W202 - Homework # 4

Problem # 1 - Once Iranian Secret Police had procured a bogus certificate, what would they have techniqually done to monitor Gmail users using a man-in-the-middle attack?

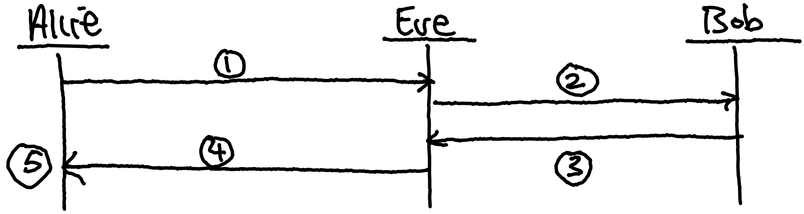
Answer:

At a high level, the Iranian Police would need to setup infrastructure, such as a fake website that looks like Google.com, redirect internet traffic to the fake website, and utilize the bogus certificate to trick browsers into believing the fake website is secure and authentic.

At a more detailed level, the Iranian Police would need to install the bogus certificate on servers hosting the fake website, which enables a complete SSL handshake between the user's browser and the server hosting the fake website. As the internet address of the fake website is different than Google.com, the address must be modified across internet service providers (ISP) and general DNS resolvers that serve as the address book for the internet. Modification techniques include directly updating DNS entries at the ISP (requires coercion) or deploying a DNS poisioning attack to trick DNS resolvers into caching the fake websites address over Google. Once done, the Iranian Police would have the ability to monitor Gmail users, and would simply need to replay login credentials entered at fake website to Google in order to operate un-detected.

Problem # 2a - Describe man-in-the-middle attack on El-Gamal cryptosystem using notation from Hoffstein textbook.

Answer:



1. Alice computes her public key A from $A = g^a \bmod p$, then sends her public key to Eve (thinking its Bob).

2. Eve recieves Alice's public key A, and stores it for future use. Eve then computes another public key A_1 from $A_1 = g^x \bmod p$.

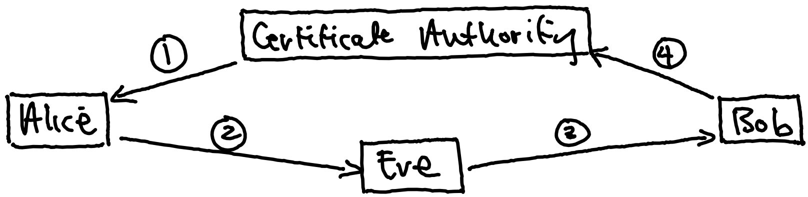
3. Bob chooses random element k, then computes C_1, C_2 from equations $C_1 \equiv g^k \bmod p$ and $C_2 \equiv m A_1^k \bmod p$, where A_1 is public key from Eve (not Alice). Bob sends C_1, C_2 to Eve (thinking its Alice).

4. Eve can decrypt ciphertext from Bob by computing $m = (C_1^x)^{-1} \cdot (C_2) \bmod p$. Once done, Eve can choose to modify message (or just read it) before encrypting message by computing $C_3 \equiv g^h \bmod p$ and $C_4 \equiv m_1 A^h \bmod p$, where h is random element from Eve; m_1 is message by Eve (can be same message); and A is Alice's public key.

5. Alice recieves C_3, C_4 from Eve, and is able to decrypt by computing $m_1 = (C_3^a)^{-1} \cdot (C_4) \bmod p$.

Problem # 2b - Discuss how the use of public key certificates could solve the attack.

Answer :



① Alice recieves a certificate from a public certificate authority, which contains her public key.

② Alice attempts to share her public key with Bob but some how, its intercepted by Eve whom tries to mount a MITM attack.

③ Eve performs same actions as described in Problem 2a, then sends modified public key to Bob.

④ Bob independently verifies the public key he has recieved from Eve with the public certificate authority. Bob will notice the public key from Eve (and digital signature) does not match the public keys maintained by the trusted certificate authority, thus informing Bob of a problem.

Problem # 2c - What prevents Eve from using a second MITM attack to pretend to be the certificate authority issuing the bogus certificate?

Answer:

While it is possible for Eve to mount a second MITM attack with a certificate authority (ie DigiNotar), the level of effort and existing security mechanisms prevent Eve from doing so. More specifically, Eve might try to issue bogus certificate on behalf of Alice from the same certificate authority that issued Alice's original certificate. In doing so, several issues may arise that includes errors with duplicate certificates, different digital signatures (bogus and real certificate) and changes across internet infrastructure described in Problem 1.

