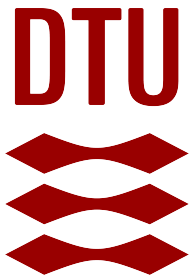
DS - Assignment1 - Report

Group 17

s233578 - Marc Franke
s232101 - Adrian Lopez
s232892 - Qingwen Zeng
s233852 - Peter Zajac

September - October 2023



1 Exercise 1

Consider the example AMP-given.AnB which is a simple Single-Sign-On protocol: A, a normal Internet user, can connect to a webserver B, and authenticate to B using a trusted third party s with whom A has a secret $pw(A, s)$. Assume this is a cryptographically strong symmetric key for beginning.
The given protocol has some vulnerabilities, which can be discovered using OFMC. The following sections are answering questions regarding this vulnerabilities.

1.1 - 1.2 Description of attack and broken goal

1. Problem: Role confusion

In the following attack trace the intruder identifies itself as A against B. With this achievement, the intruder receives the data from B with a self-chosen key, so i can decrypt the data, which was actually for A. So the goal of authentication A against B is broken. The attack trace is followed by a short description of each step of the attack trace.

Attack trace - role confusion

1. $(A,1) \rightarrow i : \{A, i, Request(1), K(1)\}\_(pk(i))$
2. $i \rightarrow (B,1) : \{A, B, Request(2), K(2)\}\_(pk(B))$
3. $(B,1) \rightarrow i : \{|A, s, B, ReqID(2)|\}\_K(2)$
4. $i \rightarrow (A,1) : \{|A, s, i, ReqID(3)|\}\_K(1)$
5. $(A,1) \rightarrow i : \{|A, s|\}\_(pw(A, s))$
6. $i \rightarrow (s,1) : \{|A, s|\}\_(pw(A, s))$
7. $(s,1) \rightarrow i : \{\{A, s\}\_inv(pk(s))\}\_(pk(B))$
8. $i \rightarrow (B,1) : \{\{A, s\}\_inv(pk(s))\}\_(pk(B))$
9. $(B,1) \rightarrow i : \{|Request(2), Data(5)|\}\_K(2)$

Description

1. A starts communication to i as intended for B
2. i starts communication to B pretending to be A choosing a symmetric key (K(2))
3. B responses to i as intended from the protocol if i was A
4. i answers to A with a new chosen ReqID(3) encrypted with symmetric key (K(1))
5. A responses to i as if i is s
6. i forwards the message of A to the real s
7. s responses to i as intended for B
8. i forwards this message to B
9. B responses to i with Request(2) and Data(5) encrypted with K(2)

Solution:
Extend Messages between A $\rightarrow$ s and s $\rightarrow$ B with B (AMP-given_Confusion.AnB)

2. Problem: Freshness problem

This above-mentioned solution fixes the problem regarding the role confusion, but offers the possibility for a new attack. The attack is shown and described in the following. The problem is a freshness problem, where the intruder can reuse messages from the first session in a second session. The message from A $\rightarrow$ s and s $\rightarrow$ B includes no nonce, so B does not know to which ReqID the answer from s belongs to. Therefore, the intruder can get the request at the server in session 1 and introduce itself as A in the second session.

Attack trace - freshness problem

1. $(A,1) \rightarrow i : \{A, B, Request(1), K(1)\}\_(pk(B))$
2. $i \rightarrow (B,1) : \{A, B, Request(1), K(1)\}\_(pk(B))$
3. $(B,1) \rightarrow i : \{|A, s, B, ReqID(2)|\}\_K(1)$
4. $i \rightarrow (A,1) : \{|A, s, B, ReqID(2)|\}\_K(1)$
5. $(A,1) \rightarrow i : \{|A, s, B|\}\_(pw(A,s))$
6. $i \rightarrow (B,2) : \{A, B, Request(2), K(2)\}\_(pk(B))$
7. $(B,2) \rightarrow i : \{|A, s, B, ReqID(4)|\}\_K(2)$
8. $i \rightarrow (s,2) : \{|A, s, B|\}\_(pw(A,s))$
9. $(s,2) \rightarrow i : \{\{A, s, B\}\_inv(pk(s))\}\_(pk(B))$
10. $i \rightarrow (B,2) : \{\{A, s, B\}\_inv(pk(s))\}\_(pk(B))$
11. $(B,2) \rightarrow i : \{|Request(2), Data(6)|\}\_K(2)$

Description

1. A starts communication to i with a message for B
2. i forwards the message to B
3. B replies to i with a message for A
4. i forwards the message to A
5. A sends message to i with a message for s
6. i starts communicating with B in the second session with a self-chosen symmetric key K(2)
7. B responses to this message with a new ReqID (ReqID(4))
8. i now forwards to message from A to the s from step 5 to s in the second session
9. s responses to i with a message for B
10. i forwards this message to B
11. B responses to i with Request(2) and Date(6) encrypted with the symmetric key K(2)

Solution:
Extend Messages between A $\rightarrow$ s and s $\rightarrow$ B with ReqID. To overcome both vulnerabilities, the protocol is extended as follows.

