CRYPTOGRAPHY: ANSWERS TO HOMEWORK 1

VISHVAS VASUKI

1

Remark 1.0.1. Collision resistant hash function H(x): Hard for an attacker to find $m' \neq m : H(m) = H(m')$.

Theorem 1.0.2. *$H : Z_p \times Z_p \rightarrow G$. G has prime order p. u, v are chosen by a trusted setup authority. $H(x_1, x_2) = u^{x_1} v^{x_2}$.*

Then, if there exists an attacker A that can find a collision, then the attacker can be used to break the discrete log problem in the same group.

Proof. Every element, other than the identity, of a prime order group is a generator. So, $v = u^a$ for some $a \in Z_p$. So, $H(x_1, x_2) = u^{x_1 + a x_2}$.

We now construct an algorithm L to break the discrete log problem in G.

L is given $n = u^b$ challenged to find b. L responds by doing the following: L uses A to find a collision for u^b and obtains $(x_1, x_2), (y_1, y_2)$. L then solves the linear equation $x_1 + a x_2 = y_1 + a y_2$ to find a. L then computes $y_1 + a y_2 = b$ and returns b.

If A works with a non negligible probability p, L succeeds in finding b with the same probability p. □

2

Fact 2.0.3. *If DDH assumption is true, ElGamal encryption is semantically secure under CPA.*

Theorem 2.0.4. *An attacker on DDH can break ElGamal.*

Proof. Let A be the ElGamal challenger, B the ElGamal attacker, C the DDH attacker.

A initially announces the PK: g, g^y. B randomly picks and sends plaintexts m_0, m_1 to A.

A picks random $r \in G$ and based on a fair coin flip b, sends $g^r, m_b g^{yr}$ to B.

B finds $T = m_0^{-1} m_b g^{yr}$, and sends T, g^r, g^y to C. With non negligible advantage over random guessing p, C distinguishes g^{yr} from $m_0^{-1} m_1 g^{yr}$.

Using this response, with non negligible advantage over random guessing p, B determines whether $m_1 g^{yr}$ or $m_0 g^{yr}$ was sent by A and responds correctly with the value of b used by A.

Hence the ElGamal cryptosystem is now vulnerable to CPA. □

Remark 2.0.5. This theorem implies that, in groups where there exist efficiently computable bilinear maps, DDH assumption is false, and consequently ElGamal is vulnerable to CPA.

3

Remark 3.0.6. Decisional linear problem: Given group G of prime order p and elements g, u, v, g^a, u^b, distinguish $T = v^{a+b}$ from a random number. T is chosen to be v^{a+b} based on a random bit s.

Decisional linear assumption (DLA): No poly-time attacker can guess s with non negligible advantage over random guessing.

Definition 3.0.7. The public key encryption system X is defined by the algorithms described below.

Setup(l): $SK = (x, y)$.$PK = (g, u = g^x, v = g^y)$.

Encrypt(m): Select random a, b. $c_1 = g^a, c_2 = u^b, c_3 = m.v^{a+b}$. Cyphertext $c = (c_1, c_2, c_3)$.

Decrypt(c): Find $v^a = c_1^y, v^b = c_2^{y/x}, v^{a+b} = v^a v^b$. Then find $m = c_3 v^{-(a+b)}$.

Theorem 3.0.8. *If DLA is correct, X is secure against CPA.*

Proof. Assume that DLA is correct. Assume that there exists an CPA attacker for X, called C. Consider the algorithm B described below.

Let A be DLA challenger and B be DLA attacker/ X challenger.

A sends g, u, v, g^a, u^b, T to B, where T is chosen to be v^{a+b} based on whether random bit s = 1.

B sends $PK = (g, u, v)$ to C.

C sends B plain text messages m_0 and m_1.

B picks random bit r and sends $m_r T$ to C.

If T is a valid blinding factor of the form v^{a+b}, C correctly guesses r with non negligible advantage over random guessing, p.

If C's guess is correct, B replies to A: s = 1. Otherwise, B tells A: s = 0.

Now, we analyze the probability with which B correctly solves the Decisional linear problem.

