Module	CSCI368 Network Security
Assignment	Assignment 2
Name	YUN MINSEO
Student ID	8225746
Date of Submission	27 Aug 2024

Table of Contents

1. Question 1: Analysis for the protocol in Assignment 1 …… p.3
2. Question 2: Key exchange protocol analysis …… p.4
3. Question 3: TLS/SSL vulnerability …… p.6
4. Question 4: Security threats countered by SSH …… p.7
5. Question 5: IPSec solution …… p.8
6. Question 6: Mobile roaming system design …… p.9

Question 1:
Perform an analysis for the authentication and key exchange protocol in Assignment 1

Protocol in Assignment 1:

1. B → A: “Bob”
2. A → B: E(H(PW), p, g, g^a mod p)
3. B → A: E(H(PW), g^b mod p)
4. A → B: E(K, N_A)
5. B → A: E(K, N_A+1, N_B)
6. A → B: E(K, K_B+1) or "Login Failed"

a) Show that the protocol is subject to offline password guessing attack.

An offline password guessing attack occurs when an attacker can capture some messages and use those messages to guess the passwords without further interaction with legitimate parties.

Steps 2 and 3 of the protocol contain the Diffie-Hellman parameters p, g, g^a mod p, and g^b mod p, encrypted with the hashed password PW.
If the attacker can capture these two messages E(H(PW), p, g, g^a mod p) and E(H(PW), g^b mod p), the attacker can try to decrypt these messages by guessing the password PW since the encryption key is based on H(PW).
If the attacker guessed password PW’, and the hashed guessed password H(PW’) can decrypt the message successfully and decrypted p, g, g^a mod p, and g^b mod p have meaningful values, then they have successfully guessed the password PW.

Therefore, the protocol is subject to offline password guessing attack.

b) Modify the protocol to resist the offline password guessing attack.

1. B → A: “Bob”
2. A → B: A, $E_{H(PW)}$(p, g, g^a mod p)
3. B → A: B, $E_{H(PW)}$(g^b mod p), $MAC_{H(K)}$(B, A, g^a mod p)
4. A → B: $E_{H(PW)}$(N_A), $MAC_{H(K)}$(A, B, g^b mod p)
5. B → A: E_K(N_A+1, N_B)
6. A → B: E_K(N_B+1) or "Login Failed"

Where K= g^{ab} mod p

Since the H(K) is used in MAC, the session key and Diffie-Hellman secrets are tied, it makes attackers hard to verify the guessed password is correct or not.

Question 2:

Consider the following key exchange protocol which is a variant of the Diffie-Hellman protocol…

a) Show the key derivation formulas of User A and User B (i.e., how does each user compute the shared key?

User A:

After receiving g^{rB} mod p from B, User A has rA (her own random number), g^{rB} mod p (from B), xA (her own private key), and g^{xB} mod p (B's public key).
So, User A can compute a session key K by using these values:

$$K = ((g^{xB} \bmod p)^{rA} \bmod p + (g^{rB} \bmod p)^{xA} \bmod p) \bmod p$$
$$= g^{rAxB} + g^{rBxA} \bmod p$$

User B:

After receiving g^{rA} mod p from A, User B has rB (his own random number), g^{rA} mod p (from A), xB (his own private key), and g^{xA} mod p (A's public key).
So, User B can compute a session key K by using these values:

$$K = ((g^{rA} \bmod p)^{xB} \bmod p + (g^{xA} \bmod p)^{rB} \bmod p) \bmod p$$
$$= g^{rAxB} + g^{rBxA} \bmod p$$

Since User A and User B can derive the same session key K, they can compute the shared key with the formulas above.

b) Does the man-in-the-middle attack against the textbook Diffie-Hellman protocol work against the above protocol? Justify your answer.

A Man-in-the-Middle attack against the textbook Diffie-Hellman is effective because the protocol doesn't have authentication.

Since this protocol also doesn't authenticate the communication parties, the same Man-in-the-Middle attack can also be used for this protocol.

Scenario: Let's call attacker Eve E.

1. A → E: g^{rA} mod p
2. E → B: g^{rE} mod p
3. B → E: g^{rB} mod p
4. E → A: g^{rE} mod p

Eve intercepts g^{rA} sent by A, and replaces it with g^{rE}.
Also, Eve intercepts g^{rB} sent by B, and replaces it with g^{rE}.

Now, the shared key derivation would process as:
$K_{AE}=(g^{rAxE}+g^{rExA})$ mod p (Between A and E)
$K_{BE}=(g^{rBxE}+g^{rExB})$ mod p (Between B and E)

Since Eve is impersonating A and B for each party, they can't know exactly who they are communicating with. Therefore, this protocol is vulnerable to MitM similar to the textbook Diffie-Hellman.

c) Does the protocol provide Forward Secrecy? Justify your answer.

In this protocol, if the attacker compromised the private key xA and xB and has intercepted the previous communications (g^{rA} mod p and g^{rB} mod p), the attacker can compute the past session keys K=(g^{rAxB} + g^{rBxA}) mod p.

