Computer Science Journal of Moldova, vol.**11**, no.2(32), 2003

Mobile Communications World: Security Implementations Aspects – A State of the Art

Nicolas Sklavos Odysseas Koufopavlou

Abstract

Wireless Communications have become a very attractive and interesting sector for the provision of electronic services. Mobile networks are available almost anytime, anywhere and the user's acceptance of wireless hand-held devices is high. The services, are offered, are strongly increasing due to the different large range of the users' needs. They vary from simple communications services to special and sensitive purposed applications such as electronic commerce and digital cash. It is obvious that in future wireless protocols and communications environments (networks), security will play a key role in the transmitted information operations. This paper summarizes key issues that should be solved for achieving the desirable performance in security implementations and focuses alternative integrations approaches for wireless communications security. It gives an overview of the current security layer of wireless protocols and presents the performance characteristics of implementations in both software and hardware. We also propose some efficient methods to implement security schemes in wireless protocols with high performance. The purpose of this paper is to provide the state-of-the-art and research trends on implementations of wireless protocols security for current and future wireless communications.

Keywords: Cryptography, Wireless Communications, Encryption Algorithms, Network Security, Hardware Implementations.

1 Introduction

While the wireless devices are coming to the offices and houses, the need for strong secure transport protocols seems to be one of the most important issues in the mobile standards. From email services to cellular provided applications, from secure internet possibilities to banking operations, cryptography is an essential part of the today's users needs [1]. Recent and future mobile communication systems have special needs for cryptography. They must support the three basic types of cryptography: Bulk Encryption, Message Authentication and Data Integrity [2]. Most of the widely used wireless systems support all the above different types of encryption. Additionally, some systems offer to the users the choice to select among two or three alternative ciphers for each encryption operation. The user can select the best-suited algorithm according to the application needs. In most of the cases, the same encryption system implementation supports all the three different types of cryptography. The standards for mobile applications and services are maturing and new specifications in security systems are being defined. This leads to a large set of possible technologies that a service provider can choose. Although organizations and forums seem to agree to the increasing need for secure systems with wide strength, cryptography is still a big black hole in the wireless networks because of the implementation difficulty. The security layers of many wireless protocols use outdated encryption algorithms, which have been proved unsuitable for hardware implementations, especially for wireless hand-held devices. In general, the ciphers use large arithmetic and algebraic modifications, which are not appropriate for hardware implementations. That's why ciphers implementations allocate many of the system resources, in hardware terms, in order to be implemented as components. So, in many cases software applications have been developed, in order to support the security and cryptography needs. But, the software solution is not acceptable for the cases of hand held devices and mobile communications with high speed and performance specifications.

2 Security Layers of Wireless Protocols

In the recent years, many wireless protocols and unwired communications systems were proposed. Some of them are already in use with a wide range of customers. In order to study the implementation issues of their security layers, a detailed analysis of most of these is needed. In the next paragraphs, the most widely used wireless protocols and their security layers are described briefly.

Wireless Application Protocol (WAP) is the de-facto world standard for the presentation and delivery of wireless information and telephony services on mobile phones and other wireless terminals. The Wireless Transport Layer Security (WTLS) is the layer of the WAP protocol dedicated to the security. It supports privacy, data integrity and message authentication. Applications such as e-commerce and online banking demand advanced level of wireless communications security. The WTLS is based on the philosophy of the well known TLS (Transport Layer Security). In Figure 1 the WTLS architecture is shown. Powerful encryption algorithms have been chosen to support the three different cryptographic operations. Bulk encryption uses DES, IDEA and RC5, message authentication is based on RSA, Diffie-Hellman and Elliptic Curve. MD5 and SHA are the used hash functions.

