

# On the Hardware Implementation Cost of Crypto-Processors Architectures

**Nicolas Sklavos**

Technological Educational  
Institute of Patras, Greece

---

**ABSTRACT** A variety of modern technologies such as networks, Internet, and electronic services demand private and secure communications for a great number of everyday transactions. Security and cryptography provide a huge set of primitives, methods, and operation modes to support the special needs of data transmission. This paper aims to introduce aspects of design, architecture, and implementation of crypto-processors. It is aimed to demonstrate efficient realizations of cryptographic mechanisms and tools in terms of hardware integration. Computational methodologies, computer arithmetic, and encryption algorithms need deep investigation and research to obtain efficient integrations of crypto-processors, with desirable improvements and optimizations. Approaches on silicon achieve high values of speed and bandwidth. VLSI design is determined with FPGA and ASIC devices, which are two alternative design methodologies for implementing crypto-processors, with several advantages such as flexibility, high performance, and fast time to market. Reconfigurable computing techniques can change the system architectures to several different modes of operations without sacrificing design efficiency or performance.

**KEYWORDS** crypto-processor, computer architecture, VLSI design, network security, cryptography

---

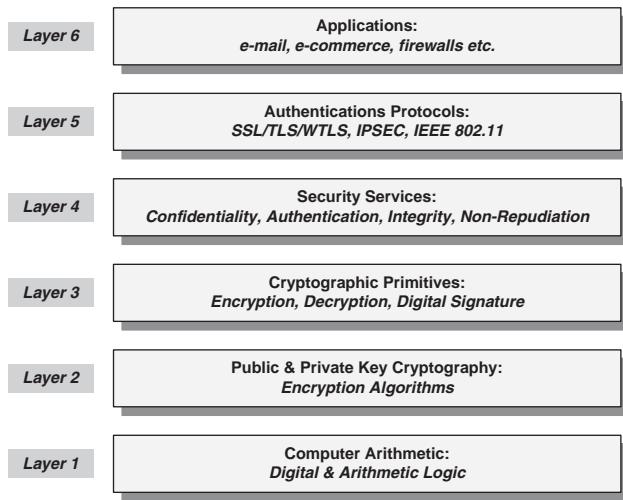
## INTRODUCTION

The special needs for secure communications are triggered not only in traditional sectors such as military and government services but also in all aspects of everyday life, in both business and private transactions. The continued growth of electronic services and the rising number of communications subscribers must be secured against attackers, which commonly aim to unauthorized usage of devices and services or modification of the transmitted data.

In Figure 1, the six-layer hierarchical model of information security and its applications are shown.

Computer arithmetic is the fundamental primitive in this model (Figure 1), and it is placed in the first layer. With the term computer arithmetic, we mean all the operations of arithmetic and digital logic such as addition, subtraction, multiplication, division operations, and ADD, OR, NOT, and XOR transformations. All security systems of high performance are built on flexible and efficient implementation of such arithmetic and logic operations.

Address correspondence to Dr. Nicolas Sklavos, Informatics & MM Dept, Technological Educational Institute of Patras, Branch of Pyrgos, Rhga Feraiou Street, 27100, Pyrgos.  
E-mail: nsklavos@ieee.org



**FIGURE 1** Six-layer hierarchical security model.

Private and public key algorithms such as RSA, DSA, elliptic curve, and AES, DES, and IDEA have been designed as a mix of layer 1 operations. Encryption algorithms belong to the layer 2 of this model. They form the two basic cryptographic primitives one layer above. Encryption/decryption and signature/verification are the straight and converse transformation of layer 3. Layer 4 contains the fundamental security services that a wireless protocol must support: confidentiality, integrity, authentication, and nonrepudiation. Different authentication protocols can be implemented to support the special needs of networks security: SSL, TLS, WTLS, IPSEC, IEEE, and 802.11. These security protocols define the layer 5. They are the fundamental base of the common use applications such as e-mail, e-commerce, electronic cash, and so forth. These applications are placed at the top, layer 6, of the model.

