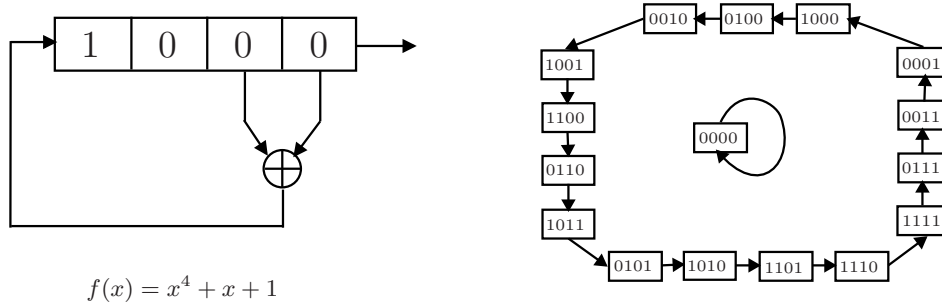


Solutions to Assignment 2

Xinxin Fan

Solutions

1. (a) Since the PRSG is an LFSR with the characteristic polynomial $x^4 + x + 1$ with an initial state $(a_0, a_1, a_2, a_3) = 0001$, we can obtain the following state transition diagram:



Assume that Alice and Bob generate the challenge numbers $R_A = 1010$ and $R_B = 1111$, respectively, from the above PRSG. Then the authentication tags can be calculated as follows:

$$\begin{aligned} \text{Tag}_B &= \text{MAC}(K, ID_B, ID_A, R_B, R_A) = f(1100 \oplus 1101 \oplus 0101 \oplus 1111 \oplus 1010) \\ &= f(0001) = 0001, \\ \text{Tag}_A &= \text{MAC}(K, ID_A, R_B) = f(1100 \oplus 0101 \oplus 1111) = f(0110) = 0111. \end{aligned}$$

(b) Firstly, an attacker E can use the vulnerability of the given MAC function to derive the pre-shared key between A and B and therefore impersonates A as follows:

- (i) The attacker E first chooses a random challenge number R_A (for example $R_A = 0000$) and sends ID_A and R_A to B .
- (ii) B chooses a random challenge number R_B (for example $R_B = 0001$) and calculates the authentication tag Tag_B as follows:

$$\begin{aligned} \text{Tag}_B &= \text{MAC}(K, ID_B, ID_A, R_B, R_A) = f(1100 \oplus 1101 \oplus 0101 \oplus 0001 \oplus 0000) \\ &= f(0101) = 1011. \end{aligned}$$

Then B sends ID_B, ID_A, R_B, R_A and the authentication tag Tag_B to A .

- (iii) Since the MAC function is known to everyone, the attacker E can look up the table and find the preimage of the Tag_B which is 0101. Therefore, the attacker E can derive the pre-shared key K between A and B as follows:

$$\begin{aligned} K &= f^{-1}(\text{Tag}_B) \oplus ID_B \oplus ID_A \oplus R_B \oplus R_A \\ &= 0101 \oplus 1101 \oplus 0101 \oplus 0001 \oplus 0000 = 1100. \end{aligned}$$

- (iv) With the pre-shared key K , the attacker E can impersonate A to generate the correct authentication tag Tag_A as follows:

$$\text{Tag}_A = \text{MAC}(K, ID_A, R_B) = f(1100 \oplus 0101 \oplus 0001) = f(1000) = 1111.$$

Then E sends ID_A, R_B and the authentication tag Tag_A to B .

- (v) B will accept the attacker E as A .

Secondly, since the period of the given PRSG is very small, A can only generate 16 different authentication tags for a given pre-shared key K . Therefore, the probability that a tag will be reused by A is $\frac{1}{16}$, which is not negligible. So an attacker E can impersonate A by launching the replay attack in this case.

(c) To thwart the above attacks, we should choose a secure hash function (i.e., SHA-1) to generate authentication tags and a secure PRSG (i.e., RC4) to generate challenge numbers.

5. (a) The random number R_A and A 's ephemeral public key g^a function as the random challenges in the identification scheme.

