CS395T Advanced Cryptography 2/17/2009

Lecture 9: CCA security

Instructor: Brent Waters *Scribe: Vishvas Vasuki*

1 Semantic security against CCA

1.1 Motivation

Chosen cyphertext attacks (CCA) are a realistic possibility. Consider the following scenarios.

Suppose that A and B are exchanging embarassing Valentines day messages using a public key cryptosystem. C can intercept a cyphertext c from A to B, mangle it to get cyphertext $c' = Enc(m', r)$ and send it to B. B, mystified by the message m' he got from C, contacts him with m' and asks him what is going on. In such ways, C can concievably choose cyphertexts and get them decrypted.

If the cryptosystem being used by A and B is vulnerable to such attacks, it is possible that, C will be able to deduce the content of the messages being sent by A to B.

1.2 The formal notion of security from CCA

The notion of semantic security of a public key cryptosystem against CCA can be specified by a game similar to the CPA security definition. The new security game is identical to the CPA game (described in previous scribe notes), except that the attacker can now make a polynomial number of decryption requests in two phases. The first phase occurs before the challenger challenges the attacker by supplying a cyphertext c; and the second phase occurs after this event. In the first phase, the attacker may request the decryption of any abritrary cyphertext; but in the second phase, he may not ask for the decryption of the challenge cyphertext c.

It is possible that the cyphertext the attacker asks to decrypt is not in the space of valid cyphertexts, in which case, the challenger responds with the message 'invalid', denoted by $\perp$.

2 Vulnerability of ElGamal cryptosystem under CCA

Consider the ElGamal cryptosystem, which was discussed in previous scribe notes. Note that the challenge cyphertext is of the form $Enc(m, r) = (c_1, c_2)$, where $c_1 = g^r,\ c_2 = g^{ry}m$, where r and y are unknown to the attacker. Given this, an attacker can pick a random number r' and generate the cyphertext $c'_1 = g^{r+r'},\ c'_2 = g^{(r+r')y}m$, as the values g and g^y are known. The attacker can then query for the decryption of this modified cyphertext and receive the message m as answer. Thus, the ElGamal cryptosystem is vulnerable to CCA.

A stronger attack may be mounted by picking a random value h and querying for the decryption of cyphertext $c'_1 = g^{r+r'},\ c'_2 = g^{(r+r')y}mh$. Upon receiving a reply mh, the

attacker can then distinguish m_0h from m_1h. This attacker is stronger as it works even against challengers who might refuse to send the challenge plaintext m to the attacker.

3 Making a CCA secure scheme from a secure IBE scheme

We now show a generic way to build a CCA secure public key cryptography scheme from a secure IBE scheme.

Consider an IBE system with public parameters PP and a master secret key MSK.

3.1 A first attempt

Consider the scheme defined by the algorithms below:

- Setup(l): PK = PP. SK = MSK.
- Enc(m): Choose a random id, find $C_1 = Enc_{IBE}(PP, ID, m)$, then compose the cyphertext $CT = C_1, ID$.
- Dec(CT): Decompose CT into C_1 and ID, then get $Keygen(MSK, ID) = SK_{id}$; then do $Dec_{IBE}(C_1, SK_{id}) = m$ to get the plaintext.

Now we attempt to show that this algorithm is CCA secure by showing that a successful CCA attacker can be used to construct an attacker for the supposedly secure IBE scheme. Let A be the IBE challenger, B the IBE attacker/ CCA challenger, and C the CCA attacker.

Initially, the IBE system in A is setup. A announces PP to B. B communicates the public key PK = PP to C. C can make decryption requests in two phases. Whenever C makes a decryption request, the Dec algorithm described above is used to answer it. At some point B chooses ID^* from the ID's which were not involved in a decryption request in the previous step. B tells A that it intends to attack ID^*.

When C chooses the plaintexts to be encrypted, B passes them on to A. Then, A replies by sending the cyphertext C_1 of a randomly chosen message using the SK_{ID^*}. B passes it on to C after appending ID to C_1.

Now consider the second phase of C's decryption requests. What will B do if C asks for the decryption of some message of the form $C_1'.ID^*$? B cannot call $keygen(ID^*)$ and respond to C's request.

Thus, our attempt to show that this scheme is CCA secure has ended in failure.

3.2 The correct reduction

We solve the problem mentioned in the previous section by signing encrypted messages. This idea is due to Canetti, Halevi and Katz. First, we introduce some notation:

Let SigPK and SigSK denote the public and private keys used by a secure signature algorithm. Let Sign(x) denote a signature for the message x created using SigPK. Let Verify(x, y) denote the result of the verification of a signature y for the message x using SigSK.

The algorithms used in the truly CCA secure scheme are as follows:

- Setup(l): PK = PP. SK = MSK, SigSK, SigPK.
- Enc(m): Choose a random id, find $C_1 = Enc_{IBE}(PP, ID, m)$, then compose the cyphertext $CT = C_1, ID, Sign(C_1)$.
- Dec(CT): Decompose CT into C_1, ID and s, and check if s is the correct signature for C_1 by calling $Verify(C_1, s)$. If this fails, reply with $\perp$ and abort. Otherwise, get $Keygen(MSK, ID) = SK_{id}$; then use $Dec_{IBE}(C_1, SK_{id}) = m$ to retrieve the plaintext.

In this way, the problem encountered with the previous attempt at a provably secure scheme is averted. This can be seen by considering the scenario in which C requests the decryption of a message of the form C $CT = C_1, ID^*, s$ in phase 2, where ID^* is the IBE ID being attacked by B. Now, as the signature scheme is assumed to be secure, the probability of C guessing a valid signature is negligibly small. So, when encountered with this request, B can call $Verify(C_1, s)$, and respond with $\perp$ if the result is negative and abort otherwise. So, the proof goes through and we have a CCA secure cryptosystem.