Eve could also pretend to be her own certificate authority, however, it's unlikely anyone will trust a new certificate authority nobody has hear of. Should Eve be successful in getting others to trust her certificate authority, mechanisms like certificate pinning help mitigate use of unauthorized public keys.

Problem # 3

Explain how to attack RSA if you are able to compute arbitrary discrete logs (mod N). That is, explain how to recover "d" from $s = hash(M)^d \mod N$ or from $C = M^e \mod N$.

Solution:

① Definition of discrete log problem:

↳ $g^x \equiv h \mod N$ (Hoffstein)

where g, h, N are known values and x is unknown value (private key)

② We have a discrete log machine that can calculate "x" in equation $g^x \equiv h \mod N$. More specifically, the machine calculates $x = \log_g h \mod N$, where g, h, N are known values.

③ Referring to RSA equation $C = m^e \mod N$, we can easily obtain Bob's public key (ie: from a public certificate authority), which gives us values: e, N.

④ Using Bob's public key, we can encrypt an arbitrary message and derive C (ciphertext) using RSA equation $C = m^e \mod N$.

⑤ The RSA equation $C \equiv m^e \mod N$, can be rewritten

as: $C^d \equiv m \mod N$

or: $d = \log_c m \mod N$ (discrete log problem)

(next page)

⑥ From steps 3, 4, 5, we know the following:

- ↳ Values e, N (Bob's public key)
- ↳ Values C, m (Arbitrary message encrypted with Bob's public key)
- ↳ Equation : $d = \log_c m \mod N$

Answer: Given we have a way to compute arbitrary discrete logs (mod N), which allows us to solve for "d" in RSA equation:

$$d = \log_c m \mod N$$

We proved we were able to compute the values C, m, N (required by equation) by simply obtaining Bob's public key, and encrypting an arbitrary message with Bob's public key. This allows us to compute "d" without Bob knowing and break RSA.

Problem #4.1

- Let p be an odd prime and let g be a primitive root modulo p. Prove a has a square root modulo p if and only if its discrete logarithm ($\log_g(a) \pmod{p-1}$) is even.

- Assume a has a unique square root (mod p), then show that it's discrete log must be even.

Proof:

① Assume $\log_g a \pmod{p-1}$ has square root:

↳ $x^2 \equiv a \pmod{p-1}$

* For a to have a square root, it must be congruent to some value x^2 such that the square root of x^2 will result in a.

② Simplify equations:

↳ $x^2 \equiv a \pmod{p-1}$

↳ $\log_g x^2 \equiv \log_g a \pmod{p-1}$ —> take log base g

↳ $2\log_g x \equiv \log_g a \pmod{p-1}$ ⟶ power rule

↳ $\underline{\underline{2}}(\underline{\underline{\text{some value}}}) \equiv \log_g a \pmod{p-1}$

Answer: By showing "a" has a square root ($x^2 \equiv a \pmod{p-1}$) we simplified the equation and proved that the discrete log of "a" must be even, given $2(\log_g x) \equiv \log_g a \pmod{p-1}$ or something that is multiplied by 2 is by definition even.

Problem # 4.2

- Let p be an odd prime and let g be a primitive root modulo p. Prove a has a square root modulo p if and only if its discrete logarithm ($\log_g(a) \pmod{p-1}$) is even.
- Assume that the discrete log of a is even, then show that a must have a square root mod p.

Proof :

① Assume $\log_g a \pmod{p-1}$ is even :

↳ $\log_g a \equiv 2x \pmod{p-1}$

* Definition of even number - when two whole numbers are divided by two, it produces two whole numbers. Thus, by multiplying some value "x" by 2, the logarithm of a will be even.

② Simplify equation :

↳ $\log_g a \equiv 2x \pmod{p-1}$

↳ $a \equiv g^{2x} \pmod{p-1}$ ⟶ power rule

↳ $a \equiv (g^{x})^2 \pmod{p-1}$

↳ $a \equiv (\text{some value})^2 \pmod{p-1}$

Answer : By showing the discrete log of "a" is even ($\log_g a \equiv 2x \pmod{p-1}$), we simplified the equation and proved that "a" must have a square root given $a \equiv (g^x)^2 \pmod{p-1}$ or something squared has a square root.

References

Student Discussions

- Chidanand Bangalore
- Satya srinivas
- Naphi Tang
- Nick Keith
- Jyotsna Sharma
- Aaron Crouch
- Nahid Farhady
- Lauren Ayala

Office hours (Slack

Tools and Resources

- Hoffstein textbook
- Stallings textbook
- Async video (module 4)
- Google searches
- Youtube
- Wikipedia
- Fox IT DigiNotar Investigations Report