CRYPTOGRAPHY ASSIGNMENT 4

Submitted To:
Kunwar Singh
Assistant Professor
Computer Science and Engineering
National Institute of Technology, Tiruchirappalli-620015

Submitted By:

1. Saksham Agarwal (106117082)
2. Ashutosh Kumar Singh (106117012)
3. Mandeep Singh (106117046)
4. Karan Puruswani (106117038)
5. Prajwal Hanu (106117070)

Q1. (i) Prove formally that hardness of CDH problem relative to cyclic group G implies hardness of discrete logarithm problem relative to G.
(ii) Prove formally that hardness of DDH problem relative to G implies hardness of CDH problem relative to G.

(i) Let $(G, q, g) \leftarrow G(1^n)$, where G is a cyclic group of order q with bit-size $||q|| = O(n)$ and g a generator of G.
To prove that hardness of the CDH implies hardness of the discrete-logarithm problem, we show that any algorithm that solves the discrete-logarithm can be used to solve CDH.
Let A be an arbitrary PPT algorithm for the discrete-logarithm problem with respect to G, i.e., on input (G, q, g, g^x) it outputs $x` \in Z_q$ and wins the game if $g^{x`} = g^x$, i.e., $x` = x$.

We construct an algorithm A` for CDH as follows:
Given a CDH instance (G, q, g, g^x, g^y), A` queries A on $(G, q, g, g^x)$ and receives $x` \in Zq$. Then A` computes $(g^y)^{x`}$. Clearly, A` succeeds if and only if A succeeds:

$$(g^y)^{x`} = DH_g(g^x, g^y) \Leftrightarrow x` = x.$$

Hardness of CDH relative to G now implies that the success probability of every PPT algorithm in particular that of A` is bounded by some negligible function negl(n).
Thus, we get

$$Pr[DLogA,G (n) = 1] = Pr[A\ 0\ (G, q, g, g^x, g^y) = g^{xy}] \leq negl(n).$$

(ii) To prove that CDH is harder than the DDH problem,
let A be an arbitrary PPT algorithm for CDH with respect to G,
i.e., on input (G, q, g, g^x, g^y) it outputs $h \in G$ and wins the game if $h = DH_g(g^x, g^y) = g^{xy}$.

We construct an algorithm A` for DDH as follows:
Given access to A and a DDH instance $(G, q, g, g^x, g^y, h`)$,
where either $h` = g^{xy}$
or $h` = g\ z$ for a $z \in Zq$ chosen uniformly at random, the algorithm A` queries A on (G, q, g, g^x, g^y) and receives h.

A` outputs 1 if $h` = h$ and 0 else.

Thus,

$$Pr[A`\ (G, q, g, g^x, g^y, g^{xy}) = 1] = Pr[A(G, q, g, g^x, g^y) = g^{xy}]$$

On the other hand,

$$Pr[A`\ (G, q, g, g^x, g^y, g^z) = 1] = 1/q\ .$$

Assuming that DDH is hard with respect to G, we get

$$|Pr[A`\ (G, q, g, g^x, g^y, g^z) = 1] - Pr[A\ 0\ (G, q, g, g^x, g^y, g^{xy}) = 1]| \leq negl(n).$$

This implies $Pr[A(G, q, g, g^x, g^y) = g^{xy}] \leq negl(n) + 1/q$, which is negligible since $||q|| = n$. This proves hardness of CDH

Ques 2: Prove that text-ElGamal is not semantically secure (IND-CPA).
Answer:

Informally, this is how the ElGamal Cryptosystem works:
The plaintext x is "masked" by multiplying it by β^k, yielding y2. The value α^k is also transmitted as part of the ciphertext. Bob, who knows the private key, a, can compute β^k from α^k . Then he can "remove the mask" by dividing y2 by β^k to obtain x.

Let p be a prime such that the Discrete Logarithm problem in $(Z_p^*, .)$ is infeasible, and let $\alpha \varepsilon Z_p$ be a primitive element. Let $P = Z_p^*$, $C = Z_p^* \times Z_p^*$, and define

$K = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod p\}$.
The values p, α, β are the public key, and a is the private key.
For $K = (p, \alpha, a, \beta)$, and for a (secret) random number $k \varepsilon Z_{p-1}$, define
$e_K(x, k) = (y1, y2)$,
where
$y1 = \alpha^k \bmod p$
and
$y2 = x.\beta^k \bmod p$.
For $y1, y2 \varepsilon Zp^*$
Define $d_K(y1, y2) = y2(y1^a)^{-1} \bmod p$.