Bluetooth is a wireless communication system that defines the way that portable computers, cellular telephones and a variety of other devices can be connected using low-power and short range wireless links. The Bluetooth specification includes security features. The system supports authentication and encryption processes. These features are based on a secret link key that is shared by a pair of devices. To generate this key a pairing procedure is used when the two devices communicate for the first time. The encryption of the payloads is carried out with a stream cipher called Eo that is re-synchronized for every payload. This algorithm is based on a method derived from the summation stream cipher generator, attributable to Massey and Rueppel. The authentication function proposed for Bluetooth, is a computationally secure authentication code, often called MAC. The algorithm used for this type of encryption is SAFER+. It was one of the contenders for

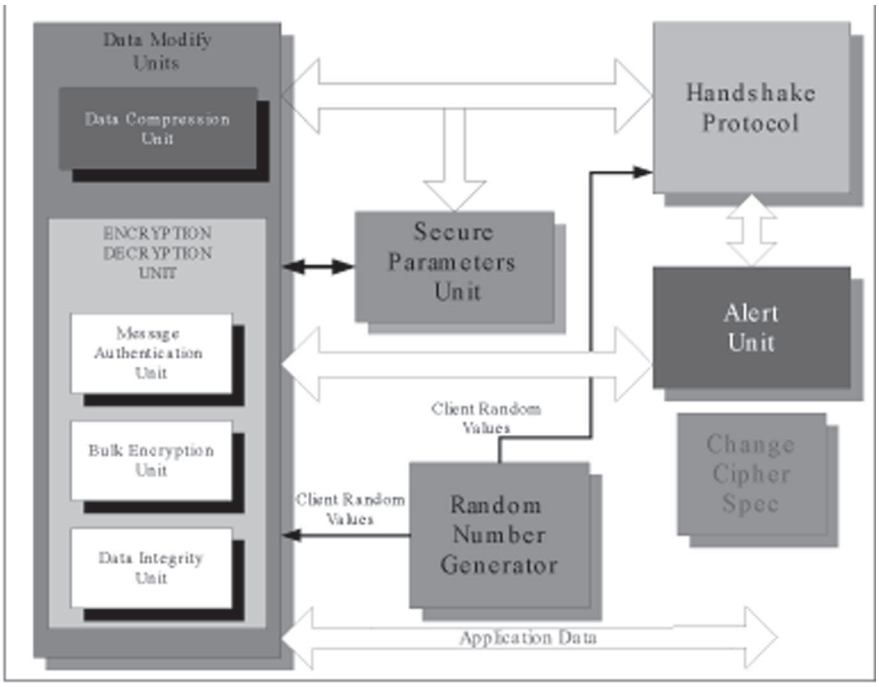

Figure 1. WTLS Security Layer Architecture

the Advanced Encryption Standard (AES) submitted by Cylink, Corp Sunnyvale, USA. This algorithm is an enhanced version of an existing 64-bit block cipher SAFER-SK 128.

HIPERLAN is ETSI's wireless broadband access standard. The specifications of this communication protocol define an encryption/decryption part for optional use. All the HIPERLAN MAC entities use common keys for the encryption algorithms that the protocol supports (Figure 2). These are called HIPERLAN key set. Every key of the set is described unique with its own identifier. In the security part of the protocol the ciphertext is produced by an XOR procedure over the plaintext. The encryption function except the keys, requires also a random sequence generation. ETSI claims that the security part

that has been adopted in HIPERLAN utilizes the level of protection of a wired LAN. It is not easy to give details of the security level and the strength of the privacy that the HIPERLAN supports because the specified ciphers are not commercially available yet.

