3. Problem Set

Secure Channels

(Winter Term 2014/2015)

Bauhaus-Universität Weimar, Chair of Media Security
Prof. Dr. Stefan Lucks, Jakob Wenzel (jakob.wenzel@uni-weimar.de)
URL: http://www.uni-weimar.de/de/medien/professuren/mediensicherheit/teaching/

Date: 18.11.2014 (1:30 pm) **Room:** SR013, B11

Information: Remember that every student has to present the solution of at least one task in front of the class! If you want to present a certain task, let me know when submitting your solution.

Nonce Generators

- **deterministic**
  - easy (counter, ...)
  - difficult to main the internal state (non-volatile memory)
- **probabilistic**
  - easy: no internal state
  - difficult, as a good random generator is needed

Task 1 (4 Credits) Nonce Generator
A nonce generator NG consist of two functions

`Init()`: Invoked upon the setup phase of NG.

`Next()`: Output is the next nonce value and updates the internal state.

a) **NG** `Mixed 1`**.** This is the combination of the two approaches explained above (deterministic/probabilistic) by concatenating an ℓ_1-bit counter C and an ℓ_2-bit random value R into an ℓ-bit nonce with $\ell = \ell_1 + \ell_2$. Each time `Next()` is called, C is increased by one and R is set to a new l_2-bit random value. The function `Init()` resets both parameters C and R to zero. Specify the collision probability (failure probability) for q invocations of `Next()` and come up with a reasonable upper bound for q.

b) **NG** `Mixed 2`**.** This NG is a variation of `Mixed 1` from a) were an ℓ-bit nonce is also derived by the concatenation of an ℓ_1-bit counter C and an ℓ_2-bit random value R. Each invocation of `Next()` increases C by one (but does **not** update R). The function `Init()` resets C to zero and R to a new random value. Specify the collision probability (failure probability) for q invocations of `Next()` and come up with a reasonable upper bound for q.

Task 2 (6 Credits) Padding Oracle
Lets consider the following CBC-Padding scheme $MMM \ldots M0RR \ldots R$ where M denotes a message byte of the last message block, 0 denotes a sequence of eight zero-bits, and R a uniformly at random chosen byte value. You can assume that a message to encrypt is an ANSI-C string, i.e., a message byte can never be equal 0 ($M \neq 0$). Come up with a Padding-Oracle Attack against the $MMM \ldots M0RR \ldots R$ padding scheme.

Task 3 (8 Credits) Fun with the CBC Encryption Scheme
Analyze the RoR-CPA security against nonce-respecting adversaries (Slide 52) of the CBC encryption scheme with $C_0 := E_L(N)$, where the underlying block cipher E has the following key independent property.

a) $E_K(0) = 0$.

b) There exists a well known fix point X with $E_K(X) = X$.

c) $\forall X : E_K(E_K(X)) = X$.

d) $\forall X : \overline{E_K(X)} = E_K(\overline{X})$ where $\overline{X}$ is the bitwise complement of X, i.e., $\overline{X} = X \oplus 1, \ldots, 1$

Task 4 (3 Credits) Secure Shell
Consider the following simplified representation of the Secure Shell (SSH) protocol between a client (C) and a server (S):

1. C requests for a connection to S.
2. S answers with its public key K_p.
3. C picks a session key K_{se} and sends $y = E_{K_p}(K_{se})$ to S, where E denotes a secure block cipher.
4. S decrypts y using K_p.

The session key K_{se} is then used to encrypt the communication between C and S.

a) Why it is advisable to secure the session with symmetric encryption instead of asymmetric encryption?

b) If the first connection between a server S and a client C is not authenticated, explain how an adversary can impersonate S.

Task 5 (4 Credits) Message Authentication Code
Let MAC be a secure and deterministic MAC algorithm and let MAC' be defined as:

$$A = \text{MAC'}_K(M) = \text{MAC}_K(M) \mid\mid 1,$$

where M is a $b \cdot n$-bit message $M = (m_1, \ldots, m_n)$, $m_i \in \{0,1\}^b$ with $1 \leq i \leq n$. The verification algorithm of a pair $(M, (A \mid\mid x))$ works as follows:

- If $x = 1$: authentic $\iff \text{MAC}(M) = A$.
- If $x = 0$: authentic $\iff \text{MAC}(M) = A$ and $m_1 \oplus m_2 \oplus \cdots \oplus m_n = 0$.

Show that MAC' is a secure MAC if MAC is a secure MAC.