

Introduction to Cryptography

TEK 4500 (Fall 2024)

Problem Set 4

Problem 1.

Read Chapter 7 in [BR] (Section 7.8 can be skipped, as can the proof of Theorem 7.5).

Problem 2.

The hex-string

b393d9ab1410d9c73660a1b18b18e56e0b60d664cb547f31

represents a ciphertext of the ASCII-encoded message $M = \text{"Transfer : \$000010 to Bob"}$, encrypted using the OTP. Modify the ciphertext so that it decrypts to \$123456.

Note: For reference, the above ciphertext was created using the following Python 3 code.

```
import secrets

m = bytes("Transfer: $000010 to Bob", 'ascii')
k = secrets.token_bytes(len(m))
c = bytes([a ^ b for (a,b) in zip(m,k)])
print(c.hex())
```

Problem 3. [Problem 10.1 in [Ros]]

Consider the following MAC scheme, where $F: \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a secure PRF.

$\Sigma.$ KeyGen:	$\Sigma.$ Tag($K, M_1 \parallel \dots \parallel M_\ell$): // each M_i is n bits
1: $K \xleftarrow{\$} \{0, 1\}^k$	1: $M \leftarrow 0^n$
2: return K	2: for $i = 1, \dots, \ell$ do
	3: $M \leftarrow M \oplus M_i$
	4: return $F(K, M)$

Show that the scheme is *not* a secure MAC. Describe an adversary and compute its UF-CMA advantage (see Fig. 1 for the formal definition).

$\text{Exp}_{\Sigma}^{\text{uf-cma}}(\mathcal{A})$ 1: $K \xleftarrow{\$} \Sigma.\text{KeyGen}$ 2: $\text{won} \leftarrow 0$ 3: $S \leftarrow []$ 4: $\mathcal{A}^{\text{TAG}(\cdot), \text{VF}(\cdot)}$ 5: return won	$\text{TAG}(M)$: 1: $S.\text{add}(M)$ 2: $T \leftarrow \Sigma.\text{Tag}(K, M)$ 3: return T $\text{VF}(M, T)$: 1: $d \leftarrow \Sigma.\text{Vrfy}(K, M, T)$ 2: if $d = 1$ and $M \notin S$ then 3: $\text{won} \leftarrow 1$ 4: return d
---	--

$\text{Adv}_{\Sigma}^{\text{uf-cma}}(\mathcal{A}) = \Pr[\text{Exp}_{\Sigma}^{\text{uf-cma}}(\mathcal{A}) \Rightarrow 1]$

Figure 1: UF-CMA security experiment and UF-CMA-advantage definition.

Problem 4. [Problem 10.3 in [Ros]]

Suppose $\text{MAC} = (\text{Tag}, \text{Vrfy})$ is a secure MAC algorithm. Create a new MAC algorithm $\text{MAC}' = (\text{Tag}', \text{Vrfy}')$, where $\text{Tag}'(K, M) = \text{Tag}(K, M) \parallel \text{Tag}(K, M)$. Define the Vrfy algorithm for MAC' and explain why MAC' is also a secure MAC algorithm.

Note: Tag' is not a secure PRF (why?). This illustrates that MAC security is different from PRF security.

Problem 5. [Problem 7.1 and 7.2 in [BR]]

Consider the following variants of CBC-MAC, intended to allow one to MAC messages of arbitrary length.¹ The domain for all MACs is $\{0, 1\}^{n \cdot \ell}$ for $\ell = 0, 1, \dots$ where n is the block length of the underlying blockcipher used by CBC-MAC (thus, the MACs takes as input arbitrary multiples of the block length n). Show that all these variants are completely insecure according to the UF-CMA definition (Fig. 1): break them with a constant number of queries.

a) $\text{CBCv1}(K, M) = \text{CBC}(K, M \parallel |M|)$, where $|M|$ is the length of M , written in n bits.

Hint: You can do this with 3 TAG queries: two queries having 1 (message) block, and one query having 3 (message) blocks. The VF query should have 3 (message) blocks and use elements from all 3 TAG queries.

b) $\text{CBCv2}((K, K'), M) = \text{CBC}(K, M) \oplus K'$, where K' has n bits.

¹We're slightly abusing notation and use $\text{CBC}, \text{CBCv1}, \dots$, to denote both the MAC algorithm and the Tag function.

- c) $\text{CBCv3}(K, M) = (IV, T)$, where IV is a value drawn at random from $\{0, 1\}^n$ and XORed with the first block of message M , then normal CBC-MAC is applied to get the tag T . That is, this is an IV-based version of CBC-MAC similar to CBC\$-encryption. Note that the IV must be sent together with the tag T in order for the recipient to be able to verify the message.

Problem 6. [Problem 6.1 in [BS]]

Consider the following MAC (a variant of this was used for WiFi encryption in 802.11b WEP), where $F: \{0, 1\}^{128} \times \{0, 1\}^{128} \rightarrow \{0, 1\}^{32}$ is a PRF. Let [CRC32](#) be a simple and popular error-detecting code meant to detect random errors; CRC32 is a function that takes as input $M \in \{0, 1\}^*$ and outputs a 32-bit string. Define the following scheme Σ :

$\Sigma.\text{KeyGen}:$	$\Sigma.\text{Tag}(K, M):$	$\Sigma.\text{Vrfy}(K, M, (R, T)):$
1: $K \xleftarrow{\$} \{0, 1\}^{128}$	1: $R \xleftarrow{\$} \{0, 1\}^{128}$	1: $T' \leftarrow F(K, R) \oplus \text{CRC32}(M)$
2: return K	2: $T \leftarrow F(K, R) \oplus \text{CRC32}(M)$	2: if $T' = T$ then
	3: return (R, T)	3: return 1
		4: else
		5: return 0

Show that this MAC system is insecure

Hint 1: One possible adversary creates a forgery by making one $\text{Tag}(\cdot)$ query and running for about 2^{32} time.

Hint 2: Another possible adversary makes one $\text{Tag}(\cdot)$ query, but runs in virtually no time using the following property of CRC32: $\text{CRC32}(M \oplus M') = \text{CRC32}(M) \oplus \text{CRC32}(M')$.

Problem 7.

Let $F: \mathcal{K} \times \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$ be a UF-CMA secure MAC.²

- a) Define another MAC $F': \mathcal{K} \times \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n}$ as follows:

$$F'_K(M \| N) \stackrel{\text{def}}{=} F_K(M \| N) \| M,$$

where $M, N \in \{0, 1\}^n$. That is, F' first applies F to the entire message, then appends the *first half* of the input (M) to the *end* of the output. Show that F' is UF-CMA secure.

- b) Suppose instead of a secure PRF, CBC-MAC was instantiated with a fixed-length UF-CMA secure MAC as its internal building block. Show that this variant of CBC-MAC is not necessarily a secure MAC (even for fixed-length messages).

Hint: Use a).

²We're slightly abusing notation and use F to denote both the MAC algorithm and the Tag function.

References

- [BR] Mihir Bellare and Phillip Rogaway. *Introduction to Modern Cryptography*. <https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf>.
- [BS] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography*, (version 0.6, Jan. 2023). <https://toc.cryptobook.us/>.
- [Ros] Mike Rosulek. *The Joy of Cryptography*, (draft Jan 3, 2021). <https://web.engr.oregonstate.edu/~rosulekm/crypto/crypto.pdf>.