Most of the widely used networks support alternative security schemes. Additionally, some systems offer to the users the choice of selection among two or three ciphers for each one of the encryption operations. The user can select the best-suited algorithm according to the application needs. In most cases, the same security system implementation supports all the different aims 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 from which a service provider can choose.

Although organizations and forums seem to agree on the increasing need of securing systems with high

security strength, cryptography is still a big black hole in wireless networks because of implementation difficulties. The security layers of many wireless protocols use outdated encryption algorithms, which have proven unsuitable for hardware implementation, especially for wireless handheld devices. In general, the ciphers use large arithmetic and algebraic modifications, which are not appropriate for hardware implementation. That is why ciphers implementations allocate many system resources, in hardware terms, in order to be integrated as components. In many cases, software applications have been developed to support the security and cryptography needs. However, the software solution is not acceptable for handheld devices and mobile communications with high speed and performance specifications.

## DESIGN AND IMPLEMENTATION APPROACHES OF CRYPTO-PROCESSORS

Current applications demand high speed processors for a great amount of data that must be transformed in real-time terms. Software approaches could be a good choice since they have low cost and require a short development time. The low values of performance are a forbidden factor for possible software implementation.

On the other hand, hardware alternatives could be selected for implementing Crypto-Processors architectures. Both Application Specific Integrated Circuits (ASICs) and Field Programmable Gate Arrays (FPGAs) can support high data rates, although such designs are more time consuming and expensive compared with the software alternative.

A detailed comparison of Hardware Vs Software solutions for implementing Crypto-Processors architectures is introduced in Figure 2.

Based on the elements of the comparison, hardware solutions are better in most of the cases than the software alternatives. The main advantages of software are the low cost and the short time to market. Low performance is a fundamental drawback of software integrations.

### Crypto-Processors Architectures of Wireless Protocols

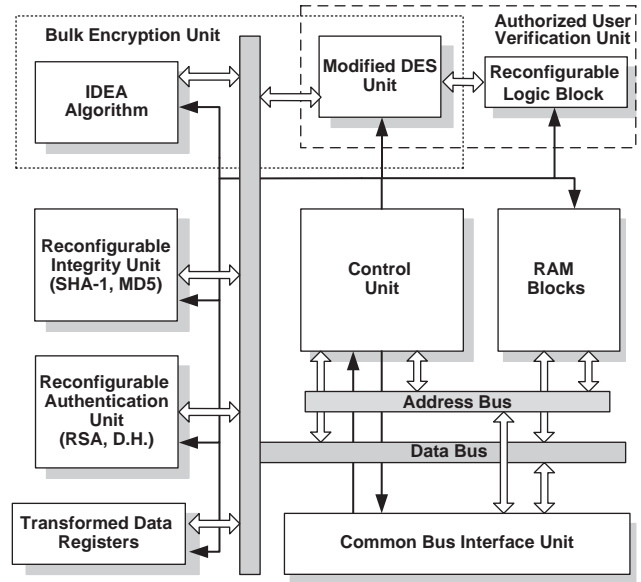
In the recent years, many wireless protocols and unwired communications systems were proposed.

	SOFTWARE	HARDWARE	
		FPGAs	ASICs
Performance	Low	Low	High
Power Consumption	Depends	Very High	Low
Logic Integration	Low	Low	High
Tool Cost	Low	Low	Low
Test Development Complexity	Very Low	Very Low	High
Density	High	Very Low	High
Design Efforts	Low-Medium	Low-Medium	High
Time Consumed	Short	Short	High
Size	Small-Medium	Small	Large
Memory	Fine	Fine	Fine
Flexibility	High	High	-
Time to Market	Short	Short	High
Run Time Configuration	-	High	-

**FIGURE 2 Hardware Vs software alternatives for Crypto-Processors.**

Some were already in use with a wide range of customers. 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 security. It supports privacy, data integrity, and message authentication. Applications such as e-commerce and online banking demand advanced levels of wireless communications security. The WTLS is based on the philosophy of the well-known Transport Layer Security (TLS). A proposed Crypto-Processor for the WTLS implementation is presented in Figure 3.