Clearly the ElGamal Cryptosystem will be insecure if Oscar can compute the value $a = \log_\alpha \beta$, for then Oscar can decrypt ciphertexts exactly as Bob does. Hence, a necessary condition for the ElGamal Cryptosystem to be secure is that the Discrete Logarithm problem in Zp* is infeasible.

Ques 3: Prove that modified ElGamal is semantically secure (IND-CPA) if the DDH assumption holds. But modified ElGamal is not IND-CCA secure.
Answer: : Assume, by contradiction, that we have an adversary that breaks El Gamal, i.e. that it has significant advantage by a real-or-random definition,

$$AdvA = Pr[A^{E_{pk}}pk(pk) = 1] - Pr[A^{Epk\,o\,\$}(pk) = 1]$$

Since El Gamal is a public key encryption scheme, if it is secure against a single query it is secure against q queries, so we only need to show that it is (t, q,) secure for q = 1; we can thus assume that the adversary A makes exactly one query.

Given such an adversary A that runs in time t and has advantage δ, we can construct an adversary B for DDH that runs in time t + O(1) and has avantage δ. Algorithm B(a, b, c) is as follows:

1. Run A^{Eb} (a), where B's version of the encryption oracle E_b answers its one query m with (b, c · m).
2. Output the same result as A does.

In the case where B is called on a triple of the form (g^x, g^r, g^{rx}), what A sees is identical to interacting with a "real" encryption oracle, $B(g^x, g^r, g^{rx}) = A^{Epk}(pk)$. In the case where B is called on a tuple of the form (g^x, g^r, g^{xz}), A sees the values $a = g^x$ and $(b, c \cdot m) = (g^r, g^z \cdot m)$. Since g^z is selected uniformly at random, $g^z \cdot m$ is also a uniform random value and is thus completely indistinguishable from gr^x. $(m). As $(g^r, g^z \cdot m)$ is the same distribution as $(g^r, g^{rx} \cdot$ $(m), This makes B a perfect simulator of a random oracle in this case, $B(g^x, g^r, g^z) = A^{Epk\,o}$ $^{\$}(pk)$.

This construction thus turns an adversary that breaks El-Gamal, into one that breaks DDH with the same advantage, adding constant time complexity.

Modified ElGamal encryption is not IND-CCA secure because it is homomorphic. IND-CCA allows the adversary to see decryptions of ciphertexts via a decryption oracle. Informally, encrypt-then-prove schemes require an adversary to prove knowledge of a plaintext as part of a valid ciphertext. But then a decryption oracle which the adversary can only call for ciphertexts on already known messages is intuitively redundant.
Hence, the modified ElGamal is not IND-CCA secure.

Ques 4: Explain RSA signature scheme. In RSA digital signature, suppose signatures of Alice for the messages 5 and 9 are respectively 6 and 24. Given the public keys, can you find the signature of Alice for message 405. Explain.
Answer:

Basic Version of the Scheme

Let n = pq, where p and q are primes. Let $P = A = Z_n$, and define $K = \{(n, p, q, a, b) : n = pq,$ where p and q are prime, $ab \equiv 1 \pmod{\Phi(n)}\}$.
The values n and b are the public key, and the values p, q, and a are the private key.
For K = (n, p, q, a, b), define

$$sig_K(x) = x^a \bmod n$$

And

$$ver_K(x, y) = true \Leftrightarrow x \equiv y^b \pmod{n},$$

for $x, y \in Z_n$.

In the given question, the attack on the RSA scheme will be called chosen plaintext attack as we have got the plaintext and we should find out the signature of that plaintext.

In order to do so , we shall follow the RSA Algorithm and see what unknowns we have .

$C = M^d \bmod n$.----------1

Here
C=6, M=5
Similarly

C=24,M=9.

Now at the decryption/receiver side , we have:
$M1 = C^e \bmod n$

Which means $M1 = 6^e \bmod n$ ----------2
And $M1 = 24^e \bmod n$---------- 2

Where e is the public key known.

