6.876/18.426: Advanced Cryptography April 7th 2003

Lecture 15: A Practical CCA-2 PK Cryptosystem

Scribed by: Javed Samuel
Guest Lecturer: Matthew Lepinsiki

1 Introduction

During this lecture we looked at a practical public key cryptosystem which was provably secure against an adaptive chosen ciphertext attack. We will first define the decisional Diffie-Hellman (DDH) problem which we assume to be hard. Then we present a modified version of El-Gamel assuming that DDH which is hard which is secure against a passive adversary. We then modify this protocol and prove that it is secure against a lunch-time attack. Finally we modify the protocol yet again and prove that it is adaptive chosen ciphertext secure (CCA-2).

2 Decisional Diffie-Hellman

2.1 Definition

Let us consider the following two distributions where q is a large prime and (g_1, g_2, r, r_1, r_2) are random elements in Z_q^*.

Type 1: (g_1, g_2, g_1^r, g_2^r)
Type 2: $(g_1, g_2, g_1^{r_1}, g_2^{r_2})$

We define an algorithm that can solve the decisional Diffie-Hellman problem as a statistical test that can effectively distinguish between these two distributions. The Diffie-Hellman decision problem[1] is considered hard if there is no such polynomial-time statistical test.

2.2 Equivalence

If we have such an algorithm then we can easily distinguish between a Diffie-Hellman triple (g^x, g^y, g^xy) and a non-Diffie-Hellman triple (g^x, g^y, g^z). The DDH problem is also equivalent to the worst-case decision problem: given g^x, g^y, g^z), decide with negligible error probability if z = xy mod p. This equivalence is a result of the random self-reducibility property.

[1]The DDH is related to the Diffie-Hellman problem (given g, g^x and g^y compute g^{xy}) and the discrete logarithm problem (given g and g^x, compute x). There are obvious polynomial time reductions from the DDH problem to the Diffie-Hellman problem, and from the Diffie-Hellman problem to the discrete logarithm problem, but reductions in the reverse direction are not known.

3 Modified El Gamel

3.1 Defintion

This public key cryptosystem[2] is secure against passive attacks but is vulnerable to lunchtime attacks (CCA-1) and adaptive chosen ciphertext attacks (CCA-2).

We have a public key consisting of a prime q, two randomly selected generators g_1, g_2 and h = $g_1^{z_1} g_2^{z_2}$.

The secret key consists of two randomly selected values in Z_q^* namely z_1 and z_2.

We encrypt a message m by computing

$$E(m, r) = g_1^r, g_2^r, h^r m$$

Let u_1 =g_1^r, u_2 =g_2^r and $e = h^r m$.

Decrypting is done as follows

$$D(u_1, u_2, e) = \frac{e}{u_1^{z_1} u_2^{z_2}} = \frac{h^r m}{g_1^{r z_1} g_2^{r z_2}} = m$$

We note that h^r still looks random even after seeing g_1^r, g_2^r.

3.2 Proof Of Security

As always we will prove this by contradiction. We assume that we have an adversary that can break El-Gamel (ADV) which we can use as a sub-routine and we will show that we can construct an adversary that can break the Decision Diffie-Hellman assumption.

Let us assume that we are given (g_1, g_2, u_1, u_2). We will use g_1, g_2 as the public key and we will have $h = g_1^{z_1} g_2^{z_2}$. The secret key is as before two randomly chosen values from Z_q^* namely z_1 and z_2.

Our algorithm sends a message to ADV and then ADV responds by sending two messages m_0 and m_1. One of the messages is randomly chosen by selecting $b \in \{0, 1\}$ and send u_1, u_2 (as obtained from above) and $e = u_1^{z_1} u_2^{z_2} m_b$. ADV then sends us g.

We then let $g = b$ which we claim would allow us to break the DDH assumption. This fact relies on two claims which we prove. We are trying to determine whether

$$(g_1, g_2, u_1, u_2) = (g_1, g_2, g_1^{r_1}, g_2^{r_2})$$

3.2.1 Claim 1

If Type 1 ($r_1 = r_2$) then $g = b$ with probability ¿ $\frac{1}{2} + \frac{1}{poly(k)}$.

This claim is very easy to see since in this case ADV is answering the exact question which we want the answer hence we have the same probability of success as ADV does.

[2]The El-Gamel cryptosystem consists of a public key with a prime p, generator g, g^x. The secret key is x. You send a message m by choosing a random number y in Z_p^* and then sending over $g^y, g^{xy} * m$. Decrypting is trivial and we simply use the secret key x to calculate g^{xy} and then obtain the message m.

