The Journal of Supercomputing, 36, 33–50, 2006


# Design, Architecture and Performance Evaluation of the Wireless Transport Layer Security

N. SKLAVOS nsklavos@ieee.org
P. KITSOS
*Electrical and Computer Engineering Department, University of Patras, Patras, Greece*

K. PAPADOPOULOS
*Siemens, A.E., Athens, Greece*

O. KOUFOPAVLOU
*Electrical and Computer Engineering Department, University of Patras, Patras, Greece*

**Abstract.** Communication protocols for wireless networks have specified security layers, with high-level encryption strength. The dedicated to security layer of Wireless Application Protocol (WAP), is the Wireless Transport Layer Security (WTLS). In this paper, an efficient architecture for the hardware implementation of WTLS is proposed. The introduced system supports bulk encryption, authentication and data integrity. The proposed architecture operates alternatively for a set of ciphers, IDEA, DES, RSA, D.H., SHA-1 and MD5. It is based on two reconfigurable design units: the Reconfigurable Authentication Unit and the Reconfigurable Integrity Unit. These units operate alternatively for different ciphers and achieve to allocate minimized resources, at the same time. The introduced security system has been implemented in an FPGA device. The supported ciphers performance is compared with previously published works, and it has been proven superior to them, in most of the cases. The system's synthesis results prove that the proposed architecture is a flexible and powerful solution for WTLS integration of today's and future wireless networks. The system can be applied to wireless communications servers and mobile devices also. Finally, the proposed architecture can be used as a powerful security engine, in WAP communication networks, with special security demands.

**Keywords:** WTLS implementation, WAP security, wireless networks, wireless communications, security system

## 1. Introduction

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 devices is high. One of the major scopes of the wireless protocols and especially of WAP [43] is to bring the Internet applications to mobile devices [16, 23, 25, 27]. Security is a key issue in the world of electronic communication, especially for the sensitive purposed services such as electronic commerce and online banking. New ciphers have been developed [1], in order to support the networks defense against to the increasing range of attacks. Optimizations of the security layers specifications have also been published the last year [46]. However, the time overhead due to the data encryption should not impose an intolerant penalty in the communication process. Almost all today's ciphers have an important drawback: the slowness of their operation, due to the mathematic and logic transformations. Since

software implementations are too slow, even running in fast processors, the use of specific hardware modules seems to be the only reasonable solution for the requested high performance. Different designs have been proposed for the hardware implementation of ciphers [4–7, 10, 12, 13, 18, 21, 29, 31, 36, 41, 44, 48]. These works present the implementation of one cipher in a hardware device at a time. Modern applied cryptography in the wireless communications networks, demands powerful encryption engines with special purposes of privacy, authentication and integrity. In order to support efficiently these security needs, hardware security engines have to be implemented in a single hardware module, due to the mobile devices restricted available resources. It is necessary for the security layers of the wireless protocols, a set of ciphers to be integrated in the same chip. The ciphers implementations in different hardware devices, one for each algorithm, are proved insufficient and forbidden solution in wireless world.

An efficient architecture, for WAP security layer implementation, is proposed in this paper. The introduced system supports six different ciphers, in both architecture and security purposes: IDEA, DES, RSA, Diffie-Hellman, MD-5 and SHA-1, in the same hardware module. The proposed architecture has been implemented in an FPGA device. The synthesis results prove that the integrated ciphers performance is very high, and better compared with ciphers separate implementations [4–7, 10, 12, 13, 18, 21, 29, 31, 36, 41, 44, 48], in most of the cases. IDEA architecture uses a modified transformation round, which minimizes the allocated area resources by about 30%, compared with other works [7, 36, 48]. DES proposed implementation performs better, with a range from 200 to 400%, compared with the other related works [5, 13, 21, 44]. Especially, the proposed DES architecture has been designed with a slight modification, in order to operate alternatively as an Authorized User Verification Unit. The introduced Reconfigurable Authentication Unit performs efficiently for both RSA and Diffie-Hellman and decreases the covered area by about 70% in total, compared with the case of two separate hardware implementations, one for each cipher. The proposed Reconfigurable Integrity Unit, achieves to minimize the allocated area resources and performs efficiently for two different operation modes: SHA-1 and MD5. Both SHA-1 and MD5 performance is better at about 50 to 300% compared with the conventional implementations [10, 12, 41]. The proposed system can be applied as a powerful solution for the WTLS implementation in wireless networks. It can be used for both server providers, and mobile devices. Furthermore, the system can be used as a powerful security core, in wireless communications supporting privacy, authentication and integrity. The high-speed performance and the minimized resources ensure the suitability and usability of this architecture, due to the limitations, that wireless networks specify.

