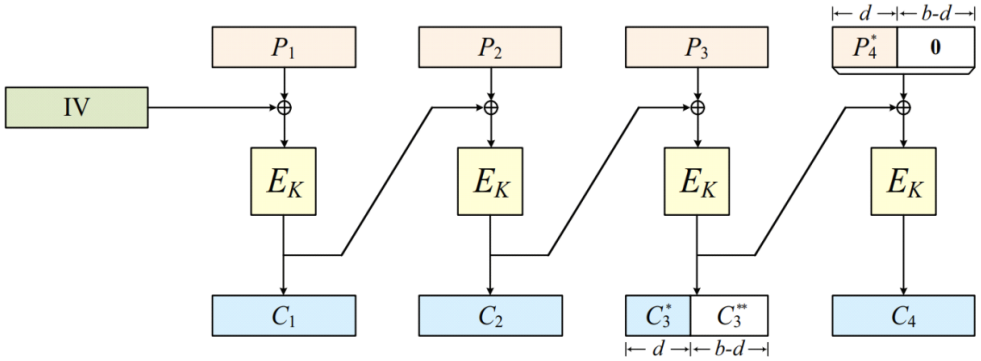
Homework 4

David Miller & Hannah McLaughlin
CIS 5371: Cryptography

October 22, 2018

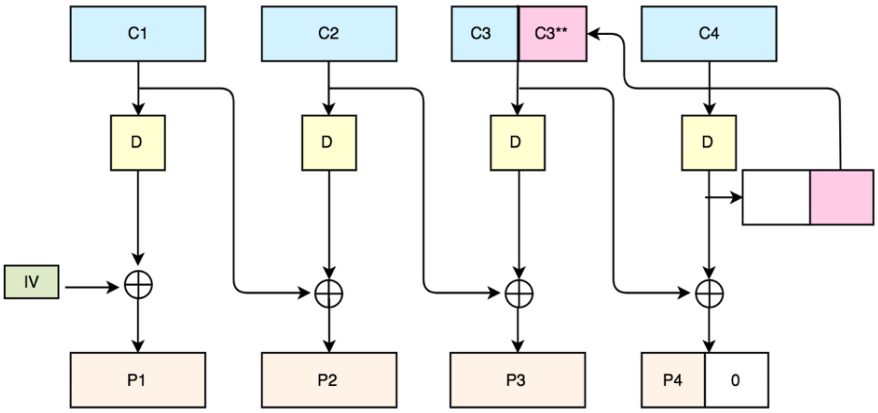
Problem 1. *(CBC with ciphertext stealing) One problem with CBC encryption is that messages need to be padded to a multiple of the block length and sometimes a dummy block needs to be added.*



The method pads the last block with zeros if needed (a dummy block is never added), but the output ciphertext contains only the shaded parts of C_1, C_2, C_3, C_4. Note that, ignoring the IV, the ciphertext is the same length as the plaintext. This technique is called ciphertext stealing.

(a) Explain how decryption works.

CBC ciphertext stealing (CBC-CS) decryption works normally until the fragmented block C_3. C_4 is decrypted before C_3 and then the last few bits of the block are appended to C_3 to make a full block. C_3 can then be decrypted normally and XOR with $E_K^{-1}(C_4)$ The result, P_4, will still have zeros appended to the end of it to make a full block.



(b) Can this method be used if the plaintext contains only one block?

Yes, if the plaintext is one block long or less (a fragmented block), the IV can act as the prior block of ciphertext.

Problem 2. (A wrong way to extend the CBC MAC) *Consider the following variant of the CBC-MAC, intended to allow one to MAC messages of arbitrary length. The construction uses a blockcipher* $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ *which you should assume to be a good PRF. The domain for the MAC is* $(\{0,1\}^n)^+$ *(namely the set of non-empty binary strings whose length is a multiple of n). To MAC a message M under key* $K_1 \| K_2$*, (with* $|K1| = |K2| = n$*), first compute the ordinary CBC-MAC of M, keyed by* K_1*, and then xor into the result the key* K_2*. Show that this MAC is completely insecure: break it (getting advantage of about 1) by a simple adversary that asks a constant number of queries.*

