Cryptography for Computer Scientists 2018/19	Nov 21, 2018

Exercises 8

Stefan Dziembowski MIM UW

Exercise 1: Håstad low exponent attack

Let $(N_1, d_1), (N_2, d_2)$, and (N_3, d_3) be the RSA secret keys (chosen randomly), and let $(N_1, 3), (N_2, 3)$, and $(N_3, 3)$ be the corresponding public keys (i.e. the public exponent e is always set to 3). Suppose the adversary learns ciphertexts of some message $m < N_1, N_2, N_3$ with respect to these keys, i.e., $m^3 \bmod N_1, m^3 \bmod N_2$, and $m^3 \bmod N_3$. Show how he can compute m from this information.

Exercise 2: Fault attacks on RSA

Let $N = pq$ be an RSA modulus, and let $\mathsf{CRT} : \mathbb{Z}_N \to \mathbb{Z}_p \times \mathbb{Z}_q$ be the function from the Chinese Remainder Theorem, i.e., $\mathsf{CRT}(x) = (x \bmod p, x \bmod q)$. Consider the following algorithm for computing RSA decryption.

$\mathsf{Dec}_{d,N}(c)$	
1 :	$(a, b) := \mathsf{CRT}(c)$
2 :	$a' := a^d \bmod N$
3 :	$b' := b^d \bmod N$
4 :	**return** $\mathsf{CRT}^{-1}(a', b')$

Suppose the adversary gets input-output access to a device that contains (d, N) and decrypts RSA according to the above algorithm. Assume that later he gets access to the same device that makes *one* error during computation in Step 2 or 3, i.e., computes wrong a' or wrong b'. Show how the adversary can factor N.

Exercise 3: Random self reducibility of discrete log

Let $(G, \times)$ be a finite group and let g be its generator. Suppose M is an oracle that on random input $h \in G$ produces as output a value $\log_g h$ with probability p. Show an algorithm that takes as input a value $f \in G$, runs in time linear in a parameter k, asks k queries to M, and outputs $\log_g f$ with probability $1 - (1 - p)^k$ (for every f).

Exercise 4: Baby-step giant-step algorithm

Let $(G, \times)$ be a cyclic group with a generator g. Show an algorithm that computes the discrete log in G (with base g) using $O(\sqrt{|G|})$ exponentiations and $\tilde{O}(\log_2 |G|)$ space.

Exercise 5: Backdoor in the Dual Elliptic Curve Deterministic Random Bit Generator

Let E be an elliptic curve group with prime order defined over some $\mathbb{Z}_p$ in which computing discrete log is hard. For $(x, y) \in E$ define $\varphi(x, y) = x$. For $x \in \mathbb{Z}_p$ let $\mathsf{suffix}(x)$ be the binary representation of x without the first 16 bits, i.e.:

$$\mathsf{suffix}(x) = (y_{17}, \ldots, y_m),$$

where $(y_1, \ldots, y_m)$ is the binary representation of x. Let P, Q be generators of E. Consider the following algorithm $\mathsf{Dual\_EC\_DRBG}$ that takes as input $s_0 \in E$, and produces as output blocks of bits $w_0, w_1, \ldots$ (where each $w_i \in \{0,1\}^{m-16}$):

$\mathsf{Dual\_EC\_DRBG}(s_0)$

for $i = 0, 1, \ldots$
  $r_i := \varphi(s_i \times P)$
  $t_i := \varphi(r_1 \times Q)$
  $s_{i+1} := \varphi(r_i \times P)$
  output $w_i := \mathsf{suffix}(t_i)$

Suppose the adversary knows the discrete log of P with base Q, i.e., he knows e such that $e \times Q = P$. Show how he can compute $w_2, w_3, \ldots$ from (w_0, w_1) (with high probability).