Advanced Topics in Cryptography

Lecture 6: El Gamal. Chosen-ciphertext security, the Cramer-Shoup cryptosystem.

Benny Pinkas

based on slides of Moni Naor

Related papers

– Lecture notes of Moni Naor, http://www.cs.ioc.ee/yik/schools/win2004/naor-slides-2.5.ppt

– Lecture notes of Jonathan Katz, http://www.cs.umd.edu/~jkatz/gradcrypto2/NOTES/lecture2.pdf

To specify security of encryption

- The power of the adversary
  - computational
    - Probabilistic polynomial time machine (PPTM)
  - access to the system
    - Can it change the messages?
- What constitutes a failure of the system
  - What it means to break the system.
    - Reading a message
    - Forging a message?

El Gamal Encryption

- We will show that El Gamal encryption provides semantic security under the DDH assumption.
- Before doing that, let's discuss the DDH assumption.

Discrete Log Problem

- A finite cyclic group G of order n. A generator g.
- DL problem for G to the base g:
  - given $Y \in G$ find $0 \leq a \leq n-1$ such that $Y = g^a$

<u>DL Assumption</u> for group G to the base g :

- No *efficient* algorithm can solve whp the DL problem for $Y = g^x$, with $x \in_R [0..n-1]$
- Very useful group for DL:
  - Z_P^*. P and Q: Large primes, s.t. $Q \mid P-1$. g is an element of order Q in $\mathbf{Z}_P^*$. Best known algorithms run in time $\sqrt{\mathbf{Q}}$ or subexponential in log P.
- Randomized reduction
  - Given a specific instance generate a random instance: given y generate $Y' = Yg^r$ for $r \in_R [Q]$
  - Therefore worst case is the same as average case

Diffie-Hellman Search Problem

For $a,b \in_R [Q]$

Given $Y = g^a$ and $X = g^b$ find $Z = g^{ab}$.

Assumption - no algorithm can succeed with high probability

No harder than DL - but not much easier.

Decisional Diffie-Hellman Problem (DDH)

For for generator g and $a,b \in [Q]$

Given g, $Y=g^a$, $X=g^b$ and Z decide whether $Z = g^{ab}$ or $Z \neq g^{ab}$

Equivalent: is $\log_g Y = \log_X Z$

DDH-Assumption:

- The DDH-Problem is hard in the **worst** case.

Average DDH

For $a,b \in_R [Q]$ and c which is either

- $c = ab$
- $c \in_R [Q]$

Given $Y = g^a$ and $X = g^b$ and $Z = g^c$ decide whether $Z = g^{ab}$ or $Z \neq g^{ab}$

DDH-Assumption average case:

- The DDH-Problem is hard for above distribution

Worst to Average case reduction

Theorem:The average case and worst case of the DDH-Assumption are equivalent (solving the DDH problem is no easier on the average case than in the worst case)

Proof:

- Given g^a and g^b and g^c (and P, Q)
- Sample $r, s_1, s_2 \in_R [Q]$
- compute
- $g^{a'} = (g^a)^r g^{s_1}$
- $g^{b'} = (g^b) g^{s_2}$
- $g^{c'} = (g^c)^r (g^a)^{r s_2} (g^b)^{s_1} g^{s_1 s_2}$

…Worst to average

If $c = ab + e \mod Q$ then

- $a' = ra + s_1 \mod Q$
- $b' = b + s_2 \mod Q$
- $c' = a'b' + e\,r \mod Q$

- Always: a' and b' are uniformly distributed.
- If $e = 0$, then $c' = a'b'$. Otherwise c' is uniform and independent in [Q]

Evidence to Validity of DDH

- Endured extensive research for DH search
  - DH-search related to discrete log
- Hard for generic algorithms
  - that work in a **black-box** group
- Computing the most significant bits of g^{ab} is hard
- Random-self-reducibility

El-Gamal Cryptosystem:

- Private key $a \in_R [Q]$
- Public key $Y=g^a$ and P, Q
- To encrypt M
  - choose $r \in_R [Q]$ compute $X=g^r$ and Y^r
  - send $\langle X , Y^r \cdot M \rangle$