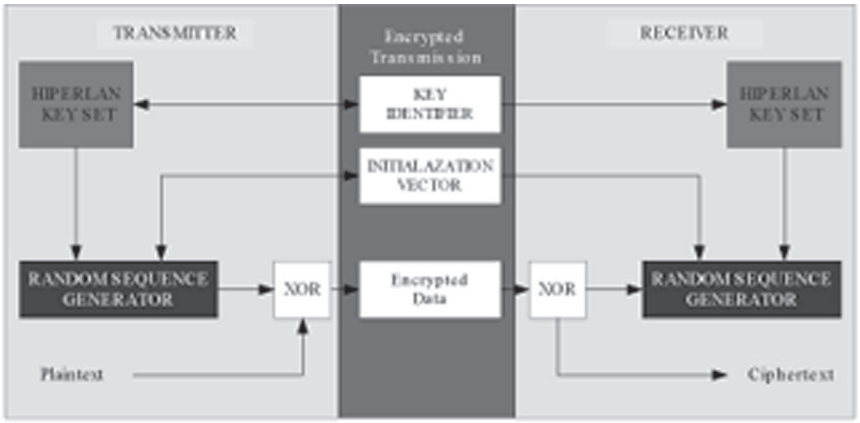

Figure 2. HIPERLAN Security Scheme

The IEEE 802 standards committee formed the 802.11 Wireless Local Area Networks Standards Working Group in 1990. The Wireless Local Area Network (WLAN) specifies an optional encryption part that is named Wireless Encryption Privacy (WEP) [4], which is illustrated in the next Figure 3. This security part supports the defence line of the protocol against the external attacks. Theoretically, an eavesdropper with the appropriate compatible radio modem could listen the transmissions of the protocol users. WEP tries to keep out these unwelcome interferences in the protocol's established communications. The encryption strength that WEP offers is under US export control. In external markets, like Singapore, the 64-bit RC4 is used with a 40 bit secret key. The latest attempts of the US show that the 128-bit encryption is going to be adopted for the encryption mode. The authentication function uses the same key with the encryption. The use of the common key for both security operations imports a high level

risk for the protocol. The authentication works efficiently only if the WEP is supported by the WLAN. Without the encryption mode the authentication procedure is cancelled.

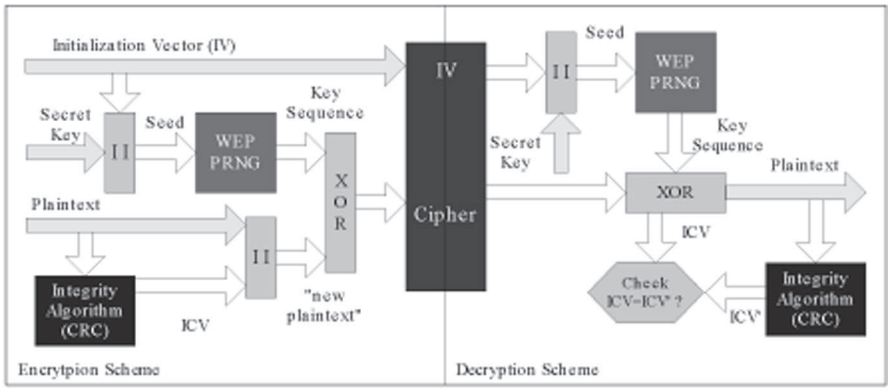

Figure 3. Wireless Encryption Privacy (WEP)

3 Security Software Developments

Today, the most complicated cryptographic systems have been implemented in software than in hardware. One major reason is the implementers increased knowledge in software programming, than in hardware design. Software tools are widely spread with low prices, while VLSI CAD commercial tools are only on interest of large companies and specified research groups. Individual users and class projects are restricted to software possibilities. Almost all the encryption algorithms have been implemented in assembly and in other languages compilers such as C++ and Java. For many years, the majority of the applied development techniques were related to the sequential applications than those related to the real time systems. Lately, programmers made hard efforts in order to find efficient solutions to the implementation problem by using different software compilers. Today, a programmer can develop a real time system in many software languages such as

Ada, Modula2 or Occam. These languages contain constructs for programming concurrency, which make them suitable for large real time systems. At the same time, new compilers appear in the foreground, especially with many possibilities for real time developed systems. Erlang is a good example of such a language compiler. A team of employees at the Swedish communications firm Ericsson developed this language. It's used to write huge real-time control programs for telephone exchanges and network switches. One basic advantage of Erlang is the language support code replacement in a running system. It allows new versions of code's functions to be executed at the same time. This is very useful in the non-stop systems, telephones exchanges, air-traffic control systems, etc, where the operation cannot be stopped to make changes in software. In such systems encryption algorithms and security schemes are also included, and can be easily modified and updated "on the fly" due to this certain language specifications. Another primitive characteristic is that this language supports three constructs for detecting run time errors.

