CMPT479 Assignment2

Zhuo Ning

July 2022

1

a

Problem

In this case, Bob is unable to decrypt the message sent by Alice because he does not own any private key.

Solution

In the case Alice wants to send Bob encrypted email, Bob should be the one who generates the key pairs. Bob should already have a pair of signature keys, and then Bob should generate another pair of keys for encryption. Bob should use the signature private key to sign the encryption public key, and send his signature public key, encryption public key, encryption public key signature to Alice. If this is under SSL/TLS, Bob should also attach his signature public key signature generated by Root CAs. Or if they are using PGP, Alice can go to a trusted server to obtain that signature. Alice will use the signature from server or CA to verify Bob's signature public key, and then use Bob's signature public key to verify Bob's encryption public key signature. If there is no problem, Alice can then use the encryption public key to encrypt the email content and send the encrypted content back to Bob. Bob can then use his encryption private key to decrypt the message.

b

Problem

A man in the middle (attacker) can intercept the communication between Alice and Bob, and send a forged encrypted password to Bob. Assume the attacker does not have the secret key and they are unable to know what the plain text password is. However, based on the previous communication, the attacker might know that this particular communication is setting up Alice's password. Bob is unable to verify whether the message is really coming from Alice or not. Once Bob treats the forged encrypted password as a real password and stored it, the attacker can use the same content as password to access next time until Alice and Bob change their secret key.

Solution

Alice and Bob should use Message Authentication Code (MAC) during communication. Along with the encrypted content, Alice should also send the hash of (key || content) as MAC. Since the attacker does not have the key, they cannot forge the MAC. Bob can then easily verify that the content is really sent by Alice.

c

Problem

Similar to (b), a man in the middle (attacker) can intercept the communication between Alice and Bob, and send a random encrypted content to Bob. Assume the attacker does not have the secret key and they are unable forge their own bank account into the encrypted content. However, if the attacker's purpose is to cause Alice and Bob to lose their money, they will probably succeed. Bob is unable to verify who sent the message. If Bob decrypt the content he will get a random string. If the bank account is not formatted, Bob could send his money to some random account.

Solution

Alice and Bob should use Message Authentication Code (MAC) during communication. Along with the encrypted bank account, Alice should also send the hash of (key || content) as MAC. Since the attacker does not have the key, they cannot forge the MAC. Bob can then easily verify that the content is really sent by Alice.

d

Problem

Bob's public key is exposed to the public for a long time. A long-lived key is safe for signature/verification but not safe for encryption/decryption, because once the key is leaked, all the previous encrypted message will be leaked as well. It is necessary for Bob to create another pair of short-lived encryption/decryption keys and send that public key to Alice. Also, in this case, if Alice directly uses Bob's signature public key to encrypt messages, Bob has no way to verify that the sender is Alice or not. Moreover, using PKE directly for communication is inefficient and also violates the short-lived principle. PKE should be used for symmetric key exchange.

Solution

Similar to (a), when Alice wants to communicate with Bob, she should first send a request to Bob. Bob should generate another pair of keys for encryption. He needs to sign the encryption public key and send the signature and encryption public key to Alice. Alice then should first verify the signature using the public key she downloaded from the trusted server, generate a secret key for SKE, encrypt the secret key with Bob's encryption public key and send it back to Bob. Now they share a pair of symmetric keys and they have established a secure channel.

2

a

RSA's security is related to the difficulty of integer factorization. Currently, integer factorization is a NP-hard problem and there is no algorithm in classical computing better than exponential time. A discovery of a quick polynomial algorithm in classical computing will destroy the security of RSA. Nowadays, RSA is used in almost all internet connections. RSA becoming insecure means all of these connections become insecure. RSA should be abandoned in most use cases, and cryptosuite developers should use other PKE algorithms to replace RSA.

b

We can assume the unknown group has used the stolen private key to pretend to be Facebook for two month. In the meantime, they could trick millions of Facebook users into leaking their personal information. They could also intercept many communications between users and Facebook, forge contents for profit. Facebook should immediately update their signing key. They should force all the users to change their password. They should warn all the users that their private information might be leaked. Users' data might be sold in the Dark web and the loss is unrecoverable.

c

The main property of cryptographic hash is irreversible. If SHA-2 can be easily reversed, we can consider that all the content hashed by SHA-2 in the past is exposed. The password SHA-2 hash stored in any web server will be easily reversed and all the passwords could be leaked. All the secret keys used in MAC could be leaked. Cryptosuite organizations should warn users to abandon SHA-2. All companies and groups previously used SHA-2 should identify the possible leaked data. They should replace SHA-2 with another hash algorithm, and update all the keys they previously used with SHA-2. Also, they are responsible for warning their users for data leakage and force users to update their keys as well.

d

In quantum computing, integer factorization can be done in polynomial time using Shor's algorithm with a large scale quantum computer. If such computers have been constructed, it means RSA is no longer secure. Similar to (a), RSA becoming insecure means all connections using SSL/TLS nowadays become insecure. However, due to the short-lived principle, the data leakage should be minimal. However, all RSA related functionality should be updated using other algorithms that do not rely on integer factorization.