The introduced system has been designed like a typical processor with data path, memory, I/O interface, and control unit. Six different ciphers are supported by the proposed Crypto-Processor. DES and IDEA algorithms are selected for the Bulk Encryption Unit. The Reconfigurable Integrity Unit performs efficiently in two different operation modes, for SHA-1 and MD5 hash functions. The operations of both RSA and Diffie-Hellman are performed by the Reconfigurable Authentication Unit. An extra security scheme is also supported by the proposed Crypto-Processor. A Reconfigurable Logic block, in cooperation with the



**FIGURE 3 WTLS Crypto-Processor architecture.**

Modified DES Unit, implements the Authorized User Verification Unit. A common data bus of 64-bit and a 32-bit address bus are used for the internal data transfer purposes. Two different storage units have also been integrated. The appropriate for the algorithms keys are stored and loaded in the RAM blocks, while the transformed data are kept as long as necessary in the Transformed Data Registers. A Common Bus Interface Unit, which supports 32-bit input data and 32-bit address buses, has been implemented for the Crypto-Processor to communicate efficiently with the external environment. This environment may be a general purpose processor or a special CPU.

The Wired Equivalent Privacy (WEP) scheme has been adopted by IEEE 802.11 standard to ensure security for the transmitted information. The basic two components of WEP are the Pseudorandom Number Generator (PRNG) and the Integrity Algorithm. The PRNG is the most valuable component because it actually is the original encryption core. WEP adopts RC4 cipher as the PRNG unit and Cyclic Redundancy Check (CRC-32) as the Integrity Algorithm. Although WEP is a good security scheme, the offered security in some cases cannot satisfy user demands. To ensure a higher security level, 802.11i working group is introduced as the protocol security scheme, the Advanced Encryption Standard (AES). The proposed architecture for the implementation of the Wired Equivalent Privacy (WEP) scheme is illustrated in Figure 4.

