Cryptology I (spring 2015) Dominique Unruh

Exercise Sheet 4

Out: 2015-04-11 Due: 2015-04-22

Problem 1: Hash functions

Let E be a block cipher with key and block length n. Let $F(x\|y) := E(x, y)$ (the compression function). Let H be the Merkle-Damgård construction using F as compression function.

Show how to find a collision for F. Show how to find a collision for H.

Note: If finding a collision for H is too difficult, you might first try to find a collision for H when H is constructed using the Iterated Hash construction. (This will give points, too.) In this case, however, the colliding messages have to have the same length.

Problem 2: MACs and encryption

Consider the following symmetric encryption scheme (KG, E, D). KG chooses an AES key. $E(k, m) := E_{AES}(k, m)\|0^{32}$. ($0^{32}$ stands for a string consisting of 32 zeros.) And the decryption $D(k, c)$ does the following: Let $c'\|p := c$ where p has length 32 bit and c' is all but the last 32 bits of c. $m := D_{AES}(k, c')$. If $p = 0^{32}$, then $D(k, c)$ returns m. If $p \neq 0^{32}$ and $k_p = 0$ (here k_p is the p-th bit of the key k), then $D(k, c)$ returns m. If $p \neq 0^{32}$ and $k_p = 1$, then $D(k, c)$ aborts.

(a) Show that (KG, E, D) can be totally broken using a chosen ciphertext attack.[1] That is, show that it is possible to recover the key k using a chosen ciphertext attack.

(b) To avoid the issue, we try to use authentication: Let MAC be an EF-CMA secure MAC. We construct a new encryption scheme E'. The key of this scheme consists of an AES key k_1 and a MAC-key k_2. Encryption is as follows: $E'(k_1k_2, m) := E(k_1, (MAC(k_2, m), m))$. Decryption D' checks the tag $MAC(k_2, m)$ and aborts if it is incorrect.[2] (This is called MAC-then-encrypt.)

Does E' withstand chosen ciphertext attacks that reveal the whole key k_1? If yes, explain why (without proof). If no, how to attack?

(c) We try to use authentication in another way: Let MAC be an EF-CMA secure MAC. We construct a new encryption scheme E''. The key of this scheme consists of an AES

[1] In a chosen ciphertext attack, the adversary is also allowed to submit plaintexts for encryption, not only ciphertexts for decryption.

[2] We assume that you cannot distinguish between an abort due to a wrong tag or an abort of the underlying algorithm D.

key k_1 and a MAC-key k_2. Encryption is as follows: $E''(k_1k_2, m) := MAC(k_2, c)\|c$ with $c := E(k_1, m)$. Decryption D' checks the tag $MAC(k_2, c)$ and aborts if it is incorrect.[3] (This is called encrypt-then-MAC.)

Does E'' withstand chosen ciphertext attacks that reveal the whole key k_1? If yes, explain why (without proof). If no, how to attack?

Hint: One of (b), (c) is secure, the other is insecure.

Problem 3: Authentication in WEP

In the WEP-protocol (used for securing Wifi, now mostly replaced by WPA), messages are "encrypted" using the following procedure: First, a key k is established between the parties A and B. (We do not care how, for the purpose of this exercise we assume that this is done securely.) Then, to transmit a message m, A chooses an initialization vector IV (we do not care how) and sends IV and $c := \mathit{keystream} \oplus (m\|\mathit{CRC}(m))$. Here *keystream* is the RC4 keystream computed from IV and k (we do not care how).

The function CRC is a so-called cyclic redundancy check, a checksum added to the WEP protocol to ensure integrity. We only give the important facts about CRC and omit a full description. Each bit of $\mathit{CRC}(m)$ is the XOR of some of the message bits. Which messages bits are XORed into which bit of $\mathit{CRC}(m)$ is publicly known. (In other words, the i-th bit of $\mathit{CRC}(m)$ is $\bigoplus_{j\in I_i} m_i$ for a publicly known I_i.)

An adversary intercepts the ciphertext c. He wishes to flip certain bits of the message (i.e., he wants to replace m by $m \oplus p$ for some fixed p). This can be done by flipping the corresponding bits of the ciphertext c. But then, the CRC will be incorrect, and B will reject the message after decryption! Thus the CRC *seems* to ensure integrity of the message and to avoid malleability. (This is probably why the designers of WEP added it here.)

Show that the CRC does not increase the security! That is, show how the adversary can modify the ciphertext c such that c becomes an encryption of $m \oplus p$ and such that the CRC within c is still valid (i.e., it becomes the CRC for $m \oplus p$).

Hint: Think of how the i-th bit of $\mathit{CRC}(m \oplus p)$ relates to the i-th bit of $\mathit{CRC}(m)$. (Linearity!)

[3] We assume that you cannot distinguish between an abort due to a wrong tag or an abort of the underlying algorithm D.