Efficiency Analysis and Comparison of Public Key Algorithms

Csilla Endrődi and Zoltán Hornák

Public key cryptography provides the theoretical background for most data security services (e.g. digital signature, non-reputation, key-agreement algorithms etc.), which became nowadays, as electric administration is spreading widely, quite indispensable. Public key algorithms are based on *mathematical hard problems*. Their essence is a *one-way trapdoor function*, which is very hard to be solved without knowing a specific information, but easy when having this secret. Up to now three hard problems seem to be suitable for this purpose in practice: *Integer Factorisation Problem* (IFP), *Discrete Logarithm Problem* (DLP) and *Elliptic Curve Discrete Logarithm Problem* (ECDLP). The most commonly known and applied public key algorithm, the *RSA* [1] is based on the IFP. Another promising alternative is the *ECC* [2]. It is getting into the lime-light in our days, while there can be found just less efficient method for breaking ECC than other algorithms; that is to say that by ECC the "security-per-key-bit" rate is higher. It flatters nice applicability, but we must not forget that this aspect should not be the only one, when the most appropriate algorithm for a specific application is to be chosen.

For an information system the needed security level must be clearly defined as an assumption. This level depends on the sensitivity of the transferred data (e.g. commercial transaction or a personal digital postcard), the environment the system will work in (e.g. through the Internet or on a separated LAN), etc. The security requirements determine the *data security services* that should be implemented (e.g. authentication, encryption) and the necessary minimal strength of the applied cryptography algorithms (practically the key size). The aim of the security engineer is to create the *most efficient system* satisfying these security requirements, which is usually a great challenge. Each data security service can be implemented by using different cryptography algorithms and the corresponding cryptography protocols. These implementations have different efficiency features and limitations, and these parameters moreover depend on the key size.

The different algorithms are not entirely interchangeable, since they need e.g. different type of environmental variables, different source data for key generation, have different limitations etc. For example, at ECC a sufficient common elliptic curve is needed, which is very critical. To test the goodness of a curve is difficult, that is why when a research group finds a suitable curve, patents it, and others should use these probed curves. Using ECC it is also significant, whether hardware support is used or not. On the other hand RSA works seamlessly without specific hardware, but its critical point is the prime testing, as RSA needs large primes for key-generation. Besides RSA's behaviour mighty depends on the chosen public exponent. Small exponent can radically speed up some operation, but choosing a too small value makes chance for some types of attacks.

Through the various aspects and the diverse behaviours of the algorithms, there does not exist a "clear winner". *The optimal solution for a given system can be determined by both knowing the target application's specialities and the potential cryptography algorithms' behaviour.*

The next difficulty emerges during the comparison of the algorithms. While the efficiency parameters considerably depend on the applied key size, it is important to make clear that besides *which key sizes* the comparison of the measured parameters should be done. With equal key-sizes the algorithms ensure different security level, thus instead the corresponding key sizes, which provide *adequate strength* should be applied. The security of an algorithm is determined by the fastest, generally working breaking method against it. "Fastest" means the *order of its speed* depending on the input data – namely here the key size. For RSA there exists *subexponent-time* breaking method, while against ECC just *exponent-time* method is known. For this reason generally a smaller key size is sufficient for ECC than RSA, and when raising security, the rate of growing the keys by RSA is greater. Now it is generally accepted, that RSA with a 1024 bit-long key is adequate to ECC with a 160 bit-long key.

When comparing efficiency parameters of the algorithms, we should do it besides these adequate key sizes. Another possibility is to compare the security level of the algorithms when they perform the same parameters in the case of a given operation.

To gain experience about the behaviour of the most relevant public key algorithms, RSA and ECC, we made measurements in practice with a chosen implementation (Crypto++ open source crypto library). We gauged the execution time and the *size of generated data* during *key generation, encryption, decryption, data signing* and *signature verification*. The result's dependence was also specially examined on some other parameters, e.g. the *type of the key*, the *size and type of the source data*, in case of ECC the *type of the common curve* and others. For clear observation, when examining the effect of a given parameter, we fixed all other parameters in some relevant combinations successively. The total number of executed measures was approximately 4000 for ECC and more than 2000 for RSA.

After analysing the database of the measurement results, some general conclusion could be unambiguously laid down. These conclusions correspond with the expectations knowing the mathematical background of these algorithms, but some new features were uncovered, as well.

About *speed* it can be claimed that the *key-generation* easy and fast with ECC, but with RSA it takes different and longer times showing an exponential distribution. At *encryption* and *signature checking* RSA with small exponent is the faster, but ECC with pentanomial based curve is not much weaker. However ECC performs higher speed at *decryption* and *data signing*, where RSA is not dependent on the value of the exponent.

About *sizes* the general experience is that in case of ECC the key sizes, the encrypted text and the size of the signature is smaller, therefore the amount of data to be transferred during a communication is less, too. However it must be considered that according to the existing standards [3] the common curve's parameters also must be included into the public key, thus the effective key size becomes bigger. Moreover for better efficiency for ECC a pre-computed table is required, which enlarges the data to be stored, as well.

These above-mentioned statements are representative examples of the collection, which were established about the ECC's and the RSA's behaviour.

References

[1] RFC 2437, PKCS #1: RSA Cryptography Specifications Version 2.0. B. Kaliski, J. Staddon. October 1998.

[2] A. Menezes, "Elliptic Curve Public Key Cryptosystems", Kluwer Academic Publishers, Boston, 1993.

[3] Standards for Efficient Cryptography Group (SECG), SEC1: Elliptic Curve Cryptography, SEC2: Recommended Elliptic Curve Cryptography Domain Parameters