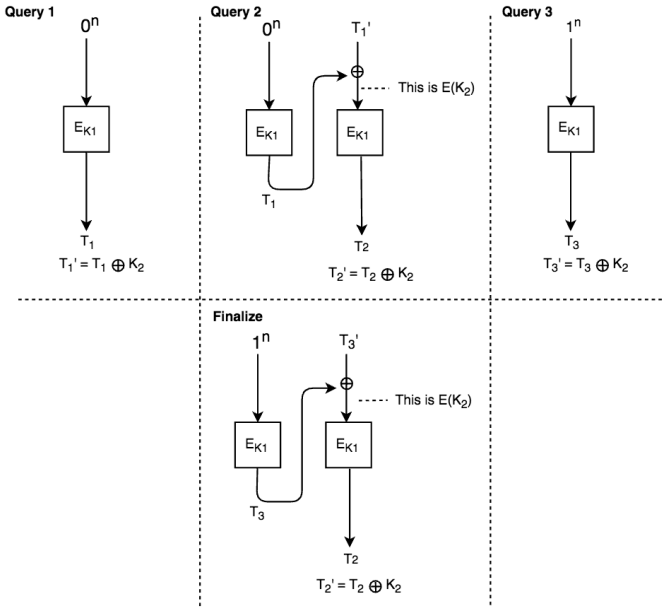
To prove that this extension of CBC-MAC is completely insecure, we will show and attack with advantage 1. First, we show the UF-CMA game.

Game UF-CMA$_T$
Let T : Keys $\times D \to R$ be a message authentication code. Let A be an adversary.

procedure Initialize
$K \xleftarrow{\$} K$; S $\leftarrow \emptyset$
procedure Tag(M)
$T \leftarrow T_K(M); S \leftarrow S \cup \{M\}$
return T
procedure Finalize(M, T)
If $M \in S$, then Return false
If $M \notin D$, then Return false
Return $T = T_K(M)$

In the diagram, Adversary A makes three queries to the oracle: (O^n, $O^n T_1'$, and 1^n). Once Adversary A recognizes that in the second block, K_2 is encrypted independent of the contents of the first block, A can call Finalize($1^n T_3'$, T_2').This attack is done in constant time, since it is done in 3 queries. Therefore we have

$$\mathbf{Adv}_T^{uf-cma}(A) = \Pr[\text{UFCMA}_T^A \Rightarrow \text{true}] = 1$$



Problem 3. (Another way to build AU hash) *Let $f : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a hash function such that for any strings $x, y, z \in \{0,1\}^n$, if $x \neq y$, then*

$$\Pr_{K \xleftarrow{\$} \mathcal{K}}[f_K(x) \oplus f_K(y) = z] \leq c/2^n,$$

where c is a small constant. (For example, f can be realized using some reduced-round version of AES, and in that case $c = 2^{15}$.) Let $\ell \geq 1$ be an integer, and let $N = 2^\ell n$. Our goal is to construct an almost-universal hash function $H : \{0,1\}^{k\ell} \times \{0,1\}^N \to \{0,1\}^n$ using f.

(a) First consider the case that $\ell = 1$. Consider the following construction $h : \{0,1\}^k \times \{0,1\}^{2n} \to \{0,1\}^n$. First, parse $x = x_1 \| x_2$, where $|x_1| = |x_2| = n$. Then $h_K(x) = f_K(x_1) \oplus x_2$. Prove that this h is indeed an almost-universal hash function.

To prove that $h_k(x)$ is indeed an almost-universal hash function, we must show that

$$\Pr_{K \xleftarrow{\$} \mathcal{K}}[h_K(x) = h_K(y)] \leq \varepsilon$$

where $x \neq y$ and $\epsilon > 0$. Let $x, y \in \{0,1\}^{2n}$, such that $x = x_1 || x_2$, where $|x_1| = |x_2| = n$, and $y = y_1 || y_2$, where $|y_1| = |y_2| = n$, and also let $z \in \{0,1\}^n$. Expanding the above we get