Since the session key computation involves both long-term private keys, compromising these keys would lead to compromising all past session keys which attacker has captured the rA and rB sharing message.

Therefore, this protocol does not provide forward secrecy.

Question 3:
Study a TLS/SSL related vulnerability or attack that happened in history. Describe the details of the vulnerability or attack and its consequence **using your own words**.

FREAK (Factoring RSA Export Keys) [CVE-2015-0204]

This vulnerability allowed a Man-in-the-Middle (MitM) to trick the SSL connection to use an insecure 'export-grade RSA algorithm', and then use a brute-force attack to obtain the RSA key.

In the past, the United States has required lower encryption levels for exporting its software, which was an 'export-grade' RSA algorithm.
RSA_EXPORT is an export-grade RSA algorithm that uses the maximum length of 512-bit encryption keys, making it more vulnerable to brute-force attacks than the 2,048-bit or higher encryption keys.
Due to improvements in computational performance, the 512-bit RSA encryption key can be compromised very fast, the RSA_EXPORT algorithm is no longer secure.

The attacker can intercept the standard RSA cipher suite request to export RSA cipher suite request. When the server replies with a 512-bit export RSA key, the attacker can brute-force the key and decrypt the communication between the server and client, observing every plain text.

This vulnerability was in OpenSSL version <=0.9.8zd and OpenSSL 1.0.0~1.0.0p, OpenSSL 1.0.1~1.0.1k. It has been patched with OpenSSL 1.0.2.

References:
https://www.redhat.com/en/blog/factoring-rsa-export-keys-freak-cve-2015-0204
https://nvd.nist.gov/vuln/detail/CVE-2015-0204

Question 4:
Consider the following security threats and describe how each is countered by SSH.

a. Impersonation Attack: The attacker impersonates a SSH server to clients.

SSH Key Fingerprints:

When the client connects to an SSH server for the first time, the server sends a public key to the client to identify itself. The client uses this key to verify with trusted key lists or store it for future sessions.

In future sessions, the client compares the server's public key with a stored key. If the key doesn't match, the client considers it as a possibility of an impersonation attack.

b. Replay Attack: The attacker replays a command from the client to the server.

Message Authentication Code (MAC):

SSH uses MAC to integrate cryptographic checksum and sequence numbers to communication.

If an attacker tries to replay the message, MAC checksum verification will fail due to the mismatched sequence number or session key.

c. Reflection Attack: An attacker reflects a message sent by a client back to the client.

Session Identifier:

SSH uses a shared secret K and exchange hash H to identify the session and who (the server or client) sends the message. This means that the hashed session identifier value is different when the server sends it to the client and when the client sends it to the server.

Therefore, if an attacker reflects a message sent by a client back to the client, it will fail to verify the hashed session identifier value and reject the message.

d. Password Sniffing: Passwords in user authentication are eavesdropped.

Encryption:

All communication in SSH is encrypted using a strong encryption algorithm such as AES. All data, including authentication, is encrypted before transmission, making it very difficult for attackers to decrypt without the correct key.

Question 5:
Alice and Bob are employees residing in two dispersed branches, D1 and D2, of the same company… Design a security solution for the above scenario. Describe the format of an IP packet when it is delivered at different sections of the network.

To ensure authenticity in the intranet and ensure confidentiality in the external network, we can use the IPSec with Encapsulating Security Payload (ESP) Protocol, which provides an encrypted data transaction.

We have two communication sections, between Alice and Bob in the intranet and between D1 and D2 in the external network.
ESP in Transport Mode can be used for the intranet (inside D1 and D2).
ESP in Tunnel Mode can be used for the external network (between D1 and D2).

ESP in Transport Mode (inside D1 and D2, intranet)

X	Authenticated	Authenticated & Encrypted			X
IP Header	IPSec ESP Header	TCP/IDP Header	Payload	IPSec ESP Trailer	IPSec ESP Auth

ESP in Tunnel Mode (between D1 and D2, external network)

X	Authenticated	Authenticated & Encrypted				X
Transit IP Header	IPSec ESP Header	Original IP Header	TCP/UDP Header	Payload	IPSec ESP Trailer	IPSec ESP Auth

Since the ESP in Tunnel Mode encrypts the whole original IP packets, it protects the internal IP and payload data from being exposed when transferred through the external network.

