CMSC 858K — Advanced Topics in Cryptography February 26, 2004

Lecture 10

Lecturer: Jonathan Katz *Scribe(s): Jeffrey Blank, Chiu Yuen Koo, Nikolai Yakovenko*

1 Summary

We had previously begun to analyze the Cramer-Shoup encryption scheme by looking at a simplified version of the scheme that is secure only against *non-adaptive* chosen-ciphertext attacks. These notes revisit this discussion, and then go on to show the full Cramer-Shoup scheme that is secure against *adaptive* chosen-ciphertext attacks [1].

2 Simplified Cramer-Shoup

Recall the simplified Cramer-Shoup scheme:

- $PK = (g_1, g_2, h = g_1^x g_2^y, c = g_1^a g_2^b)$
- $SK = (x, y, a, b)$
- $\mathcal{E}_{PK}(m)$: choose random $r \in \mathbb{Z}_q$; set $C = (g_1^r,\ g_2^r,\ h^r \cdot m,\ c^r)$
- $\mathcal{D}_{SK}(u, v, w, e)$
  - If $e \neq u^a v^b$ then output $\perp$ (i.e., invalid)
  - else output $\frac{w}{u^x v^y}$

A proof of the following was given in the last lecture, but we briefly review it here.

Theorem 1 *Under the DDH assumption, the simplified Cramer-Shoup Scheme is secure against non-adaptive chosen-ciphertext attacks.*

Proof Based on a PPT adversary $\mathcal{A}$ attacking the simplified Cramer-Shoup scheme, we construct an adversary $\mathcal{A}'$ as follows:

```
$\underline{\mathcal{A}'(g_1, g_2, g_3, g_4)}$
$x, y, a, b \leftarrow \mathbb{Z}_q$
$PK = (g_1, g_2, h = g_1^x g_2^y, c = g_1^a g_2^b)$
$(m_0, m_1) \leftarrow \mathcal{A}(PK)$
$b \leftarrow \{0, 1\}$
$b' \leftarrow \mathcal{A}(PK,\ g_3,\ g_4,\ g_3^x g_4^y \cdot m_b,\ g_3^a g_4^b)$
output 1 iff $b' = b$
```

Let Rand be the event that (g_1, g_2, g_3, g_4) are chosen from the distribution of random tuples, and let DH be the event that they were chosen from the distribution of Diffie-Hellman tuples. We have the following claims:

Claim 2 $\big|\Pr[\mathcal{A}' = 1|\mathsf{DH}] - \Pr[\mathcal{A}' = 1|\mathsf{Rand}]\big| = \mathsf{negl}(k)$.

Proof This follows from the DDH assumption and the fact that $\mathcal{A}'$ is a PPT algorithm. ■

Claim 3 $\Pr[\mathcal{A}' = 1|\mathsf{DH}] = \Pr_{\mathcal{A}}[\mathsf{Succ}]$.

Proof By inspection, if (g_1, g_2, g_3, g_4) is a DH tuple, then $g_3 = g_1^r$ and $g_4 = g_2^r$ for some (randomly distributed) $r \in \mathbb{Z}_q$. The claim follows. ■

Claim 4 $|\Pr[\mathcal{A}' = 1|\mathsf{Rand}] - \frac{1}{2}| = \mathsf{negl}(k)$.

Proof Since $\Pr[\mathcal{A}' = 1] = \Pr[\mathcal{A} = b]$, we will show that

$$|\Pr[\mathcal{A} = b \mid \mathsf{Rand}] - \frac{1}{2}| = \mathsf{negl}(k). \tag{1}$$

In fact, we will show that this is true even if $\mathcal{A}$ has unlimited computational power (but can only access the decryption oracle polynomially-many times). If $\mathcal{A}$ is all-powerful, we may assume it knows $\log_{g_1} g_2$; call this γ. From the public key, $\mathcal{A}$ learns that $h = g_1^x g_2^y$, or, equivalently (taking discrete logarithms of both sides):

$$\log_{g_1} h = x + y \cdot \gamma. \tag{2}$$

This collapses the space of (x, y) into q possible pairs, one for each value of $x \in \mathbb{Z}_q$ (and similarly for y). The following sub-claims show that $\mathcal{A}$ cannot learn any additional information about x and y using its decryption queries:

Sub-Claim Except with negligible probability, for all decryption queries (u, v, w, e) made by $\mathcal{A}$ such that $\log_{g_1} u \neq \log_{g_2} v$, decryption returns $\perp$.

