Introduction to Cryptography

TEK 4500 (Fall 2020)
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

5a8a55d2dea40512da9d0dd64863107aa3eb5a4d8f46967f

represents a ciphertext of the ASCII-encoded message $M =$ "`Transfer:     $10 to Bob`", encrypted using the OTP. Modify the ciphertext so that it decrypts to $100000.

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

```
import os

m = bytes("Transfer:     $10 to Bob", 'ascii')
k = os.urandom(len(m))
c = bytearray([a ^ b for (a,b) in zip(m,k)])  # note: bytarray is mutable
print(c.hex())
```

Problem 3. [Problem 10.1 in [Ros]]
Consider the following MAC scheme, where $F\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is a secure PRF.

Σ.KeyGen:	Σ.TAG$(K, M_1\| \cdots \|M_\ell)$: // each M_i is n bits
1: $K \xleftarrow{\$} \{0,1\}^k$	1: $M \leftarrow 0^n$
2: **return** K	2: **for** $i = 1, \ldots, \ell$ **do**
	3: $\quad M \leftarrow M \oplus M_i$
	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).

$\mathbf{Exp}^{\text{uf-cma}}_{\Sigma}(\mathcal{A})$

$K \xleftarrow{\$} \Sigma.\mathsf{KeyGen}$
$\mathsf{won} \leftarrow 0$
$S \leftarrow [\,]$
$\mathcal{A}^{\textsc{Tag}_K(\cdot),\mathrm{VF}_K(\cdot)}$
return won

$\textsc{Tag}_K(M)$:

$S.\mathsf{add}((M,T))$
$T \leftarrow \Sigma.\textsc{Tag}(K,M)$
return T

$\mathrm{VF}_K(M,T)$:

$d \leftarrow \Sigma.\textsc{Vrfy}(K,M,T)$
if $d = 1$ **and** $(M,T) \notin S$ **then**
    $\mathsf{won} \leftarrow 1$
return d

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

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

Problem 4. [Problem 10.3 in [Ros]]
Suppose MAC is a secure MAC algorithm. Create a new MAC algorithm $\mathsf{MAC}'(K,M) = \mathsf{MAC}(K,M)\|\mathsf{MAC}(K,M)$. Define the Vrfy algorithm for MAC' and explain why MAC' is also a secure MAC algorithm.

Note: MAC' 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. The domain for both MACs is $\{0,1\}^{n\cdot\ell}$ for $\ell = 0, 1, \ldots$ 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).

a) $\mathsf{CBCv1}(K,M) = \mathsf{CBC}(K, M\||M|)$, where $|M|$ is the length of M, written in n bits. Show that CBCv1 is completely insecure according to the UF-CMA definition (Fig. 1): break it with a constant number of queries.

b) $\mathsf{CBCv2}((K,K'),M) = \mathsf{CBC}(K,M) \oplus K'$, where K' has n bits. Show that CBCv2 is completely insecure according to the UF-CMA definition (Fig. 1): break it with a constant number of queries.

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 $F\colon \{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{32}$ is a 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:

Σ.KeyGen:	Σ.TAG(K, M):	Σ.VRFY$(K, M, (R, T))$:
1: $K \xleftarrow{\$} \{0,1\}^{128}$ 2: **return** K	1: $R \xleftarrow{\$} \{0,1\}^{128}$ 2: $T \leftarrow F(K, R) \oplus \mathsf{CRC32}(M)$ 3: **return** (R, T)	1: $T' \leftarrow F(K, R) \oplus \mathsf{CRC32}(M)$ 2: **if** $T' = T$ **then** 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 $\text{TAG}_K(\cdot)$ query and running for about 2^{32} time.

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

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 0.5, Jan. 2020). https://toc.cryptobook.us/.

[Ros] Mike Rosulek. *The Joy of Cryptography*, (draft Feb 6, 2020). https://web.engr.oregonstate.edu/~rosulekm/crypto/crypto.pdf.