(b) For entity B , the signature $\sigma_B = \text{Sig}_B(R_B, R_A, g^a, g^b)$ is served as the response to the challenge of A . Roughly speaking, signing the random challenges provides mutual authentication and the message authentication code generated using K_S provides the key confirmation. If the tag σ_B (i.e., the response from B) is removed, then A cannot authenticate B . Moreover, if the tag $\text{MAC}(K_S, R_A \| g^a \| g^b \| \sigma_B)$ is removed, then A has no way to be confirmed that B has generated the shared key S .

(c) When the protocol is finished, A and B share the key $S = g^{ab} = (g^b)^a = (g^a)^b$, which includes the contributions from both parties.

(d) If A 's long term secret key is compromised at the time instance t , then an attacker E can impersonate A to initiate the protocol 5 and establish a key on behalf of A with B . However, the keys generated before the time instance t are still secure due to the intractability of *Decision Diffie-Hellman* (DDH) problem. Similar argument is applicable to the party B . Therefore, the protocol 5 can provide perfect forward secrecy.

6. (a) Assume that A selects an integer $a = 23 = (0010111)_2 \in [0, 106]$ and a random number $R_A = 30 = (0011110)_2$. Then A computes the ephemeral Diffie-Hellman key $g^a = 4^{23} \pmod{107} = 56 \pmod{107} = (0111000)_2$. Hence, the message of the first round is

$$\langle R_A, g^a \rangle = \langle 0011110, 0111000 \rangle.$$

(b) In the second round, B selects an integer $b = 47 = (101111)_2 \in [0, 106]$ and a random number $R_B = 73 = (1001001)_2$. Then B computes his ephemeral Diffie-Hellman key $g^b = 4^{47} \pmod{107} = 25 \pmod{107} = (0011001)_2$ and the shared key $S = (g^a)^b = 56^{47} \pmod{107} = 57 \pmod{107} = (0111001)_2$. From the shared key S , B derives a key $K_s = S = (0111001)_2$. B also generates the DSS signature on the message $m_B = \langle R_B, R_A, g^a, g^b \rangle$ as follows:

- Randomly pick a state generated by PRSG: $k_B = 10 = (0001010)_2 \in [0, 106]$.

- Compute $r_B = (g^{k_B} \bmod p)(\bmod q) = (4^{10} \bmod 107)(\bmod 53) = 83 \bmod 53 = 30 = (011110)_2$.
- Compute $h(m_B) = MAC(R_B, R_A, g^a, g^b) = MAC(1001001, 0011110, 0111000, 0011001) = f(1001 \oplus 0010 \oplus 0111 \oplus 1001 \oplus 1100 \oplus 0001 \oplus 1001) = f(0001) = 1$. (Note that since we use the simplified AES 4-bit S -box as the MAC function we divide the message into 4-bit block starting from the least significant bit.)
- Solve for s_B in the equation: $h(m_B) \equiv sk_B \cdot r_B + k_B \cdot s_B \pmod{q} \Rightarrow s_B = k_B^{-1}(h(m_B) - sk_B \cdot r_B) \pmod{q} = 10^{-1}(1 - 7 \times 30) \pmod{53} = 48 = (110000)_2$.

Therefore, $\sigma_B = Sig_B(R_B, R_A, g^a, g^b) = (r_B, s_B) = (011110, 110000)$. B also computes the MAC with the key K_S : $MAC(K_S, R_A || g^a || g^b || \sigma_B) = MAC(0111001, 0011110, 0111000, 0011001, 011110, 110000) = f(0111 \oplus 0010 \oplus 0111 \oplus 1001 \oplus 1100 \oplus 0001 \oplus 1001 \oplus 0111 \oplus 1011 \oplus 0000) = f(0011) = (1110)_2$. Finally, B sends the following messages in the second round:

$$\langle R_B, g^b, \sigma_B, MAC(K_S, R_A || g^a || g^b || \sigma_B) \rangle = \langle 1001001, 0011001, 011110, 110000, 1110 \rangle.$$

(c) After receiving the above messages from B , A first verifies B 's DSS signature as follows:

- Compute $h(m_B) = MAC(R_B, R_A, g^a, g^b) = MAC(1001001, 0011110, 0111000, 0011001) = f(1001 \oplus 0010 \oplus 0111 \oplus 1001 \oplus 1100 \oplus 0001 \oplus 1001) = f(0001) = 1$.
- Compute $s_B^{-1} = 48^{-1} \bmod 53 = 21$, $u_B = s_B^{-1} h(m_B) \bmod q = 21$, and $v_B = -r_B s_B^{-1} \bmod q = -30 \times 21 \bmod 53 = 6$.
- Compute $w_B = (g^{u_B} p k_B^{v_B} \bmod p)(\bmod q) = (4^{21} \times 13^6 \bmod 107)(\bmod 53) = 30$.
- Check $w_B = r_B$ and accept B 's signature.