One basic advantage of software is that the development of an operation such as encryption is significantly easy process. Many cryptographic libraries exist and someone can prototype ciphers with no special effort and no long waste of time. On the other side, a hardware implementation needs much effort and a lot of time for designing and testing. Generally the software developments performance is much slower than typical network bandwidths. Fast hardware systems are implemented and are projected to achieve speeds for encryption/decryption processes, comparable to the network bandwidths. In Table I, the software performance characteristics of the most well-known cryptographic algorithms are shown.

The ciphers have been implemented in Java (using the Cryptix Library) in one run interpreted on a Pentium II/266 Linux System [5]. The encryption transformed plaintext data is 1 Mbyte. The works from the other research groups [6-7] have also been included into the above table.

Another comparison study of ciphers software implementation has been done in C++ [8]. The algorithms' performance has been analyzed

Algorithm	Encryption Time (ms)	Rate(Kbit/s)
IDEA	43409	193
SAFER	41442	202
Blowfish	20506	409
TripleDES	160807	52
Loki91	31071	269
RC2	43329	193
Square	29610	283
RC4	12945	648
DES	48629	172
CAST5	23772	352
SHA-1 [6]	-	4.23 Mbps 41.51 Mbps
SAFER+ [7]	-	25.6

Table I. Software Implementation Performance of Encryption

in a Pentium Processor and their performance characteristics are illustrated in Table II. These measurements have been taken for needed clock cycles per output byte.

Encryption Algorithm	Clocks/Byte
RC4	7
SEAL	4
BLOWFISH	18
KNUFU	20
RC5	23
DES	45
IDEA	50
TRIPLE DES	108
RIJNDAEL	32

Table II. Ciphers Software Performance Comparison

4 Encryption Algorithms and Hardware Devices

The applications increasing demand for computation power, and the power reduction requirements for portable devices, force us to consider

that general-purpose processors are no longer an efficient solution for mobile systems. So, new hardware approaches are needed in order to implement some computational heavy and power consuming functions in order to meet the current network speed requirements. Such approaches are:

Recent Application-Specific Integrated Circuits (ASIC) technology was the solution that created better opportunities for implementing real-time and more sophisticated systems. ASICs devices guarantee better performance, with enough small dedicated size. The reliability reaches high limits and the turnaround time is fast. The implementations in these modules are characterized of tighter design security than any other type of devices. ASICs include several custom and simicustom hardware designs.These devices are based on Programmable Logic Devices (PLD), Gate Arrays (GA) and Standard Cells (SC). In our case ASICs can be described as follows: custom-designed hardware, specially tailored to one particular encryption process. They require a significant initial investment for design and testing. If such a device doesn't produce in mass quantities, it is not economical for the market. ASICs seem to be more suitable for dedicated applications and not for an extended purposed encryption system.

Between the software applications and the ASICs devices there is a middle ground. This area is covered by the Field Programmable Gate Arrays (FPGAs) and Complex Programmable Logic Devices (CPLDs). These components provide reconfigurable logic and they are commercially available at low prices. They support the benefits of the customisable hardware and they are software-driven implementations. Of course, these devices vary in capacity and performance. The main disadvantage of them is that they are not suitable for the implementation of large functions. Programmable logic has several advantages over custom-hardware. It is less time-consuming, for the development and the design phase, than the custom-hardware approach. These devices are more flexible than ASICs. They can be reused for cryptanalysis of many different encryption algorithms with little extra effort.

Another solution to the implementation platform problem is smart cards. This issue has to do more with fit than with performance.

In smart cards the RAM requirements are more important, than the clock's frequency. Most commodity smart cards CPU today include 128 to 256 bytes of on-board RAM. Each CPU family contains members with more RAM capacity and a correspondingly higher cost. Although some CPUs include enough RAM to hold the keys of the algorithms, it is often not realistic to assume that such a large fraction of RAM can be dedicated solely to the encryption function. In a smart-card operating system, encryption is a minor small part of the system and will not be able to use more than half of the available RAM. Obviously, if an encryption/decryption system does not fit on a certain CPU, with particular configuration of its components, the performance of the system is unrealistic. Even if an algorithm fits onto smart card, the encryption function will not be able to use all of the RAM capacity. For example, an algorithm that needs at about 100-bytes RAM seems to fit in a 128-bytes smart card. Of course this is a theoretical result because there are RAM requirements also for the control procedure that handles the total security process and these requirements increase the need of the memory-limited capacity. It is clear that the devices of this category are not proper for large encryption systems with special specifications.

