CS 4501-6501 Topics in Cryptography	January 26, 2015
Lecture 4	
Lecturer: Mohammad Mahmoody	*Scribe: Saba Eskandarian*

1 Overview

- Formal Defintion of Fully Homomorphic Encryption
- FHE vs CCA Security
- FHE: from private key to public key

2 Formal Definition of Fully Homomorphic Encryption

2.1 Definition from last lecture

Fully Homomorphic Encryption:

- $\text{KeyGen}(1^n) \to (ek, dk)$
- $\text{Dec}_{dk}(\text{Enc}_{ek}(m)) = m$, $m \in \mathcal{M}$
- Single message eval: for any "given function" f it holds that:
  $\text{Dec}_{dk}(\text{Eval}_{ek}(f, \text{Enc}_{ek}(m)) = f(m)$ (with probability $1 - negl(n)$).
- Semantic Security: for any polytime adversary ADV and any two messages m_1, m_2 it holds that $\Pr[\text{ADV}(ek, \text{Enc}_{ek}(m_0) = 1] - \Pr[\text{ADV}(ek, \text{Enc}_{ek}(m_1) = 1] \leq negl(n)$.

This definition is almost what we want, but there are a couple more properties which we must add before arriving at a complete definition of fully homomorphic encryption that truly satisfies all the properties we want.

2.2 Multi-message Eval

Multi-message eval: For any "given function" $f : \mathcal{M}^k \to \mathcal{M}$ and $c_1, ..., c_k$ where $c_i = \text{Enc}_{ek}(m_i)$, $\text{Dec}_{dk}(\text{Eval}_{ek}(f, c_1, ..., c_k)) = f(m_1, ..., m_k)$.

We will concern ourselves with the case of two-message evaluation.

Theorem 1. *Two-message eval implies single-message eval (or more generally, k message eval implies $k - 1$ eval).*

Proof. Given $f : \mathcal{M} \to \mathcal{M}$, convert f to $f' : \mathcal{M}^2 \to \mathcal{M}$ such that $f'(x, y) = f(x)$. □

Research Question: *Is the reverse of the above also true?*

2.3 Trivial FHE?

We can define a trivial FHE that satisfies our current definition by using $\text{Eval}(c, f) = (c, f)$ and pushing all the work to the decryption with $Dec(c_1, f_1, f_2, ..., f_k) = f_k(...f_1(\text{Dec}(c))...)$. Note that none of the applications of the FHE that we discussed actually works with this cheat (e.g. think of the delegation of computation).

Compactness. A correct definition of fully homomorphic encryption should not allow the above trick, so we need to add a compactness requirement: $|\text{Eval}(...)| \leq poly(n)$. This polynomial is independent of k. We call this condition the "compactness" condition, and is the standard way of defining FHE.

Statistical variant. There is also a harder to achieve, statistical variant of this restriction: $(pk, \text{Eval}(f, \text{Enc}(x))) \equiv_{negl(n)} (pk, \text{Enc}(f(x)))$.

2.4 Final Definition

The final definition of homomorphic encryption is the same as the original one from last last lecture shown above but with the addition of multi-message eval and the compactness requirement.

3 No CCA Security

Recall that in CPA or semantic security, the adversary who knows pk is given an encryption y and has to distinguish between whether y Enc(1) and Enc(0). The CCA security game is the same except that before attempting to distinguish between the encryptions, the adversary can ask for anything other than y to be decrypted. Fully homomorphic encryption is not CCA secure because the adversary could ask for the decryption of $y' = \text{Enc}(f(x))$ by using Eval(...) to modify x without decrypting it. The adversary can then take $f^{-1}(\text{Dec}(y'))$ to recover x. f in this case could be as simple as $x \oplus 1$.

4 From Private Key to Public Key Encryption

Given a private key encryption scheme (Enc, Dec, Eval), we want to get a public-key scheme $(\overline{Enc}, \overline{Dec}, \overline{Eval}, \overline{KeyGen})$. This, in general is very hard, and some "impossibility" results against this are known [IR89]. However, when the private key scheme is already homomorphic, it is indeed possible to elevate it to the public-key setting.

Theorem 2 ([RV10])**.** *Assuming a given CPA-secure private key encryption scheme, the following is a secure public key encryption scheme. Use public key (Enc(b_1),Enc(b_2),...,Enc(b_k), $r = (b_1, ..., b_n)$) for random $b_1, \ldots, b_k$. To encrypt bit b, choose $s \in \{0,1\}^k$ at random so that $\langle s, r\rangle = \oplus_{s_i=1} b_i$. Then use Eval to get the encryption of the XOR of $\{b_i\}_{s_i=1}$.*

Proof. We will not prove the theorem here, but the "left-over hash lemma" that we will see in the next sessions is at the heart of the proof.

A special case of Left-over hash lemma: if we are given $r = (b_1, ..., b_k)$ chosen at random and given only m bits of information about $s = (s_1, ..., s_k)$ which is also random, the distribution of $\langle s, r\rangle$, the inner product of s and r, remains a random bit. □

References

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 44–61. ACM Press, 1989.

[RV10] Guy N. Rothblum and Salil P. Vadhan. Are PCPs inherent in efficient arguments? *Computational Complexity*, 19(2):265–304, 2010.