Then A computes the shared key $S = (g^b)^a = 25^{23} \pmod{107} = 57 \pmod{107} = (0111001)_2$, derives the key $K_s = S = (0111001)_2$, and verifies the MAC value $MAC(K_S, R_A || g^a || g^b || \sigma_B) = MAC(0111001, 0011110, 0111000, 0011001, 011110, 110000) = f(0111 \oplus 0010 \oplus 0111 \oplus 1001 \oplus 1100 \oplus 0001 \oplus 1001 \oplus 0111 \oplus 1011 \oplus 0000) = f(0011) = (1110)_2$. A finds the MAC is valid and then generates the DSS signature on the message $m_A = \langle R_A, R_B, g^a, g^b \rangle$ as follows:

- Randomly pick a state generated by PRSG: $k_A = 20 = (0010100)_2 \in [0, 106]$.
- Compute $r_A = (g^{k_A} \bmod p)(\bmod q) = (4^{20} \bmod 107)(\bmod 53) = 41 \bmod 53 = 41 = (101001)_2$.
- Compute $h(m_A) = MAC(R_A, R_B, g^a, g^b) = MAC(0011110, 1001001, 0111000, 0011001) = f(0011 \oplus 1101 \oplus 0010 \oplus 0101 \oplus 1100 \oplus 0001 \oplus 1001) = f(1101) = 4$.
- Solve for s_A in the equation: $h(m_A) \equiv sk_A \cdot r_A + k_A \cdot s_A \pmod{q} \Rightarrow s_A = k_A^{-1}(h(m_A) - sk_A \cdot r_A) \pmod{q} = 20^{-1}(4 - 3 \times 41) \pmod{53} = 2 = (000010)_2$.

Therefore, $\sigma_A = Sig_A(R_A, R_B, g^a, g^b) = (r_A, s_A) = (101001, 000010)$. A also computes the MAC with the key K_S : $MAC(K_S, R_B || g^a || g^b || \sigma_A) = MAC(0111001, 1001001, 0111000, 0011001, 101001, 000010) = f(0111 \oplus 0011 \oplus 0010 \oplus 0101 \oplus 1100 \oplus 0001 \oplus 1001 \oplus 1010 \oplus 0100 \oplus 0010) = f(1011) = (0101)_2$. Finally, B sends the following messages in the third round:

$$\langle \sigma_A, MAC(K_S, R_B || g^a || g^b || \sigma_A) \rangle = \langle 101001, 000010, 0101 \rangle.$$

(d) The shared key between A and B is $S = (g^a)^b = (g^b)^a = 0111001$.

8. (a) The ciphertext C is the following:

$$C = \text{Enc}(K_C, m) = f(K_C \oplus m) = f(1010 \oplus 1111) = f(0101) = 1011.$$

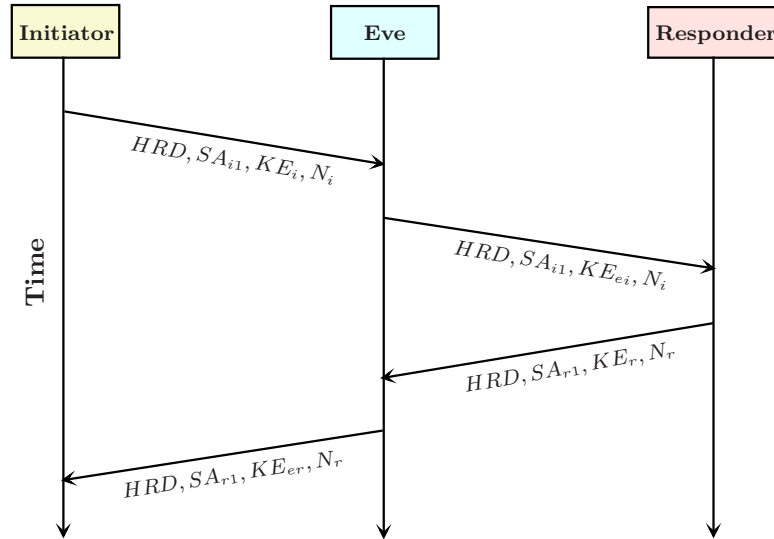
(b) The authentication tag of the header $H = 1000$ and ciphertext $C = 1011$ with key K_I is the following:

$$\text{Tag} = \text{MAC}(K_I, H, C) = f(K_I \oplus H \oplus C) = f(0001 \oplus 1000 \oplus 1011) = f(0010) = 1001.$$

