Lecture 22 - Cramer-Shoup Cryptosystem

Boaz Barak

December 11, 2007

CCA security Recall that in many applications of encryption schemes, we need security against chosen-ciphertext attack (CCA). Therefore constructing public key encryption scheme with CCA security is one of cryptography's most important goals. The first such construction was given by Dolev, Dwork and Naor in 1991, but it used general purpose (non-interactive) zero knowledge for **NP** and hence was not efficient enough for implementations. We saw the encryption schemes of Bellare and Rogaway and Shoup that are conjectured to be CCA secure, but have only a proof of security in the random oracle model. In this lecture we show the 1998 Cramer-Shoup cryptosystem, that was the first practical scheme to be proven CCA secure in the standard model (without assuming random oracles).

CCA security Recall the definition of the CCA security game:

1. The keys (e, d) are chosen and Adversary gets the public key e.
2. Adversary gets to make queries to the decryption box.
3. Adversary chooses a pair of messages m_0, m_1 and gets an encryption y^* of m_b for $b \leftarrow_R \{0, 1\}$.
4. Adversary gets to make more queries to the decryption box as long as it doesn't ask the exact string y^*.
5. Adversary outputs a guess b' and *wins* if $b = b'$.

The scheme is said to be *CCA secure* if for every poly-time A and poly-bounded function ϵ, the probability that A wins in this game is at most $1/2 + \epsilon$.

CCA1 One relaxation of CCA security is called *non-adaptive CCA*, also known as CCA1 or "lunchtime attack" (in contrast, CCA security is known as CCA2 or adaptive CCA). In this relaxation, we drop Step 4 in the above attack: the adversary doesn't get to interact with the decryption box after he receives the challenge. Although this is provides weaker security, which is not necessarily sufficient for all applications, as demonstrated by Bleichenbacher, simple encryption schemes such as padded RSA do not satisfy even this notion. So it is challenging to show even a CCA1 secure encryption scheme (and no such practical proven-secure scheme was known before Cramer and Shoup's work).

Plan We start by describing a "reduced" Cramer Shoup scheme, that achieves only CCA1 security, and then modify it to get full CCA2 security.

DDH and El-Gamal We use the DDH assumption, and so work in some group G with a generator g and assume that it is infeasible to distinguish between (g^x, g^y, g^{xy}) and (g^x, g^y, g^z) where x, y, z are randomly chosen in $\{0, .., |G| - 1\}$.

Thus, we are going to prove the following theorem:

Theorem 1 (Cramer-Shoup 98)**.** *Assume that the DDH assumption holds. Then there exist a CCA secure public key encryption scheme.*

El-Gamal encryption The starting point for the Cramer-Shoup system is the El-Gamal / Diffie-Hellman encryption scheme. Recall that the latter operates as follows:

Keys: Public key: $X = g^x$, private key: x

Encryption: To encrypt m (encoded as a group element), select $r \leftarrow_{\mathrm{R}} \{0, .., |G| - 1\}$ and output $g^r, X^r \cdot m$.

Decryption: To decrypt R, P output P/R^x.

"Reduced" Cramer-Shoup We start with an encryption scheme that is only CCA1 secure. It operates as follows:

Keys: Private key is $x, y, z, w \leftarrow_{\mathrm{R}} \{0..|G|-1\}$, public key is $A = g^x \hat{g}^y, B = g^z \hat{g}^w$ where $\hat{g} = g^\alpha$ where α is chosen at random in $\{0..|G1\}$ ($\hat{g}$ is a generator with very high probability—assume the group has prime order and hence every element other than 1 is a generator).

Encrypt: Choose $r \leftarrow_{\mathrm{R}} \{0..|G| - 1\}$, output $g^r, \hat{g}^r, A^r \cdot m, B^r$.

Decrypt: To decrypt $(R, \hat{R}, P, T)$ check that $T = R^z \hat{R}^w$ (otherwise output $\perp$). If so, output $P/(R^x \hat{R}^y)$.

It is not hard to see that given an encryption of a message m, the decryption algorithm's check will pass and the algorithm will output m.

Intuition: We can think of the tag $T = B^r$ as some kind of a zero knowledge proof that the ciphertext was formed properly. We will see below that this is crucial to the analysis.

