CS 4501-6501 Topics in Cryptography March 2, 2015

Lecture 14

Lecturer: Mohammad Mahmoody *Scribe: Collin Berman*

1 Introduction

Recall with homomorphic encryption, we had encrypted data and wanted to perform some computation over the encrypted data. The result of this computation would result in a piece of ciphertext. We were able to find a construction that would let us compute any polynomial-time function, which we called *fully* homomorphic encryption. Now we are interested in a related problem, *functional encryption*, where we are no longer interested in keeping things encrypted after the computation. Now we receive the plaintext result of functional computation. That is, given an encryption $\text{Enc}(x)$ of some message x, we wish to efficiently compute $f(x)$ for some function $f(\cdot)$. Because we are given an encryption of the message x, we feel we should need some kind of secret key to be able to extract information in this manner.

2 Motivating Scenario

Consider a scenario where we have a number of stored ciphertexts, and we want some form of access control for accessing the data. For example, we might have a firm, and access control is based off of job position. The CEO should of course be able to decrypt everything, but other employees should only be able to decrypt parts that relate to them. Another option for access control would be to have a boolean formula determine whether or not a specific person should be able to access a piece of data.

With both of these methods of access control we see we will need different keys for each level of access. We can even imagine users moving around in the hierarchy, so that they may be given new keys over time. Even in this case we want to have to encrypt the data only once.

3 Definition

3.1 Schheme

We consider a public-key variant of this primitive. We will have a master public key MPK for encryption, and a master secret key MSK for generating keys for other users with lower access level. These keys will be generated by a key-generation phase. MPK can then be used to get $\text{Enc}(x)$ from message x, and MSK gives us K_f for a function $f(\cdot)$. After giving K_f to a user, they will be able to compute $f(x)$ from $\text{Enc}(x)$. This final step requires a source of randomness and is called decryption.

This definition is sufficient for the latter boolean-based method of access control we considered, but we will also introduce new notation for it. This method of allowing decryption by identity is called *attribute-based* or *predicate* encryption [BSW11]. Now when we encrypt a message, we also need some index that will determine whether or not a given access level is sufficient to decrypt.

The decryption keys are now based on predicates (boolean functions) $P(\cdot, \cdot)$ that will take an index and an ID. The holder of a key for a certain ID, K_{ID} will be able to learn x from Enc(index, x) if $P(\text{index}, ID) = 1$. This formulation will be called *ciphertext-policy* attribute-based encryption.

We will later be interested in an alternative formulation known as *key-policy* attribute-based encryption where the predicate $p(\cdot)$ is stored in the user's key. Now the key K_p will give us all of x, but only if $p(x) = 1$.

Finally we define *identity-based* encryption [Sha85]. This is ciphertext-policy attribte-based encryption, where the index of a message corresponds exactly to an ID. Thus if we have the encryption Enc(ID^*, x), we will be able to use our key K_ID to decrypt and recover x only if $ID^* = ID$.

3.2 Security

First let's recall the security of public-key encryption, then extend it to functional encryption. With public-key encryption, if we encrypt any two messages, they should be indistinguishable without the secret key. Now we can take this idea and apply it to identity-based encryption. We will consider an adversary who has keys for any IDs he wants. This adversary should still not be able to distinguish between Enc(ID^*, x) and Enc(ID^*, x') for a new ID ID^* for which the adversary does not have a key, and $x \neq x'$.

References

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *TCC*, pages 253–273, 2011.

[Sha85] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.