Problem Set 3
Course **Secure Channels**
(Summer Term 2018)

Bauhaus-Universität Weimar, Chair of Media Security
Prof. Dr. Stefan Lucks, Eik List
URL: `http://www.uni-weimar.de/de/medien/professuren/mediensicherheit/teaching/`
Due Date: 17 May 2018, 11:00 AM, via email to `eik.list(at)uni-weimar.de`.

Task 1 – Combinations of Universal Hash Functions (4 Credits)
Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ be non-empty sets. Let $\mathcal{F} = \{F \,|\, F : \mathcal{X} \to \mathcal{Y}\}$ be a family of ϵ_1-almost-universal hash functions and let $\mathcal{G} = \{G \,|\, G : \mathcal{Y} \to \mathcal{Z}\}$ be a family of ϵ_2-almost-universal hash functions. We define a third family of hash functions $\mathcal{H} = \{H \,|\, H : \mathcal{X} \to \mathcal{Z}\}$ as follows:

$$H(X) \stackrel{\text{def}}{=} G(F(X))$$

Show or disprove that $\mathcal{H}$ is $(\epsilon_1 + \epsilon_2)$-almost-universal.

Task 2 – Transformations of Universal Hash Functions (4 Credits)
Let $\mathcal{G} = \{G \,|\, G : \mathcal{X} \to \{0,1\}^n\}$ be a family of ϵ-almost-XOR-universal hash functions. Show or disprove that the family of hash functions $\mathcal{H} = \{H \,|\, H : \mathcal{X} \times \{0,1\}^n \to \{0,1\}^n\}$ whose elements are defined as $H(X,Y) \stackrel{\text{def}}{=} G(X) \oplus Y$ is ϵ-almost-universal.

Task 3 – Polynomial Hashing (4 Credits)
Let $\mathbb{GF}(2^n)$ denote the Galois-Field of elements of degree at most $n-1$. For this task, we focus on $n = 8$ and use $p(\mathtt{x}) \stackrel{\text{def}}{=} \mathtt{x}^8 + \mathtt{x}^4 + \mathtt{x}^3 + \mathtt{x} + \mathtt{1}$ as the irreducible polynomial for multiplying elements in the field. We define the family of polynomial hash functions $\mathcal{H} = \{H \,|\, H : (\mathbb{GF}(2^n))^* \to \mathbb{GF}(2^n)\}$ as follows. For any given input $M = (M_1, \ldots, M_m)$, with $0 \le m < 2^n$, it splits M into blocks of n bits each, appends a final block $|M|$ that encodes the number of blocks m as integer, and computes the hash Y as

$$Y = H_K(M) \stackrel{\text{def}}{=} \left(\bigoplus_{i=1}^{m} K^{i+1} \cdot M_i \right) \oplus |M| \cdot K.$$

The secret key K defines the instance $H \in \mathcal{H}$. All multiplications are in $\mathbb{GF}(2^n)$ modulo $p(\mathtt{x})$. Implement H in Python3. We treat every byte $M_i = (a_7, a_6, \ldots, a_0)$ as polynomial $\sum_{i=0}^{7} a_i \cdot \mathtt{x}^i$. For example, the byte $(\mathtt{00000101})_2$ that represents the integer `5` is interpreted as $(\mathtt{x}^2 + \mathtt{1})$ in $\mathbb{GF}(2^8)$.
Implement also the Wegman-Carter MAC, that computes an authentication tag T for a given n-bit nonce and an arbitrary-length message M, with the hash function above and a call to an n-bit permutation $\pi : \{0,1\}^n \to \{0,1\}^n$:

$$T = \pi(N) \oplus H_K(M)$$

You can use `numpy.random.permutation(1 << n)` to mimic a pseudorandom n-bit permutation. You can find an API and test case that your code shall pass on the website of the problem session.

Task 4 – Almost Uniformity (7 Credits)
The notion of almost-uniformity is closely related to almost-universality. It considers the probability that certain outputs occur. Let $\mathcal{X}, \mathcal{Y}$ denote two non-empty sets, and $\mathcal{H} = \{H | H \in \mathcal{X} \to \mathcal{Y}\}$ be a family of hash functions. We call $\mathcal{H}$ ϵ-almost-uniform if and only if, for all inputs $X \in \mathcal{X}$ and all outputs $Y \in \mathcal{Y}$, it holds that

$$\Pr_{H \leftarrow \mathcal{H}}[H(X) = Y] \leq \epsilon.$$

a) Determine comprehensively the smallest possible ϵ for the almost-uniformity of the polynomial hash function in Task 3.

b) Let $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a secure block cipher with n-bit key and n-bit inputs. Let H be the polynomial hash function from Task 3. For a secret random key $K \in \mathbb{GF}(2^n)$, consider the following MAC that uses a hash of a given message M as key:

$$T = MAC_K(M) \stackrel{\text{def}}{=} E_{H_K(M)}(0)$$

Show briefly efficient forgery attack on this MAC.

c) Which properties among {ϵ-almost-universality, ϵ-almost-uniformity, ϵ-almost-XOR-universality} do you need to make this MAC secure (up to $2^{n/2}$ queries suffices) and why?

d) Modify the hash function (with a change as minimal as possible) so that it provides the security properties from c) that are needed.

Task 5 – Bonus (1 Credits)
The following logical question is borrowed from a book by William Poundstone: Assume, you are an conductor, and strive to get the best sound produced by an orchester of n musicians. You want to test all 2^n combinations of musicians at some point of time. You can always ask one of the musicians to either enter the stage or to leave it (they can come back later, of course). The stage entry is very small, so only one person can go through it at a time. Every time a person enters or leaves the stage is called a move. Briefly describe (1 sentence suffices) an efficient strategy to test all $2^n - 1$ different combinations of musicians on the stage with as few moves as possible.