In general, hardware implementations have been proved better approaches compared with the software developments, in the terms of throughput, and operating frequency. Of course the covered area resources is a factor that have to be under consideration. The covered area and the performance results of some good hardware implementation examples of ciphers are shown in Table III.

For all the hardware devices there are some common factors that make the implementation of the ciphers in powerful hardware engines a very hard process. The most critical of them is the large number of registers for key storage, which are used by most of the algorithms. As it has been already mentioned above, the security of ciphers is a function of two factors: the strength of the algorithm and the length of the key. While the strength of a cipher is a fixed factor since algorithm's definition, the key length is a parameter that can vary. Ciphers' introducers and cryptographers use large keys for more secure operations. This means larger number of buffers and storage units and larger

Encryption Algorithm	Device Type	Area Cost	Frequency (MHz)	Throughput (Mbit/sec)
RSA [9]	ASIC	47.61 mm^2	80	0.301
IDEA [10]	ASIC	50.01 mm^2	25	178
DES [11]	FPGA	741 CLBs	100	400
Elliptic Curve [12]	FPGA	1290 CLBs	45	0.031
SAFER + [13]	FPGA	6068 CLBs	50	640
Rijndael [14]	ASIC	32.50 mm^2	55	610
Triple-DES [15]	ASIC	1225 mm^2	105	6.7 Gps
Twofish [16]	ASIC	35000 gates	66	200
Kasumi [17]	FPGA	749 CLBS	35.35	71

Table III: Hardware Implementation Characteristics of Encryption Algorithms

memory requirements, for hardware integration. This event has finally cost in the chip's covered area and sometimes in the I/O devices of the system. In order to face this problem RAM blocks are mainly used in hardware implementations. However, in many cases the availability of RAM usage is restricted. The internal memory capacity of many hardware devices is limited. The use of external RAM reduces the total system performance and increases the system's covered area. All these factors are critical items, which must be taken care of special attention of the designers. The application itself defines each time the impact grade of these factors.

5 Alternative Solutions for Security Implementations

The problem of hardware implementation is a function of two different factors: cryptographic algorithms architectures and the efficient integration of them. All forums and organizations in the wireless communication world have specified security layers/systems and have published the selected ciphers that these systems are based on. In order the security with high-level strength to be ensured, three schemes of encryption must be applied in a communication handshake: Bulk Encryption, Message Authentication and Data Integrity. The wireless protocols have defined alternative ciphers in each type of the above schemes. Large encryption systems have been mainly implemented only in software. In hardware devices have been integrated only encryption algorithms separately in different devices and only a few simple encryption systems [18], [19].

The previous years, the hardware integration approach to the issue of security implementation was the ASICs solution. Implementations on these modules achieve high-speed performance and have been proved confidant solutions. Although in the case of wireless protocols, this implementation aspect is proved unfeasible. The hardware integration of a set of ciphers, that a protocol defines, consults to a very large circuit. Encryption algorithms implementations, that have been published until know in ASICs, cover an area of 40-60 mm^2 each. For example, the WAP cipher set integration (eight algorithms in total), in one or more ASICs needs an area about 400-480 mm^2, plus the space needed for the total control unit and routing allocated area. Such an ASIC device is very difficult to be designed and manufactured. Of course the cost of the chip is increased dramatically in this case.