This work is organized as follows: In Section 2 the Wireless Transport Security Layer (WTLS) is introduced. In Section 3 the proposed architecture is presented. Section 4 is dedicated to verification and testing. The synthesis results of the FPGA implementation and comparisons with other published works are given in the Section 5. Finally, in Section 6 conclusions and observations are given.

## 2. Wireless Transport Layer Security (WTLS)

The Wireless Application Protocol (WAP) is completely a new concept, specified by an industry consortium, the WAP Forum [43]. The protocol layer, dedicated to security, is

the Wireless Transport Layer Security (WTLS) [19, 43]. WTLS is based on the transport layer security (TLS) [39], which is the internet security standard. A number of modifications in TLS were needed, in order for the WTLS to meet finally the specifications, due to the nature of the wireless networks. On the other hand there are a lot of restrictions, in the implementations of the selected ciphers because of the limited processing power, memory and the bandwidth. In a wireless device, only a set of these encryption algorithms can efficiently be integrated, due to these implementation limitations.

In the WTLS, three different schemes of security have been defined: privacy, authentication and data integrity [19, 43]. With the term privacy (bulk encryption), an applied transfer method that ensures a private end-to-end transfer is defined. The specified ciphers for the WTLS bulk encryption process are: IDEA, DES and RC5 [26, 37, 40]. Authentication ensures the identity of every communication party. This is achieved by using digital signatures or electronic certificates. After the authentication, the service provider is sure that the supported service is available to the user who requests to use it. On the other hand, the user can be confident about the service provider with the same way of authentication. The RSA, Diffie-Hellman and Elliptic Curve ciphers [26, 37, 40] serve authentication in WAP. Integrity is used in order to verify that the transmitted data have not been modified, during transmission through the network. This can be guaranteed by calculating checksums from the original transmitted information. Hash functions are widely used for data integrity. In the WTLS, SHA-1 and MD5 hash functions [26, 37, 40] are introduced by the specifications.

## 3. Proposed WTLS architecture

### *3.1. Proposed Crypto-Processor architecture*

The proposed Crypto-Processor architecture, for the WTLS hardware implementation is illustrated in the following Figure 1. The introduced system has been designed like a typical processor with data path, memory, I/O interface, and control unit [30, 32].

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 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 all the transformed data are kept as long as it is necessary in the Transformed Data Registers. A Common Bus Interface Unit, which supports 32-bit input data and 32-bit address buses, has also been implemented, in order the Crypto-Processor to communicate efficiently with the external environment. This environment may be a general purpose processor or a special CPU.

It has to be mentioned, that WAP is intended to be applied mainly in mobile devices. Due to their hardware integration limitations, only a set of ciphers and not all of the

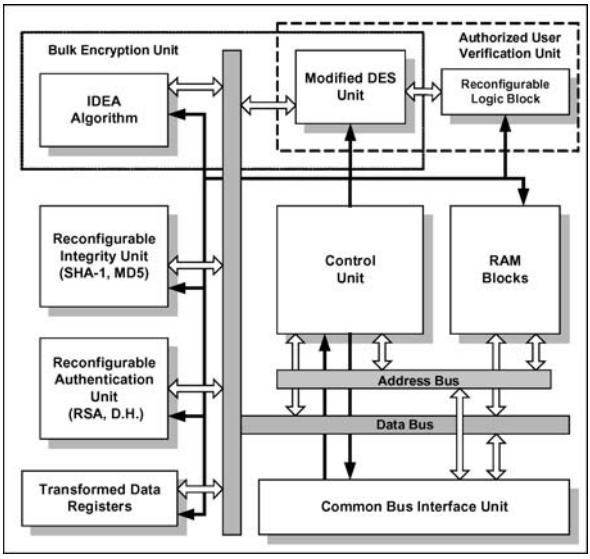


*Figure 1*. Crypto-Processor Proposed architecture.