Extended protocol (AMP-given_Freshness.AnB)

- $A \rightarrow B : \{A, B, Request, K\}pk(B)$
- $B \rightarrow A : \{|A, s, B, ReqID|\}K$
- $A \rightarrow s : \{|A, s, B, ReqID|\}pw(A,s)$
- $s \rightarrow B : \{\{A, s, B, ReqID\}inv(pk(s))\}pk(B)$
- $B \rightarrow A : \{|Request, Data|\}K$

1.3 - 1.4 Description of attack and broken goal

Now the shared-secret $pw(A, s)$ is considered as a badly chosen and guessable password, which leads to a new attack which is shown and described below.

Attack trace

1. $(A, 1) \rightarrow i : \{A, i, Request(1), K(1)\}\_(pk(i))$
2. $i \rightarrow (A, 1) : \{|A, s, i, ReqID(1)|\}\_K(1)$
3. $(A, 1) \rightarrow i : \{|A, s, i, ReqID(1)|\}\_(pw(A, s))$
4. $i \rightarrow (i, 17) : guessPW$
5. $i \rightarrow (i, 17) : guessPW$

Description

1. A starts talking to i as to B
2. i responds to A as the protocols requires for B
3. A sends a message to i like talking to the server s encrypted with $pw(A, s)$
4. Because i gets a message which is encrypted with the password, he can guess it, because it is poorly chosen

The problem in this scenario is, that i can easily guess our shared secret between A and s, therefore we do not have a secure way of communication between them. To improve the protocol, a new secure way has to be found.
By extending the protocol as follows, it is ensured that A and s can communicate secure.

Extended protocol (AMP-given_Password1.AnB)

1. $A \rightarrow B : \{A, B, Request, K\}pk(B)$
2. $B \rightarrow A : \{|A, s, B, ReqID|\}K$
3. $A \rightarrow s : \{A, s, B, ReqID, S\}pk(s)$
4. $s \rightarrow A : \{|A, s, B, ReqID|\}S$
5. $A \rightarrow s : \{|pw(A, s)|\}S$
6. $s \rightarrow B : \{\{A, s, B, ReqID\}inv(pk(s))\}pk(B)$
7. $B \rightarrow A : \{|Request, Data|\}K$

Description

3 A initiates the communication to the server. In this message, A includes the sender, receiver, communication partner, nonce (ReqID), and a symmetric key (S).

4 s can now use S to respond to A and confirm the parameters.

5 To authenticate A to s, A uses the shared secret and encrypts it with S to ensure that i does not get hold of pw(A,s)

Alternative solution

An alternative way to solve this issue is adding a nonce NS to the message.
By adding another element that the intruder must guess, guessing the secret, even if the password is not strong, becomes much more difficult. The reason is, even if the intruder manages to guess the password correctly, without having the nonce NS, he won't be able to create the same ciphertext that A had sent to s.

Extended protocol (AMP-given_Password2.AnB)

1. $A \rightarrow B : \{A, B, Request, K\}pk(B)$
2. $B \rightarrow A : \{|A, s, B, ReqID|\}K$
3. $A \rightarrow s : \{A, s, B, ReqID, pw(A, s), NS\}pk(s)$
4. $s \rightarrow B : \{\{A, B, s, ReqID\}inv(pk(s))\}pk(B)$
5. $B \rightarrow A : \{|Request, Data|\}K$

1.5 Combination with TLS

Q: *Suppose one wants to implement the single-sign-on with a TLS channel from exercise 1 to secure the connection between A and B. Describe how that relates to the current formulation of the protocol, in particular if that has any advantages/disadvantages.*

Implement:
The formulation we are using now (SSO) strongly relies on the authentication of public keys, at the same time, there is a risk that the servers may be impersonated and tampered with, causing information leakage. If the SSO protocol wants to add a TLS channel the first thing that will be adopted is the digital signature of the certificate authority and more amount of the use of the cryptographic hash to reduce the risk of keys being transmitted and the uncertainty of the server, meanwhile a third-party certificate authority may provide some more flexible ways for users to log in.

Advantage:
The most significant advantage of combining the single-sign-on with a TLS channel is that it ensures the security of data transmission and makes it more difficult to steal information. Even if the theft is successful, it will be difficult to decipher due to the characteristics of the hash algorithm.

Disadvantage:
A large number of encryption algorithms make the protocol more lengthy and complex. The verification of the public key by the certificate authority requires more resources, which lengthens the overall response time of the system.

2 Exercise 2

A new protocol Selfie.AnB is given for Exercise 2. It is designed as a key update protocol. In the next sections, the protocol is described and extended is such a way that it is secure.

2.1 Description of the protocol

The shape of the initial key is exp(exp(g,secretk(A)),secretk(B)) to generate a key like this can follow these steps:

A generates a random number secretk(A)
B generates a random number secretk(B)

Make exchange respectively

$$A \rightarrow B \quad exp(g, secretk(A))$$

$$B \rightarrow A \quad exp(g, secretk(B))$$

B exponentiate the value exp(g, secretk(A)) from A with his exp(g,secretk(B)) ,

$$exp(exp(g, secret(A)), secret(B))$$

New Key:
kdf(exp(exp(g,secretk(A)),secretk(B)),N1,N2),N1,N2))

Nonce:
The fundamental purpose of random numbers is to generate keys to ensure that the identities of both parties in the connection are correct.Whether it is the client or the server, a random number is needed so that the generated key will not be the same every time. Since the certificate in the protocol is static, it is necessary to introduce a random factor to ensure the randomness of the negotiated key.

