 \Leftrightarrow (f(v), f(w)) \in E_1$$

Protocol