specified by the WTLS can be implemented in a flexible embedded system [14, 19]. The ciphers set, in the proposed Crypto-Processor was selected by considering security and hardware implementations parameters. As it will be presented in the rest of the paper, according to our study, the integration units of the selected ciphers ensure the highest provided security and the best hardware performance at the same time.

### *3.2. Bulk Encryption Unit*

In the proposed Crypto-Processor, the Bulk Encryption Unit provides the capability of selection between two ciphers: IDEA and DES. According to the security experts' opinion, IDEA is one of the most secure block algorithms available to the public [37]. On the other hand, DES has been established as the Data Encryption Standard and has been proved flexible enough design for VLSI implementations. These algorithms have been preferred, for several reasons, instead of RC5, which is the third cipher, specified by the WTLS. RC5 is a trademark of RSA Data Security [33] and it is expected to be patented in the near future. This cipher has a set of parameters, like key rounds, operation word-lengths and secret key variables, which have to be specified during initialization. All the possible set values seem to be available and usable in theory, but many of them in practice may be forbidden [20]. Detailed analysis is needed, to prove the security level and the hardware performance of each set. Unlike the other encryption algorithms, the parameterized RC5 permits upgrades in the operation, with main goal to increase the supported security level, but with a major disadvantage in the performance and vice versa. In order for the proposed Crypto-Processor to provide alternative capabilities, both IDEA and DES have been integrated. Due to the large amount of data, that a bulk

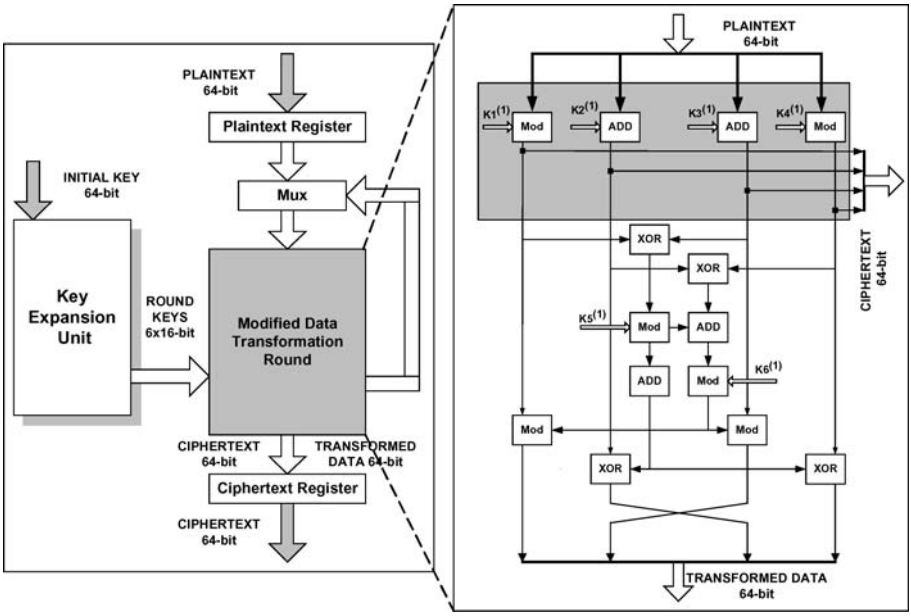


*Figure 2.* (a) IDEA architecture and (b) Modified data transformation round.

encryption unit has to transform, these two ciphers can operate at the same time, or one only at a time. In the case of parallel operation of course the Crypto-Processor performance is increased by a great factor.

