LASEC
SECURITY AND CRYPTOGRAPHY LABORATORY

EPFL / I & C
CH-1015 Lausanne
Phone: ++41 (0) 21 693 76 03
Fax: ++41 (0) 21 693 68 79
URL: http://lasecwww.epfl.ch

Advanced Cryptography

Final Exam

(July 11, 2005)

1 Wired Equivalent Privacy (WEP)

In this exercise, we study some real security flaws in the Wired Equivalent Privacy (WEP) protocol used in 802.11 networks to protect the data at the link-layer during wireless transmission. WEP relies on a 40-bit secret key K shared between two communicating parties to protect the data of each transmitted frame. In this exercise, we assume that K is a permanent key which never changes its value. When the user A wants to send a frame of data to B, he proceeds in the following 3 steps

- CRC encoding: Given an n-bit message M (n is a constant), A computes the 32-bit parity check $L(M)$, where L is a linear function that does not depend on K (Note that the linear property of the function L satisfies $L(X \oplus Y) = L(X) \oplus L(Y)$ for any X, Y). The plaintext is $(n + 32)$-bit $P = M \| L(M)$.
- Encryption: A encrypts P with the stream cipher $\mathsf{RC4}$ using the secret key K and a 24-bit initial vector IV assigned to each frame. The ciphertext is $C = P \oplus \mathsf{RC4}(\mathrm{IV}, K)$.
- Transmission: A sends (IV, C) in clear to B over the radio link.

1. Some marketing media advertise that WEP encryption enforces a total of $40 + 24 = 64$ bits security strength. What do you think about this statement? Justify your answer.
2. Explain how the receiver B uses K to extract the original message M upon receipt of (IV, C).
3. In some poor implementations, the 24-bit IV is assigned at random to each frame. Show that it leads to a serious security problem, when one user sends or receives a large amount of data. Propose a better solution.
4. Now we examine another security issue of WEP. Assume that an attacker sitting in-the-middle has intercepted one frame of traffic data (IV, C) from A destined for B. Show that the attacker, who does not know K and does not bother to find K, can *easily* compute a valid C' ($C' \neq C$) such that he can send the modified data (IV, C') to B without fear of detection. How many different choices of such C' does he have? Which property of cryptography is violated here?

2 Batch Verification of DSS Signatures

In this exercise, we consider a variant of the DSS signature from which we remove some modulo q operations. Namely, r is computed as $r = g^k \bmod p$ and the verification consists in checking that

$$r = g^{\frac{H(m)}{s} \bmod q} y^{\frac{r}{s} \bmod q} \bmod p.$$

All the other operations of this DSS variant are identical to those of the original DSS. For the sake of simplicity, this variant will simply be called DSS throughout the exercise.

We recall that g generates a subgroup of $\mathbf{Z}_p^*$ of order q. We denote by ℓ_p and ℓ_q the respective sizes of p and q in bits.

Assume that we have n DSS signatures to verify. We need to check n triplets (m_i, r_i, s_i), where m_i is the ith message and (r_i, s_i) is the corresponding signature, for $1 \leq i \leq n$. We assume that all signatures come from the same signer and correspond to the same public key y and the same parameters p, q, and g.

1. What is the complexity of sequentially verifying all the signatures in terms of ℓ_p, ℓ_q, and n? (You can neglect the computation time of the hash function.)

In order to speed up the verification of the signatures, we will perform a "batch verification", namely we will check all the signatures at the same time. We consider a set $\mathcal{A}$ of N pairwise coprime numbers in $\mathbf{Z}_q^*$ which are smaller than an upper bound $B < \sqrt{q}$. Then, we pick n different elements $a_1, \ldots, a_n$ in $\mathcal{A}$. We define

$$\begin{aligned} R &= r_1^{a_1} r_2^{a_2} \cdots r_n^{a_n} \bmod p, \\ G &= \tfrac{a_1 H(m_1)}{s_1} + \tfrac{a_2 H(m_2)}{s_2} + \cdots + \tfrac{a_n H(m_n)}{s_n} \bmod q, \\ Y &= \tfrac{a_1 r_1}{s_1} + \tfrac{a_2 r_2}{s_2} + \cdots + \tfrac{a_n r_n}{s_n} \bmod q. \end{aligned}$$

A batch verification of these n signatures consists in verifying that

$$R = g^G y^Y \bmod p.$$

2. Show that the batch verification succeeds when all the signatures (m_i, r_i, s_i) for $1 \leq i \leq n$ are valid.

3. What is the complexity of the verification in terms of n, ℓ_p, ℓ_q, and B?

4. Let γ_1 and γ_2 be two elements of the subgroup generated by g such that $\gamma_1 \neq 1$ and $\gamma_2 \neq 1$. Show that there exists at most one pair $(a_1, a_2) \in \mathcal{A} \times \mathcal{A}$ with $a_1 \neq a_2$ satisfying

$$\gamma_1^{a_1} \gamma_2^{a_2} \equiv 1 \pmod{p}.$$

Hint: Given two such pairs (a_1, a_2) and (a'_1, a'_2) deduce that $a'_1 = a_1$ and $a_2 = a'_2$ from $a_1 a'_2 = a'_1 a_2$.

5. Let α_1, β_1, α_2, β_2 be arbitrary elements of the subgroup generated by g, such that $\alpha_1 \neq \beta_1$ and $\alpha_2 \neq \beta_2$. Using result of the previous question, show that there exists at most one pair $(a_1, a_2) \in \mathcal{A} \times \mathcal{A}$ with $a_1 \neq a_2$ satisfying

$$\alpha_1^{a_1} \alpha_2^{a_2} \equiv \beta_1^{a_1} \beta_2^{a_2} \pmod{p}.$$

In what follows, for any invalid signature triplet (m, r, s) we assume that r lies in the subgroup generated by g.

