GODZUK Cryptographic Algorithm for 3rd Generation Mobile Systems

A. C. C. Vieira[2,3], S. L. C. .Salomão[1,3], A. C. Pinho[1,3] and F. França[3]
{castanon, apinho, salomao}@lpc.ufrj.br and felipe@cos.ufrj.br

[1] Information Thecnology Department
Brazilian Army Research and Development Institute – IPD
Av. Das Américas 28705, 23020-470, Rio de Janeiro, RJ Brazil

[2]Department Engineering
Military Military Institute of Engineering
Pca. Gen Tibúrcio 80, 22290-270 Rio de Janeiro, RJ, Brazil

[3]Department of Electrical Engineering
COPPE/Federal University of Rio de janeiro
P.O. Box 68504 21945-970 Rio de Janeiro, RJ, Brazil

Abstract - **Nowadays, data security is an important issue in telecommunication network to be considered in its implementation and operation. This paper presents the GODZUK cryptographic algorithm in a civilian and military mobile communication applications, as a possible standards chosen by 3GPP for the third generation of cellular telephony and by the Brazilian Army as its symmetric cipher algorithm in tactical and strategic communications.**

I. Introduction

In the last years, we have witnessed a fast increase in the number of individuals and organizations using mobile communications for personal and professional activities. Recent studies on the evolution of the mobile telephony indicate that the percentage of multimedia mobile users will significantly increase after 2000 [2]. In accordance with the UMTS Forum, in 2010, 60 percent of the traffic in Europe will be created by multimedia applications [2]. A similar growth is waited worldwide, with a year growth rate in order of 70% in the next 5 years, starting with 3 million users in 1998 for 77 million users foreseen in 2005.

Observing this increasing demand, the services to be provided by third generation systems are:

- internet access, e-mail, e-commerce, real time image transfer, multimedia file transfer, mobile computation.
- Videoconference, ISDN service, videophone.
- Interactive video services, entertainment, news, TV and etc.

Analyzing these services can be noticed that a prominence factor for the perfect functioning of third generation systems bases on the emphasis given to the security question of theses systems. The main reason for that is the great amount of data traffics became very important, opening path for the process of digital fraud of information.

This work presents the symmetric cipher algorithm GODZUK and its implementation in hardware according the 3GPP requirements for mobile communications applications.

GODZUK hardware implementation is suited as 3GPP standard for two main reasons. The first one is its high performance on cipher and decipher data operations, since all information used on third generation network have to be protected with cryptography, needing a algorithm faster than 2 Mbits/s. The second one is the circuit size required for this application. Many others symmetric cryptographics algorithms are as fast as GODZUK [3,4]. However, for archiving such rate, they need more area. The GODZUK is a possible solution for this problem.

This work is organized in six sections. Section II describes the security method used in third generation systems. Section III describes the Scheduling by Multiple Edge Reversal – SMER. Section IV states the GODZUK cryptographic algorithm. Section V shows the performance of GODZUK. Finally, section VI presents the conclusions.

II. Security in Third Generation Mobile System

The studies for the globalization of the personal communications have had beginning in 1986 for ITU (International Telecommunication Union) identifying the necessary frequency band for the future systems of mobile telecommunications of third generation. In 1992, ITU identified to 230MHz band around 2GHz to implement IMT-2000 (International Mobile Telecommunication-2000) system in a worldwide scale for terrestrial and satellite components.

The IMT-2000 features a vast gamma of voice, data and multimedia services with equal or upper quality to fixed telecommunication networks in a distinct environment of RF. The main intention of the IMT-2000 is to provide universal covering making possible roaming through multiple networks. Moreover, one of the main requirements for the third generation system is the transmission rates of, 144Kbps for vehicular environments, 384Kbps for pedestrian environments and 2Mbps for indoor and picocell environments.

For the harmonization of the diverse proposals to cellular telephony of third generation, two international

organisms have been created. 3GPP (Third Generation Partnership Project) to harmonize and to standardize in details the proposals of ETSI, ARIB, TTA related to the WCDMA and the 3GPP2 for the proposal based in cdma2000 of TIA and TTA. 3 GPP elaborated a series of technical specification [5] related the information security.