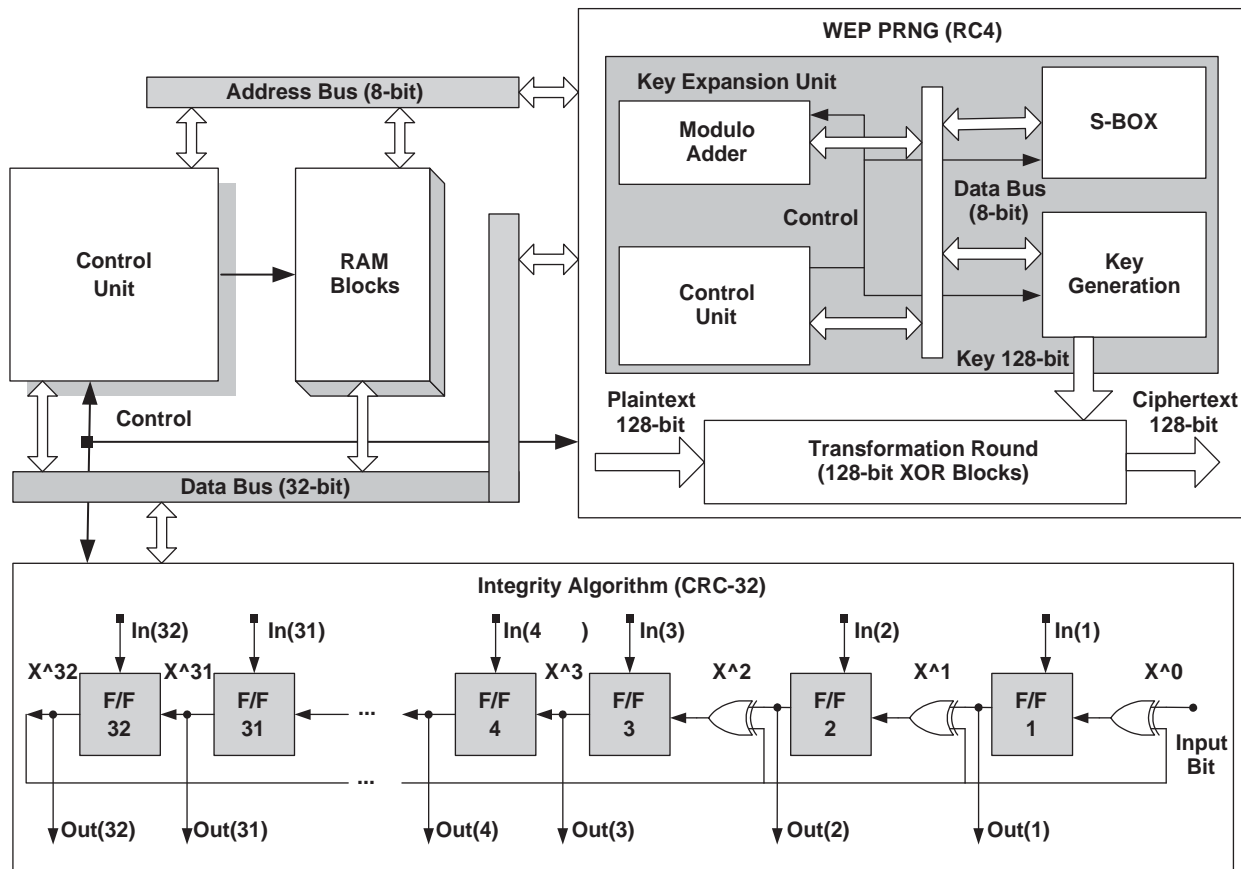


FIGURE 4 WEP Crypto-Processor architecture.

## CRYPTO-PROCESSORS OF SECURITY STANDARDS

AES (Figure 5) is the latest world standard for cryptography. The AES scheme architecture operates each time on a column of 32-bit data. It needs 41 clock cycles to complete the transformation of a 128-bit plaintext block. The column subunit is composed of 4 basic building blocks: S-Box, DataShift, MixColumn, and KeyAddition. The RAM-based design for the S-BOXes ([256x8]-bit) guarantees high performance. This “column”-based architecture minimizes the area resources compared with “State”-based architectures.

A proposed Crypto-Processor for SHA-2 hash family standard implementation is illustrated in Figure 6. It performs the three different SHA-2 functions (–256, –384, and –512). The control unit is totally responsible for the system operation. It defines the proper constants and operation word length, manages the ROM blocks, and controls the proper algebraic and digital logic functions for the operation of SHA-2 hash functions. The Hash Computation Unit is the main datapath

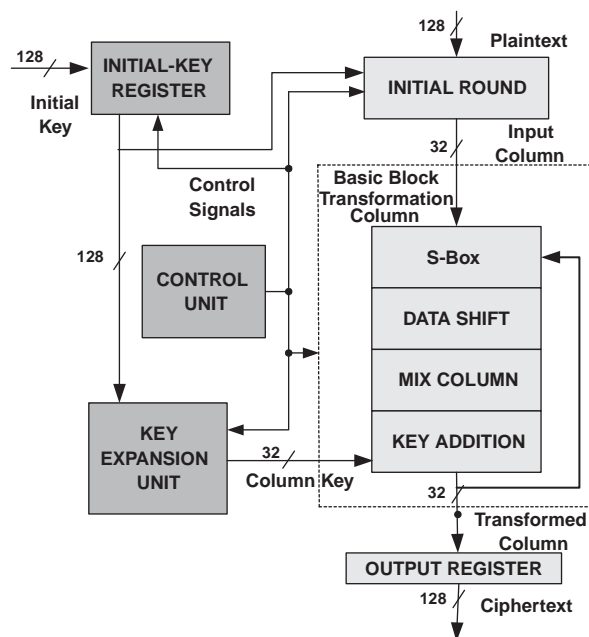
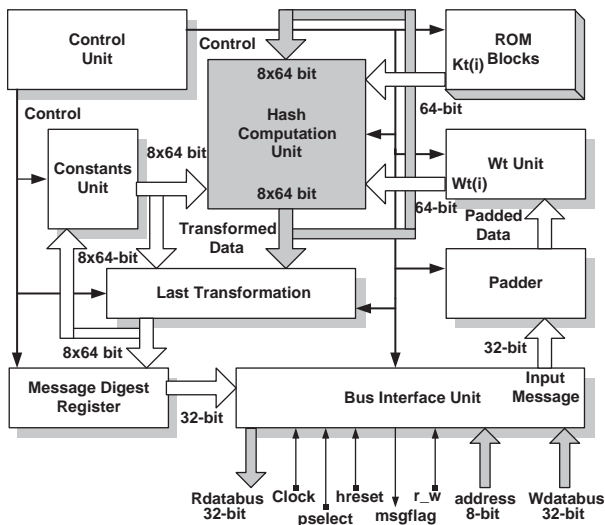