Nowadays a flexible encryption system, which would support the operation of a set of ciphers integrated in the same module, can be implemented with hardware and software cooperation. This type of cooperation could be achieved efficiently by the principles of reconfigurable computing. A proposed solution is the design of a reconfigurable cryptographic system, which will support at least bulk and message

authentication encryption. Reconfigurable computers are those machines that use the reconfigurable aspects of Reconfigurable Processing Units (RPUs) and FPGAs to implement a system/algorithm. The algorithms are partitioned into a sequence of hardware implementable objects (hardware objects). This type of objects represent the serial behaviour of the algorithm and can be executed sequentially. The design technique based on hardware objects offers to the developer/designer a logic-on-demand-capability that is based on the reconfigurable computing. The appropriate software, in order to suit the application at hand, modifies the architecture of these computing platforms. This means that within the application program a software routine has been written to download a digital circuit (chip design) directly into the RPU. The main idea of these designs is the alternation among static and dynamic performance of the system.

Static circuitry is the part of the operation performance that remains in action between the different configurations of the system. This must be in the maximum of the design possibilities and much attention must be paid for its optimization. General-purpose blocks, such as adders, belong to this part of performance. Another example of the static parts is the storage units. These are the parts of each system that never are been changed during different operations. Always they maintain the characteristics that the initialization process has set. On the other hand, there is the dynamic circuitry. With this term, we mean the parts of the system that change during configuration. These blocks must be minimized in order to increase the system performance. If there are not basic common parts between the selected algorithms, the dynamic circuitry is high enough and this is bad for the system operation. Dynamic circuitry increases the demands for the systems resources and decreases its attribution. It has been cleared that the selection of similar ciphers would be approved a critical factor of the design hardware implementation.

Of course there are no many choices for similar ciphers' architectures in the technical literature. In order to achieve this, the designer of a powerful security system has to choose one flexible algorithm for bulk encryption with the ability to operate as a hash function (data

integrity). The addition of some extra parameters in the algorithm's architecture is necessary for the efficient operation of the two encryption modes. In this way, the needs of the system resources are reduced. At the same time, we have to avoid ciphers' with heavily arithmetic functions such as multiplication and modulo processes. These operations have difficultly been implemented in hardware devices and they have no commonality.

The implementation of a security system with some common basic parts, which can be used for the implementation of ciphers' common functions, seems to be the more sophisticated alternative solution for a large encryption engine. With the term basic parts we mean "heavy" algebraic or logical components of the algorithms' architectures. In most of the cases it is difficult to implement these parts in a hardware device, with high-speed performance and minimized covered area. An example is the multiplication modulo that IDEA algorithm needs [10]. Reconfigurable computing method is proved efficient enough to solve the implementation problem of encryption engines and is suitable for the different types of architectures that ciphers' have. Of course, the specifications of the application itself would prove this method as a good or best solution.

The latest years, implementations on the smart cards devices have been very attractive for the hardware designers. Compared with the other hardware devices like ASICs and FPGAs, smart cards have limited computing power and minimized storage capacity. Therefore, security applications which allocate a huge amount of storage or which require an extensive computation power might cause conflicts.

The persistent storage of a smart card is limited to a few kilobytes today, which prevents it from storing larger items on the card. This can be circumvented if the smart card delegates the storage of the item in an external environment. The smart card receives and processes the transmitted data. It encrypts them and saves them to the sender/receiver's device external storage units (RAM, registers of general use). Later, when these data are needed again, the smart card can request it from these storage units. By using this described method the smart card internal storage requirements can be reduced significantly.

However, we have to take care that we do not create another bottleneck: the communication speed of the smart card is not very high and so we should have to handle the transmission of the same data back and forth, with special care.

Another limitation of smart cards is the small processing power. The appropriate data modifications, due to encryption/decryption, may possibly exceed the computing power of a smart card. In this case it will take unacceptably long to finish the appropriate data transformation. Thus, it is important to minimize the amount of computation power, which the smart card has to pay for the requested tasks. For such applications, it is better the design to be kept as simple as possible. The requested task can be divided in smaller parts with no hard processing specifications. The requested round keys for encryption/decryption can be generated in the initialization procedure and not at the same time with the encryption round transformation (on the fly key generation). In this way, we avoid to spend extra processing power for the key expansion unit during encryption/decryption. The same methodology can be followed for the appropriate specified constants generation.