3.2.2 Claim 2

If Type 2 ($r_1 \neq r_2$) then b is completely independent of ADV's view (PK, u_1, u_2, e) and that the probability that $g = b$ is equal to $\frac{1}{2} + \frac{1}{2^{-k}}$. In Type 2 we note that this is not a valid ciphertext since the encryption is not valid. Valid encryptions are always in the form $e = h^r m_b$

3.2.3 Proof of Claim 2

Given the public key we can see that $log(h) = z_1 + wz_2$. Also it is equally likely that $r_1 z_1 + w r_2 z_2 = \frac{e}{m_0}$ or that $r_1 z_1 + w r_2 z_2 = \frac{e}{m_1}$. We can think of the public key as defining some line in the z_1, z_2 space. Each of the messages is also constrained to a particular line in the $z_1.z_2$ space as indicated above. However that is still not enough information for the adversary, even with infinite time to be able to decide whether m_0 or m_1 was sent.

3.3 Why is it not CCA-2 secure?

Assuming we are given u_1, u_2, e which is equal to $g_1^r, g_2^r, h^r m$. The adversary can ask for the decryption of $u_1 g_1, u_2 g_2, h * e{=}g_1^{r+1}, g_2^{r+1}, h^{r+1} * m$ which would be equal to the message m.

3.4 Why is it not CCA-1 secure?

Using a lunchtime attack we can compute information about the public key that we did not know before. We can ask for the decryption of

$$(g_1^{r_1}, g_2^{r_2=r_1-1}, h_1^r) = \frac{h^{r_1}}{u_1^{z_1} u_2^{z_2}} = \frac{g_1^{r_1 z_1} g_2^{r_1 z_2}}{g_1^{r_1 z_1} g_2^{r_2 z_2}} = g_2^{z_2}$$

We gain some more information about the public key which we could use to break the security. We do not know how we can use a PPT algorithm to break security with a CCA-1 attack but we cannot prove that this protocol is CCA-1 secure.

4 Cramer Shoup CCA-1 Secure Protocol

4.1 Introduction

This Cramer-Shoup protocol is an improvement of the modified El Gamel protocol discussed earlier. This protocol is secure against a passive adversary and against a lunch-time attack. The basic idea involves having another pair of elements from Z_q^*,$x_1 and x_2$ which provides security against a lunch time attack.

4.2 Definition

We have a public key consisting of a prime q, two randomly selected generators g_1, g_2 and h = $g_1^{z_1} g_2^{z_2}$ and c = $g_1^{x_1} g_2^{x_2}$.

The secret key consists of four randomly selected values in Z_q^* namely z_1, z_2 and x_1, x_2.

We encrypt a message m by computing

$$E(m,r) = g_1^r, g_2^r, h^r m, c^r = u_1^{x_1} u_2^{x_2}$$

Let $u_1 = g_1^r$, $u_2 = g_2^r$, $e = h^r m$ and $v = c^r = u_1^{x_1} u_2^{x_2}$

Decrypting is done as follows using (u_1, u_2, e, v). First we verify that v is indeed equal to $c^r = u_1^{x_1} u_2^{x_2}$. Then we compute m which is equal to $\frac{e}{u_1^{z_1} u_2^{z_2}}$.

4.3 Proof of CCA-1 Security

Security against a passive adversary is trivial and follows immediately from the proof described earlier in the modified El-Gamel protocol. As before we will prove that this protocol is CCA-1 secure by contradiction. We assume that we have an adversary ADV which can break the security of this protocol and then we will show that this adversary can be used as a subroutine to solve the DDH problem with non-negligible probability.

Let us assume that we are given (g_1, g_2, u_1, u_2). We will use g_1, g_2 as the public key and we will have $h = g_1^{z_1} g_2^{z_2}$ and $c = g_1^{x_1} g_2^{x_2}$. The secret key consists of four randomly chosen values from Z_q^* namely z_1 ,z_2, x_1 and x_2.

Our algorithm sends a message to ADV and then ADV responds by sending a ciphertext of any message. This decryption of the ciphertext is sent back to ADV. After a polynomial number of queries ADV sends two messages m_0 and m_1. One of the messages is randomly chosen by selecting $b \in \{0,1\}$ and then u_1, u_2 (as obtained from above) ,$e = u_1^{z_1} u_2^{z_2} m_b$ and $u_1^{x_1} u_2^{x_2}$ is sent to ADV. ADV then sends us g.

We then let $g = b$ which we claim would allow us to break the DDH assumption. This fact relies on two claims which we prove. We are trying to determine whether

$$(g_1, g_2, u_1, u_2) = (g_1, g_2, g_1^{r_1}, g_2^{r_2})$$

Claims 1 and 2 from above hold here. Assuming that no invalid ciphertexts are decrypted then the above proof shows that that the protocol is CCA-1 secure.

4.3.1 Claim 3

The adversary cannot ask for the decryption of invalid ciphertext except with negligible probability. This is important since if the adversary is able to ask invalid questions with greater than non-negligible probability, then he can put extra constraints on the secret key which may allow him to get additional information.

