RUHR UNIVERSITÄT BOCHUM RUB

Lehrstuhl für Kryptologie und IT-Sicherheit
Prof. Dr. Alexander May
Elena Kirshanova

Hausübungen zur Vorlesung

Kryptanalyse I

SS 2015

Blatt 3 / 11. Juni 2015

Abgabe bis: 18. Juni 12:00 Uhr, Kasten NA/02

Aufgabe 1 (10 Punkte)**:**
Parallel Pollard's ρ.

A simple but powerful technique to parallelize collision search was proposed by van Oorschot and Wiener. It is based on a *distinguished point* approach.

Assume we have m nodes and one server. Each node starts its own collision search based on a random walk defined by a function $f : S \to S$ (e.g. $f(x) = x^2 + a$ in the factoring example from class). Namely, a node selects $x_0 \in S$ and produces a sequence of points $x_i = f(x_{i-1}), i = 1, 2, \dots$ until some *distinguished* point is reached. The distinguishing property is defined such that it is easy to test (say, numbers with d leading zeros in bit-representation). This distinguishing property also determines the proportion of points, denoted θ, that satisfy it (e.g. if the set S consists of n-bit integers, $\theta = 1/2^d$). Once a distinguished point x_d is found, a node sends it to the server, which accumulates all the received distinguished points in a central list. A collision is detected when the same distinguished point appears twice in the list. In 1, the two nodes report the same distinguished point x_5 indicating a collision $f(x_2) = f(x_2')$.

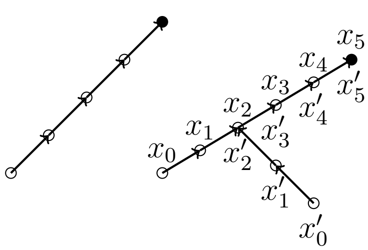

Abbildung 1: The distinguished point (solid black) x_5 indicates a collision $f(x_2) = f(x_2')$.

Running time analysis. Let $|S| = p$. We make the following assumptions to aid the complexity analysis:

1. we require only one collision and we assume that the first found one is useful (an example of a not useful collision for factorization would be $f(x) = f(x')$, s.t. $\gcd(x' - x, n) = n$).
2. all nodes lead to a distinguished point
3. f behaves like a truly random map

From the analysis of a ρ method, we expect to produce approx. $\sqrt{p}$ points before one node touches another (i.e. before a collision occurs). Since we have m nodes, we make $\sqrt{p}/m$ steps to expect a collision. Now the last question is how to get from a distinguished point (found by a server node) to a collision?

Since f is a random map, $\Pr[x_i \text{ is a distinguished point}] = \theta$, (where $\theta = \frac{\#\text{ dist.points}}{p}$), thus we expect to produce additional $1/\theta$ points after a collision occurs. (Equivalently, the number of steps from a collision to its detection is geometrically distributed random variable with mean $1/\theta$). We trace back from a distinguished point to the corresponding collision by, for instance, sending the initial x_0 to the server. Overall, the expected running time is

$$\mathbb{E}(T) = \left(\sqrt{p}\frac{1}{m} + \frac{1}{\theta}\right).$$

1. Describe a parallel version of the Pollard's ρ method for the dlog problem. Estimate its running time $\mathbb{E}(T)$ and space complexity $\mathbb{E}(S)$.
2. Assume you solve a dlog in a group of size $p = 2^{80}$ and you have $m = 128$ nodes at your disposal (and 1 server). What will be the optimal distinguishing criteria?

Aufgabe 2 (7 Punkte)**:**
Generalized k-List Problem.

1. Describe an algorithm that solves the k-List Problem for $k = 2^m + j, 0 < j < 2^m$ in $\widetilde{\mathcal{O}}(k 2^{\frac{n}{m+1}})$ with lists of size $2^{\frac{n}{m+1}}$. Prove the correctness and runtime.
2. Use the above to solve the following 5-List problem over $\mathbb{Z}_{64}$:

$$L_1 = \{31, 6, 11, 3\}, \quad L_2 = \{10, 5, 7, 21\}, \quad L_3 = \{19, 30, 13, 9\},$$
$$L_4 = \{8, 14, 4, 1\}, \quad L_5 = \{7, 12, 2, 50\}.$$

Aufgabe 3 (13 Punkte)**:**
Programming assignment: Attack on El-Gamal Signature.

The El-Gamal signature for a message $m \in \mathbb{Z}_p$ is a tuple

$$\mathrm{Sign}(m) = (\gamma, \delta) = (\alpha^k \mod p, (m - a\gamma)k^{-1} \mod (p-1)),$$

where $pk = (\alpha, \beta = \alpha^a), sk = a$ and $k \in_R \mathbb{Z}_{p-1}^*$. It is crucial that k is chosen uniformly at random, having the constant k for all messages leads to a total break. In this exercise you will exploit this breach.

You're given an access to the oracle that outputs El-Gamal signature (γ, δ) for the input message m. It uses the same k for all messages. As always, the parameters p, α, β are in 'params.txt'. The file 'ElGamalSign.o' provides

```
void ElGamalSign (mpz_t m, mpz_t gamma, mpz_t delta).
```

The parameters p, α, β are declared and initialized in the header 'ElGamalSign.h'.

Your task is to find a. You can follow the instructions from HW1. Submit your code!

EXTRA-Points: Suppose Alice chooses an initial random value k_0 and sings the i-th message with $k_i = k_0 + 2i \mod p$. Describe how Bob can easily compute Alices' secret key and recover k_0.