Security on the Internet, summer 2007

MICHAEL NÜSKEN, DANIEL LOEBENBERGER

10. Exercise sheet
Hand in solutions until Thursday, 21 June 2007.

Any claim needs a proof or an argument.

Exercise 10.1 (A simple linear attack). (4+4 points)

Each variable in the following stores one byte or eight bits. Consider the function 4

$$f(B, C) = (B \Rightarrow C) = BC \oplus B \oplus 1,$$

interpreted bitwise, $K_j = \texttt{0x42}$ for all relevant j, and let (H_1, H_2, H_3) be computed as follows

Algorithm.

Input: A message $(X_0, X_1, X_2, \ldots, X_{n-1})$.
Output: A hash value $H \in \{0,1\}^{3\times 8}$.

1. $(H_1, H_2, H_3) \leftarrow (\texttt{0x31}, \texttt{0x41}, \texttt{0x59})$.
2. For $i = 0..n-1$ do 3–7
3. $\quad (A, B, C) \leftarrow (H_1, H_2, H_3)$.
4. $\quad$ For $j = 0..R-1$ do 5–6
5. $\quad\quad t \leftarrow A \lll 2 + f(B, C) + X_{i+j} + K_j$,
6. $\quad\quad (A, B, C) \leftarrow (t, A, B \lll 3)$.
7. $\quad (H_1, H_2, H_3) \leftarrow (H_1 + A, H_2 + B, H_3 + C)$.
8. Return $H_1|H_2|H_3$.

We consider a message with $n = 1$ and for simplicity we use $R = 1$. Write one of the bits in the output as a function in the input bits in X_0 in the form $f(X_{00}, \ldots, X_{07}) = a_0X_{00} + \cdots + a_7X_{07} + a_8$ where $a_i \in \{0, 1\}$ as good as possible. Can you find coefficients a_i such that f and the chosen output bit coincide in, say 75% of all cases?

Try $R = 3$. +4

Exercise 10.2 (Keyed MAC). (5+2 points)

Along with authenticity (the request that the two peers authenticated reciprocally by using a public key signature mechanism) and privacy (secret key encrypted communication), an important issue in secure communication protocols is the *message integrity*. Assume that Alice and Bob decide that they have no need for encrypted communication (which might also slightly slow down the communication). They definitively wish to maintain the integrity of their communication and ascertain that no third party can interfere and modify the

messages they send to each other. Hashing may be helpful in this respect: if Alice adds the hash value $h(M)$ to a message M sent by her, then Bob can compute the hash of the received message and compare the to the received hash. Due to the collision resistance of hashes, M cannot be *partially* altered. An eavesdropper may however send a totally different message $M'|h(M')$ together with its valid hash: hashing is not sufficient for defending against this kind of attack. However, if Alice and Bob established a common shared secret S at session initialization, having a hash method which uses this secret would protect against eavesdropping. The term of MAC (*message authentication code*) is standard for this idea of combining hashing with a shared secret value.

In this exercise you will discover the current standard HMAC which can be used in combination with various hash methods. You can download the paper *Keying Hash Functions for Message Authentication*, in which a universal MAC function is described, which can be used together with various hash functions.

5 (i) Read the paper and give an algorithmic description of the procedure for creating a HMAC on an input message M (of, say, 1 MB) using the SHA1 hash function and the secret key *secret*.

+2 (ii) In `cryptool`, using the shared secret defined in the previous exercise, generate the HMAC of some text of own choice, by following your own algorithmic description given in (i).

Exercise 10.3 (IPsec in practice). (0+4 points)

+4 Which (common) applications do use/implement IPsec?

Where is it used in our vicinity? (Where within b-it, computer science Bonn, computer science Aachen, University of Bonn, University of Aachen? Which services there do use it?)

Exercise 10.4 (Authentication and Encryption). (4 points)

4 When you want to sent a signed and encrypted message, in which order should the operations be applied? Make a statement and argue.

Compare to IPsec/ESP with both options.