Introduction to cryptology
TD#2

2021-W06,...

Exercise 1: Arithmetic in $\mathbb{Z}/2^8\mathbb{Z}$ and $\mathbb{F}_{2^8}$

Q. 1: Compute the following in $\mathbb{Z}/2^8\mathbb{Z}$:

1. 153 + 221
2. 29 + 8
3. 64 + 31

Q. 2: Compute the following in $\mathbb{F}_2^8$ (where a decimal representation is used for the elements, i.e. the addition corresponds to the bitwise XOR):

1. 153 + 221
2. 29 + 8
3. 64 + 31

Q. 3: Under what condition on their operands are the additions in $\mathbb{Z}/2^8\mathbb{Z}$ and $\mathbb{F}_{2^8}$ equivalent?

Exercise 2: Bit-vector arithmetic

Q. 1: Write a small "naïve" C function that computes the scalar product of two vectors of $\mathbb{F}_2^{32}$. This function must have the following prototype:

```
uint32_t scalar32_naive(uint32_t x, uint32_t y).
```

Q. 2: Write another implementation of the same function, of prototype

```
uint32_t scalar32_popcnt(uint32_t x, uint32_t y),
```

that uses a *bitwise and* instruction "`&`" and the *population count* function for 32-bit words "`__builtin_popcount()`".

Q. 3 Explain why in C, assuming that `x` is of type `uint32_t`, `x << 1` computes the multiplication of `x` by two in $\mathbb{Z}/2^{32}\mathbb{Z}$.

Q. 4 Explain why in C, assuming that `x` is of type `uint32_t`, `x >> 1` is equivalent to `x / 2`.

Q. 5

1. Write the matrix $\boldsymbol{M}$ of dimension 8 over $\mathbb{F}_2$ such that $\boldsymbol{Mx} = \texttt{mul2(x)}$, where `mul2` is defined as:

```
uint8_t mul2(uint8_t x)
{
  return ((x << 1) & 0xFF);
}
```

   and $\boldsymbol{x}$ and `x` are in natural correspondence (with the encoding convention that $\boldsymbol{x} = \begin{pmatrix}\boldsymbol{x}_0 & \boldsymbol{x}_1 & \ldots \boldsymbol{x}_7\end{pmatrix}^t \mapsto \boldsymbol{x}_7 2^7 + \boldsymbol{x}_6 2^6 + \ldots + \boldsymbol{x}_0 2^0$).

2. Is this matrix invertible?

Q. 6 What are the logical formulas computed by the following functions on their inputs?

```
uint32_t f1(uint32_t x, uint32_t y, uint32_t z)
{
  return ((x & y) | (~x & z));
}

uint32_t f2(uint32_t x, uint32_t y, uint32_t z)
{
  return ((x & y) | (x & z) | (y & z));
}

uint32_t f3(uint32_t x, uint32_t y, uint32_t z)
{
  return (z ^ (x & (y ^ z)));
}
```

Which of these functions can be computed as a matrix-vector product?

Exercise 3: PRPs

Let $\mathcal{E} : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher for which there is a subset $\mathcal{K}' \subset \{0,1\}^\kappa$ of *weak keys* of size 2^w such that if $k \in \mathcal{K}'$, $\mathcal{E}(k,\cdot) : x \mapsto x$.

Q. 1: Give a lower-bound for $\mathbf{Adv}_{\mathcal{E}}^{\mathrm{PRP}}(1,1)$.

Q. 2: Some mode of operation of block ciphers rely on the fact that $\mathcal{E}(k,0)$ is an unpredictable value when k is picked uniformly at random and kept secret (with 0 denoting the all-zero binary string).

Show that this is a reasonable assumption. More precisely, give a lower-bound on $\mathbf{Adv}_{\mathcal{E}}^{\mathrm{PRP}}(1,1)$ assuming that one can predict this value with unit time and success probability p.