In IDEA [24], the required data confusion is achieved by using three different and incompatible group operations. These group components operate on pairs of 16-bit sub-blocks and mix them. The three algebraic operations are: (i) 16-bit XOR, (ii) modulo addition $2^{16}$ and )iii) modulo multiplication $2^{16}$ [8]. The proposed IDEA architecture is illustrated in detail in the following Figure 2.

IDEA defines 4 or 8 basic transformation rounds plus one half-round. In the previous published works [48], both the basic transformation round and the half round have been integrated separately. Such an implementation approach has a major cost in the covered area resources. The proposed IDEA architecture is based on a feedback logic operation mode. As the Figure 2 illustrates, the specified IDEA transformation round architecture is partially modified (Modified Data Round). It operates efficiently for both the basic and the half-round, as the specifications demands [24]. By this proposed architecture the allocated resources of IDEA implementation are reduced at about two modulo multipliers area, which are the fundamental covered area components. It has been measured that by the proposed architecture, 40% area reduction is achieved compared with the design approach applied in [48]. The analytical synthesis results and detailed comparisons will be given in Section 5.

Furthermore, in the proposed IDEA architecture the key expansion unit has also been integrated and the appropriate round keys are generated on the fly. By the Key Expansion Unit integration the needed dynamic key refreshing of the WTLS is achieved, with no extra time delay. In addition, the introduced design has no time delay in the

initialization process for the key setup like the work [48]. According to our study, the separate integrated Key Expansion Unit has a cost at about 7–10% in the total IDEA implementation area resources. In the implementation of [48] the key expansion unit has not been integrated. In our proposed work such an approach is forbidden due to the WTLS specified key refreshing. In the proposed architecture, the key refreshing process is supported during data transformation. A design methodology like [48] causes a great time delay cost for every new key setup. In this way, the system performance in [48] is been dramatically decreased. Analytical time delay cost measurements, for both operations (Key Refresh, Initialization) are presented in Section 5.

DES cipher operation is supported by a 64-bit key [26, 37, 40]. The computation of the key schedule is clearly described by the standards specifications [9]. The algorithm defines 16 rounds. Each data round uses a different key comprised of 48-bit from the initial input key (64-bit). The total key schedule process was analyzed and according to our study is proved that a certain combinational shift register can produce every round key. The key Expansion Unit design in the proposed DES architecture is built on 16 different shift registers and not with a full rolling design technique, used in previous published works [5, 13, 21, 44]. With this applied technique (shift registers), DES performance is increased at about 170% compared with any architecture with a key expansion unit, built on the defined key scheduling logical components [5, 13, 21, 44]. The only drawback of the proposed architecture is the 8% increased covered area resources. According to DES specifications, the decryption keys are the same as those used in encryption mode, if only they are processed in the reverse order. For this reason RAM blocks are used in order the encryption keys to be stored, after their generation. RAM blocks equal to $16 \times 48$-bit are allocated in total. In this way, the system does not generate extra keys for the decryption mode and no extra time for the decryption keys generation is needed.

The common DES architecture has been slightly modified and the proposed architecture, in cooperation with a reconfigurable logic block, operates as a bulk cipher and as an Authorized User Verification Unit. This unit would be analyzed in detail in the next section.

### *3.3. Authorized user verification unit*

In order to have only authorized users accessing to multi-users hand held devices and provided services, Personal Identification Numbers (passwords) are used. In order not to allocate many extra resources for the implementation of an authorized user verification unit, a UNIX method for password verification [15] could be adopted in wireless devices. This method has the advantage that is based on DES cipher, specified by WTLS. This means that it can be applied efficiently in the proposed Crypto-Processor with minimized extra covered area resources. The Authorized User Verification Unit proposed architecture is illustrated in Figure 3.

The random number generator could be implemented, by using the well-known random or pseudorandom generation techniques [37]. In the proposed architecture (Figure 3), the generation of the salt is based on the system clock. In this way, no extra allocated area resources for the integration of the random number generator are needed, compared with the resources that the implementation of the published techniques [37] demand.