FIGURE 5 Advanced Encryption Standard (AES) Crypto-Processor architecture.



**FIGURE 6** SHA-2 (256, 384, & 512) Crypto-Processor architecture.

component of the system architecture. The specified number of the data transformation rounds, for each one of the SHA-2 hash family functions, is performed in this component with the support of a rolling loop (feedback). The transformed data are finally modified in the last transformation, which operates in cooperation with the constants unit. In this way, the message digest is produced and is stored into the Message Digest Register.

## CRYPTO-PROCESSORS ARCHITECTURES AND HARDWARE DEVICES

The applications increasing demand for computing strength 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. New hardware approaches are needed to implement computational heavy and power consuming functions to meet current network speed requirements.

Recent Application-specific Integrated Circuits (ASICs) 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 with fast turnaround time. The implementations in these modules are characterized by tighter design security than any other type of devices. ASICs include several custom and semicustom hardware designs. 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 does not 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). These components provide reconfigurable logic and are commercially available at low prices. They support the benefits of the customizable hardware and are software-driven implementations. Of course, these devices vary in capacity and performance. The main disadvantage is that they are not suitable for implementation of large functions. Programmable logic has several advantages over custom-hardware. For example, it is less time-consuming for the development and the design phase than the custom-hardware approach, and the devices are more flexible than ASICs.

In general, hardware implementations have been proven better approaches compared with software developments in the terms of throughput and operating frequency. Of course, the covered area resources is a factor that must be considered.

For all the hardware devices, there are some common factors that make the implementation of the ciphers in powerful hardware engines a hard process. The most critical of these factors is the large number of registers for key storage, which are used by most algorithms. As has been mentioned earlier, 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 because of the 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 memory requirements for hardware integration. This event increases the cost of the chip in system resources terms; covered area and sometimes in the I/O devices of the system. To deal with 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 into consideration by the

designers. The application itself defines each time the impact grade of these factors.

## ALTERNATIVES APPROACHES AND EFFICIENT SOLUTIONS FOR SECURITY ARCHITECTURES

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 and have published the selected ciphers on which the systems are based. In order 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 mainly have been implemented only in software. In hardware devices, only encryption algorithms have been integrated separately in different devices and only a number of multipurpose encryption systems have been integrated.

In the previous decade, the hardware integration approach for the security implementation was only the ASICs solution. Implementations on these modules achieved high-speed performance and have been proven a confident solution. Although in the case of wireless protocols, this implementation aspect is not feasible. 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 now in ASICs cover an area of 40–60 mm<sup>2</sup> each. For example, the WAP cipher set integration (eight algorithms in total) in one or more ASICs needs an area about 400–480 mm<sup>2</sup>, plus the space needed for the total control unit and the routing allocated area. Such an ASIC device is difficult to design and manufacture, dramatically increasing the cost of the chip 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 (RPU) and FPGAs to implement a system. The algorithms are partitioned into a sequence of implementable objects (hardware objects). These types of objects represent the serial behavior of the algorithm and can be executed sequentially. The design technique, based on hardware objects, offers the developer a logic-on-demand capability 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). 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 to its optimization. General-purpose blocks, such as adders, belong to this part of the performance. Another example of the static parts is the storage units. These are the parts of each system that should never be changed during different operations. They should always 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 system performance. If there are no basic common parts between the selected algorithms, the dynamic circuitry is the greatest part of the design and this is a disadvantage for the system operation. Dynamic circuitry increases the demands for system resources and decreases system attribution. The selection of similar ciphers would be a critical factor in the design hardware implementation.