$$\begin{aligned}
\Pr_{K \xleftarrow{\$} \mathcal{K}}[h_K(x) = h_K(y)] &= \Pr_{K \xleftarrow{\$} \mathcal{K}}[f_K(x_1) \oplus x_2 = f_K(y_1) \oplus y_2] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[(f_K(x_1) \oplus x_2) \oplus (f_K(y_1) \oplus y_2) = 0] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[(f_K(x_1) \oplus f_K(y_1)) \oplus (x_2 \oplus y_2) = 0] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[f_K(x_1) \oplus f_K(y_1) = z \wedge x_2 \oplus y_2 = z] \\
&= \sum_{i=1}^{2^n} \Pr_{K \xleftarrow{\$} \mathcal{K}}[f_K(x_1) \oplus f_K(y_1) = z] \times \Pr_{K \xleftarrow{\$} \mathcal{K}}[x_2 \oplus y_2 = z] \\
&\leq \frac{c}{2^n} \sum_{i=1}^{2^n} \Pr_{K \xleftarrow{\$} \mathcal{K}}[x_2 \oplus y_2 = z] \\
&= \frac{c}{2^n} \sum_{i=1}^{2^n} \frac{1}{2^n} \\
&= \frac{c}{2^n}
\end{aligned}$$

where $0 < c/2^n \ll 1$. Therefore $h_K(x)$ is an almost-universal hash function.

(b) Next consider the case that $\ell = 2$. Design an almost-universal hash $h' : \{0,1\}^{2k} \times \{0,1\}^{4n} \to \{0,1\}^n$.

For any $x \in \{0,1\}^{rn}$, lets define $x \equiv x_{1:r}$ as $x_1||\ldots||x_r$ such that $|x_1| = \ldots\ |x_r| = n$ for $r \in \mathbb{Z}_{>0}$. Similarly, we can define sub-strings in $x_{1:r}$ as $x_{i:j} = x_i||\ldots||x_j$. Now let

$$h'_{K_1,K_2}(x) = f_{K_2}(h_{K_1}(x_{1:2})) \oplus h_{K_1}(x_{3:4})$$

where $h_{K_1}(x)$ is the hash function from part (a). This can be shown to be an almost-universal hash function the same way we showed $h_k(x)$ in part (a) to be an almost-universal hash function. Let $x, y \in \{0,1\}^{4n}$ where $x \neq y$ and $z \in \{0,1\}^n$, then

$$\begin{aligned}
\Pr_{K \xleftarrow{\$} \mathcal{K}}[h'_{K_1,K_2}(x) = h'_{K_1,K_2}(y)] &= \Pr_{K \xleftarrow{\$} \mathcal{K}}[f_{K_2}(h_{K_1}(x_{1:2})) \oplus h_{K_1}(x_{3:4}) = f_{K_2}(h_{K_1}(y_{1:2})) \oplus h_{K_1}(y_{3:4})] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[(f_{K_2}(h_{K_1}(x_{1:2})) \oplus h_{K_1}(x_{3:4})) \oplus (f_{K_2}(h_{K_1}(y_{1:2})) \oplus h_{K_1}(y_{3:4})) = 0] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[(f_{K_2}(h_{K_1}(x_{1:2})) \oplus f_{K_2}(h_{K_1}(y_{1:2}))) \oplus (h_{K_1}(x_{3:4}) \oplus h_{K_1}(y_{3:4})) = 0] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[f_{K_2}(h_{K_1}(x_{1:2})) \oplus f_{K_2}(h_{K_1}(y_{1:2})) = z \wedge h_{K_1}(x_{3:4}) \oplus h_{K_1}(y_{3:4}) = z] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[f_{K_2}(h_{K_1}(x_{1:2})) \oplus f_{K_2}(h_{K_1}(y_{1:2})) = z] \\
&\quad \times \Pr_{K \xleftarrow{\$} \mathcal{K}}[h_{K_1}(x_{3:4}) \oplus h_{K_1}(y_{3:4}) = z] \\
&\leq \frac{c}{2^n} \sum_{i=1}^{2^n} \Pr_{K \xleftarrow{\$} \mathcal{K}}[h_{K_1}(x_{3:4}) \oplus h_{K_1}(y_{3:4}) = z] \\
&\leq \frac{c}{2^n} \sum_{i=1}^{2^n} \frac{c}{2^n} \\
&= \frac{c^2}{2^n}
\end{aligned}$$

where $0 < c^2/2^n \ll 1$. Therefore $h'_{K_1,K_2}(x)$ is an almost-universal hash function.