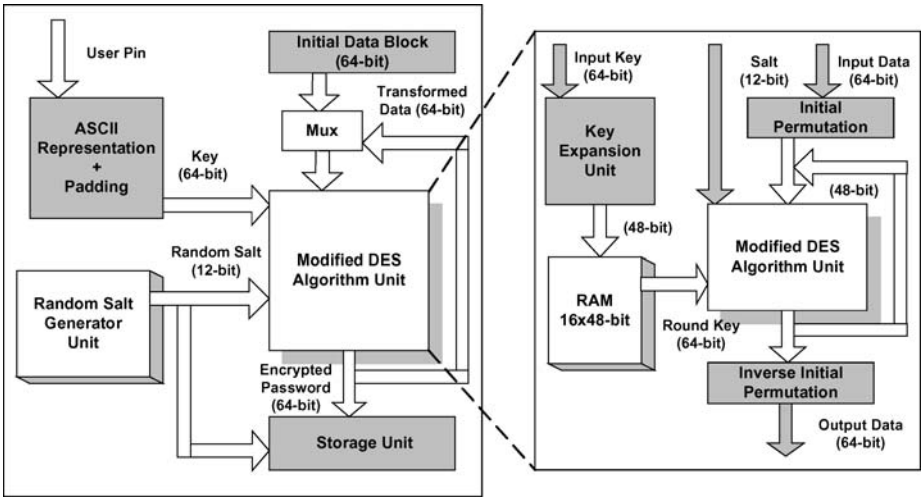


*Figure 3*. Authorized user verification unit proposed architecture.

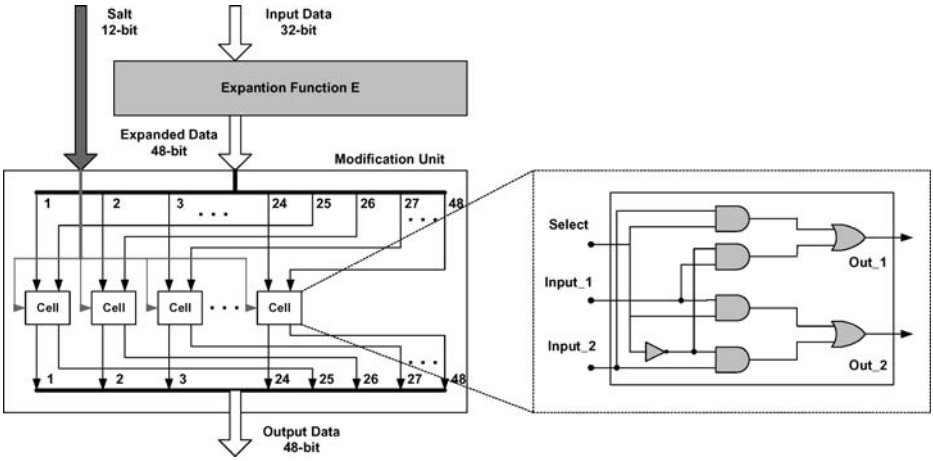


*Figure 4*. Modified function $E$ proposed architecture.

After 25 iterations of data transformation, both the encrypted password and the random salt are stored in the Storage Unit.

Especially, the salt is used to support the modified behavior of DES expansion function $E$, providing 4096 different operation modes. In this way, the security of the Authorized User Authentication Unit is increased dramatically by a factor of $2^{12}$ (=4096). In systems with 8 characters PINs, an attacker has to use a database of $2^{64} \times 2^{12}$ possible passwords for function $E$ variations. In architectural level, the desired variable operation of function $E$ is achieved by the proposed Modified Function $E$. The architecture of Modified Function $E$ is illustrated in the following Figure 4.