There are not 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 are difficult to implement in hardware devices and 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 most 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 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 for IDEA algorithm. The reconfigurable computing method is proven efficient to solve the implementation problem of encryption engines and is suitable for the different types of cipher architectures. The specifications of the application itself would prove this method a good or even best solution.

Recently, implementations on the smart card devices have been attractive for hardware designers. Compared with other hardware devices such as ASICs and FPGAs, smart cards have limited computing power and minimal storage capacity. Therefore, security applications that allocate a huge amount of storage or require extensive computation power might cause conflicts with the whole system operation.

The persistent storage of a smart card is limited to a few kilobytes today, which prevents the card from storing larger items. 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 the data and saves them to the sender/receiver's device's external storage units (e.g., RAM, registers of general purpose). Later, when the data are needed again, the smart card can request the data 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 not to create another bottleneck: the communication speed of the smart card is not very high, and we have to handle the back-and-forth transmission of the same data with special care.

Another limitation of smart cards is their 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 time 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 that the design be kept as simple as possible. The requested task can be divided into 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 spending extra processing power for the key expansion unit during encryption/decryption. The same methodology can be followed for the appropriate specified constants generation.

## SECURITY IN THE FUTURE

Technology growth offers many promises for information security's future. If the strength of the applied cryptography that is used in the wireless industry were increased enough, network security would be efficient to withstand attackers' attempts. Today many ciphers can support the defense of the communications links in external invaders. On the other hand, the implementation of these defenses is a hard process and sometimes cannot meet the wireless network requirements because the current ciphers were designed years ago and for general cryptographic reasons. They were not specialized for wireless communications. Security improvements need strong, flexible encryption algorithms with efficient performance. Modern algorithms must be designed for this type of application. In addition, the new cipher designs require invention. They are notoriously difficult to demonstrate or trust. Everything should be demonstrated in software before committing to hardware.

## REFERENCES

- Clemments, A. (2000). *The principles of computer hardware* (3rd ed.). Oxford: Oxford University Press.
- Henney, J. and Patterson, J. (2003). *Computer architecture*. Morgan Kaufmann Publishers.
- Mano, M. M. (1988). *Computer engineering: Hardware design*. Englewood Cliffs, NJ: Prentice Hall.
- Rodriguez-Henriquez, F., Saqib, N. A., Diaz Perez, A., and Kaya Koc, C. (2006). *Cryptographic algorithms and reconfigurable computing*. Springer.
- Schneier, B. (1996). *Applied Cryptography – Protocols, Algorithms and Source Code in C* (2nd ed.). New York: John Wiley and Sons.
- Sklavos, N. and Zhang, X. (2007). *Wireless security & cryptography: Specifications and implementations*. Boca Raton, FL: CRC-Press.

- Sklavos, N. and Touliou, K. (2007). A system-level analysis of power consumption & optimizations in 3G mobile devices. Proceedings of the 1st International Conference on New Technologies, Mobility & Security (NTMS'07). Springer, pp. 225–235.
- Sklavos, N. and Koufopavlou, O. (2002). Architectures and VLSI implementations of the AES-proposal Rijndael. IEEE Transactions on Computers, 51(12), 1454–1459.
- Sklavos, N. and Koufopavlou, O. (2005). Implementation of the SHA-2 hash family standard using FPGAs. Journal of Supercomputing, 31(3), 227–248.
- Sklavos, N., Kitsos, P., Papadopoulos, K., and Koufopavlou, O. (2006). Design, architecture and performance evaluation of the wireless transport layer security (WTLS). Journal of Supercomputing, 36(1).