**0368.3049** **Fall 2013**

**Introduction to Modern Cryptography**

Benny Chor and Nir Bitansky

# Assignment 2

Published November 21, 2013. Due December 5, in mailbox 372 (Schreiber building).
Submission in pairs is encouraged (submission in threes or more is not allowed). A 5-point bonus will be given to typed (as opposed to handwritten) submissions.

**Remark:** As usual, in all questions, assume that algorithms may be probabilistic, and in all probability statements, the probability is also taken over their random coin tosses.

**Problem 1: A non-mandatory useful exercise (will not be graded).** Let $G$ be a cyclic group of order $N$, with generator $g$. Show that the following distributions are identical:

$$z : z \leftarrow G \ (\leftarrow \text{“means sampled uniformly at random from”}) \tag{1}$$

$$g^r : r \leftarrow \mathbb{Z}_N \tag{2}$$

$$g^{r \bmod N} : r \leftarrow \{t+1, t+2, \ldots, t+N\} \text{ and } t \text{ is any integer} \tag{3}$$

$$g^r : r \leftarrow \{t+1, t+2, \ldots, t+N\} \text{ and } t \text{ is any integer } . \tag{4}$$

**Problem 2: Random self-reducibility of discrete-log.** The goal of this question is to show that if discrete-log can be solved on random instances with reasonable probability, then it can also be solved on worst-case instances. Let $g$ be a generator of $\mathbb{Z}_p^*$. Given an algorithm $A$, such that $\Pr_{x \leftarrow \mathbb{Z}_p^*}[A(g, g^x) = x] \geq \delta$, show how to construct a probabilistic algorithm $A'$ that, for **any** $x \in \mathbb{Z}_p^*$, given $(g, g^x)$, outputs $x$ in expected time $O(\frac{\mathsf{time}(A)}{\delta})$.
**Guidance**: consider running $A$ on inputs of the form $(g, g^{x+r})$ for random $r \leftarrow \mathbb{Z}_p^*$.

**Problem 3: Quadratic residues and DDH in $\mathbb{Z}_p^*$.** In what follows, we consider the multiplicative group $\mathbb{Z}_p^*$, and denote by $\mathbb{QR} \subset \mathbb{Z}_p^*$ the subgroup of quadratic residues.

(a) Show that any $s \in \mathbb{QR}$ has exactly two roots $r, r' \in \mathbb{Z}_p^*$.

(b) Show that for any $z \in \mathbb{Z}_p^* \setminus \mathbb{QR}$, $z^{\frac{p-1}{2}} = -1 \pmod{p}$.

(c) Is it true that, for any prime $p$, $-1 \notin \mathbb{QR}$? prove, or give an example.

(d) Let $p$ be a prime such that $p \equiv 3 \mod 4$. Show that for any $s \in \mathbb{QR}$, $s^{\frac{p+1}{4}}$ is a square root of $s$.

(e) Let $g$ be a generator of $\mathbb{Z}_p^*$. Show that there exists a one-bit output algorithm $A$ that decides DDH with a $1/2$ distinguishing gap; that is,

$$\left| \Pr_{(x,y) \leftarrow (\mathbb{Z}_p^*)^2}[A(g, g^x, g^y, g^{xy}) = 1] - \Pr_{(x,y,z) \leftarrow (\mathbb{Z}_p^*)^3}[A(g, g^x, g^y, g^z) = 1] \right| \geq \frac{1}{2} \ .$$

**Problem 4: CBC-MACs and variable length messages.** In this problem, we explore the (in)security of CBC-MACs of variable message length. The suggested constructions rely on a block cipher,

$E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, namely $E_K(B)$ is the encryption of a block $B \in \{0,1\}^n$ under a key $K \in \{0,1\}^k$.

Let $\mathbf{x} = x_1 x_2 \cdots x_\ell$, where $x_i \in \{0,1\}^n$ for each $i = 1, \ldots, \ell$. For all the variants considered in this problem, the authentication of the message $\mathbf{x}$ will be denoted by $\mathrm{MAC}_K(\mathbf{x})$, where $K$ is the secret key (shared by Alice and Bob), and $\mathrm{MAC}_K(\mathbf{x})$ will be of fixed length $n$.

We say that Fred, the adversary, succeeds in forging a MAC, if it can choose a small (e.g., constant) number of messages $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_s$, referred to as *Fred's queries*, obtain their MACs under the *unknown* secret key $K$, and then produce a new message $\mathbf{w} \notin \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_s\}$ together with it MAC $\mathrm{MAC}_K(\mathbf{w})$. $\mathbf{w}$ may (and typically will) be constructed out of pieces depending on the $\mathbf{z}_i$'s.

Show how to break each of the following MACs with a constant number of queries:

(a) Consider the application of 'standard' CBC-MAC to messages of arbitrary length. Formally, given $\mathbf{x} = x_1 x_2 \cdots x_\ell$, we define $y_0 = 0^n$ and $y_i = E_K(y_{i-1} \oplus x_i)$ for $1 \le i \le \ell$. Then $\text{CBC-MAC}_K(\mathbf{x}) = y_\ell$.

(b) In order to overcome the problem of applying 'standard' CBC-MAC to messages of arbitrary length, consider the following patch.

$$MAC_K(x_1, x_2, \ldots, x_\ell) = \text{CBC-MAC}_K(x_1, x_2, \ldots, x_\ell, \ell) \ ,$$

where the number $\ell$ of blocks in $\mathbf{x}$ is written in binary using $n$ bits.