The Modified Function $E$ is based on a dynamic combinational circuit, which is called Modification Unit. This unit consists of 24 similar cells. By using the 12-bit salt, $2^{12}$ (=4096) different modification cases on the Expanded Data are achieved. Each bit of the salt is used in two basic cells. For example, salt bit ($k$) is used for basic shells ($k$) and ($k + 12$). In the proposed architecture, every bit of Modification Unit output is predetermined, by a pair of the input data bits (Expanded Data). The basic cell determines the appropriate modification by multiplexing every pair of input bits, with the salt bit as the Select input (Figure 4). If the Select is equa1 to logic one the two associated bits are swapped. Otherwise (Select is zero), no modification takes place in the certain pair of bits. The security strength of the encrypted password is been augmented with the use of the randomly generated salt, and so any possible dictionary attacks become less effective.

The proposed DES architecture (Figure 3) is used alternatively as the original DES cipher core with the appropriate commands of the Crypto-Processor control unit. By the applied design (Figure 4) that was described above, the proposed Crypto-Processor is supported with one more extra security scheme (Authorized User Verification) and the provided security of the system gets higher. The major advantage of the proposed Authorized User Verification Unit is that uses only 2% extra resources compared with the original DES core implementation for bulk encryption, with no performance penalty at the same time.

### *3.4. Reconfigurable message authentication unit*

In order to support the demanded authentication in WAP, a Reconfigurable Authentication Unit is proposed. It operates in two different modes: RSA [35] and D.H. [11]. Due to the fact that their major operations are in common, both ciphers are implemented in the same unit, based on a reconfigurable design. EC algorithm [17] has no major common parts with the other two ciphers. For this reason, EC operation is not supported in the proposed Crypto-Processor, in order to minimize the allocated area resources.

The Reconfigurable Authentication Unit is presented in Figure 5. This proposed architecture is reconfigurable in the sense that it performs efficiently for both RSA and D.H. upon to the user selection and it is not predefined by the Crypto-Processor (Figure 1).

This proposed reconfigurable unit is based on the Array Multiplier. The most widely known algorithm in order both encryption and decryption processes of RSA to be performed is the square and multiply algorithm [22]. A number of works have been published reporting systolic array architectures for modular multiplication. One of the most well-known modular multiplication algorithms is the Montgomery algorithm [28]. Different architectures have been published [4, 6, 18, 29, 42], with alternative implementation criteria (performance, covered area, run time), for modular multiplication applied to RSA hardware integration. Based on work [28] and in combination with the previous works [2, 3] on systolic multiplication, T. Blum and C. Paar [4] introduced a systolic array modular multiplication suitable for hardware implementations. The proposed Reconfigurable Authentication Unit, which is illustrated in Figure 6, is based on square and multiply algorithm [22, 28] and uses the modular multiplication systolic array architecture proposed in [4].

The supported plaintext and keys word-length is equal 512-bit, defined by the WTLS specifications. RSA uses one exponent (A) of 512-bit, while D.H. architecture is based

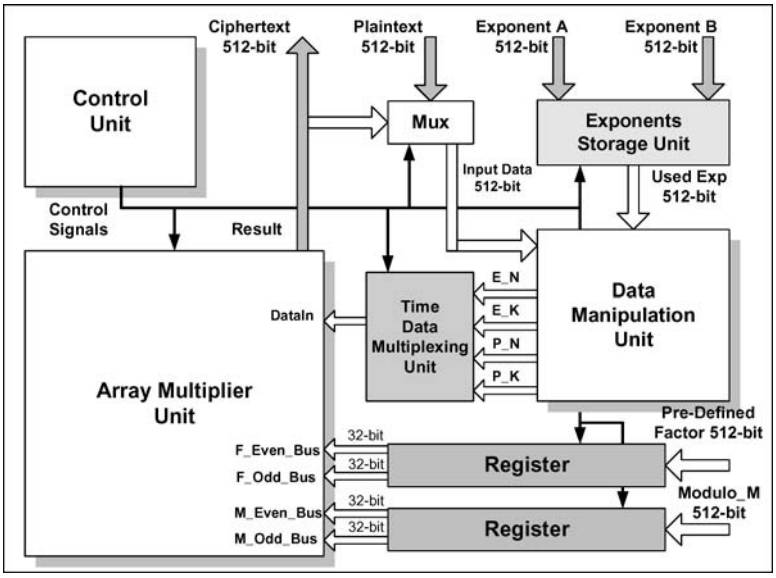


*Figure 5*. Reconfigurable authentication unit proposed architecture.

on two 512-bit exponents (A, B). Both algorithms are based on modular multiplications on an input modular base (M). The applied multiplier architecture of [4] demands an extra pre-computational factor input. (More details for the multiplier operation could be found in [4]).