M1=M means C is the signature hence
Using equation 2 we can find out the value of n and then use that value in equation 1 to find out the private key(d).

Once we have got d and n , we can find the signature of 405.

Ques 5: Define existential forgery. Show that RSA signature scheme is vulnerable for existential forgery under known message attack.Write two methods to prevent existential forgery.
Answer:
Existential forgery is a certain type of attacker goal that is used to formally define the security of digital signature schemes, in particular the unforgeability part of security. Existential forgery is a weak message related forgery against a cryptographic digital signature scheme. Given a victim's verifying key, an existential forgery is achieved, if the attacker finds a signature s for at least one new message m, such that the signature s is valid for m with respect to the victim's verifying key. The message m need not be sensical or useful in any way. Existential forgery defines the outcome of an attack, not the way how or how often the attacker can interact with the attacked signer while the attack is performed

OR

Adversary is able to create a valid signature for at least one message. In other words, Adversary can create a pair (x, y), where x is a message and $ver_K(x, y) = true$. The message x should not be one that has previously been signed by Alice.

RSA signature scheme is vulnerable for existential forgery under known message attack

Suppose $y = sig_k(x)$ and $y' = sig_k(x')$
We can check $e_k(y\ y' \bmod n) = x\ x' \bmod n$
So $y\ y' \bmod n = sig_k(x\ x' \bmod n)$

Two methods to prevent existential forgery are:

1- Alice sends (A, M, S = DA(H(M))) where H is a public pre-image resistant hash function on M.
2- Bob computes EA(S) and H(M), and accepts the signature if and only if they match.

Ques 6: Define selective forgery. Show that RSA signature scheme is vulnerable for selective forgery under chosen message attack.
Answer:

Selective forgery is a message related forgery against a cryptographic digital signature scheme. Given a victim's verifying key, a selective forgery is successful if the attacker finds a signature s for a message m selected by the attacker prior to the attack, such that the signature s is valid for m with respect to the victim's verifying key.

OR
With some non-negligible probability, Adversary is able to create a valid signature on a message chosen by someone else. In other words, if Oscar is given a message x, then he can determine (with some probability) a signature y such that $ver_K(x, y) = true$. The message x should not be one that has previously been signed by Alice.

Q7. Suppose Alice wants to sign a message x. She rst constructs the message digest z = h(x),and then computes the signature on z,namely, y = sigK(z). Then she transmits the ordered pair(x,y) over the channel. Suppose hash function is not secondary image resistant. Then show that Adversary can make existential forgery using known message attack.

Ans:
Adversary will start with a valid signed message (x, y), where $y = sig_K(h(x))$.
(The pair (x, y) could be any message previously signed by Alice.)

Then he computes z = h(x) and attempts to find $x` \neq x$ such that $h(x`) = h(x)$.
If Adversary can do this, (x`, y) would be a valid signed message, so y is a forged signature for the message x`.
This is an existential forgery using a known message attack.
In order to prevent this type of attack, we require that h be secondry image resistant.

Q8. Suppose Alice wants to sign a message x. She rst constructs the message digest z = h(x),and then computes the signature on z,namely, y = sigK(z). Then she transmits the ordered pair(x,y) over the channel. Suppose hash function is not secondary image resistant. Then show that Adversary can make existential forgery using known message attack.

Ans:
Adversary will start with a valid signed message (x, y), where $y = sig_K(h(x))$.

(The pair (x, y) could be any message previously signed by Alice.)

Then he computes z = h(x) and attempts to find x`≠ x such that h(x`) = h(x).
If Adversary can do this, (x`, y) would be a valid signed message, so y is a forged signature for the message x`.
This is an existential forgery using a known message attack.
In order to prevent this type of attack, we require that h be secondry image resistant.

Q9. Suppose Alice wants to sign a message x. She rst constructs the message digest z = h(x), and then computes the signature on z,namely, y = sigK(z). Then she transmits the ordered pair(x,y) over the channel. Suppose the hash function is not collision resistant. Then show that Adversary can make existential forgery using chosen message attack.
Ans:

Adversary first finds two messages x≠ x`such that h(x) = h(x`).
Adversary then gives x to Alice and persuades her to sign the message digest h(x), obtaining y.
Then (x`, y) is a valid signed message and y is a forged signature for the message x`.
This is an existential forgery using a chosen message attack;
It can be prevented if h is collision resistant.