Note: We assume that the $|G|$ is prime which means that the numbers $0..|G|-1$ form a field with addition and multiplication modulo p. We also have the following useful fact: let ℓ be a random line over $0..|G| - 1$: $\ell(s) = x + sy$ for random x, y. Then the value $\ell(\alpha)$ doesn't give any information about the value $\ell(\alpha')$ for $\alpha' \neq \alpha$.

Analysis We now analyze the security of the CS lite scheme:

Theorem 2. *Under the DDH assumption, the above scheme is CCA1 secure.*

Proof of Theorem 2 The final challenge ciphertext has the form

$$g^r, \hat{g}^r = g^{\alpha r}, A^r \cdot m = (g^r)^x (g^{\alpha r})^y \cdot m, B^r = (g^r)^z (g^{\alpha r})^w \tag{1}$$

Note that even the honest party (one that generates keys, decrypts and prepares the challenge) never needs to know the value α. Furthermore, by (1), given $R = g^r$, $\hat{R} = g^{\alpha r}$, we can prepare the challenge even without knowing r (assuming we know x, y, z, w. But since the DDH assumption tells us that we cannot distinguish between a tuple $(g, g^\alpha, g^r, g^{\alpha r})$ and a tuple $(g, g^\alpha, g^r, g^{\alpha r'})$ for a random r', then the experiment is equivalent if the ciphertext is generated as follows:

$$g^r, \hat{g}^{r'} = g^{\alpha r'}, (g^r)^x (g^{\alpha r'})^y \cdot m, (g^r)^z (g^{\alpha r})^w \tag{2}$$

Claim: If the challenge ciphertext is prepared as in (2) then even an all-powerful adversary has only negligible information on the value of m.

Proof of claim One convenient notion when working with all powerful adversaries, is that we can assume without loss of generalities that they actually know the discrete logarithms of all quantities involved. So in the proof of this claim, whenever the adversary gets an element of the form g^s we assume it also gets s.

Say that a ciphertext $(R, \hat{R}, P, T)$ is *proper* if there exists r such that $R = g^r, \hat{R} = \hat{g}^r$ (i.e., $\log_g R = \log_{\hat{g}} \hat{R}$). The idea behind the check $T = R^z \hat{R}^w$ is to ensure the ciphertexts are proper. Assume that the adversary does not ask any improper ciphertexts, then we can implement the decryption algorithm as follows:

- To decrypt $(R, \hat{R}, P, T)$ check that $T = R^z \hat{R}^w$ (otherwise output $\perp$). If so, output $P/R^{x+\alpha y}$.

This is enough to prove the claim, since it means that the decryption box can be implemented with the following information: $\alpha, x + \alpha y$, z, w. Looking at (2), the challenge ciphertext gives the adversary also the additional information r and r'. But, given $x + \alpha y$ still doesn't tell the adversary anything about what is $rx + \alpha r' y$ (if $r \ neqr'$ then this is another point on this line). This means that the "pad" $g^{rx+\alpha r' y}$ is completely random as far as the adversary is concerned unless the negligible probability event that $r = r'$ happens.

We still need to prove that the adversary cannot ask any improper ciphertext and get a non-$\perp$ response from the decryption box. But the adversary only information on z, w that the adversary gets from the public key is $z + \alpha w$. An improper ciphertext will consist of values $r \neq r'$ and some number s such that $rz + r' \alpha w = s$, or equivalently, $z + \alpha(r/r')w = s/e'$. That is, the adversary gets $f(\alpha)$ where $f(x) = z + xw$ and needs to guess $f(\alpha')$ for some $\alpha' \neq \alpha$. We know this can happen with probability at most $1/|G|$. $\square$

Reflection Why does this proof break down for CCA2 security? Can you show that Cramer Shoup "lite" is *not* CCA2 secure?

Magic of proof Part of the "magic' of the proof is that we had two separate arguments in the analysis. One argument used the fact that we can simulate the experiment knowing x, y, z, w but not α or the random r used in the challenge ciphertext. The other argument used the fact that we can simulate the experiment knowing r, α, z, w but only $x + \alpha y$ rather than both x and y.