(c) Consider the following attempt to allow one to MAC messages of arbitrary length. The domain for the MAC is $(\{0,1\}^n)^*$. To MAC the message $\mathbf{x} = x_1 x_2 \cdots x_\ell$ under the secret key $(K, K')$, compute $\text{CBC-MAC}_K(\mathbf{x}) \oplus K'$, where $K$ has $k$ bits and $K'$ has $n$ bits.

**Problem 5: Collision-resistance implies one-wayness.** Let $H : \{0,1\}^{n+s} \to \{0,1\}^n$ be a compressing function (in particular, $s \ge 1$). Let $A$ be a poly-time algorithm that inverts $H$, specifically assume $\Pr_{x \leftarrow U_{n+s}}[A(H(x)) = x' : H(x) = H(x')] \ge \epsilon$. Show that there exists a probabilistic poly-time algorithm $A'$ that can find a collision in $H$ with probability at least $\epsilon/2 - 2^{-s}$.

**Problem 6: Irreducible polynomials.** Use Sage to find out the number of degree three polynomials of the form $f(x) = x^3 + a_2 x^2 + a_1 x + a_3$ over $\mathbb{Z}_3$ that are irreducible, and the number of such polynomials that factor into 3 linear factors (of the form $ax + b$, where $(a, b) \in \mathbb{Z}_3^* \times \mathbb{Z}_3$).

**Problem 7: 12 Point bonus - The Blum-Micali hardcore bit.** Let $g$ be a generator of $\mathbb{Z}_p^*$. In class, we defined the predicate $\mathsf{Half} : \mathbb{Z}_p^* \to \{0,1\}$, as follows:

$$\mathsf{Half}(x) = \begin{cases} 1 & 1 \le x \le \frac{p-1}{2} \\ 0 & \frac{p+1}{2} \le x \le p-1 \end{cases} \qquad (g^x \text{ is the principle root of } g^{2x}) \ .$$

We have seen that given an algorithm $H$ that perfectly predicts $\mathsf{Half}$, namely $\Pr_{x \leftarrow \mathbb{Z}_p^*}[H(g, g^x) = \mathsf{Half}(x)] = 1$, we could construct an efficient algorithm $A^H$ that makes subroutine calls to $H$, and can always solve discrete-log, namely $\Pr_{x \leftarrow \mathbb{Z}_p^*}[A^H(g, g^x) = x] = 1$.

In this question, we are given a noisy algorithm $\tilde{H}$ that only predicts $\mathsf{Half}$ with some advantage over a random guess; specifically, we assume $\Pr_{x \leftarrow \mathbb{Z}_p^*}[\tilde{H}(g, g^x) = \mathsf{Half}(x)] \ge \frac{1}{2} + \delta$. We would show that such an algorithm can still be used to solve discrete-log with high probability.

(a) **Step 1: Enough to solve DL for small (but not too small) exponents.** Assume a (probabilistic) algorithm $A_t$ such that for $t < p-1$, and **any** $x \in \{1, 2, \ldots, t\}$: $\Pr[A_t(g, g^x) = x] \geq 3/4$, where the probability is over the random coin tosses of $A_t$. Show there is a (probabilistic) $A'$ that, for **any** $x \in \mathbb{Z}_p^*$, given $(g, g^x)$ outputs $x$ in expected time $O(\mathsf{time}(A_t) \cdot \frac{p-1}{t})$.

(b) **Step 2: Making $\tilde{H}$ work for any small exponent** $\mod \frac{p-1}{2}$**, rather than only for random exponents.** Given $\tilde{H}$ such that $\Pr_{x \leftarrow \mathbb{Z}_p^*}[\tilde{H}(g, g^x) = \mathsf{Half}(x)] \geq \frac{1}{2} + \delta$, and any $t < \frac{\delta}{4} \cdot (p-1)$, construct a (probabilistic) algorithm $\tilde{H}_t$ such that for **any** $x \in \{1, 2, \ldots, t\} \cup \left\{\frac{p-1}{2}, \frac{p+1}{2}, \ldots, \frac{p-1+2t}{2}\right\}$: $\Pr[\tilde{H}_t(g, g^x) = \mathsf{Half}(x)] \geq \frac{1}{2} + \frac{\delta}{2}$.

**Guidance:** Given $(g, g^x)$, run $\tilde{H}(g, g^{x+r})$ for a random $r \leftarrow \mathbb{Z}_p^*$. For which $r \in \mathbb{Z}_p^*$ can you compute $\mathsf{Half}(x)$ from $\mathsf{Half}(r + x \bmod p - 1)$?

(c) **Step 3: Amplifying $\tilde{H}_t$.** Consider a new algorithm $\tilde{H}_t^{\otimes N}$ that given input $(g, g^x)$, runs $\tilde{H}_t(g, g^x)$ $N$ times independently, and outputs the majority vote. Using a standard tail bound, it can be shown that, for any $x$ such that $\Pr[\tilde{H}_t(g, g^x) = \mathsf{Half}(x)] \geq \frac{1}{2} + \frac{\delta}{2}$, it holds that $\tilde{H}_t^{\otimes N}(g, g^x) = \mathsf{Half}(x)$, except with probability $2^{-\delta^2 N/2}$.
Fix $N = \frac{5 \log p}{\delta^2}$. Use $\tilde{H}_t^{\otimes N}$, and the procedure we described in the recitation, to construct an algorithm $A_t$ as required for Step 1.