[Aidan Mcdonald](mailto:Aidan.McDonald@uconn.edu)

Tommy Jiang

Luke Pepin

CSE 3400 Problem Set 4

4/11/24

| 1. Show a man-in-the-middle attack against ElGamal. In particular, assume an adversary captures a cipher-text c = (c1, c2) which is the encryption of message m. Now, the adversary cannot learn m, but show an attack allowing the adversary to construct a new ciphertext c ′ = (c ′ 1 , c′ 2 ) which is a valid encryption for a message α · m, for some number α **of the adversary’s choosing**. Your answer should show how an adversary, on input a ciphertext c, and after choosing any α, can send an encryption for the message α · m even though m is never learnt by the adversary.   Multiplication is considered problematic in certain encryption schemes, like ElGamal, because it can introduce vulnerabilities such as the man-in-the-middle attack.  In ElGamal encryption, the vulnerability arises from the fact that the encryption process involves exponentiation, which makes it susceptible to manipulation through multiplication. Specifically, when an adversary intercepts a ciphertext encrypted using ElGamal, they can exploit the multiplicative property of the encryption process.  The exploit of the multiplicative property in encryption, like ElGamal, allows adversaries to manipulate ciphertexts by multiplying them with chosen scalars. This manipulation enables attackers to create new ciphertexts that decrypt to modified messages without needing knowledge of the original plaintext. This vulnerability underscores the importance of safeguarding against such attacks to maintain the security of encrypted communications.  By multiplying the intercepted ciphertext by a chosen factor, the adversary can effectively manipulate the encryption of the original message. This allows them to create a new ciphertext that encrypts a modified version of the original message, without knowing the original plaintext. |
| --- |
| 1. (Based on Ex. 5.11 of the text): Let Fk be a PRF. Consider the protocol shown in Figure 1. Show that it is insecure against a man-in-the-middle attack. In particular, show how an adversary, who can intercept and tamper with any message sent between A and B, can decrypt and learn message m1 sent on the last step. This protocol is a variant of the DH key exchange protocol. Namely, g is a publicly known number and all arithmetic is done modulo a publicly known N. First A picks a random number a and sends the message g a . (Again, g is publicly known.). Next, B picks a random b and sends g b . Both parties can now compute g ab which they use as a shared secret key. The extension in Figure 1 adds an extra step: using this key k = g ab, run the message g b through a PRF to “verify” that Bob really sent that message and not an adversary; if this does not check out, parties abort before sending the message. Show that this is not secure and the adversary can pass this additional test and learn m1 assuming the adversary can tamper with messages and send messages of his/her own. |
| 1. Exercise 6.17 (page 392 of the text).     Problem 1.  Given DH Protocol is insecure due to an efficient discrete log algorithm, a proposal public key cryptosystem key-agreement protocol that is secure against an eavesdropping adversary is as follows: |
| 1. Explain why it is important to ensure that you have an authentic public key before encrypting anything. In particular, consider the protocol shown in Figure 2. Assume that AES-Enc is CBC mode using the AES cipher (thus it is CPA secure); you may assume “Enc” is any secure public key encryption system (e.g., RSA-OAEP). Show how a man-in-the-middle adversary may learn all 3 messages. The adversary may eavesdrop and tamper with any messages transmitting between A and B. Give exact details on what the adversary does. Explain why, even though we are using two secure systems (AES and secure public key system), the overall system is insecure.   Next, show how the adversary can send any message she likes to either A or B. Again, give exact details on the attack. Do your attacks work if the adversary can only eavesdrop?    This shows why it is important to have a public key that is authenticated before encrypting anything, as Eve is able to tamper with anything sent over the channel. Even though we are using two secure systems, AES and secure public key encryption are used within this protocol, the system is insecure as there is no authentication on the public keys that are exchanged. Without authentication, Eve is able to intercept and tamper with the messages and send them without being caught.  In the event that Eve, our adversary, can only eavesdrop and not intercept/manipulate the channel, the attacks would not work because Eve wouldn’t be able to substitute her own public key for Alice’s. Authentication does not matter here, as Eve is unable to alter their messages in any way. Eve cannot decrypt messages sent over the channel here as she cannot find (gb\*a) bcause ga and gb do not provide enough information. |
| 1. Consider the following scenario: A has a secret key and public key pair for textbook RSA (denoted sk and pk respectively); that is RSA without OAEP. B has an authentic copy of pk; as does an adversary. Now, B wants to send his PIN to A, 1 which is a four digit number. He encrypts it as follows: First he chooses a nonce N0, a number chosen randomly from a large domain. He then sends the encryption: N0||RSA(pk, [P IN||N0]) = N0||([P IN||N0] e (mod N)) where (e, N) is the RSA public key. That is, he constructs the new message P IN||N0 (you may assume that he is able to embed [P IN||N0] as a number in Z/NZ ∗ ) and encrypts that with text-book RSA; and he also sends N0 to A “in the clear”. Show an attack which allows the adversary to learn the PIN **using only eavesdropping abilities**. |
| 1. Simulate RSA Key-Gen: Pick your primes to be p = 5 and q = 29. Let e = 3; Find the secret key d. Encrypt the message m = 11, what is the cipher text? Decrypt it to check that it works (i.e., take your cipher text and raise it to the power d, modulo pq, to show that it goes back to the message 11). Refer to Section 6.5.1 in the text for the RSA algorithm. Note, you may want to use a calculator program that can do exponents modulo a number - e.gPowerMod in Mathematica.   ` |