Q12. Show that Schnorr Signature Algorithm is existentially unforgeable under known message attack.

Ans:
Given g and $A = g^a$ (by the discrete log challenger).
Part 1: Generate Signatures:
1. Choose e_i , $s \in Z_p$ randomly.
2. Let $R = A^{-ei} \times g^s$
3. σ = (R,S)
4. Set $H(M_i, R) = e_i$

Part 2: Use Forgery:
Note that we need two separate forgeries such that:
$s1 = k + a \times e1$
$s2 = k + a \times e2$
where k is the same in both forgeries.

To accomplish this, we will need to rewind the attacker. Think of the attacker as an internal algorithm we are using, in which we are able to dive into the code of the algorithm and take snapshots of its state after every step. This way, we can backtrack to any state that the attacker was in at any point during its execution.

Assume the attacker makes Q oracle queries. We will guess that the attacker will forge on its i'th oracle query (we have a 1/Q, i.e. non-negligible, chance of guessing correctly). Once we receive the first forgery (s1), rewind the attacker to right AFTER it made the i'th oracle query, but BEFORE it received its answer. This way, the attacker has already fixed the k it will use, but it now receives a different forgery. Since we have a non-negligible chance of guessing which query the attacker will forge on, we also have a non-negligible chance of breaking discrete log.

Since we have a non-negligible chance of guessing which query the attacker will forge on, we also have a non-negligible chance of breaking discrete log.

Q13. Users Alice and Bob use the Die-Hellman key exchange protocol with common prime p = 23 and a primitive root g = 7. Alice has private key 3 and Bob has private key 5. Find the shared symmetric key. Can you make man in middle attack on Die- Hellman key exchange protocol with one private key and the corresponding public key.

Ans: Given :

p=23
g=7
a=3
b=5

Alice will compute $A \equiv g^a \pmod p$
$= 7^3 \pmod{23}$
$= 21$

Bob will compute $B \equiv g^b \pmod p$
$= 7^5 \pmod{23}$
$= 17$

Shared Symmetric Key

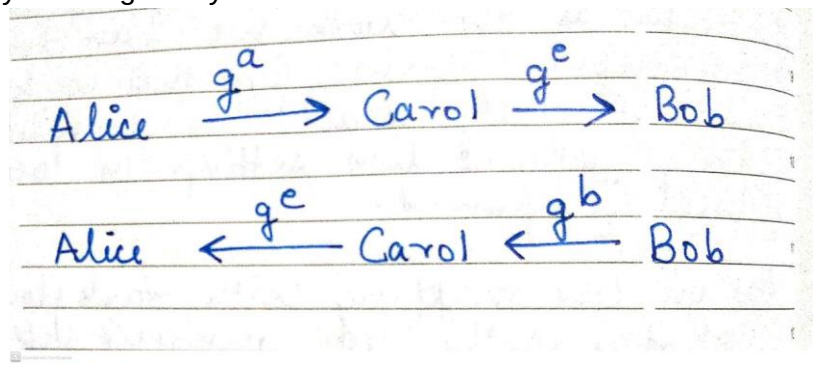
$A' \equiv B^a \pmod p$
$= 17^3 \pmod{23}$
$= 14$
$B' \equiv g^b \pmod p$
$= 21^5 \pmod{23}$
$= 14$

A'=B'
∴ Shared symmetric key is 14

The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack.

In this attack, an opponent Carol intercepts Alice's public value and sends her own public value to Bob. When Bob transmits his public value, Carol substitutes it with her own and sends it to Alice. Carol and Alice thus agree on one shared key and Carol and Bob agree on another shared key. After this exchange, Carol simply decrypts any messages sent out by Alice or Bob, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the other party. This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants.

The following man-in-the-middle attack allows Carol to exchange keys with Alice and Bob while making them believe that they exchanged keys with each other.



Now, whenever Alice sends a message to Bob encrypted with g ae, Eve can decrypt it, read it, and re-encrypt it with g^{be} before sending it to Bob. The other direction works accordingly.