VERY HIGH THROUGHPUT CRYPTO-SYSTEM ARCHITECTURES: THE RPK SOLUTION

A. Romeo, G. Romolotti, M. Mattavelli, D. Mlynek

Swiss Federal Institute of Technology, Integrated Systems Center C3I,
CH-1015 Lausanne Switzerland.

Abstract

RPK is a new cryptographic algorithm based on the discrete logarithm problem implemented using discrete exponentiation over finite Galois fields GF[2^n]. This paper presents a high throughput RPK architecture, suitable for encrypting high bit-rates multimedia contents.

Introduction

Crypto block systems such as RSA or Diffie-Hellman[1], presents a high level of security, but require large computational resources when applied to high bit rate applications. Conversely, stream ciphers systems are low cost and fast, but they are not considered secure enough [1]. The RPK [4,5] approach to public key encryption combines the best characteristics of both classical crypto block systems and stream ciphers systems yielding a secure system that require very low processing resources at very high throughput rates.

The RPK Algorithm

All RPK [4,5] system basic operations are mathematically equivalent to exponentiation in finite fields [2]. Private and public operation scheme is depicted in Figure 1.
Main features of the scheme are:

- the choice of the private key without restrictions, thus making possible the creation of the private key with a simple random generator,
- RPK is a non-deterministic crypto-system, therefore if the same key is used to encrypt a given plain-text twice, the two resulting cipher-text differ in a random fashion,
- the cipher-text has exactly the same length of the plain-text, provided that a short header block enabling the correct set-up of the system for the decryption is transmitted before the cipher-text,
- the state of the machine has no memory of its history.

The two main desirable features of block and stream ciphers systems, respectively security and low complexity processing, are combined in the RPK system in two distinct phases: an initialization phase and a streaming phase.

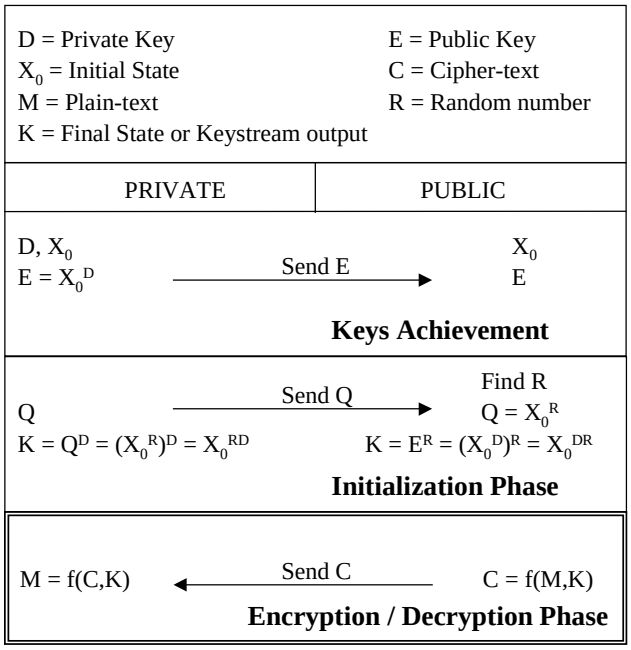

Figure 1. *Schematic representation of the RPK public key Crypto-system.*

The Phases

In more details the two distinct phases consist of:

- the crypto block:
  exponentiations over finite Galois fields GF[2^n] is used to find the final state K. The processing architecture is based on shift registers, relatively simple to be implemented in hardware and/or software. No complex arithmetic operations such as multiplication or division are necessary, complexity only depends on the length of the private and public keys
- the stream cipher block:
  n-different bit streams generated by the crypto block are combined a single key-stream output. Different architectures such as the “Alternative stop and go" [3] or other mixing generators are an example of this processing. The complexity of this block can be orders of magnitude lower than that of the crypto block.

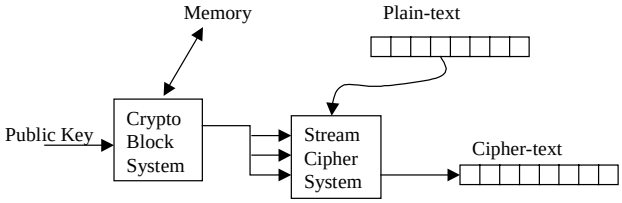

Figure 2. *Schematic representation of the RPK public key Crypto-system.*

The core RPK Crypto block processing is based on the use of the Galois fields representation, so it is possible to reduce the exponentiation to a much simpler multiplication of polynomials. This operation can be implemented in very simple and efficient ways using simple shift registers and a memory for the storage of the pre-defined polynoms.