D.H. operation is based on the same array multiplier unit used in RSA. The only basic difference is that D.H. uses two exponents compared with the one used in RSA. This, results in doubled needed number of clock cycles for a produced cipher of D.H., compared with the RSA requested time for a complete encryption/decryption process. Although, the operating frequency is common for both operation modes, the proposed Reconfigurable Authentication Unit decreases the covered area compared with the case of two separate implementations, one for each cipher, by about 70%. Furthermore, the proposed WTLS Crypto-Processor implementation (Figure 1) is able, without sacrificing the system performance or using extra resources, providing two alternative operation modes for authentication (RSA, D.H.). The analytical synthesis results of the proposed Reconfigurable Authentication Unit implementation are given in Section 5.

### *3.5. Reconfigurable integrity unit*

The proposed architecture for the Reconfigurable Integrity Unit implementation is presented in the next Figure 6.

This proposed architecture is reconfigurable in the sense that operates efficiently for both SHA-1 [38] and MD5 [34]. The selected operation mode (SHA-1 or MD5) in

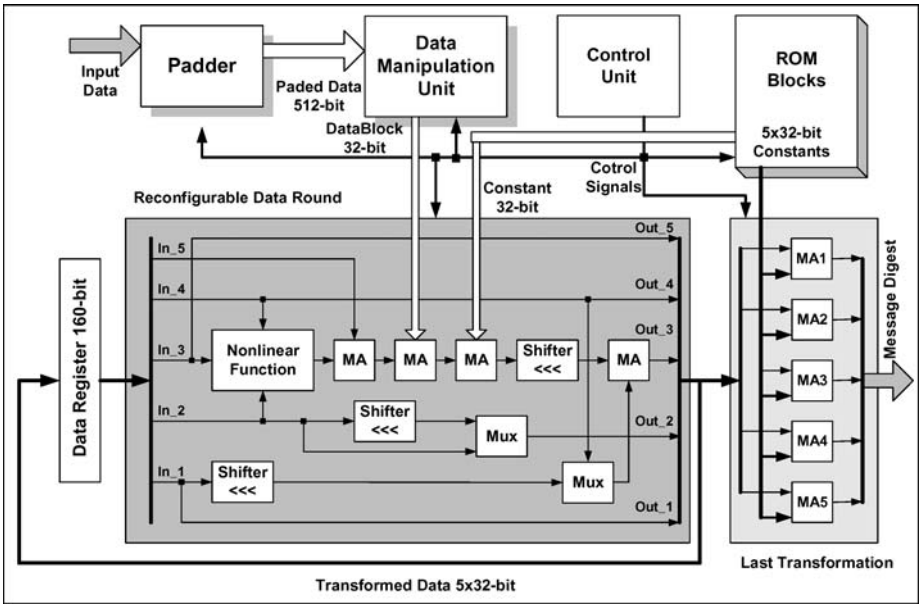


*Figure 6*. Reconfigurable integrity unit proposed architecture.

