LASEC
SECURITY AND CRYPTOGRAPHY LABORATORY

EPFL / I & C
CH-1015 Lausanne
Phone: ++41 (0) 21 693 76 03
Fax: ++41 (0) 21 693 68 79
URL: http://lasecwww.epfl.ch

Solution of the Midterm (Crypto Part)

1. Two famous hash function based on the Merkle-Damgård scheme are MD5 and SHA-1. In both cases, the block length ℓ is equal to 512 bits. The length n of the hashed value is 128 bits in the case of MD5 and 160 bits in the case of SHA-1.

2. It should be hard (computationally speaking) to find collisions on a cryptographic hash function. If h denotes some cryptographic hash function, this means that is should be hard to find two *distinct* messages m and m' such that $h(m) = h(m')$. According to the birthday paradox, there exists a generic attack which complexity (i.e., the number of hash computations) is close to $2^{n/2}$. Assuming that 2^{60} hash computations can indeed be performed in a "reasonable" time, we conclude that the hash length n cannot be lower than 120 bits. Keeping a (too small) security margin, we see that the hash length cannot be less that 128 bits.

3. The solution is given in Algorithm 1. Instead of outputting the correct tag, this algorithm could also output the correct key.

Algorithm 1 Forging a MAC by an exhaustive search on the key

Input: a message M
Output: the valid MAC of M under the key k
Processing:

```
for all k ∈ {0,1}^64 do
    c ← MAC_k(M)
    if Oracle(c) is valid then
        display c and exit
    end if
end for
```

4. Algorithm 2 is very similar to the previous one, except that this time the exhaustive search is performed on the MAC value itself, not on the

key. This time, it is not possible to recover the key, but only a valid MAC for a given message.

Algorithm 2 Forging a MAC by an exhaustive search on the MAC value

Input: a message M
Output: the valid MAC of M under the key k
Processing:

```
for all c ∈ {0,1}^64 do
    if Oracle(c) is valid then
        display c and exit
    end if
end for
```

5. We denote $|k|$ and $|c|$ the respective bit length of the key k and of the output of the MAC c. From the previous questions, we can deduce that there always is a generic attack against a MAC scheme in a complexity in the order of $2^{\min(|k|,|c|)}$. Consequently, it is not useful to have a key longer than the MAC output size. Considering that about 2^{60} MAC computations can be performed in a "reasonable" time, both the key and the MAC output size should be larger than 60 bits. We a small security margin, this gives a minimum size of 64 bits.

6. We denote by $M = M_1 \| M_2 \| \cdots \| M_N$ the ℓ bit blocks of the message that are successively processed by the reduction function f in the Merkle-Damgård scheme, where M_N corresponds to the last bits of the message concatenated with the padding[1]. We denote by $c = \mathrm{MAC}_k(M) = h(\widetilde{k} \| M)$ the MAC of M. Figure 1 represents the MAC computation of the original message M.

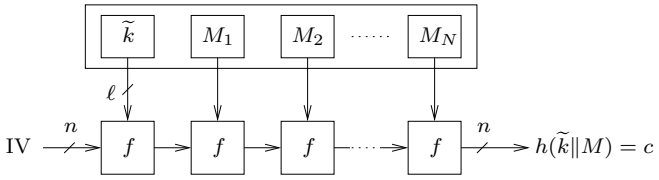

Figure 1: Computing $\mathrm{MAC}_k(M) = h(\widetilde{k} \| M)$

Given M and c, the adversary can easily forge a MAC for any message M' such that $M' = M_1 \| M_2 \| \cdots \| M_N \| B$, where B is any ℓ bit block (including the padding). This is represented on Figure 2. Clearly, using c, the adversary does not need the key to compute the valid MAC of the message M', as $c' = \mathrm{MAC}_k(M') = f(c, B)$.

[1] In case the message length is a multiple of ℓ, M_N only corresponds to the padding.

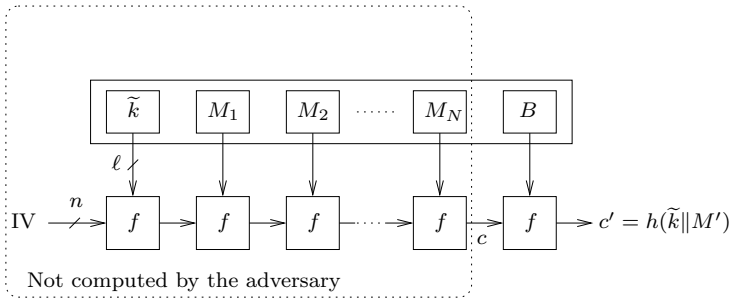

Figure 2: Forging $\mathrm{MAC}_k(M') = h(\widetilde{k}\|M')$

7. To simplify the notations, we will consider that all the messages sent by Alice are exactly ℓ bits long, that is, they all have the length of one block. The adversary can store the $2^{n/2}$ messages and their MAC in a table and then try to find another block M' such that $f(IV, B) = f(IV, M_i)$ for some M_i sent by Alice. According to the birthday paradox, this can be done with a complexity close to $2^{n/2}$. Obviously, for such a B with have $\mathrm{MAC}_k(B) = \mathrm{MAC}_k(M_i)$ as shown on Figure 3.

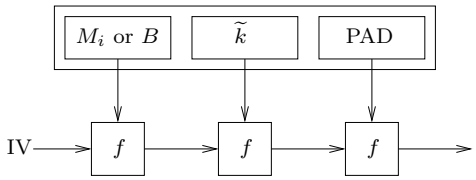

Figure 3: Forging $\mathrm{MAC}_k(M') = h(M'\|\widetilde{k})$

Note that the attack can still be performed with messages of arbitrary length (for example by applying the same technique on the first block of the messages).

8. One possibility is to use a construction that uses both previous ones. More precisely, we define

$$\mathrm{MAC}_k(M) = h(\widetilde{k}\|M\|k)$$

using the previous notations.

The attack proposed on the first construction does not apply as a previous MAC value cannot be used directly as the input of the compression function to compute a new MAC value. In this case the key k is needed in the last block.

The attack proposed in the second construction does not apply either as the collision search could not be performed by the adversary.

Indeed, (assuming once again that M is of length ℓ), the adversary would have to look for two blocks B, B' such that $f(f(IV, B), B') = f(f(IV, \widetilde{k}), M)$ which cannot be done as the computation of $f(IV, \widetilde{k})$ is required.