Five security feature groups are defined. Each of these feature groups meets certain threats accomplishes certain security objectives:

- Network access security (I)
- Network domain security (II)
- User domain security (III)
- Application domain security (IV)
- Visibility and configurability of security (V): the set of features that enables the user to inform himself whether a security features is in operation or not and whether the use and provision of services should depend on the security feature. Within the mobile communication system UMTS specified by 3GPP there is a need to provide security features. It was decided that two cryptographic algorithms, f8 (the confidentiality algorithm) and f9 (the integrity algorithm) need to be standardised.

A. f8 – Confidentiality Algorithm

As decided by 3GPP [6,7], function f8 shall only be used to protect the confidentiality of user data and signaling data sent over the radio access link between UE (User Equipment) and RNC (Radio Network Controller). The algorithm should be designed to accommodate a range of implementation options including hardware and software implementations. For hardware implementation, it should be possible to implement one instance of the algorithm using less than 10,000 gates. It must be possible to implement the algorithm to achieve an encryption speed in order of 2Mbps on downlink and uplink.

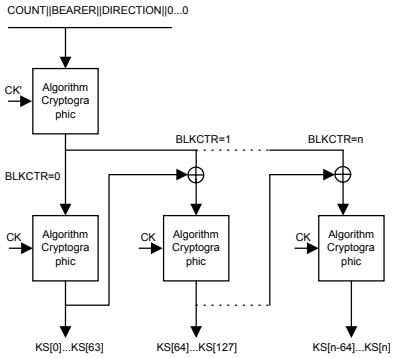

Figure 1:Function f8

The function f8, showed in figure 1, will be used to encrypt frames of variable length up to approximately 5,000 bits[6]. As each instance of the algorithm produces a 64-bit block, it is necessary that the algorithm be executed 79 times (5,000/64 = 78.125) in order to produce a data cipher block up to 5.000 bits required by function f8. For such, the typical performance for the GODZUK cipher operation shall be at least 158 Mbps, i.e., 2Mbps x 5,000/64. Moreover, the function f8 must be a synchronous and symmetric algorithm within 128-bit cipher key.

B. f9 – Integrity Algorithm

As decided by 3GPP [6,7], the function f9, as shown in figure 2, shall be a MAC function (Message Authentication Code) and shall be used to authenticate the data integrity and data origin of signaling data transmitted between UE and RNC. The algorithm should be designed to accommodate a range of implementation options, including hardware and software implementations. Moreover, this function will be used to encrypt frames of variable length up to approximately 5,000 bits [6]. The key length is 128 bits. The algorithm shall output 32-bit MAC.

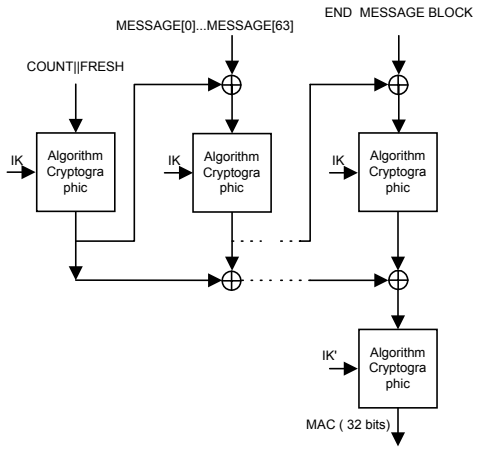

Figure 2: Function f9

III - SCHEDULING BY MULTIPLE EDGE REVERSAL (SMER)

A. Scheduling by Edge Reversal (SER)

Consider a neighborhood-constrained system composed of a set of processes and a set of atomic shared resources represented by a connected graph G = (N,E) where N is the set of processes, and E, the set of edges defining the interconnection topology. An edge is present between any two nodes if and only if the two corresponding processes share at least one atomic resource.

SER works in the following way: starting from any acyclic orientation *w* on G there is at least one sink node, i.e., a node that has all its edges directed to itself. All sink nodes are allowed to operate while other nodes remain idle. This obviously ensures mutual exclusion at any

access made to share resources by sink nodes. After operation a sink node will reverse the orientation of its edges, becoming a source and thus releasing the access to resources to its neighbors. A new acyclic orientation is defined and the whole process is then repeated for the new set of sinks [8]. Let $w' = g(w)$ denote this greedy operation, SER can be regarded as the endless repetition of the application of $g(w)$ upon G. Assuming that G is finite, it is easy to see that eventually a set of acyclic orientations will be repeated defining a *period* of length *p*. This simple dynamics ensures that no deadlock or starvation will ever occur since at every acyclic orientation there is at least one sink, i.e., one node allowed to operate. Also, it is proved that inside any period every node operates exactly *m* times [8].