When s = 0, C's guess is no better than random guessing in the worst case. So, B's reply to A will be correct with probability 1/2.

When s = 1, C's guess is correct with probability 1/2 + p. So, B is correct with probability 1/2 + p.

So, B's overall probability of success is 1/2 + p/2, which is a non negligible advantage over random guessing. This violates the DLA assumption. So, the assumption that C exists was incorrect. □

4

Acknowledgement. The solution to this problem emerged during a conversation with Rashid Kaleem.

Definition 4.0.9. CPA2 security game is defined as follows:

The game is similar to CPA security game except the attacker submits 1 message m^*. The challenger picks a random bit g and sends m^* if g = 0 and some random m otherwise.

Remark 4.0.10. An attacker A successful in the CPA2 security game can be modified to be successful in the CPA security game: When it has to choose a pair of plaintext messages, the modified algorithm A' simply chooses $m_0 = m^*, m_1 = m$ where m is random, and proceeds with rest of the game as A would. So, CPA security $\implies$ CPA2 security.

Theorem 4.0.11. *CPA2 security $\implies$ CPA security. In other words, an attacker which can win the CPA security game can be used to win the CPA2 security game.*

Proof. The following construction illustrates the creation of a CPA2 attacker from a CPA attacker.

Let A be CPA2 challenger, B be the CPA2 attacker/ CPA challenger, C the CPA attacker.

A tells B the public key PK.

B sends PK to C.

C sends m_0, m_1 to B.

B picks a random bit b and sends m_b to A.

A picks a random bit a and sends B the cyphertext $c = e(m_b)$ if a = 0, and c = e(m) of random message m otherwise.

B sends c to C.

C guesses the bit b and responds to B. C is correct with non negligible advantage over random guessing p, in case $c = e(m_b)$.

If C's guess is correct, B tells A that a = 0. Otherwise, it says a = 1.

Now, we analyze the success rate of B.

A sends c = e(m) with probability 1/2. In this case, C has no advantage over random guessing, and B is correct 1/2 the time.

With probability 1/2, A sends $c = e(m_b)$. In this case, B is correct with probability 1/2 + p.

So, overall, B has an advantage p/2 over random guessing. □

Remark 4.0.12. So, CPA2 security is equivalent to CPA security.

5

Theorem 5.0.13. *Selective IBE security does not imply full IBE security.*

Proof. Assume that there exists an IBE system X secure under the standard definition of security with algorithms: Setup, KeyGen, Encrypt, Decrypt. Identities are in the space $\{0,1\}^l$.

Now, we create a new system Y which is selectively secure, but not fully secure.

In Y, the Encrypt' and Decrypt' algorithms remain the same. Setup' accepts an id, called tid, as an additional parameter. The Setup' calls the Setup algorithm, but also identifies a pair of special id's: tid and oid. These are made part of the public parameters.

The keygen' algorithm is altered: keygen'(id) = keygen(id) if id = tid, and keygen(oid) otherwise.

Y is selectively secure, but not fully secure: as shown in the lemmas below.

□

Lemma 5.0.14. *If X is secure, Y is selectively secure.*

Proof. Let C be the Y attacker, B the Y challenger/ X attacker and A the X challenger. We show that if C exists, then B exists.

We illustrate the construction of an X attacker from a Y attacker with the following game:

The interactions between A and B are as expected from the definition of full IBE security. In general, B acts as a relay between A and C with the following changes:

1. When it is B's turn to declare the target, it simply declares tid, the id declared by C to be the target.

2. When B is to provide C with the public parameters, PP, it appends tid and some arbitrary oid. B finds out keygen(oid) from A before this.

3. For all keygen queries by C, B responds using the specified tid and oid in the definition of keygen' algorithm.

□

Lemma 5.0.15. *Y is not fully secure.*

Proof. We show this by constructing an attacker B. Let A be the Y challenger.

A specifies two id's: tid and oid in the PP.

B finds out keygen'(oid) from A. It then chooses to attack some other id, $soid \notin \{tid, oid\}$. keygen'(soid) = keygen'(oid): So B knows the secret key of soid.

So, Y is vulnerable to CPA.

□