MAC:
The purpose of MAC is to verify the authenticity and integrity of a message and to ensure that the message has not been tampered with during transmission.

The goal is to establish a secure connection between A and B, which requires A and B to authenticate each other respectively while ensuring that the information will not be tampered with. Firstly, A sends a nonce N1 to B then B generates another nonce N2 to build a public key through kdf(kdf(exp(exp(g,secretk(A)),secretk(B)),N1,N2)) for authentication, and B use mac to make sure information is not tampered with during transmission. B sends the public key, N1 and N2 to A, after A authenticates B, A will generate another nonce outer and use mac to re-encapsulate all information sent to B. There is a secure channel between A and B to be built afterwards which makes the goals meaningful. However, this protocol is not complete, which will be described in 2.2 and 2.3.

2.2 - 2.3 Description of attack and extension of protocol

Problem:
The goal of (weak) authentication was breached, where A is talking to the intruder, because no authentication was done, and identity confusion happens. There is no authentication requested in the protocol, so we cannot be sure who we are talking to. (Diffie-Hellman)

Solution:
By adding the desired communication recipient to the MAC, when the intruder tries to replay this message back to the sender, the hashed value will not match the expected value.

Attack trace

1. $(A, 1) \rightarrow i : N1(1)$
2. $i \rightarrow (A, 1) : N1(1)$
3. $(A, 1) \rightarrow i : N2(2), mac(kdf(exp(exp(g, secretk(B)), secretk(A)), N1(1), N2(2)), N1(1), N2(2))$

4. $i \rightarrow (A,1) : N2(2), mac(kdf(exp(exp(g, secretk(B)), secretk(A)), N1(1), N2(2)), N1(1), N2(2))$
5. $(A,1) \rightarrow i : A, B, mac(kdf(exp(exp(g, secretk(B)), secretk(A)), N1(1), N2(2)), outer, N1(1), N2(2),$ $mac(kdf(exp(exp(g, secretk(B)), secretk(A)), N1(1), N2(2)), N1(1), N2(2)))$

Description

1. A starts a session with B by sending a nonce N1(1)
2. the intruder intercepts this and replays/reflects the message back to A in a new session.
3. A responses to i following the protocol

Extended protocol (Selfie_Sol1.AnB)

1. $A \rightarrow B : N1$
2. $B \rightarrow A : N2, mac(\mathbf{A}, kdf(exp(exp(g, secretk(A)), secretk(B)), N1, N2), N1, N2)$
3. $A \rightarrow B : A, B, mac(kdf(exp(exp(g, secretk(A)), secretk(B)), N1, N2),$

   $outer, N1, N2, mac(kdf(exp(exp(g, secretk(A)), secretk(B)), N1, N2), N1, N2))$

Alternative Solution (Selfie_Sol1.AnB)

1. $A \rightarrow B : N1$
2. $B \rightarrow A : N2, mac(\mathbf{A}, kdf(exp(exp(g, secretk(A)), secretk(B)), N1, N2), N1, N2)$
3. $A \rightarrow B : A, B, mac(\mathbf{B}, kdf(exp(exp(g, secretk(A)), secretk(B)), N1, N2),$

   $outer, N1, N2, mac(kdf(exp(exp(g, secretk(A)), secretk(B)), N1, N2), N1, N2))$

2.4 Secret guessed after protocol

Q: *Suppose that after the agents have executed the protocol, the intruder would find out* $secretk(A)$ *(for some honest agent* $A \neq i$*). Explain why this would break the secrecy goal of the protocol*

The problem comes down to the way Diffie-Hellman generates the shared *secret key*, i.e. $exp(exp(g, X), Y)$, where g is the generator function, and X and Y are the respective secret keys and X, Y are interchangeable. This means that if one of the private keys was discovered, the intruder would be able to breach the secrecy goal by applying the exponential function with that secret key on the other agent's public key, thus producing the shared secret.

Part II *Is it possible to modify the protocol so that secrecy would still hold as long as secretk(A) is only discovered after the execution of the protocol?*

One such solution would be for each participant to encrypt their nonces with the public key of the other party, this would keep the shared key secret even if one of the secret keys were compromised.

Extended protocol (Selfie_After.AnB)

1. $A \rightarrow B : \{|N1|\}exp(g, secretk(B))$
2. $B \rightarrow A : \{|N2|\}exp(g, secretk(A)), mac(A, kdf(exp(exp(g, secretk(A)), secretk(B)), N1, N2), N1, N2)$
3. $A \rightarrow B : A, B, mac(B, kdf(exp(exp(g, secretk(A)), secretk(B)), N1, N2), outer,$

   $N1, N2, mac(kdf(exp(exp(g, secretk(A)), secretk(B)), N1, N2), N1, N2))$