(c) Generalize the method above for the case $\ell = 3$.

Let

$$h''_{K_1,K_2,K_3}(x) = f_{K_3}(h'_{K_1,K_2}(x_{1:4})) \oplus h'_{K_1,K_2}(x_{5:8})$$

where $h'_{K_1,K_2}(x)$ is the hash function from part (b). We will follow suit from the previous parts to show $h''_{K_1,K_2,K_3}(x)$ is an almost-universal hash function. Let $x, y \in \{0,1\}^{8n}$ where $x \neq y$ and $z \in \{0,1\}^n$, then

$$\begin{aligned}
\Pr_{K \xleftarrow{\$} \mathcal{K}}[h''_{K_1,K_2,K_3}(x) = h''_{K_1,K_2,K_3}(y)] &= \Pr_{K \xleftarrow{\$} \mathcal{K}}[f_{K_3}(h'_{K_1,K_2}(x_{1:4})) \oplus h'_{K_1,K_2}(x_{5:8}) \\
&\quad = f_{K_3}(h'_{K_1,K_2}(y_{1:4})) \oplus h'_{K_1,K_2}(y_{5:8})] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[(f_{K_3}(h'_{K_1,K_2}(x_{1:4})) \oplus h'_{K_1,K_2}(x_{5:8})) \\
&\quad \oplus (f_{K_3}(h'_{K_1,K_2}(y_{1:4})) \oplus h'_{K_1,K_2}(y_{5:8})) = 0] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[(f_{K_3}(h'_{K_1,K_2}(x_{1:4})) \oplus f_{K_3}(h'_{K_1,K_2}(y_{1:4}))) \\
&\quad \oplus (h'_{K_1,K_2}(x_{5:8}) \oplus h'_{K_1,K_2}(y_{5:8})) = 0] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[f_{K_3}(h'_{K_1,K_2}(x_{1:4})) \oplus f_{K_3}(h'_{K_1,K_2}(y_{1:4})) = z \\
&\quad \wedge h'_{K_1,K_2}(x_{5:8}) \oplus h'_{K_1,K_2}(y_{5:8}) = z] \\
&= \Pr_{K \xleftarrow{\$} \mathcal{K}}[f_{K_3}(h'_{K_1,K_2}(x_{1:4})) \oplus f_{K_3}(h'_{K_1,K_2}(y_{1:4})) = z] \\
&\quad \times \Pr_{K \xleftarrow{\$} \mathcal{K}}[h'_{K_1,K_2}(x_{5:8}) \oplus h'_{K_1,K_2}(y_{5:8}) = z] \\
&\leq \frac{c}{2^n} \sum_{i=1}^{2^n} \Pr_{K \xleftarrow{\$} \mathcal{K}}[h'_{K_1,K_2}(x_{5:8}) \oplus h'_{K_1,K_2}(y_{5:8}) = z] \\
&\leq \frac{c}{2^n} \sum_{i=1}^{2^n} \frac{c^2}{2^n} \\
&= \frac{c^3}{2^n}
\end{aligned}$$

where $0 < c^3/2^n \ll 1$. Therefore $h''_{K_1,K_2,K_3}(x)$ is an almost-universal hash function.

The main idea behind building an almost-universal hash function via the construction in parts (a) - (c) for some h_ℓ is using the previous hash function for $h_{\ell'}$. This is because we can split in half the longer strings for h_ℓ and parse them via $h_{\ell'}$. Let $h_\ell : \{0,1\}^{\ell k} \times \{0,1\}^{\ell n} \to \{0,1\}^n$ be an almost-universal hash function. Via the construction in parts (a) - (c) we can recursively define h_ℓ as

$$h_\ell(x) = \begin{cases} f_{K_1}(x) \oplus x, & \ell = 1 \\ f_{K_\ell}(h_{\ell-1}(x_{1:\ell n/2})) \oplus h_{\ell-1}(x_{\ell n/2+1:\ell n}), & \ell \geq 2 \end{cases}$$