- To decrypt **<X**, W>**:**
  - compute $X^a = Y^r$ and
  - output W / X^a

Semantic security of El Gamal encryption

- Semantic security = indistinguishability of encryptions = indistinguishability of an encryption of M from an encryption of a random element
- Suppose that an adversary can
  - Choose M
  - Receive either an encryption of X ($\langle g^r, Y^r \cdot M \rangle$) or an encryption of a random element ($\langle g^r, Y^r \cdot R \rangle$), and distinguish between these cases.
- Then we can use the adversary to break the DDH
  - We are given g^a and g^b and g^c (where g^c is either g^{ab} or random)
  - Define the public key as $Y=g^a$
  - The adversary chooses M
  - We send it $(g^b, g^c \cdot M)$

El-Gamal Security

Under the **DDH assumption** the cryptosystem is *semantically secure* against chosen *plaintext* attacks

but...

- Scheme is malleable
  - To change M to M'=M·C:

    change ⟨**X**, W⟩ to ⟨**X**, W·C⟩
- Therefore the scheme is insecure against chosen ciphertext attacks
  - Given an encryption of M, change it to an encryption of M' and ask to see its decryption.
  - Why is this important?

Security against chosen-ciphertext attacks

- Adversary can ask to receive decryptions of messages of his choice
- Adversary chooses two messages m_0,m_1 (possibly based on the answers he previously received)
- Adversary is given an encryption $E(m_b)$, where $b \in_R \{0,1\}$
- Adversary can issue further decryption queries
- Adversary guesses b

- Adversary succeeds if its probability of guessing b correctly is not negligibly close to ½

The Cramer-Shoup cryptosystem

- Cramer and Shoup suggested (in 1998) an encryption scheme which is practical and provably secure against chosen ciphertext attacks
- Security is based on the DDH assumption
- The overhead is only a few exponentiations

- The basic idea:
  - Add redundancy to the cryptosystem.
  - A ciphertext with the right redundancy is “valid”. Otherwise it is invalid.
  - Decryption is only performed for valid ciphertexts.

Non-adaptive chosen ciphertext security, aka security against lunch-time (or preprocessing) attacks

- Adversary can ask to receive decryptions of messages of his choice
- Adversary chooses two messages m_0,m_1 (possibly based on the answers he previously received)
- Adversary is given an encryption $E(m_b)$, where $b \in_R \{0,1\}$
- ~~Adversary can issue further decryption queries~~
- Adversary guesses b

- Adversary succeeds if its probability of guessing b correctly is not negligibly close to ½

Cramer-Shoup "Lite"

- A simplification of the Cramer-Shoup cryptosystem, which is only secure against non-adaptive chosen ciphertext attacks.

Cramer-Shoup "Lite"

- Setup:
  - A subgroup G of order q, with generators g_1,g_2
- Key generation:
  - $x,y,a,b \leftarrow_R Z_q$
  - $h = (g_1)^x \cdot (g_2)^y \quad c = (g_1)^a \cdot (g_2)^b$
  - Public key = $\langle g_1,g_2,h,c \rangle$
  - Private key = $\langle x,y,a,b \rangle$
- Encryption of m:
  - $r \leftarrow_R Z_q$
  - Ciphertext is $\langle g_1{}^r, g_2{}^r, h^r \cdot m, c^r \rangle$
- Decryption of $\langle u,v,e,w \rangle$:
  - If $(w=u^a v^b)$ then output $e/(u^x v^y)$, otherwise no output.

Correctness?

Overhead?

Security proof (against non-adaptive chosen ciphertext attacks)

- Assume that A attacks the cryptosystem. We build an A' which breaks the DDH assumption.
- We are given an input to A' and we generate a setting for A to work in. We want the following to hold:
  - If the input to A' is a DDH tuple, then the setting of A is exactly as in the case it is attacking the cryptosystem.
  - If the input to A' is a random tuple, then the setting of A provides it with an encryption of a random element.
  - The queries that A' makes to the decryption oracle do not reveal anything.