It turns out that the fact we can simulate an experiment using two different pieces of information is very powerful. A somewhat simpler example of this phenomenon (to someone who knows non-interactive zero knowledge) appears in the Naor-Yung CCA1 encryption scheme (and its extension to CCA2 by Sahai).

Full-fledged Cramer-Shoup The full CS encryption operates as follows:

Keys: Private key is $x, y, z, w, z', w' \leftarrow_{\mathrm{R}} \{0..|G|-1\}$, public key is $A = g^x \hat{g}^y, B = g^z \hat{g}^w, B' = g^{z'} \hat{g}^{w'}$ where $\hat{g} = g^\alpha$ where α is chosen at random in $\{0..|G1\}$ ($\hat{g}$ is a generator with very high probability— assume the group has prime order and hence every element other than 1 is a generator). We assume that the public key also contains a collision-resistant hash function H.

Encrypt: Choose $r \leftarrow_{\mathrm{R}} \{0..|G|-1\}$, compute $s = (g^r, \hat{g}^r, A^r \cdot m)$, $\beta = H(s)$ and output $(s, (BB'^{\beta})^r)$.

Decrypt: To decrypt $(R, \hat{R}, P, T)$, compute $\beta = H(R, \hat{R}, P)$ and check that $T = R^{z+\beta z'}\hat{R}^{w+\beta w'}$ (otherwise output $\perp$). If so, output $P/(R^x \hat{R}^y)$.

Analysis The challenge ciphertext has the form

$$g^r, \hat{g}^r = g^{\alpha r}, A^r \cdot m = (g^r)^x (g^{\alpha r})^y \cdot m, (BB'^{\beta})^r = (g^r)^{z+\beta z'}(g^{\alpha r})^{w+\beta w'} \tag{3}$$

where β is the value of the hash function applied to the first three components. Once more, the DDH assumption tells us that this cannot be distinguished from the case that the challenge has the form

$$g^r, g^{\alpha r'}, (g^r)^x (g^{\alpha r'})^y \cdot m, (g^r)^{z+\beta z'}(g^{\alpha r'})^{w+\beta w'} \tag{4}$$

Which means that once more, it suffices to show that if the challenge is as in (4), then the adversary has almost no information on the plaintext even if it is all powerful. In fact, since in real life the adversary is not all powerful, it suffices to show that an all powerful adversary that can do anything except find collisions in the hash function has no information on the plaintext.

As before, if the adversary is only restricted to asking *proper* ciphertexts, both before and after seeing the challenge, then this is true, since the decryption algorithm can then be implemented knowing only $x + \alpha y$, and hence the adversary doesn't get any information on the line $s \mapsto x + sy$ other than the point α. But now it's more challenging to argue that the adversary is only restricted to asking proper ciphertexts, especially since it is given an example of one improper ciphertext passing verification - the challenge ciphertext (4).

Let's keep track on what information the adversary gets on w, z, w', z'. The public key tells him $z + \alpha w$ and $z' + \alpha w'$. The challenge ciphertext tells him

$$rz + r\beta z' + \alpha r' w + \alpha r' \beta w'$$

or (dividing by r and setting $\alpha' = \alpha r'/r$)

$$z + \alpha' w + \beta(z' + \alpha' w') \tag{5}$$

That is, if we let $\ell(s) = z + sw$ and $\ell'(s) = z' + sw'$, then the adversary gets $\ell(\alpha), \ell'(\alpha)$ and then $\ell(\alpha') + \beta\ell'(\alpha')$. But since the values $\ell(\alpha'), \ell'(\alpha')$ are completely random to the adversary (unless $\alpha' = \alpha$ which happens with negligible probability), if we call $z'' = \ell(\alpha')$ and $w'' = \ell'(\alpha')$ and define the line $\ell''(s) = z'' + sw''$ then the adversary gets $\ell''(\beta)$ but $\ell''(\beta')$ is still completely random for him, for every $\beta' \neq \beta$. But note that the fourth component of the ciphertext is a deterministic function of the first three and so, unless the breaks the hash, he can only ask the decryption box with ciphertexts with different hash value than β, which means he will have to predict ℓ'' on a different point. (More precisely, we have four unknowns z, w, z', w': the public key gives the adversary the value of two linear equations on these unknown and the challenge gives him the value of one more equation, but the value of a fourth independent equation is still completely random given this information.)