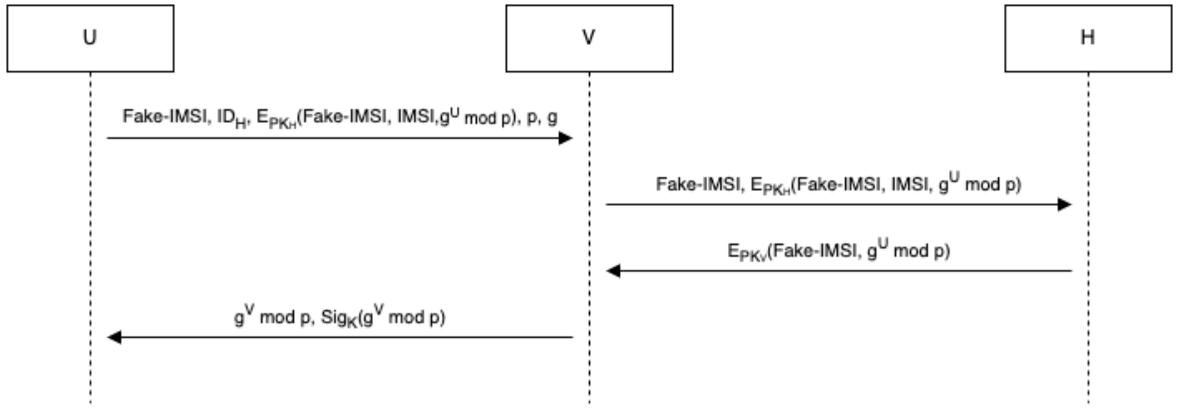
Following is the sample flow:
In D1 (intranet):
Alice’s IP packet is secured using ESP in Transport Mode, so it ensures authenticity and integrity.

From D1 to D2 (external network):
Alice’s IP packet is encapsulated using ESP Tunnel Mode. A transit IP header is added to indicate D1 and D2’s gateway. The entire original packet is encrypted.

In D2 (intranet):
The packet decrypted in D2 and reveals Alice’s original IP packet which was secured with ESP in Transport Mode. This will ensure authenticity within D2 until delivered to Bob.

Question 6:

In a mobile roaming system, there are three entities: a mobile user U, its home server H, and a foreign server V… Suppose that the mobile user U shares a symmetric-key K_{UH} with its home server H. The home server H and the foreign server V also have public keys PK_H and PK_V that are certified by a trusted CA. Your task is to design a secure authentication and key establishment protocol which can satisfy the following security requirements.


U
V
H
Fake-IMSI, ID_H, E_{PK_H}(Fake-IMSI, IMSI,g^U mod p), p, g
Fake-IMSI, E_{PK_H}(Fake-IMSI, IMSI, g^U mod p)
E_{PK_V}(Fake-IMSI, g^U mod p)
g^V mod p, Sig_K(g^V mod p)

Protocol Design

Step 1:

U sends random Fake-IMSI, an ID of H ID_H, (Fake-IMSI, IMSI, and g^U mod p) encrypted with a public key of H, and Diffie-Hellman parameter p and g to V.

Step 2:

V sends Fake-IMSI and encrypted (Fake-IMSI, IMSI, and g^U mod p) to H with ID_H for verification. (Since the (Fake-IMSI, IMSI, g^U mod p) is encrypted with PK_H, V cannot decrypt it.)

Step 3:

H decrypts the message and checks the user is a legitimate subscriber, using the IMSI.

If it is a legitimate subscriber, H replies with (Fake-IMSI, g^U mod p) encrypted with a public key of V.

Step 4:

V decrypts the message from H and reveals Fake-IMSI of the legitimate subscriber and g^U mod p.

Now V can calculate K=g^{VU} mod p. V sends g^V mod p and Sig_K(g^V mod p) for authentication.

Step 5:

U receives the message from V and calculates K=g^{UV} mod p. If Sig_K(g^V mod p) is valid, U can use K as the session key for U and V.

Security Analysis

a. User Authentication

In step 3, the foreign server V will get a reply from H about whether the user U is a legitimate subscriber. If user U is a legitimate subscriber, foreign server V can get the Diffie-Hellman key of U, g^U mod p.

Therefore, the foreign server V can sure that U is a legitimate subscriber of H.

b. Server Authentication

In step 1, the user U's Diffie-Hellman public key g^U mod p was encrypted with the public key of home server H. So the foreign server V cannot reveal the g^U mod p before it gets a reply from H.

In step 3, the message containing public key g^U mod p will be encrypted with PK_V.

The foreign server V can reveal the g^U mod p after home server H re-encrypts the plain data with PK_V.

If the $Sig_K(g^V$ mod p) in step 4 is valid, it means that foreign server V was able to successfully decrypt the message in step 3 using a private key of V.

Therefore, U can trust that V is indeed the real foreign server.

c. Secure Key Agreement

In this protocol, home server H can only reveal the user U's Diffie-Hellman public key g^U mod p. Since the V's Diffie-Hellman public key is only transmitted between U and V, H cannot get g^V mod p.

Therefore, the session key $K=g^{UV}$ mod $p=g^{VU}$ mod p can only be computed by U and V.

d. Anonymity & Unlinkability

User U sends Fake-IMSI as plaintext and encrypts the real IMSI using PK_H. This IMSI will be revealed in the home server H after it decrypts with H's private key.

Home server H will use the decrypted IMSI to check the legitimacy of a user, but they will reply with Fake-IMSI to the foreign server V, not the real IMSI.

Therefore, foreign server V can know that which U is a valid user of H, with Fake-IMSI, but does not know U's real identity since they can't reveal the real IMSI of U.