Proof of Sub-Claim Before the first decryption query, the only thing $\mathcal{A}$ knows about a and b from the public key is that $c = g_1^a g_2^b$. Taking discrete logarithms of both sides gives:

$$\log_{g_1} c = a + b \cdot \gamma. \tag{3}$$

Consider some ciphertext (u, v, w, e) submitted to the decryption oracle, where $u = g_1^r$ and $v = g_2^{r'}$ with $r \neq r'$. For every $z \in \mathcal{G}$, there is exactly one pair $(a, b) \in \mathbb{Z}_q \times \mathbb{Z}_q$ satisfying Equation (3) and

$$z = u^a v^b \Leftrightarrow \log_{g_1} z = ar + br' \cdot \gamma \tag{4}$$

(this is because Equation 3 and Equation 4 are linearly independent in the unknowns a and b). Therefore, from the point of view of $\mathcal{A}$ the value of $u^a v^b$ is uniformly distributed in $\mathcal{G}$. Furthermore, the ciphertext (u, v, w, e) is rejected unless $e = u^a v^b$. Thus, the ciphertext is rejected except with (negligible) probability $1/q$ (i.e., unless $\mathcal{A}$ happens to choose e correctly).

Assuming the first decryption query of this form (i.e., with $\log_{g_1} u \neq \log_{g_2} v$) is rejected, all $\mathcal{A}$ learns is that $e \neq u^a v^b$. This eliminates one of the q possibilities for (a, b), but there are still $q - 1$ possibilities remaining. Thus, the same argument as above shows that for the second decryption query of this form, the query will be rejected except with probability (at most) $1/(q-1)$. Continuing in this way, the n^{th} decryption query of this form will be rejected except with probability (at most) $1/(q-n+1)$. Thus, the probability that *one* of these queries is *not* rejected is at most $n/(q-n+1)$. Since q is exponential in k while n (the number of decryption queries) is polynomial in k, this proves the claim.

<u>Sub-Claim</u> Assuming all "bad" decryption queries of the form described above are rejected, $\mathcal{A}$ learns no additional information about x and y.

<u>Proof of Sub-Claim</u> Note that when a "bad" decryption query is rejected, $\mathcal{A}$ learns nothing about x and y (since the ciphertext is rejected based on a and b alone). So, we need only look at the "good" decryption queries (u, v, w, e); i.e., those for which $\log_{g_1} u = \log_{g_2} v = r$ (for some r). From the response m to such a query, $\mathcal{A}$ learns that $w/m = (g_1^r)^x (g_2^r)^y$, or:

$$\log_{g_1}(w/m) = xr + yr \cdot \gamma. \tag{5}$$

However, the above equation and Equation (2) are linearly *dependent*. Thus, no extra information about (x, y) is revealed.

The above claims show that, with all but negligible probability, when $\mathcal{A}$ gets the challenge ciphertext $(g_3,\ g_4,\ g_3^x g_4^y \cdot m_b,\ g_3^a g_4^b)$, there are q equally-likely possibilities for (x, y). Thus, with all but negligible probability $g_3^x g_4^y \cdot m_b$ is uniformly distributed over $\mathcal{G}$ and independent of b and hence $\Pr[\mathcal{A} = b] = \frac{1}{2}$. But this is equivalent to Equation (1), completing the proof for Claim 4 and Theorem 1. ■ ■

It is worth noting that the proof above fails to extend for the case of adaptive chosen-ciphertext attacks. The reason is as follows: in trying to extend the proof of Claim 4, we see now that the challenge ciphertext $(g_3,\ g_4,\ g_3^x g_4^y \cdot m_b,\ g_3^a g_4^b)$ gives an additional, independent linear constraint on the pair (a, b); namely:

$$\log_{g_1} e = a \log_{g_1} g_3 + b \log_{g_1} g_4. \tag{6}$$

From Equations 3 and 6, $\mathcal{A}$ could (at least in theory) compute the values of a and b. Thus, the first sub-claim — though still true for the decryption queries made by $\mathcal{A}$ *before* seeing the challenge ciphertext — no longer holds for decryption queries made by $\mathcal{A}$ *after* seeing the challenge ciphertext (indeed, here we see exactly an example of the extra power given by an *adaptive* chosen-ciphertext attack). In particular, $\mathcal{A}$ can potentially make a query of the form $(g_1^r, g_2^{r'}, w, (g_1^r)^a (g_2^{r'})^b)$ (with $r \neq r'$), receive in return the answer m, and thus learn that:

$$\log_{g_1}(w/m) = xr + yr' \cdot \gamma. \tag{7}$$

Since Equations 2 and 7 are linearly independent, $\mathcal{A}$ can compute the values of x and y and decrypt the challenge ciphertext.