Exercise 4: Format-preserving encryption *(Adapted from M2's exam, 2021)*

A *format-preserving* block cipher is a block cipher $\mathcal{E} : \{0,1\}^\kappa \times \mathcal{S} \rightarrow \mathcal{S}$ where $\mathcal{S}$ is an arbitrary finite set (that is $\mathcal{S}$ is not necessarily equal to $\{0,1\}^n$ for some n). For instance, $\mathcal{S}$ could be $\Pi_{\leq 2^{128}}$, the set of primes less than 2^{128}.

The *cycle walking* algorithm is a method to convert a block cipher $\mathcal{E} : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$ into $\mathcal{E}' : \{0,1\}^\kappa \times \mathcal{S} \rightarrow \mathcal{S}$ for any $\mathcal{S} \subseteq \{0,1\}^n$ as long as it is efficient to test if an element of $\{0,1\}^n$ is in $\mathcal{S}$. It works as follows: to encrypt $x \in \mathcal{S}$ with the key k, compute $x' := \mathcal{E}(k,x)$. If $x' \in \mathcal{S}$ then return x'; otherwise iterate the process by computing $x'' = \mathcal{E}(k,x')$ and testing if it is in $\mathcal{S}$, etc.

Q.1

1. Give an algorithm for the inverse $\mathcal{E}'^{-1} : \{0,1\}^\kappa \times \mathcal{S} \rightarrow \mathcal{S}$ of a block cipher $\mathcal{E}'$ over $\mathcal{S}$ obtained from cycle walking applied to some suitable block cipher $\mathcal{E}$.
2. Show that the condition that $\mathcal{S} \subseteq \{0,1\}^n$ be efficiently testable is not enough to guarantee that cycle walking will result in an efficient block cipher.

We now suppose the existence of a black-box algorithm that efficiently converts a block cipher $\mathcal{E} : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$ into $\mathcal{E}' : \{0,1\}^\kappa \times \{0,1\}^{n'} \rightarrow \{0,1\}^{n'}$ for any $0 < n' < n$.

Q.2

1. How does the existence of this black-box allow to remedy the efficiency problem from the previous question?
2. Are there still sets for which cycle walking is inefficient?

Exercise 5: CTR mode

Let $\mathcal{E} : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher. The CTR encryption of a message $m = m_0||m_1||\ldots$ (where all of the m_is are n-bit long) with $\mathcal{E}$ and a key k is given by $m_0 \oplus \mathcal{E}(k,t_0)||m_1 \oplus \mathcal{E}(k,t_1)\ldots$, where the t_is are n-bit pairwise-distinct values (for instance one can take $t_0 = 0$, $t_1 = 1$, etc.). In other words, one is encrypting a message with a pseudo-random keystream generated by $\mathcal{E}$.

Q. 1 : Show that the keystream used to encrypt a message of 2^n blocks (that is $n2^n$-bit long) is easy to distinguish from one drawn uniformly at random random, if it is generated with a single key.

Hint: Exploit the fact that $\mathcal{E}(k,\cdot)$ is invertible, and the bound $n! < (n/2)^n$ (valid for $n > 5$).

We may try to solve the problem of the previous question by defining $\mathcal{F}(k,x) := \mathcal{E}(k,x) \oplus x$. This makes $\mathcal{F}$ non-injective. One may then still encrypt a message $m = m_0||m_1||\ldots$ as $m_0 \oplus \mathcal{F}(k,t_0)||m_1 \oplus \mathcal{F}(k,t_1)\ldots$.

Q. 2 : Show that if the t_i values are public, then $\mathcal{F}$ suffers from the same problem as $\mathcal{E}$ in Q. 1.

Remark: It can be shown that if the t_is are secret and "random" enough (for instance $t_i = \mathcal{E}(k',t'_i)$ where the t'_is are pairwise distinct), then $\mathcal{F}$ does not suffer from the same limitation as $\mathcal{E}$ in CTR mode any more.