6 Wireless Communications Security in the near Future

The needs for personal wireless communications systems are growing rapidly. Coupled to this increase is the telecommunication-related crime. In unwired networks, an invader with suitable receiver can intercept the transfer data. It is clear that such systems, although have specified a satisfactory level of security, are vulnerable. Security is a primary requirement of all wireless cryptographic protocols. Cryptographic algorithms are meant to provide secure communications applications. However, if the system is not designed properly, it may fail. Although there are many well know ciphers, with different specifications and characteristics, the security of some of them is under consideration. Many works, from different research groups, have been

published in technical literature in which cryptanalysis methods have been applied in order to find out any existing black holes in the security strength of the encryption algorithms. From many points of view, such attempts offer valuable knowledge in the growth and the improvement of cryptography. Encryption algorithms have to perform efficiently in a variety of current and future applications, doing different encryption tasks. The algorithms should be used to encrypt streaming audio and video data in real time. They would have to work correctly in 32 and 64-bit CPUs. Many of today's applications run with smart cards based on 8-bit CPUs, such as burglar alarms, pay-TV and general other meters. All hardware implementations have to be efficient, with less allocated area resources. This means simplicity in algorithm's architectures with enough "clever" data transformation components. A wireless protocol implementation demands low power devices and fast computation components, which imply that the number and complexity of the encryption operations should be kept as simple as possible. A basic transformation in the operation of the encryption algorithms is needed, including modifications in the data blocks and key sizes.

The ciphers of the future have to be key agile. Many applications need a small amount of text to be encrypted with keys that are frequently changed. Many well known applications, like IPsec, use this way of algorithm's operation. Although the most widely used mode of operation is encryption with the same key for all the amount of transport data, the previous mode is also very useful for future applications. Ciphers that require subkeys precomputation have a lower key agility due to the computation time, and they also require extra RAM to hold the subkeys. This RAM requirement does not exist in the implementations of algorithms, which compute their keys during the encryption/decryption operation. Cellular phones technology demands hard specifications of the cryptography science. Ciphers have to be compatible with wireless devices restricted standards in hardware resources. New mobile phones will have proper encryption part built in them. In these devices there is not enough room for a large integrated security layer. A solution in order to decrease the required hardware resources is to use the ciphers for both bulk encryption and

data integrity, with a simple change of their operating mode. All the above make the effort of design of new security algorithms for wireless applications a real hard process. But this is the only way to change the pure today's security wireless standards in order to succeed in solving the implementation problem. The AES and SHA-2 hash function standards are very good steps for the design of communications security schemes, for the next decades.

The technology growth gives many promises for the security future. If the strength of the applied cryptography that is used in wireless industry were increased enough, the protocol security would be efficient to withstand the attempts of the attackers. Today many ciphers can support the defense of the communications links in external invaders. On the other hand, the implementation of these is a hard process and sometimes cannot meet the wireless network requirements. This is due to the fact that today's used ciphers have been designed some years ago and for general cryptography reasons. They are not specialized for the wireless communications. Security improvement needs strong, flexible encryption algorithms with efficiently performance. New algorithms must be designed for this type of applications. In addition the new ciphers' designs require invention. They are notoriously difficult to demonstrate or otherwise establish trust in. On the other hand, everything should be demonstrated in software before committing to hardware.

References

[1] Ueli Maurer, *"Cryptography* 2000 ± 10*"*, Informatics - 10 Years Back, 10 Years Ahead, Lecture Notes in Computer Science, Springer-Verlag, vol. 2000, pp. 63–85, 2001.

[2] Bruce Schneier, *"Applied Cryptography – Protocols, Algorithms and Source Code in C"*, second edition, John Wiley and Sons, New York, 1996.

[3] ISAAC Research Group at the University of California, Berkeley, http://www.isaac.cs.berkeley.edu/isaac/gsm.html.

[4] S.H. Park, A. Gaz and Z. Ganz, *"Security protocol for IEEE 802.11 wireless local area network"*, Mobile Networks and Applications 3 (1998), pp. 237–246.