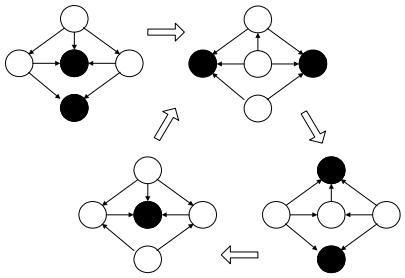

Figure 3: SER dynamics

SER is a fully distributed graph dynamics algorithm. A very interesting property of this algorithm lies in its generality in the sense that any topology will have its own set of possible SER dynamics [8]. The Figure 3 illustrates the SER dynamics.

B. Scheduling by Multiple Edge Reversal (SMER)

SMER is a generalization of SER where pre-specified access rates to atomic resources are imposed to processes in a distributed resource-sharing system which is represented by a multigraph M(N,E). Differently from SER, in the SMER dynamics a number of oriented edges can exist between any two nodes. Between any two nodes *i* and *j* , $i,j \in N$, there can exist e_{ij} unidirected edges, e_{ij}. The reversibility of node *i* is r_i, i.e., the number of edges that shall be reversed by *i* towards each of its neighbouring nodes, indiscriminately, at the end of operation. Node *i* is an r-sink if it has at least r_i edges directed to itself from each of its neighbours. Each r-sink node *i* operates by reversing r_i edges towards its neighbours; a new set of r-sinks will operate and so on. Similarly to sinks under SER, only r-sink nodes are allowed to operate under SMER. It is easy to see that with SMER, nodes are allowed to operate more than once consecutively.

The following lemma states a basic topologic constraint towards the definition of *M*, where gcd is the greatest common divisor and f_{ij} is the sum of the greatest multiple of $gcd\ (r_i,r_j)$ that does not exceed the number of shared resources oriented from n_i to n_j, and from n_j to n_i, respectively in the initial orientation.

Lemma: Let nodes *i* and *j* be two neighbors in *M*. If no deadlock arises for any initial orientation of the shared resources between *i* and *j* , then $\max\{r_i, r_j\} \leq e_{ij} \leq r_i + r_j - 1$ and $f_{ij} = r_i + r_j - \gcd(r_i,r_j)$ [15].

When we have two nodes, $e_{ij} = r_i + r_j - 1$. This property is shown in Figure 3(b). A full example of the SMER dynamics is presented in Section 4.

IV. GODZUK ALGORITHM

A. - Feistel Network

Some cryptographic algorithms are based on Feistel Network concept [11]. This concept dates of the beginning of the 70's. The basic structure of the Feistel Network is based on possibility of defining a cipher operation of blocks, where the output of I^{th} iteration phase is dependent on the previous phase output.

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \text{ XOR } F(R_{i-1,} K_i)$$

K_i is the subkey used in the i^{th} algorithm phase and F is any function that operates in each phase. For the perfect working of Feistel Network it is necessary there be the reverse operation, i.e., that the F function shows the following feature:

$$L_{i-1} \text{ XOR } F(R_{i-1,} K_i) = L_{i-1}$$

The above concept is largely used in many cryptographic algorithms.

B. GODZUK Cryptographic Algorithm

The cryptographic GODZUK algorithm is a version of the GODZILLA [9] algorithm, developed to be used in the third generation of cellular (according to the norms of the 3GPP).

As GODZILLA, GODZUK is an algorithm of symmetrical key, whose operation structure is based on Feistel Network. The algorithm operates with blocks of 64 bits and with key of 128 bits, according to demands of the 3GPP. The algorithm is executed in eight rounds on Feistel Network. In the same way that in GODZILLA, the internal operations of the algorithm are operations of OR - exclusive, executed now in only two levels of functions SMER.

