Applied Cryptography

Symmetric Cryptography, Assignment 1, Tuesday, September 10, 2024

Exercises with answers and grading.

1. **(15 points)** A well-known message authentication code not discussed in the lectures is LightMAC. It is defined on top of AES-128 instantiated with two independent and random secret keys K_1, K_2, which we denote as E_{K_1} and E_{K_2}. For $a \in \mathbb{N}$ and $X \in \{0, \ldots, 2^a - 1\}$, $\langle X \rangle_a$ denotes the encoding of X as an a-bit string. We will consider a simplified version of LightMAC (hint: make a drawing of the scheme!):

   - On input of an arbitrary-length message $M \in \{0,1\}^*$, it is padded with a 1 and a sufficient number of 0s so that its length is a positive multiple of 96:

     $$X' \leftarrow M \| 1 \| 0^{-|M|-1 \bmod 96}.$$

     This string is then split into 96-bit blocks $X_1, \ldots, X_\ell$.
   - The ℓ message blocks are concatenated with a counter, and encrypted using E_{K_1}, and the outcome is added into a checksum:

     $$V \leftarrow \bigoplus_{i=1}^{\ell} E_{K_1}(X_i \| \langle i \rangle_{32}).$$

   - The result is encrypted using E_{K_2}, and this gives the tag:

     $$T \leftarrow E_{K_2}(V).$$

   (a) **(3pt)** What is the key size of LightMAC?

   (b) **(4pt)** Is this construction a Wegman-Carter MAC or a Protected Hash MAC?

   Suppose you find two messages M, M' *of the same length* such that $\text{LightMAC}(M) = \text{LightMAC}(M')$.

   (c) **(3pt)** What can you conclude about the corresponding intermediate values V and V'?

   (d) **(5pt)** Suppose you *additionally* know that, for some known arbitrary-length message X, $\text{LightMAC}(M \| 1 \| 0^{-|M|-1 \bmod 96} \| X) = T^\star$. Describe a forgery for LightMAC using this additional knowledge, and explain why this forgery is valid.

   Begin Secret Info: .....

   (a) *The key is made of K_1 and K_2, each of 128, so the total size is 256.*

   (b) *Protected hash MAC, since T is computed by the encryption of V.*

   (c) *Taking $T = \text{LightMAC}(M), T' = \text{LightMAC}(M')$, then $T = T'$ is the same as $E_{K_2}(V) = E_{K_2}(V')$. By taking $E_{K_2}^{-1}$ on both sides, we conclude that $V = V'$.*

   (d) *Then also $\text{LightMAC}(M' \| 1 \| 0^{-|M'|-1 \bmod 96} \| X) = T^\star$ is valid. This is because $V = V'$.*

   End Secret Info .....

2. **(20 points)** In the `PyCryptodome` package, you can find an implementation for `AES-128-ECB`. `AES-128-ECB` applied to a 128-bit input is just the block cipher `AES-128`. Different *modes* can be built using this block cipher, such as `OFB`.

(a) **(8pt)** Implement `AES-128-OFB` and its inverse using the implementation for `AES-128-ECB`. Remember that in `OFB` mode, the initial value (IV) needs to be random. In particular, your implementation of `AES-128-OFB` needs to take as input a 128-bit value as an IV, and a plaintext which can be of any length. Encrypt a plaintext with an IV and key of your choice, and decrypt the ciphertext again to verify your implementation. The plaintext must be at least 512 bits. Note that the decryption function also takes IV as an input.

(b) **(4pt)** Argue whether `AES-128-OFB` or `AES-128-OFB`$^{-1}$ can be implemented in parallel.

(c) (**8pt**) Note that in `OFB` mode, it is necessary to use a random IV. Suppose we replace this random IV with a user-chosen nonce N. Explain that this makes the `OFB` mode insecure.

Begin Secret Info:.....
See `implementation_exercise_2.py` for the script we wrote for the exercise.

(a) Implementing `OFB` given the block cipher `AES`$_K$ is straightforward, simply follow the diagram from lecture 1 slide 38. From which we conclude:

$$\begin{aligned} C_1 &= M_1 \oplus E_K(IV)\,, \\ C_2 &= M_2 \oplus E_K^2(IV)\,, \\ &\vdots \\ C_n &= M_n \oplus E_K^n(IV)\,. \end{aligned}$$

Where $E_K^\ell = E_K \overset{\ell}{\cdots} E_K(M)$ denotes the iterated application of E_K.
To implement the inverse note that `AES-128-OFB` = `AES-128-OFB`$^{-1}$.