6. For $n = 2$, show that for any two triplets of DSS signatures (m_1, r_1, s_1) and (m_2, r_2, s_2) such that at least one of them is invalid, the probability that the batch verification fails is greater than or equal to

$$1 - \frac{1}{N^2 - N}.$$

Hint: Separate the cases where one or two signatures are invalid. For the latter case, use the previous question.

7. Using the parameters $p = 11$, $q = 5$, $g = 4$, $y = 3 = 4^4 \bmod 11$, $n = 2$, $a_1 = 1$, and $a_2 = 2$, exhibit an example, where at least one signature is invalid but the batch verification passes. We do not require to find the m_i's here, but only the digests $h_1 = H(m_1)$ and $h_2 = H(m_2)$.

3 Conference Key Distribution System

We study a synchronous Conference Key Distribution System (CKDS) for $m > 2$ users denoted by $U_0, U_1, \ldots, U_{m-1}$. Those m users are connected in a ring network (see Figure 1), such that U_i can only send messages to U_j, where $j = i + 1 \bmod m$ for any $i \in \{0, 1, \ldots, m-1\}$. This means that U_i can receive messages from U_j only, where $j = i - 1 \bmod m$, for any $i \in \{0, 1, \ldots, m-1\}$.

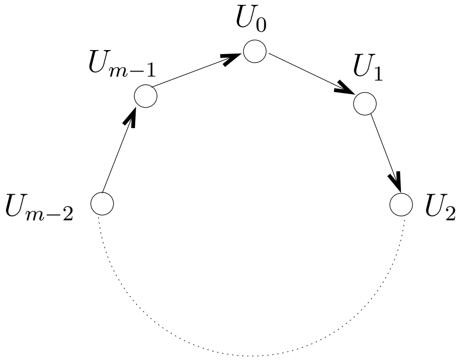

Figure 1: The CKDS ring network

The purpose of the CKDS is to derive one common communication key K for all users over authenticated channels, so that they can hold a confidential conference online. K is generated after several synchronized rounds among the users: during the kth round, U_i sends out two messages denoted by $(S_i^{k,a}, S_i^{k,b})$ and receives two messages $(R_i^{k,a}, R_i^{k,b})$. Thus, according to the message transmission rule, we know that

$$\begin{cases} S_i^{k,a} &= R_j^{k,a} \\ S_i^{k,b} &= R_j^{k,b} \text{ where } j = i + 1 \bmod m. \end{cases}$$

Let us first examine a CKDS for $m = 3$ users U_0, U_1, U_2. The protocol proceeds in 2 synchronized rounds as shown in Algorithm 1.

1. Give the name of a famous protocol to solve the key distribution problem between $m = 2$ users?

2. Express K computed by each user in Algorithm 1 in terms of user secrets N_0, N_1, N_2 and public parameters g, p only.

3. Prove that each user does share the same conference key K.

4. Now, we extend the above CKDS to a CKDS for $m = 4$ users $U_0, \ldots, U_3$ as follows. The setup and the first two rounds of the algorithm are the

Algorithm 1 The key generation algorithm of the CKDS for three users

```
Public parameters:
  1: a large prime $p$, a generator $g$ of $\mathbf{Z}_p^*$
Setup:
  2: Each $U_i$ chooses a random number $N_i \in \mathbf{Z}_p^*$ and keeps it secret.
Key generation:
  3: At the first round, each $U_i$ computes $S_i^{1,a} = g^{N_i} \bmod p$ and sends $(S_i^{1,a}, 1)$.
  4: At the second round, each $U_i$ computes $S_i^{2,a} = R_i^{1,a} \cdot S_i^{1,a} \bmod p$ and $S_i^{2,b} = (R_i^{1,a})^{N_i} \cdot R_i^{1,b} \bmod p$. $U_i$ sends $(S_i^{2,a}, S_i^{2,b})$.
  5: Each $U_i$ computes $K = (R_i^{2,a})^{N_i} \cdot R_i^{2,b} \bmod p$
```

same as in Algorithm 1. After that, we add a third round in which each U_i computes

$$\begin{aligned} S_i^{3,a} &= R_i^{2,a} \cdot S_i^{1,a} \bmod p, \\ S_i^{3,b} &= (R_i^{2,a})^{N_i} \cdot R_i^{2,b} \bmod p, \end{aligned}$$

and sends $(S_i^{3,a}, S_i^{3,b})$. At the end, each U_i computes

$$K = (R_i^{3,a})^{N_i} \cdot R_i^{3,b} \bmod p. \tag{1}$$

Prove that K computed by each user in Equation (1) is the same.

5. We investigate the security of the above CKDS protocol for $m = 4$. Show that given $S_0^{2,b}, S_0^{3,b}, S_1^{3,b}, S_2^{2,b}$, the attacker (wire-tapper) can reconstruct K without the knowledge of user secrets N_i.

6. For an arbitrary m-node CKDS communication network where all the channels are assumed to be authenticated (yet insecure), we define the Multi-Tap Resistance (MTR) by

$$\text{MTR} = \frac{\tau - 1}{m},$$

where τ is the minimum number of physical wires the wire-tapper needs to tap in order to recover K. From the previous question derive an upper bound of MTR for the above CKDS with $m = 4$.

7. Generalize the CKDS protocol for an arbitrary number $m > 2$ of users $U_0, U_1, \ldots, U_{m-1}$. What is the exact total number of multiplications over $\mathbf{Z}_p^*$ that each user must compute to obtain K? And what is the exact total number of exponentiations over $\mathbf{Z}_p^*$ that each user must compute to obtain K?