CS 4501-6501 Topics in Cryptography January 21, 2015

Lecture 3

Lecturer: Mohammad Mahmoody *Scribe: Will Hawkins, Sakura Lim*

1 Homomorphism from Exponentiation

1.1 Multiplicative Homomorphism from RSA

Recall that it is possible to create a multiplicative homomorphic scheme using RSA. In RSA, the encryption function is based on exponentiation in a group. Given an encryption key e, $Enc(x) = f(x) = x^e$. Therefore,

$$\begin{aligned} Enc(x_0) * Enc(x_1) &= f(x_0) * f(x_1) \\ &= x_0^e * x_1^e \\ &= (x_0 * x_1)^e \\ &= f(x_0 * x_1) \\ &= Enc(x_0 * x_1) \end{aligned}$$

1.2 Additive Homomorphism from Exponentiation

In a similar manner, define $f(x) = g^x$ for some g. In this case,

$$\begin{aligned} f(x+y) &= g^{x+y} \\ &= g^x * g^y \\ &= f(x) + f(y) \end{aligned}$$

This means that $f(x+y)$ can be evaluated homomorphically. Unfortunately, $f(x)$ cannot be used as an encryption function because there is no known way to decrypt it privately and efficiently (using a trapdoor)! The challenge is to find an encryption technique that is additively homomorphic.

2 Diffie-Hellman Breakthrough

Diffie-Hellman[1] used similar ideas to homomorphism of exponentiation and published the first method for key exchange over an insecure channel [DH76]. Their work pushed modern cryptography beyond private key cryptographic regimes and forced researchers to consider public key encryption. It is possible to do key agreement using public key encryption. However, the DH method is a “direct” key agreement protocol.

2.1 DH Protocol

Alice and Bob want to generate a shared, private key by exchanging messages over an insecure channel. Eve is listening on the insecure channel and is able to read any messages sent from Alice to Bob and vice versa. The protocol is secure if Alice and Bob generate a key that is “hidden” from Eve. In other words, Eve must not be able to use the intercepted messages to reconstruct the secret key.

Before starting the key negotiation, Alice and Bob agree on a group. This agreement is conducted publicly. The group must have certain properties for the protocol to be secure. See below.

[1] or Merkle, depending on who you talk to.

After agreeing on a group, g, Alice and Bob generate random numbers r_a and r_b, respectively. Alice sends g^{r_a} to Bob; Bob sends g^{r_b} to Alice. From these two messages, Alice and Bob construct the private key:

$$g^{r_a * r_b} = (g^{r_a})^{r_b} = (g^{r_b})^{r_a}.$$

As mentioned previously, g must have special properties. Define group g as $(\mathbb{Z}_p^*, \odot).\mathbb{Z}_p^*$ is $\{1...p-1\}$ where p is prime. $\odot$ is multiplication modulo p: $x \odot y = x * y \ mod \ p$.

A group G is defined as follows:

1. closure: $x \odot y \in G$ for $x, y \in G$
2. associativity: $x \odot (y \odot z) = (x \odot y) \odot z$
3. identity: there is some $1 \in Z$ such that $1 \odot x = x \odot 1 = x$ for all $xinG$
4. invertibility: for any $x \in G$ there is some $y \in G$ such that $x \odot y = y \odot x = 1$.

Proofs of (1-3) are straightforward.

To show that g is invertible, use

Lemma 2.1.1. *$\forall a, b \ \exists \ \alpha, \beta$ such that $\alpha * a + \beta * b = gcd(a, b)$.*

Therefore, $\forall x, p \ \exists \ x', p'$ such that $x * x' + p * p' = 1$ which implies that $x \odot x' = 1$.

2.2 Hardness Assumption of DH Protocol

To demonstrate the security of DH, we want an assumption that implies DH is secure. In other words, if it is possible to break the security of the protocol, it is also possible to break the assumption. One might first think that this assumption could simply be the hardness of computing discrete logarithm in Z_p^*, but that is not the right one, since it only is the case that: "If the discrete log problem is easy,[2] then DH is insecure." which is a reduction in the wrong direction! In the latter case, breaking the discrete log is *one possible way* to one to invalidate DH but it does not prove that it is the only way.

The assumption whose truth guarantees the security of DH protocol is the Decisional DH (DDH). Eve is given either:

$$(g^x, g^y, g^{xy}) \tag{1}$$

or

$$(g^x, g^y, g^r) \tag{2}$$

where $r \Leftarrow \$$ and g is a generator for $(\mathbb{Z}_p^*, \odot)$. Eve's goal is to distinguish between (1) and (2) with probability more than $1/2 + \epsilon$ where $\epsilon > negl(n)$ where n is the security parameter that determines the size of the group (think of $n \approx \log(p)$). If Eve can do so, then she can break DDH. And if she can break DDH, then she can break DH.

2.3 ElGamal

2.3.1 PKE from DDH

ElGamal is a PKE scheme based on DDH. It is a straightforward application of DDH. Bob wants to send an encrypted message m to Alice in the presence of Eve, an eavesdropper who can intercept messages between them.

The protocol involves two messages. In what follows Alice is the person who generates the public key, Bob is the one who encrypts and again Alice is the one who decrypts. The intuition behind using DH is that Alice sends a public key to Bob and he responds with the rest of the key and the encrypted message.

[2] Here "easy" means computable in polynomial time.

1. Alice determines a group $G = (S, \odot)$ and a generator of that group g.
2. Alice generates x, a hidden secret using some local randomness.
3. Alice transmits G, g and g^x to Bob — the triple is the public key.
4. Bob generates y using some local randomness.
5. Bob encrypts m as $c = Enc(m) = g^{xy} \odot m$.
6. Bob sends (g^y, c) to Alice.
7. Alice decrypts c as $Dec(g^y, c) = g^{xy} \odot m \odot (g^{xy})^{-1}$.

The inverse of g^{xy}, $(g^{xy})^{-1}$, exists by definition of a group.

2.3.2 Multiplicative Homomorphism for ElGamal

Given $Enc(m_1)$ and $Enc(m_2)$, compute $Enc(m_1 * m_2)$.

$$\begin{aligned}(g^{xy_1}, Enc(m_1)) &= (g^{xy_1}, g^{xy_1} \odot m_1)\\ (g^{xy_2}, Enc(m_2)) &= (g^{xy_2}, g^{xy_2} \odot m_2)\end{aligned}$$

Therefore,

$$\begin{aligned}Enc(m_1) * Enc(m_2) &= (g^{xy_1} \odot m_1) * (g^{xy_2} \odot m_2)\\ &= g^{x(y_1+y_2)} * m_1 * m_2\\ &= g^{xr'} * m'\\ &= Enc(m')\end{aligned}$$

where $r' = y_1 + y_2$ and $m' = m_1 * m_2$.

ElGamal's weakly multiplicatively homomorphic property is an improvement over the same property in textbook RSA. Textbook RSA is not secure as a cryptography protocol and ElGamal is. In other words, ElGamal serves two functions in one scheme.

References

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.