Report – first phase of the project

Martin Bulák, Michal Čech, Šimon Doucha

Scripts and generator: https://github.com/michal-cech/PV204-Project/
Keys: https://drive.google.com/drive/folders/1efLlZsRKriTyoDMFVR38LwZjzIF5dokN?usp=sharing

We are working with BearSSL library https://bearssl.org/ which is a cryptographic library implemented by Thomas Pornin. The author says that the library is considered beta-quality and some features are missing.

Key collection
Michal implemented key generator which is a small C program using BearSSL. All functions (both for RSA and ECC) use PRNG which was already implemented by the author. We chose HMAC generator using SHA256 (for no particular reason). If the program is compiled on Linux, then we are using urandom file as initial seed, otherwise we are not seeding it (the generation process run on the Ubuntu machine anyway).

The generation itself is straightforward because the library is quite easy to use. The only problematic part might have been getting private exponent for RSA, because author uses partial private RSA key (d mod p and d mod q). Fortunately, author already implemented function for obtaining private exponent from the partial ones, so we did not have to do it ourselves.
We used struct timespec and clock_gettime from time.h for time measurement. We saved the time in nanoseconds.

We managed to obtain 1000000 of RSA-512 and ECC keys and 10000 of RSA-1024 and RSA-2048 keys.

Python scripts
We used plotly library for python for the generation of all of the graphs https://plot.ly/python/.

Šimon created python script for histograms and scatter graph generation. We generated histograms of most significant byte and least significant byte for modulus, first prime, second prime and private exponent of RSA keys . In case of ECC, we generated histogram of MSB and LSB of private key and scatter graph with points which are stored in public key. For both RSA and ECC were also generated histograms with keygen times.

Martin created script for heatmap generation. We generated heatmaps for MSB of the modulo (RSA) and of the private key(both RSA and ECC). X-axis of the heatmap presents the value of MSB and the y-axis presents keygen time. Z-axis (colouring) shows number of occurrences constituted from collision of axis x and y.

One of the issues with the implementation of the heatmap generator was structuring data in the right order for plotly to correctly and consistently execute generation of graph. Another issue was the presentation of the data. The problematic heatmaps were the one presenting results of the RSA-1024 and RSA-2048. This issue was caused by lower number of the key

pairs(10000) resulting in somehow ugly graph with a low number of overlaps. We solved this issue by rounding the time on the y-axis from milliseconds to hundreds of milliseconds.

Discussion

We agreed that both heatmaps and histograms do not show some kind of anomaly, MSBs, LSBs and the times of the generation are distributed quite nicely. The only peculiar thing is that the maximum value of the MSB of the private exponents of RSA keys is 170 and that the LSBs of private exponent are distributed only on values 3, 19, 35, 51, 67, 83, 99, 115, 131, 147, 163, 179, 195, 211, 227, 243.

We are slightly confused about the ECC public key scatter plot. We expected at least some resemblance of the curve but all we are getting is huge blob of points.