(c) Upon receiving (H, C, Tag) , the receiver will recover the data through the following steps:

- (i) Compute a $\text{Tag} = \text{MAC}(K_I, H, C)$ with K_I and compare it with the received tag. If they are equal, go to the next step. Otherwise, indicate failure and/or abort,
- (ii) Decrypt $m = \text{Dec}(K_C, C)$ with K_C and recover the message m .

9. In **IKE SA INIT**, initiator and responder have negotiated a shared but unauthenticated IKE SA (SA_{r1}). Also, after the Diffie-Hellman key exchange, each party generates a shared but unauthenticated key, **SKEYSEED**, from which all keys are derived for that IKE SA. We should be aware that this phase is vulnerable to a man-in-the-middle attack (see the following figure) since no any protection is provided for the exchanged messages.



From the above figure, we know that after the initial exchange the attacker Eve respectively shares the following secret values SKEYSEED_{ei} and SKEYSEED_{er} with the initiator and the responder:

$$\begin{aligned} \text{SKEYSEED}_{ei} &= \text{PRF}(g^{ie_r}, N_i || N_r), \\ \text{SKEYSEED}_{er} &= \text{PRF}(g^{re_i}, N_i || N_r), \end{aligned}$$

where KEYSEED_{ei} and KEYSEED_{er} are used as keys of a key derivation function to generate the keying material $\text{SK}_{e\&i}$ and $\text{SK}_{e\&r}$ as follows:

$$\begin{aligned}\text{SK}_{e\&i} &= \text{KDF}(\text{KEYSEED}_{ei}, N_i \| N_r \| \text{SPI}_i \| \text{SPI}_e) \\ &= \text{SK}_{dei} \| \text{SK}_{ai} \| \text{SK}_{ae} \| \text{SK}_{ei} \| \text{SK}_{ee} \| \text{SK}_{pi} \| \text{SK}_{pe}, \\ \text{SK}_{e\&r} &= \text{KDF}(\text{KEYSEED}_{er}, N_i \| N_r \| \text{SPI}_e \| \text{SPI}_r), \\ &= \text{SK}_{der} \| \text{SK}_{ae} \| \text{SK}_{ar} \| \text{SK}'_{ee} \| \text{SK}_{er} \| \text{SK}_{pe} \| \text{SK}_{pr}.\end{aligned}$$

In **IKE Auth**, when the initiator sends $\text{AUTH}_i = \text{MAC}_{\text{SK}_{ai}}(\text{INIT}_i, N_r, \text{PRF}(\text{SK}_{pi}, \text{ID}_i))$ to the responder, the attacker Eve will intercept this message and replace it by $\text{AUTH}_e = \text{MAC}_{\text{SK}_{ae}}(\text{INIT}_e, N_r, \text{PRF}(\text{SK}_{pe}, \text{ID}_i))$, which will pass the verification of the responder. Moreover, Eve can impersonate the responder to communicate with the initiator in a similar way. Both the initiator and the responder think they are talking to each other and cannot detect this man-in-the-middle attack.

10. (a) If the key establishment algorithm is *RSA* and the client authentication is not conducted, an attacker can impersonate any client and use *RSA* and the server's public key to transport the pre-master secret generated by himself to the server without being detected by the server.
- (b) If the client must enter password before any further application data will be exchanged, then the attack identified in (a) will not gain access to the server since the attacker does not know the password pre-shared between the client and the server.
- (c) Note that the key establish algorithms *RSA* and *DH* use the long term cryptographic keys to generate the pre-master secret. If the long term keys are comprised, an attacker can derive the master secret and decrypt all previous communication. Therefore, *RSA* and *DH* cannot provide perfect forward secrecy.