6.875J/18.425J Cryptography and Cryptanalysis	November 21, 2001
Lecture 20	
Instructor: Anna Lysyanskaya	*Scribed by: Alexander Firshein*

1 Security for Cryptosystems

1.1 Old Scenario

The way we used to think about the situation was like this: Our world consisted of Alice, Bob, and Eve. Bob wants to send a message securely to Alice without Eve finding out. In the public key model, Alice would publish her public key, Eve would produce a message space that Bob could select messages from, Bob would select his message and encrypt it using Alice's public key, and sends it to Alice as Eve evedrops and finds out the ciphertext. The challenge then is for Eve to be able to tell what message Bob sends.

Also, for signature schemes, the adversary has even more control. The forger has total control over all messages, and succeeds if it can forge a signature. Eve can't communicate with Alice, but the forger can communicate with the signer.

1.2 Revision: Adaptive Chosen-Ciphertext Security

Now we allow Eve to communicate with Alice. This scenario is secure if the only way that Eve can decrypt is by actually asking Alice to decrypt. This is called "Adaptive Chosen-Ciphertext Security" and it goes as follows: the adversary, Eve

- Receives Alice's Public Key
- Queries Alice for decryptions of ciphertexts of her choice
- Declares that she can succeed in figuring out Bob's message if it's chosen according to the Message Space $\mathcal{M}$ (Before, $\mathcal{M}$ was fixed after looking at only PK, but now can use Alice)
- Challenge encryption is issued ($m \leftarrow \mathcal{M}, c \leftarrow E(m, PK)$)
- Query Alice for more decryptions (Except for c)
- Output a guess for m

What Eve is going to try and do is bypass $E()$ so that she gives Alice something such that she finds out something to help her crack it.

Example

Public Key: PK of a TDP, to encrypt m, with $|m| = n$.

Choose n random $r_1, ..., r_n$ and $R_1, ..., R_n$, all from the domain of the TDP; then output $p_{PK}(r_1), ..., p_{PK}(r_n)$ and $R_1 \cdot \ldots \cdot R_n$ and let $c = ((r_1 \cdot R_1) \circ \ldots \circ (r_n \cdot R_n)) \oplus m$. This creates a one-time pad using the Goldreich-Levin hard core bit.

Closer Look

Create a random pad U, with $|U| = n$, by choosing random r_i, R_i; and letting $U = (r_1 \cdot R_1) \circ \ldots \circ (r_n \cdot R_n)$. To encrypt m compute $y_i = p_{PK}(r_i)$ then output $\{R_i\}, \{y_i\}, U \oplus m$. The one time pad is hidden from the adversary, that is why this is semantically secure.

This example is not adaptive however, since $(U \oplus m) \oplus JUNK$ could be Eve's ciphertext to query to Alice, and she would decrypt giving back $m \oplus JUNK$, and then Eve can $\oplus$ that with $JUNK$ to get m, allowing Eve to break the system.

In an upcoming homework problems, we will

- Formalize this notion of security
- Show how to break more : El-Gamal and GM

but for now, we will present something that satisfies this security using this the informal definition from above.

2 Cramer-Shoup Cryptosystem

2.1 Decisional Diffie-Hellman

Recall the DDH assumption and setting. Given prime p with:

$$p = aq + 1; g \in \mathbb{Z}_p{}^*; |g| = q; G =< g >; x, y, z \in \mathbb{Z}_q{}^*$$

it is hard to distinguish, or tell whether the tuple (g_1, g_2, u_1, u_2) is of the form $(g_1, g_1{}^x, g_1{}^y, g_1{}^z)$ or $(g_1, g_1{}^x, g_1{}^y, g_1{}^{xy})$.

2.2 The Cramer-Shoup Cryptosystem

The Cramer-Shoup Cryptosystem is then as follows:

The Public Key consists of

$$p, q, g_1, g_2, G =< g_1 >=< g_2 >, |G| = q, c = g_1^{x_1} g_2^{x_2}, d = g_1^{y_1} g_2^{y_2}, h = g_1^{z_1}, H()$$

where $H()$ is a collision resistant hash function with output in $\mathbb{Z}_q{}^*$. (Actually, an even weaker primitive, called "universal one-way hash function," is sufficient.)

The Secret Key consists of

$$x_1, x_2, y_1, y_2, z_1 \in \mathbb{Z}_q{}^*$$

To encrypt a message $m \in G$, pick a random $r \leftarrow \mathbb{Z}_q{}^*$. Then let

$$u_1 = g_1{}^r, u_2 = g_2{}^r, e = h^r m, \alpha = H(u_1, u_2, e), v = c^r d^{\alpha r}$$

and then the ciphertext would be the tuple (u_1, u_2, e, v).

To decrypt a ciphtext (u_1, u_2, e, v), there are two steps, first checking that it's a valid encryption, and then the actual decryption and retrival of the message.

To validate,

$$v = c^r d^{\alpha r} = (g_1^{x_1} g_2^{x_2})^r (g_1^{y_1} g_2^{y_2})^{\alpha r} = (g_1^r)^{x_1 + \alpha y_1} (g_2^r)^{x_2 + \alpha y_2}$$

so check if $v = u_1^{x_1 + \alpha y_1} u_2^{x_2 + \alpha y_2}$. If not output INVALID and do not decrypt m. Otherwise, continue and output

$$e/(u_1^{z_1}) = e/((g_1^r)^{z_1}) = e/((g_1^{z_1})^r) = e/h^r = m$$

.

So for this cryptosystem, the correction of the decryption is easy to see, and the security, intuitively, is that any cheating will be very difficult to do because of the invalidater. This can be proven via a reduction.

2.3 The Reduction

The goal of the reduction is that on an input tuple (g_1, g_2, u_1, u_2), figure out whether there exists an r such that $u_1 = g_1^r, u_2 = g_2^r$ or whether there exists two distinct random r_1, r_2 and $u_1 = g_1^{r_1}, u_2 = g_2^{r_2}$.

2.4 Big Picture

- Form PK, SK correctly
- Answer the adversary's queries correctly
- Adversary asks for a challenge (choose $m \leftarrow \mathcal{M}$). If input is DDH-1, the challenge is a valid cipher for m. If input is of form DDH-2, the challenge cipher is distributed independently of m
- Answer some more questions
- If adversary's guess for m was correct, then output DDH-1, otherwise DDH-2

A fleshed out proving of this reduction can be found in the actual Cramer-Shoup paper, which is online on the webpage.