Conclusion

Figure 3 shows typical qualitative curves of the processing time/complexity of RPK and RSA versus bit/bit-rate of the plain/cipher-text. RPK requires an initial set-up time (Crypto-block processing) which is independent from the bitrate, but just depend on the length of the public and private key. Afterwards the dependence on the bitrate is just given by the complexity of the Streaming Block (f(M,C)) which can be orders of magnitude lower than the Crypto-block processing. Conversely RSA processing complexity is proportional to the complete RSA Crypto-block which, at the same security level, is of the same order of magnitude of the RPK one and obviously much more complex than the RPK Streaming Block.

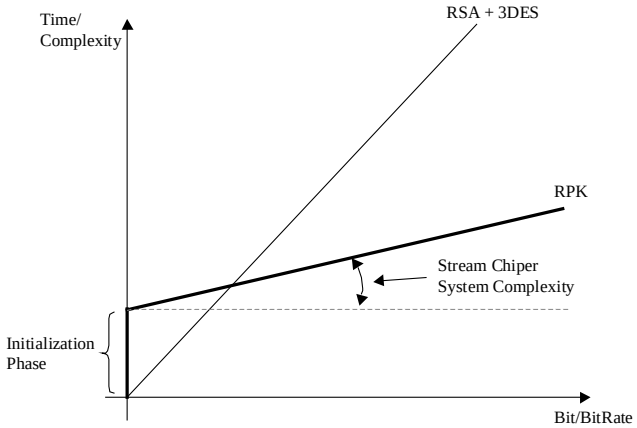

Figure 3. *Example of processing time for RPK and RSA versus the bit of the plain/cipher-text. While RSA complexity is directly proportional on the bitrate of the plain/cipher-text, RPK shows a different bitrate dependence. The slope of the curve is only dependent on the complexity of the "Stream Cipher" block that can be orders of magnitude lower than the "Crypto Block" determining the "Initialization phase".*

The RPK high throughput performances, compared with RSA performances, have been confirmed by software simulations for high bitrate multimedia content.

The operations used in the RPK initialization phase are depicted in Table 1:

INT32	UINT32	UINT64	PTR
1.E+07	2.E+07	5.E+06	2.E+07

Table 1: *Operations necessary to initialize the Stream Cipher System. This table is the result of the INSTRUMENTATION software with three keys (607-127-89 bits).*

Without the initialization phase the comparison between RSA+3DES and RPK (Stream Cipher Block) is:

	RSA+3DES	RPK
INT8	2.E+03	-
UINT8	2.E+09	-
UINT16	4.E+05	-
INT32	3.E+06	8.E+07
UINT32	5.E+08	8.E+07
INT64	1.E+08	-
UINT64	6.E+07	3.E+07
PTR	4.E+09	9.E+07

Table 2: *Comparison between the number of operations for RSA+3DES and RPK methods. This table is the result of the INSTRUMENTATION software with 1 Mbytes file.*

Is easy to show that there are many more operations in RSA hybrid method than in RPK. The Stream Cipher Block is mostly made by shift registers and xor, which are efficient and very easy to implement in hardware architectures. The Keystream output has a periodicity of 2^{823} bits (10^{247}) because of the fact that the lengths of the three different keys are Marsenne primes (607-127-89).
There is the possibility of computing the Cipher Text byte by byte or word by word. In this way the operations number, shown in Table 2, can be scaled down of a factor 8 or 32. This scalability allows several levels of security and speed according to the actual multimedia requirements.

References

[1] Bruce Schneier. "Applied Cryptography". John Wiley and Sons, New York, 2nd edition, 1996.
[2] Rudolf Lidl, Harald Niederreiter "Introduction to finite fields and their applications", Addison-Wesley Publishing Company, 1983
[3] C.G. Gunther, 1988, "Alternative Step generators controlled by de Bruijn sequences" Advances in cryptology - EUROCRYPT '87, p. 5-14
[4] "The RPK public-key cryptographic system Technical Summary" and "Detailed Supplemental Technical description of the RPK public-key cryptographic system" http://www.rpkusa.com
[5] U.S. patent number 5,799,088, August 25, 1998 or New Zealand patent number 277,128 on August 17, 1998.