4.3.2 Proof of Claim 3

A ciphertext $g_1^{r_1'}, g_2^{r_2'}, e', v'$ is rejected unless it lies on the line satisfied by the following validity check equation.

$$log(v') = r_1' x_1 + w r_2' x_2$$

where $w = log_{g_1} g_2$

The constraint from the public key is

$$log(C) = x_1 + w x_2$$

.

Therefore we note that for each invalid ciphertext where ($r_1^{'} \neq r_2^{'}$) there is only one secret key (x_1, x_2) which passes the validity check. We recall from linear algebra that a line can intersect a linearly independent hyperplane exactly once. The adversary only passes the validity check if his invalid ciphertext line intersects the public key constraint. This event has negligible probability.

4.4 Why is it not CCA-2 secure?

An equivalent attack as described previous for the El-Gamel scheme will suffice. Assuming we are given u_1, u_2, e which is equal to $g_1^r, g_2^r, h^r m$. The adversary can ask for the decryption of $u_1 g_1, u_2 g_2, h * e{=}g_1^{r+1}, g_2^{r+1}, h^{r+1} * m$ which would be equal to the message m. Another attack suggested in class was to simply the third value in our challenge ciphertext by some constant and then request that this ciphertext is decrypted. This modification will not affect the validity check and then we can divide by the constant to determine the original message.

5 Cramer Shoup CCA-2 Secure Protocol

5.1 Introduction

This Cramer-Shoup CCA-2 Secure protocol is a modification of the CCA-1 Secure Protocol discussed earlier. This protocol is secure against an adaptive chosen cipher text adversary The basic idea involves having another pair of elements from Z_q^*,$y_1 and y_2$ and a collision resistant hash function[3] H.

5.2 Definition

We have a public key consisting of a prime q, two randomly selected generators g_1, g_2 and h = $g_1^{z_1} g_2^{z_2}$, c = $g_1^{x_1} g_2^{x_2}$, d = $g_1^{y_1} g_2^{y_2}$ and a public collision resistant hash function H.

The secret key consists of six randomly selected values in Z_q^* namely z_1, z_2, x_1, x_2 , y_1 and y_2.

We encrypt a message m by computing

$$E(m, r) = g_1^r, g_2^r, h^r m, c^r d^{\alpha r}$$

Let $u_1 = g_1^r$, $u_2 = g_2^r$, $e = h^r m$ and $v = c^r d^{\alpha r}$

Decrypting is done as follows using (u_1, u_2, e, v). First we compute $\alpha = H(u_1, u_2, e)$ and then we test the validity condition.

$$v = u_1^{x_1 + \alpha y_1} u_2^{x_2 + \alpha y_2}$$

If this condition does not hold, the decryption algorithm outputs "reject"; otherwise we compute m which is equal to $\frac{e}{u_1^{z_1} u_2^{z_2}}$.

[3]We can actually use the weaker assumption of universal one-way hash functions

5.3 Proof of CCA-2 Security

This protocol is CCA-1 secure and this can be seen by the proof described earlier. As always we prove that this is CCA-2 secure by contradiction. We have an adversary ADV who can break this cryptosystem, and that the hash function is collision resistant and we will use this adversary to construct a statistical test for the DDH problem.

The validity check constraint is (let $w = log_{g_1} g_2$)

$$log(v`) = r`x_1 + wr`_2 x_2 + \alpha r`_2 + \alpha w r`_2 y_2$$

The two public key constraints are

$$log(c) = x_1 + wx_2$$

$$log(d) = y_1 + wy_2$$

The public key constraints define a plane and he validity check constraint also defines a linearly independent plane. Also we recall from linear algebra that these two linearly independent planes will intersect at a line.

When constructing invalid cipher text the adversary will either use new α or old α. We want to show that the adversary has negligible probability of constructing an invalid ciphertext.

5.3.1 Case 1

The adversary uses an α from a previous challenge in his new query. We can easily see that this cannot happen since we assumed that the hash function was collision resistant.

5.3.2 Case 2

The adversary uses a new α. The decryption algorithm will reject this invalid ciphertext unless it lies on the hyperplane defined by our validity check constraint. We can use linear algebra in 4d geometry to see that the adversary is looking for a particular line that intersects the validity check constraint. The probability of this event is negligible since each challenge ciphertext plane only intersects at one point. Basically the probability that the adversary`s invalid ciphertext contains the secret key is negligible given that it is linearly independent from the public key constraints.

Since we have argued that the adversary cannot create invalid ciphertext with more than negligible probability it follows from the previous discussion that this protocol is adaptive chosen ciphertext secure.

6 Questions

1. Can you use a similar approach to simplify the Sahai scheme?
2. Is there any other scheme that achieves adaptive chosen ciphertext security?

References

[CS98] R. Cramer and V. Shoup . A Pratical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack

[Sah98] A. Sahai Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security