CHALMERS TEKNISKA HÖGSKOLA
Datavetenskap
Björn von Sydow INN150/TDA350

Solutions to exam in Cryptography 051212

1. (a) The maximum length is $2^n - 1$. We cannot have a longer period, since the state of the generator is a bit string of length n; thus there are 2^n possible internal states. The all-zero state cannot appear in a maximal period LFSR, since that will give an all-zero output (period 1). Thus we cannot output more than $2^n - 1$ bits before we return to a previous state.

   (b) The given sequence has 7 consecutive zeros. This means that an LFSR of size seven or less would lead to an all-zero state after the first bit and then just produce zeros.

   It is easy to construct an LFSR of size 8 that produces the given output: we put $c_8 = 1$ and choose arbitrary values for the other taps. With initial state 10000000, this will generate the desired output.

2. (a) The length L of key_3 is the least common multiple of the lengths of key_1 and key_2. We form key_3 by forming two words key_1' and key_2' of length L, where key_1' consists of a number of copies of key_1 concatenated and key_2' is similarly formed from a number of conatenated copies of key_2. Then $key_3 = key_1' + key_2'$, where the addition symbols means addition letter by letter (using a=0, b=1, ...). Then we have that double encryption is $(m + key_1^\infty) + key_2^\infty$. Using associativity of letter-wise addition we see that this is the same as $m + key_3^\infty$.

   (b) The length of key_3 is much longer than those of key_1 and key_2, at least if these are chosen to be of relatively prime lengths, which is recommended. Thus the probabilities for two occurrences of the same word to be at the same position modulo the period is much smaller.

   Also, if the adversary does find two occurrences of the same ciphertext and computes their distance, it will not be as easy to guess the correct period from the factorisation of the distance.

3. (a) A hash function H should map arbitrarily long messages to bit strings of fixed length (typically 160 bits). It should be fast to compute and it should be collision-free, i.e. it should be infeasible to find messags m_1 and m_2 such that with $H(m_1) = H(m_2)$.

   (b) There are many uses of hash functions. We mention some (not all are needed for full credit):

   - Digital signatures are often computed not on a message itself, but on its hash value.
   - To produce several subkeys from a master key.
   - To construct MACs, such as e.g. the HMAC construction.
   - As parts of paddding schemes, such as e.g. OAEP.
   - To generate strong pseudo-random numbers.

(c) There are several variants but the basic idea is that to generate two messages m_1 and m_2 with $H(m_1) = H(m_2)$ one can expect to have to try in the order of $2^{n/2}$ messages. This is in contrast to generation of a message with a given hash value h, which requires in the order of 2^n messsages.

4. We need to use CBC decryption: $M_i = D_K(C_i) \oplus C_{i-1}$.

Let the messages sent in the first run of the protocol be

1. $A \longrightarrow B : N_A$

2. $B \longrightarrow A : \{N_A, K\}_{K_{AB}}$.

Then we have that $K = D_{K_{AB}}(C_2) \oplus C_1$.

Similarly, let the messages in the second run be

1'. $A \longrightarrow B : N'_A$

2'. $B \longrightarrow A : \{N'_A, K'\}_{K_{AB}}$.

Thus $N'_A = D_{K_{AB}}(C'_1) \oplus C'_0$ and $K' = D_{K_{AB}}(C'_2) \oplus C'_1$.

Now Alice receives $C'_0 C'_1 C_2$. She decrypts this and gets the two blocks $D_{K_{AB}}(C'_1) \oplus C'_0$ and $D_{K_{AB}}(C_2) \oplus C'_1$. She checks that the first of these equals her nonce N'_A, which holds. She therefore accepts the second block as the new session key (and this is not the key K' that B chose). But C can also compute this key since it is

$$D_{K_{AB}}(C_2) \oplus C'_1 = D_{K_{AB}}(C_2) \oplus C_1 \oplus C_1 \oplus C'_1 = K \oplus C_1 \oplus C'_1.$$

Comment: We can easily modify this to become the beginning of the Needham-Schroeder protocol with exactly the same weakness. So, this is a concrete example of the general fact that one should not use the encryption primitive, which is developed to provide confidentiality, when the service needed is authentication.

5. (a) A certificate contains at least the name of a user and his/her public key. In addition it will typically contain expiration time. The certificate is issued by a certificate authority (CA), which is trusted by all users. The CA also signs the certificate, so that all users can verify it; the public key of the CA is known by everyone.

The purpose of certificates and CA's is to provide a means to authenticate public keys.

(b) The attack proceeds as follows:

1. $A \longrightarrow C : \quad n_A$

  1'. $C(A) \longrightarrow B : \quad n_A$

  2'. $B \longrightarrow C(A) : \quad Cert_B, n_B, S_B\{n_A, n_B\}$

2. $C \longrightarrow A : \quad Cert_C, n_B, S_C\{n_A, n_B\}$

3. $A \longrightarrow C : \quad Cert_A, S_A\{n_B, n_A\}$

  3'. $C(A) \longrightarrow B : \quad Cert_A, S_A\{n_B, n_A\}$

C makes use of his run with A to get A:s signature on the two nonces, something he could not produce himself. For this to work, it is essential that he get an initial nonce n_A from A and that he lets B choose the reply nonce. C then just forwards the nonces, adding what is required; in particular, in 3' he adds the signature produced by A. After the protocol run, B will conclude that C is actually A.

6. (a) We have that $c_A = m^3 \bmod N$ and $c_B = m^5 \bmod N$. We note that $5*2-3*3=1$, so Deborah computes $c_B^2 \cdot (c_A^{-1})^3 \bmod N = m$. We also note that c_A^{-1} can be computed using the extended Euclidean algorithm, since $\gcd(m,N)=1$.

   (b) As long as e_A and e_B satisfy $\gcd(e_A,e_B)=1$, we know that there exist integers x and y with $xe_A + ye_B = 1$. Thus Deborah computes $c_A^x \cdot c_B^y \bmod N$ to recover m. If $\gcd(e_A,e_B) > 1$ this does not work and there does not seem to be an easy attack for Deborah.

7. (a) Bob checks that $s^{H(m)} = g \in \mathbb{Z}_N^*$. Justification: We note that according to the signing procedure $x \cdot H(M) = 1 + kr$ for some integer k; thus, computing in $\mathbb{Z}_N^*$, we get

$$s^{H(m)} = g^{x \cdot H(m)} = g^{1+kr} = g \cdot (g^r)^k = g \cdot 1^k = g.$$

   (b) Since g has order r, r must be a divisor of $\Phi(N) = (p-1)(q-1)$. Since r is prime, it must then be a factor of either $p-1$ or $q-1$.

   (c) Since $g^r = 1 \in \mathbb{Z}_N^*$, we must have $g^r = 1 \in \mathbb{Z}_q^*$. Thus g generates a subgroup of $\mathbb{Z}_q^*$ of size 1 or r, since r is prime. But the latter case is impossible, since r is not a divisor of $q-1$, the size of $\mathbb{Z}_q^*$. Hence g generates a group of size 1, i.e. $g = 1 \bmod q$. Therefore we can compute $\gcd(N, g-1) = q$ to get a factor of N.