The above just shows where the proof breaks down, but does not show an explicit (poly-time) attack. However, such an attack on the simplified Cramer-Shoup scheme is easily derived, and is left as an exercise.

3 The Cramer-Shoup Cryptosystem

The discussion at the end of the last section illustrates that for the proof to extend, we need to ensure that the adversary still cannot submit "bad" ciphertexts which decrypt "properly", even after seeing the challenge ciphertext. We will achieve this by adding two more variables so that the number of unknowns will remain greater than the number of linear equations in these unknowns. The details follows.

Before fully describing the scheme, we note that the scheme will use a *collision-resistant hash function H* hashing arbitrary-length strings to $\mathbb{Z}_q$. We do not give a formal definition here, but content ourselves with the following, *informal* definition: a function is collision-resistant if an adversary cannot find two distinct inputs hashing to the same output in any "reasonable" amount of time. (In fact, a weaker assumption suffices to prove security for the Cramer-Shoup cryptosystem, but we do not explore this further here.)

The Cramer-Shoup scheme is as follows:

- $PK = (g_1,\, g_2,\, h = g_1^x g_2^y,\, c = g_1^a g_2^b,\, d = g_1^{a'} g_2^{b'},\, H)$ where H is a collision-resistant hash function
- $SK = (x, y, a, b, a', b')$
- $\mathcal{E}_{PK}(m)$: Choose random $r \in \mathbb{Z}_q$, and set $C = (g_1^r,\, g_2^r,\, h^r \cdot m,\, (cd^\alpha)^r)$, where $\alpha = H(g_1^r, g_2^r, h^r \cdot m)$.
- $\mathcal{D}_{SK}(u, v, w, e)$
  - if $u^{a+\alpha a'} v^{b+\alpha b'} \neq e$ (where $\alpha = H(u, v, w)$) then output $\perp$ (i.e., invalid)
  - else output $\frac{w}{u^x v^y}$

For an honestly-constructed ciphertext, we have:

$$\begin{aligned} u^{a+\alpha a'} v^{b+\alpha b'} &= u^a v^b (u^{a'} v^{b'})^\alpha \\ &= (g_1^a g_2^b)^r (g_1^{a'} g_2^{b'})^{r\alpha} \\ &= c^r d^{r\alpha} = (cd^\alpha)^r \end{aligned}$$

and so the validity check always succeeds. It can then be verified that decryption recovers the encrypted message.

Theorem 5 *Under the DDH assumption, the Cramer-Shoup scheme is secure against adaptive chosen-ciphertext attacks.*

Proof We proceed as we did in the previous proof, using the same notation as there. Given a PPT adversary $\mathcal{A}$ attacking the Cramer-Shoup Scheme that succeeds with $\Pr_{\mathcal{A}}[\mathsf{Succ}]$, we construct an adversary $\mathcal{A}'$ as follows:

```
$\underline{\mathcal{A}'(g_1, g_2, g_3, g_4)}$
$x, y, a, b, a', b' \leftarrow \mathbb{Z}_q$
$PK = (g_1, g_2, h = g_1^x g_2^y, c = g_1^a g_2^b, d = g_1^{a'} g_2^{b'}, H)$
$(m_0, m_1) \leftarrow \mathcal{A}(PK)$
$b \leftarrow \{0, 1\}$
$b' \leftarrow \mathcal{A}(PK,\, g_3,\, g_4,\, g_3^x g_4^y \cdot m_b,\, g_3^{a+\alpha a'} g_4^{b+\alpha b'})$
output 1 iff $b' = b$
```

The following two claims are exactly as in the previous proof:

Claim 6 $\big|\Pr[\mathcal{A}' = 1|\mathsf{DH}] - \Pr[\mathcal{A}' = 1|\mathsf{Rand}]\big| = \mathsf{negl}(k)$.