B.1. GODZUK Cipher Operation

The Figure 4 shows the cipher operation for the GODZUK cryptographic algorithm, that is executed in eight rounds, with the following operation:

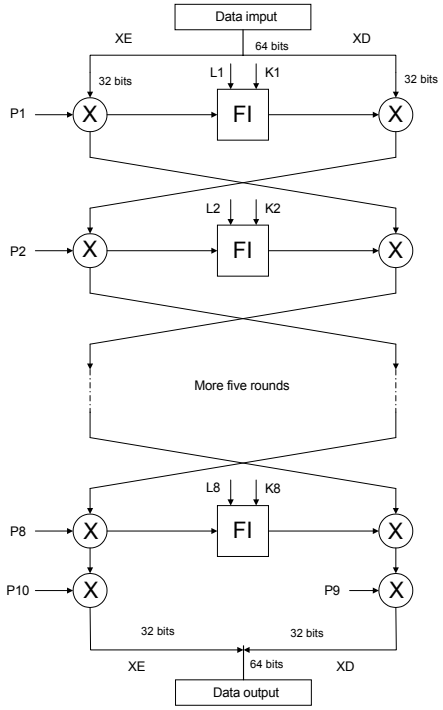

Figure 4: GODZUK cipher operations

Divide the 64 bits input block in two sub-blocks of 32 bits, XE and XD;

for i = 1 to 8 do:

XE = XE XOR Pi;

XD = FI(XE) XOR XD;

if the number of phases is different from 8:

Change XE and XD;

else

Don't change XE and XD;

XE = XE XOR P10;

XD = XD XOR P9;

Combine XE and XD to obtain the cipher data of 64 bits.

The used sub-keys are: ten sub-keys Pi of 64 bits, eight sub-keys Li of 48 bits and eight sub-keys Ki of 84 bits. The exact method for sub-keys generation can be found in [9].

Function FI

The function FI, described by Figure 5, it is also an algorithm of three phases based on Feistel Network. The operation form is plenty similar the one of the block FI of the algorithm GODZILLA.

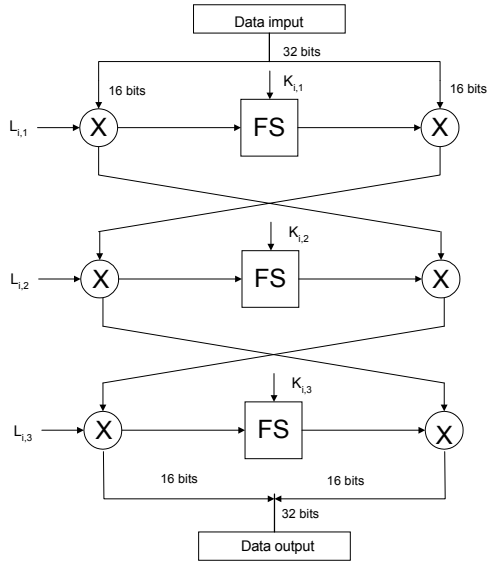

Figure 5: FI function

Divide the 32 bits input block in two sub-blocks of 16 bits, XE and XD;

for J = 1 to 3 do:

XE = XE XOR LI,J;

XD = FS(XE) XOR XD;

Combine XE and XD to obtain the cipher date of 32 bits.

The sub-keys used in the execution of each function FI are: three sub-keys Li,j of 16 bits, obtained by the division of each sub-key Li(48 bits) in three new sub-keys Li,j(16 bits); three sub-keys Ki,j of 28 bits, obtained by the division of the sub-key Ki(84 bits) in three new sub-keys Ki,j(28 bits).

Function FS

FS is a special function based on Scheduling by Multiple Edge Reversal techniques, Figure 6, it named SMER function [10]. This SMER function used has the following configuration:

- two nodes and fifteen edges.
- ri and rj are coprime numbers, making possible Nc (cycle number) be maximum, i.e., Nc = (ri + rj) / gcd(ri, rj) = 16.