[5] L. Brown, *"A current Perspective on Encryption Algorithms"*, Presented at the UniforumNZ'99 conference in NZ, April 1999.

[6] Michael Roe, *"Performance of Block Ciphers and Hash Functions-One Year Later"*, proceedings of Second International Workshop for Fast Software Encryption '94, Leuven, Belgium, December 14–16, 1994.

[7] J. L. Massey, G. H. Khachatrian, M. K. Kuregian, *"SAFER+ Cylink Corporation's Submission for the Advanced Encryption Standard"*, First Advanced Encryption Standard Candidate Conference, Ventura, CA, August 20–22, 1998.

[8] Bruce Schneier, Doug Whiting, *"Fast Software Encryption Algorithms for Optimal Software Speed on the Intel Pentium Processor"*, Proceedings of Fast Software Encryption Workshop 1997.

[9] H. Nozaki, M. Motoyama, A. Shimbo and S. Kawamura, *"Implementation of RSA Algorithm Based on RNS Montgomery Multiplication"*, Proc. Of CHES 2001, LNCS 2162, pp. 364–376, 2001.

[10] R. Zimmermann, A. Curiger, H. Bonnenberg, H. Kaeslin, N. Felber and W. Fichtber, *"A 177 Mbit/s VLSI Implementation of the International Data Encryption Algorithm"*, IEEE Journal of Solid State Circuits, Vol. 29, No 3, March 1994.

[11] J.-P. Kaps, C. Paar, *"Fast DES Implementation for FPGAs and its Application to a Universal Key-Search Machine"*, 5th Annual Workshop on Selected Areas in Cryptography (SAC '98), August 17–18, Queen's University, Kingston, Ontario, Canada.

[12] K.H. Leung, K.W. Ma, W.K. Wong and P.H.W. Leong, *"FPGA Implementation of a Microcoded Elliptic Curve Cryptographic Processor"*, proc. of Field-Programmable Custom Computing Machines (FCCM'00).

[13] Kitsos, N. Sklavos and O. Koufopavlou, *"Hardware Implementation of the SAFER+ Encryption Algorithm for the Bluetooth System"*, proceedings of IEEE International Symposium on Circuits & Systems (ISCAS'02), Vol. IV, pp. 878–881, USA, May 26–29, 2002.

[14] Gaj and Pawel Chodowiec, *"Fast implementation and fair comparison of the final candidates for Advanced Encrytpion standard using Field Programmable Gate Arrays"*, Proc. RSA Security Conferences, San Fransisco, CA, April 8-12, 2001.

[15] D.C. Wilcox, L.G.Pierson, P.J.Robertson, E.L. Witzke, and Carl Gass, *"A DES ASIC Suitable for Network Encryption at 10 GPS and Beyond"*, proc. of CHES'99, pp. 37–48, 1999.

[16] Yeong-Kang Lai, Liang-Gee Chen, Jian-Yi Lai, and Tai-Ming Parng, *"VLSI Architecture Design and Implementation for Twofish Block Cipher"*, proceedings of IEEE International Symposium on Circuits & Systems (ISCAS'02), USA, May 26–29, 2002.

[17] K. Marinis, N. K. Moshopoulos, F. Karoubalis, and K. Z. Pekmestzi, *"On the Hardware Implementation of the 3GPP Confidentiality and Integrity Algorithms"*, Proceedings of the 4th International Conference for the Information Security, ISC 2001, Malaga, Spain, pp. 248-265, October 1–3, 2001.

[18] J. Goodman and A. P. Chandrakasan, *"An Energy-Efficient Reconfigurable Public-Key Cryptography Processor"*, IEEE Journal of Solid-State Circuits, Vol 36, No 11, November 2001.

[19] S. Dominikus, *"A Hardware Implementation of MD-4 Family Hash Algorithms"*, proceedings of ICECS 2002, Croatia 16–18, 2002.

N.Sklavos, O.Koufopavlou, Received May 27, 2003

Electrical and Computer Engineering Department
University of Patras, Patras, Greece
E–mail: *nsklavos@ee.upatras.gr*