CHALMERS TEKNISKA HÖGSKOLA
Datavetenskap
Björn von Sydow DIT250/TDA351

Solutions to exam in Cryptography, December 10, 2011

1. We must first run the given LFSR to find its period. Doing so, we see that the beginning of the output sequence is 1001011 and that we are then back to the initial state. Thus the output is periodic with period 7, and it is possible that it could be generated by an LFSR of size 3. To check this, we assume tap sequence $c_1c_2c_3$ and set up the system of equations

$$\begin{array}{rcl} c_3 & = & 1 \\ c_1 & = & 0 \\ c_2 & = & 1 \end{array}$$

which is already solved. It just remains to check that the seventh bit of output agrees with the given output sequence. This is the case and we have shown that the sequence is generated by an LFSR of size three which taps the two leftmost bits (and has initial state 100).

2. (a)
   - Signing is a slow operation, so signing the hash is (much) more efficient if the message is long.
   - Signing the hash prevents forgery by breaking algebraic properties of e.g. RSA signing.

   (b) Non-repudiation is the non-ability of the signer to deny having signed the message.

   (c) A certificate contains at least the name and public key of a user. It may also contain date of expiry, the name of the issuing CA and other data. It is issued by a certification authority, who also signs the certificate. Verification consists at least of verifying the CA's signature.

3. (a) First one computes $\Phi(N) = (p-1)(q-1)$ and then one computes d as the inverse of e in $\mathbb{Z}^*_{\Phi(N)}$, using the extended Euclidean algorithm. For this to work we must have $\gcd(e, \Phi(N)) = 1$.

   (b) No. The security of RSA depends on the fact that only the key owner can compute $\Phi(N)$. If N is a prime number, then $\Phi(N) = N-1$, so anyone can compute d from the public key.

   (c) Yes. If $N = pqr$ then $\Phi(N) = (p-1)(q-1)(r-1)$ and computation of d can only be done by the key owner, who knows p, q and r.

4. (a) If we manage to find a block B, such that $h(B, IV) = IV$, then for any message M the hashes of M and $B||M$ will be the same. We can actually put any number of B's in the beginning of the message without changing the hash value.

   (b) The reason that we got collisions in (a) was that the initial B's did not change the state in the iterative computations, and that the rest of the two messages were identical. When length is added in the padding, the padding will be different for M and $B||M$, and we will not get the same hash value.

(c) The birthday attack consists in generating random messages, hashing them and keeping record of the hash values obtained. Because of the random nature of hash functions, a collision is expected in $O(2^{n/2})$ steps for a n bit hash function.

5. (a) The first block of plaintext is removed.

   (b) The last block of plaintext is removed.

   (c) For this to work, it must be the case that $C_0 = E_K(M_0 \oplus C_{-1})$. We solve this for C_{-1} to get $C_{-1} = D_K(C_0) \oplus M_0$. Someone who knows K can certainly compute C_{-1} in this way.

   (d) The adversary needs help in computing $D_K(C_0)$, i.e. to decrypt a given block. The attack in home assignment 1 provides this help:

   The adversary constructs messages of the form $R_i||C_0$ where R_i has last byte i and varies i until padding is OK. He then expects the last byte of C_0 to be i, and continues byte by byte, exactly as in home assignment 1.

6. (a) The complete attack is as follows:

$$\begin{array}{lll}
1. & C(A) \rightarrow B & : & A, B, \{N_C, C, B\}_{K_{CT}} \\
2. & B \rightarrow C(T) & : & A, B, \{N_C, C, B\}_{K_{CT}}, \{N_B\}_{K_{BT}} \\
2'. & C(B) \rightarrow T & : & C, B, \{N_C, C, B\}_{K_{CT}}, \{N_B\}_{K_{BT}} \\
3'. & T \rightarrow C(B) & : & \{N_C, K_{CB}\}_{K_{CT}}, \{N_B, K_{CB}\}_{K_{BT}} \\
3. & C(T) \rightarrow B & : & \{N_C, K_{CB}\}_{K_{CT}}, \{N_B, K_{CB}\}_{K_{BT}} \\
4. & B \rightarrow A & : & \{N_C, K_{CB}\}_{K_{CT}}
\end{array}$$

   After 2', Trent will reply to Bob. Charlie may intercept it as shown above and forward it to Bob, who will see this as message 3. (Alternatively, with the same effect, Charlie just lets Trent's message go to Bob, who sees it as message 3.) Bob will finally send the first half to Alice. Charlie may intercept, but he already has and can decrypt that message. Bob and Charlie now share K_{CB}, but Bob believes he shares it with Alice. Bob cannot see that the parts he forwards have been encrypted for Charlie rather than Alice.

   (b) If the last part of message 2 includes the names A and B in addition to N_B in the encryption, the attack as described fails.

7. (a) First the ciphertext is parsed as $C_1||C_2$ and C_1 parsed as (Y, z). Then we do Elgamal decryption of C_2, i.e. compute $K = Y^x$, then K^{-1} and finally $z \cdot K^{-1}$. This last value is then the r chosen by the sender. We can now compute the key $H_2(r)$ for the block cipher and decrypt to get m. As the final step we check that the ciphertext is valid by computing $H_1(r||m)$ and checking that $X^{H_1(r||m)} = K$.

   (b) A cipher is CCA2 resistant (resistant against adaptive chosen ciphertext attacks) if an efficient adversary has no more than a negligible advantage over guessing in the following game against Alice:

   - The adversary chooses two ciphertexts m and m' and gives them to Alice.
   - Alice chooses one of these at random and encrypts it, giving c.
   - The adversary chooses a number of ciphertexts (except c) and gets them decrypted by Alice. Finally, the adversary guesses whether m or m' was encrypted.