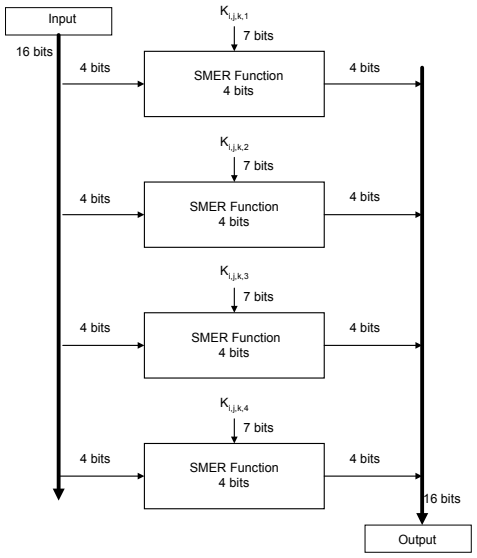

Figure 6: FS function based on SMER functions

The possible configurations of ri and rj that satisfy Nc = 16 are shown in Table 1.

Table 1. Possible Configurations of ri and rj.

ri	rj	Configuration
1	15	000
3	13	001
5	11	010
7	9	011
9	7	100
11	5	101
13	3	110
15	1	111

The ri and rj combination is called SMER configuration. Each SMER configuration has sixteen states, as shown in Figure 7 (a). The node is associated to four bits and these bits are the SMER function input, as illustrated in Figure 7 (b). The SMER function input determines the initial state. The SMER key contains two parts, as shown in Figure 7 (c). The configuration part, ``b7 b6 b5", represents the possible SMER configurations, i.e, determines the combination of ri and rj. The cycle number part, ``b4 b3 b2 b1", determines how many steps will be necessary in order to obtain the final state. This final state produces the four-bit output. More information can be found in [10].

As example, suppose that the SMER key is ``0010100". The bits ``001" determine the SMER configuration, therefore ri = 3 and rj = 13. The cycle number is supplied by the bits ``0100". Assigning ``1010" to the SMER input, we obtain the SMER output ``0110", as shown in Figure 7 (a). The process of retrieving data works on the same way, however the bits associated to cycle number need to be changed by its complement. In the above example, the new SMER key will be ``0011011". For this new key and the SMER input being ``0110", the SMER output will be ``1010",i.e., the same value that was apply on the first step of this example.

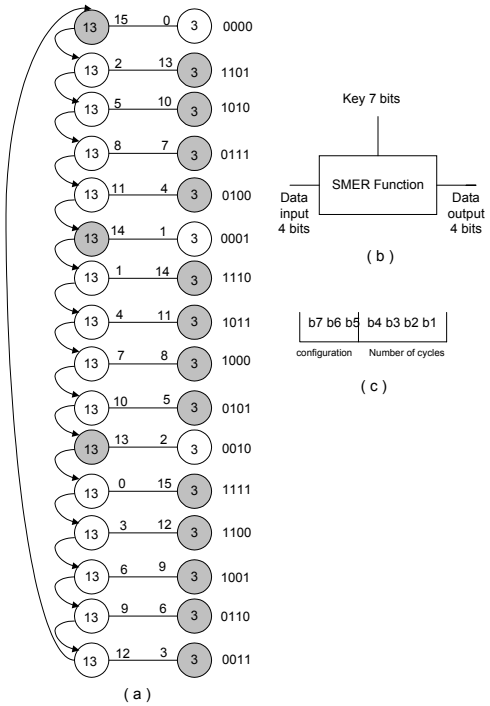

Figure 7: SMER Function

B.2 Security of GODZUK

There are three powerful cryptanalytic attacks against block ciphers. They are (I) exhaustive key search, (II) linear cryptanalysis and (III) differential cryptanalysis. GODZUK is enough strong against these attacks.

Against exhaustive key search

Suppose that there is a machine performs about 10^{+6} cipher and decipher steps per second, and a set of those machines working together in the same time. Therefore, it will take $3{,}17 \times 10^{+18}$ years for a exhaustive key search attack.

Against linear and differential cryptanalysis

The strength of a cipher against linear and differential cryptoanalysis is show by two security parameters - (I) “linear” probability [12] and (II) “differential” probability [13].

Table 2 shows that the “linear” probability and “differential” probability of GODZUK, GODZILLA and KASUMI [14], that is one of standard algorithm adopted by 3GPP (in 2000).