Constructing A'

- Our input is (g_1,g_2,g_3,g_4), which is either a DDH tuple (of the form g,g^a,g^b,g^{ab}, namely $\log_{g1}(g_3)=\log_{g2}(g_4)$), or a random tuple.
  - $x,y,a,b \leftarrow_R Z_q$
  - $h = (g_1)^x \cdot (g_2)^y \quad c = (g_1)^a \cdot (g_2)^b$
  - Public key = $\langle g_1,g_2,h,c \rangle$
  - Private key = $\langle x,y,a,b \rangle$
  - Answer decryption queries of A, and then receive m_0,m_1.
  - Choose $s \in_R \{0,1\}$.
  - Send to A the ciphertext $\langle g_3, g_4, g_3^x g_4^y \cdot m_s, g_3^a g_4^b \rangle$
  - If the response of A is equal to s then output "DDH tuple", otherwise output "random tuple"

Case 1: The input of A' is a DDH tuple

- THM: If A' receives an input which is a DDH tuple, then the view of A is the same as when it is interacting with a real cryptosystem.
- Corollary: Pr(A' outputs "DDH" | DDH input) = Pr(A succeeds when attacking a real cryptosystem)
- Proof:
  - The public and secret keys generated by A' are of the right format, and the decryption queries are answered correctly.
  - If the input of A' is a DDH tuple
  - then $\log_{g1}(g_3)=\log_{g2}(g_4)=r$
  - and then the ciphertext $\langle g_3, g_4, (g_3)^x(g_4)^y \cdot m_s, (g_3)^a(g_4)^b \rangle$ is of the form $\langle (g_1)^r, (g_2)^r, h^r \cdot m_s, c^r \rangle$, which is the right format.

Case 2: The input of A' is a random tuple

- THM: If A' receives an input which is a random tuple, then (except with negligible probability) A has no information about the bit *s* chosen by A'.

  Namely, | Pr(A guesses s | random tuple) – ½ | is negligible.

- Corollary:
  - | Pr(A' outputs "DDH" | random tuple input) – ½ | = | Pr(A guesses s | random tuple) – ½ |, and is negligible
  - | Pr(A' outputs "DDH" | DDH input) – Pr(A' outputs "DDH" | random tuple input) |

    = |Pr(A succeeds when attacking a real cryptosystem) - ½|

Proof of the theorem

- We will prove the theorem for the case of a *computationally unbounded* A
  - Therefore A knows $\gamma=\log_{g1} g_2$
- Claim 1: With all but negligible prob, all decryption queries (u,v,e,w) s.t. $\log_{g1} u \neq \log_{g2} v$, fail.
- Proof:
  - Suppose $u=g_1^{r}$, $v=g_2^{r'}$, $r \neq r'$.
  - $\forall z$, $\exists$a single pair (a,b), s.t. $w=u^a v^b$, namely $\log_{g1} w = ar+br' \cdot \gamma$.
  - Therefore, for A the value $u^a v^b$ is uniformly distributed, and its guess of w is rejected with probability 1-1/q.
  - If A performs n queries, they are *all* rejected with prob 1-n/q.

Proof of the theorem (contd)

- Claim 2: Assuming all "bad" decryption queries are rejected, A learns no information about *x* and *y.*
- Proof:
  - A knows $\gamma=\log_{g1}g_2$. The public key contains $h=g_1^x g_2^y$, and A therefore learns that $\log_{g1}h=x+y\cdot\gamma$.
  - Bad (rejected) queries reveal nothing about (x,y), since the rejection is based on the values of (a,b) alone.
  - For good queries (u,v,e,w), A learns $e/m=g_1^{rx}g_2^{ry}$. Namely, that $\log_{g1}(e/m)=xr+yr\cdot\gamma$. (Which is a relation it already knows.)
- Claims 1+ 2 $\rightarrow$ after n queries, with probability 1-n/q it holds that the ciphertext $\langle g_3, g_4, g_3^x g_4^y\cdot m_s, g_3^a g_4^b \rangle$ has (q-n) equal probability options for (x,y), and therfore for m.
- QED