Claim 7 $\Pr[\mathcal{A}' = 1|\mathsf{DH}] = \Pr_{\mathcal{A}}[\mathsf{Succ}]$.

To complete the proof, we prove the following claim. Note, however, that the proof is now more complicated than it was previously.

Claim 8 $\big|\Pr[\mathcal{A}' = 1|\mathsf{Rand}] - \frac{1}{2}\big| = \mathsf{negl}(k)$.

Proof As before, we will show that the claim is true even if $\mathcal{A}$ were able to compute discrete logarithms (which, in general, it cannot since it runs in polynomial time). Furthermore, the overall structure of the proof will be the same — namely, we show that $\mathcal{A}$ cannot make any "bad" decryption queries that do not get rejected, and that conditioned on this fact $\mathcal{A}$ has no information about the encrypted message (since it does not have enough information about x, y). Since the latter part of the proof is the same, we only consider the first part of the proof here. The discussion below assumes the reader is familiar with the proof for the simplified Cramer-Shoup scheme given earlier.

Let us look at the information $\mathcal{A}$ possibly learns about a, b, a', b' during the course of the experiment. From the public key, $\mathcal{A}$ learns:

$$\log_{g_1} c = a + b \cdot \gamma \tag{8}$$
$$\log_{g_1} d = a' + b' \cdot \gamma, \tag{9}$$

where we again let $\gamma = \log_{g_1} g_2$. Let $g_3 = g_1^r$ and $g_4 = g_2^{r'}$; as usual, with all but negligible probability we have $r \neq r'$. When $\mathcal{A}$ is given the challenge ciphertext, denoted by $(g_3, g_4, w^* = g_3^x g_4^y \cdot m_b, e^* = g_3^{a+\alpha a'} g_4^{b+\alpha b'})$, $\mathcal{A}$ additionally learns:

$$\log_{g_1} e^* = (a + \alpha a')r + (b + \alpha b')\gamma r'. \tag{10}$$

As in the previous proof, we want to show that, with all but negligible probability, any "bad" decryption queries made by $\mathcal{A}$ — i.e., decryption queries (u, v, w, e) with $\log_{g_1} u \neq \log_{g_2} v$ — will be rejected. Recall that $\mathcal{A}$ is not allowed to submit $(u, v, w, e) = (u^*, v^*, w^*, e^*)$. We look at three possible cases:

Case 1. If $(u, v, w) = (u^*, v^*, w^*)$ but $e \neq e^*$, it is easy to see that this query will always be rejected.

Case 2. If $(u, v, w) \neq (u^*, v^*, w^*)$ but $H(u, v, w) = H(u^*, v^*, w^*)$ then this means that $\mathcal{A}$ has found a collision in H. But since H is collision-resistant and $\mathcal{A}$ runs in polynomial time, we may assume that this happens with only negligible probability.

Case 3. If $H(u, v, w) \neq H(u^*, v^*, w^*)$, then let $\alpha' = H(u, v, w)$ (and, as in Equation (10), $\alpha = H(u^*, v^*, w^*)$). Let $\log_{g_1} u = \hat{r}$ and $\log_{g_2} v = \hat{r}'$, and recall that since we are considering "bad" queries we have $\hat{r} \neq \hat{r}'$. Looking at the first "bad" decryption query made by $\mathcal{A}$, we see that it is not rejected only if:

$$\log_{g_1} e = (a + \alpha' a')\hat{r} + (b + \alpha' b')\gamma \hat{r}'.$$

The key point is that this equation is *linearly independent* of Equations (8)–(10), where these are being viewed as four equations in the four unknowns a, b, a', b'. (Linear independence can be verified by "brute-force" calculation, which is worth doing at least once. In fact, if you work it out you will find that the equations are linearly independent exactly when $r \neq r'$ and $\hat{r} \neq \hat{r}'$ and $\alpha \neq \alpha'$, which are exactly the conditions we consider!)

As in the previous proof, this means that the first "bad" decryption query of $\mathcal{A}$ is rejected except with probability $1/q$. Continuing in the same way as in the previous proof, we see that *all* of the "bad" decryption queries of $\mathcal{A}$ are rejected, except with negligible probability (here, we use the fact that $\mathcal{A}$ may make only polynomially-many queries). This concludes the proof of the claim, and hence the proof of the theorem. ■ ■

References

[1] R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. Crypto '98. Full version available from http://eprint.iacr.org.