Table 2. Linear and differential cryptanalysis

Algorithm	Linear Probability	Differential Probability
GODZUK	$\leq 1.0x2^{-71}$	$\leq 2^{-56}$
KASUMI	$\leq 1.0x2^{-71}$	$\leq 2^{-56}$
GODZILLA	$\leq 1.0x2^{-132}$	$\leq 2^{-112}$

V. PERFORMANCE OF GODZUK

Initially, the GODZUK cryptographic algorithm was described in VHDL at the structural level resulting in the GODIZUK Soft-core. It can be used to synthesize hardware implementation of the GODZUK.

The soft-core was implemented using a 0,7μm two metal CMOS technology. This implementation resulted in a clock frequency (typical case) of 78 MHz and in a ASIC core, from synopsys tools, equal to 9870 equivalent gates. The final encryption rate is about 2,63 Mbits/s, which fully satisfies 3GPP rules. The table 3 compares GODZUK and KASUMI [14] performances.

Table 3: GODZUK and KASUMI performances.

Algorithm	Equivalent gates	Throughput Mbits/s
GODZUK	9870	2,63
KASUMI	9620	2,03

VI. CONCLUSIONS

The difficulty to define a standard cryptographic algorithm by 3GPP for the third generation of cellular telephony resides in the compromise between security, complexity (measure in equivalent gates) and performance (required in the executions of f8 and f9 functions).

It was shown that, GODZUK is categorized as a symmetric cipher algorithm. It operates with 64-bit data (cipher or decipher) and 128-bit secret-key. It has provable security against differential cryptanalysis and linear cryptanalisys.

GODZUK performances (complexity and throughput) are reasonably fast and small in hardware to attend 3GPP requirements.

GODZUK showed that could be used as an algorithm for authentication, confidentiality and integrity in cellular mobile systems.

VII. REFERENCES

[1] V. K. Garg, P. E. S. Halpern e K. F. Smolik, "Third Generation (3G) Mobile Communications Systems", published in International Conference on Personal Wireless Communication, 1999.

[2] W. Mhor e S. Onoe, "The 3GPP Proposal for IMT-2000", published in IEEE Communication Magazine, December, 1999.

[3] S. L. C. Salomão, A. C. C. Vieira et al, "SCOB, A Soft-core for the Blowfish cryptographic algorithm" published in the proceedings of the XII Brazilian Symposium on Integrated Circuits, October de 1999.

[4] S. L. C. Salomão, et al "Hipcrypto: A high-performance VLSI cryptographic chip", published in the proceedings of the 11th Annual IEEE International Asic Conference (ASIC98), September, 1998.

[5] 3rd Generation Partnership Project, Technical Specification Group Services and Systems Aspects, 3G Security, "Cryptographic Algorithm Requirements", 3G TS 33.105, version 3.2.0-1999

[6] "General Report on the Design, Specification and Evolution of 3GPP Standard Confidentiality and Integrity Algorithm".

[7] "Report on the Evaluation of 3GPP Standard Confidentiality and Integrity Algorithm", ETSI/SAGE 3GPP specification, number AS WG3 S3-000104-1999.

[8] Barbosa, V., An Introduction to Distributed Algorithms. MIT Press, 1996.

[9] Salomão, S. L. C., "Architectures in Hardware to Cryptographics Accelerators", Doctor Thesis, COPPE/UFRJ, Rio de Janeiro, Brazil, 2000.

[10] Salomão, S. L. C. et al, " Improved IDEA" published in the proceedings of the 13th Symposium on Integrated Circuits and System Design -–SBCCI 2000, September 2000, Brazil.

[11] Schneier, Bruce, "Applied Cryptography", Second Edition, John Wilwy & Sons inc., 1996.

[12] Nyberg, K.: "Linear Approximation of Block Ciphers", Proceedings of Eurocrypt´94. Springer-Verlag 1995.

[13] Nyberg, K., Knudsen, L.: "Provable Security against Differential Cryptanalysis", Journal of Cryptology, vol. 8, No.1, 1995.

[14} Salomão, S. L. C., et al. "Hardware Implementation of KASUMI Cryptographic Algorithm for Third Generation Mobile Systems", published in the 1st IEEE South-American Workshop on Circuits and Systems – SAWCAS´2000, nov, 2000, Brazil

[15] França, F. M. G. " Scheduling weightless systems with self-timed Boolean networks. Workshop on Weightless Neural Network, April 1996.