(b) Neither `AES-128-OFB` nor `AES-128-OFB`$^{-1}$ can be implemented in parallel, because it requires the iterated computation of $E_K^n(IV)$.

(c) Choose any nonce N and any 256 bit message $M_1 \| M_2$. Then:

$$\begin{aligned} C_1 &= M_1 \oplus E_K(N)\,, \\ C_2 &= M_2 \oplus E_K^2(N)\,. \end{aligned}$$

Now, choose a new nonce $N' = C_2 \oplus M_2 = E_K(N)$ and encrypt the 128 bit message M_2, getting:

$$C_1' = M_2 \oplus E_K^2(N)\,.$$

Hence, $C_1' = C_2$, which breaks prf security.

End Secret Info.....

3. **(15 points)** Consider the Wegman-Carter MAC function of lecture 2 slide 14. We have

$$\mathbf{Adv}^{\mathrm{unf}}_{\mathsf{WC}}(q_m, q_v) \leq q_v/2^n + \mathbf{Adv}^{\mathrm{prf}}_F(q_m + q_v)\,,$$

provided that the adversary does not query WC_K for repeated nonces.

(a) **(5pt)** Assume you can evaluate this function for repeated nonces. Mount a forgery attack in $q_m = 3$ MAC queries and $q_v = 1$ VFY query.

(b) **(5pt)** Consider the function $H : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ defined as $H_L(M) = L \otimes M$, i.e., defined as finite-field multiplication over GF(2^n). Prove that H_L is 2^{-n}-XOR-universal.

Hint: Over finite fields, multiplications can be inverted the same way that it can in the field of real numbers. I.e., the equation $A \otimes X = B$ can be solved for X, so long as $A \neq 0$.

(c) **(5pt)** Now, we consider the same setting as in (3a), except that we take the hash function to be the finite-field multiplication of (3b). Describe a forgery attack in only $q_m = 2$ MAC queries and $q_v = 1$ VFY query.

Begin Secret Info: .....

(a) Make the following three MAC queries, for arbitrary M, M', N, N':

$$\begin{aligned}(N, M) &\mapsto T\\ (N, M') &\mapsto T'\\ (N', M) &\mapsto T''\,.\end{aligned}$$

This yields:

$$H_L(M) \oplus F_K(N) \mapsto T \tag{1}$$
$$H_L(M') \oplus F_K(N) \mapsto T' \tag{2}$$
$$H_L(M) \oplus F_K(N') \mapsto T''\,. \tag{3}$$

Xoring equations (1), (2, (3) yields:

$$H_L(M') \oplus F_K(N') \mapsto T \oplus T' \oplus T''\,.$$

Hence, $\mathsf{WC}_K(N', M') = T \oplus T' \oplus T''$.

(b) Fix any $M \neq M'$ and T:

$$\begin{aligned}\mathbf{Pr}_L\big[H_L(M) \oplus H_L(M') = T\big] &= \mathbf{Pr}_L\big[L \otimes (M \oplus M') = T\big]\\ &= \mathbf{Pr}_L\big[L = T \otimes (M \oplus M')^{-1}\big]\\ &= 1/2^n\,.\end{aligned}$$

(c) Take two messages M, M', and we re-use the nonce N, we then have:

$$\big(M \otimes L\big) \oplus F_K(N) = T\,, \tag{4}$$
$$\big(M' \otimes L\big) \oplus F_K(N) = T'\,. \tag{5}$$

By xoring equations (4) and (5) we get:

$$\begin{aligned}&\big(M \otimes L\big) \oplus \big(M' \otimes L\big) = T \oplus T'\,,\\ &L \otimes (M \oplus M') = T \oplus T'\,,\\ &L = (T \oplus T') \otimes (M \oplus M')^{-1}\,.\end{aligned}$$

Hence, we have recovered L. Then we can compute $M \otimes L$ and recover $F_K(N)$ from 4, since $F_K(N) = \big(M \otimes L\big) \oplus T$.

Finally, take any message M'', we can compute its tag $T'' = \big(M'' \otimes L\big) \oplus F_K(N)$, which is a forgery.

End Secret Info .....