upon to the user need it's time. First, in the Padder, the input data are padded in order to be a multiple of 512-bit block as both MD5 and SHA-1 specifications define. The padding process is exactly the same procedure, for both SHA-1 and MD5 hash functions, according to the two algorithms specifications. The 512-bit Padded Data blocks are divided to sixteen 32-bit words in the Data Manipulation Unit. Then, these words are processed in order, according to each algorithm specified data manipulation procedures. SHA-1 demands $8 \times 32$-bit ROM, while MD5 requires $68 \times 32$-bit ROM blocks, in order for the specified constants of these hash functions, to be stored. The Reconfigurable Data Round is the most critical component of the Reconfigurable Integrity Unit proposed architecture. It has been designed as a mix of both MD5 and SHA-1 specified transformation rounds. The MA component denotes modulo adder $2^{32}$. The Nonlinear Function is a combination of mathematical functions and digital logic. It performs in two different ways, for SHA-1 and MD5 operation modes. The multiplexers and also the shifters (left circular shift), operate according to the Control Unit commands, in order the two different operation modes (SHA-1, MD5) to be performed. MD5 defines 64 data transformation rounds, while SHA-1 specifies 80 rounds. Finally, the Last Transformation modifies the data. This unit consists of 5 modulo adders $2^{32}$, where modulo additions between the five data inputs and the five 32-bit constants are performed in parallel. The 160-bit SHA-1 message digest is obtained by concatenating the 32-bit outputs of all the modulo adders. In the case of MD5 operation, the 128-bit message digest is equal to the concatenation of the first four modulo adders' 32-bit outputs (MA1 to MA4).

Our proposed Reconfigurable Integrity Unit implementation needs at about 7–10% extra covered area resources compared with a separate SHA-1 or MD5 implementa-

tion. The critical path of this proposed unit (Figure 6) is defined by the Out_3 data arrival time of the Reconfigurable Data Round. The achieved frequency for both SHA-1 and MD5 operation modes is equal to 70 MHz for the proposed implementation. It is important to be noted, that the achieved frequency is reduced by about 2% compared with one of a SHA-1 and a MD5 implementation in the case of two separate hardware devices.

## 4. Verification and testing

The proposed Crypto-Processor architecture (Figure 1) has been captured by using VHDL. All the internal components of the design were synthesized placed and routed using XILINX FPGA device [45]. The system then was simulated again, for the verification of the correct functionality. In order to verify the right operation of the developed system, the test board of Xilinx (XSA Board) was used. This board is shown in the Figure 7.

Initially, the developed architecture is downloaded to the to the FPGA device of this board by the parallel computer port. Then, the required VHDL code that permits emulation of the developed architecture is created, and downloaded to the FPGA device too. The values of the input/output signals were monitored through the help of a logic analyzer, which is connected to the whole board structure. (The PS/2 connector and the CRT port are not used in our case). The test scenarios that are applied to the board, in order to verify systems' correct functionality, are provided by the cipher standards. In addition, during the test procedure a great number of test vectors were used to verify

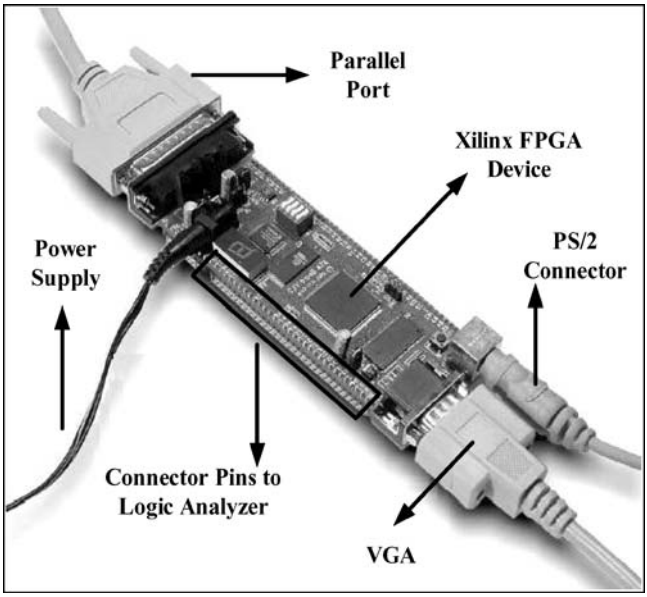


*Figure 7.* Used FPGA board.

the right operation of the received FPGA device samples. These test vectors, were mostly selected in a random way, but there have been included some special values of the input data (for example "FFF...FFF", "000...000") to ensure maximum test coverage.

The proposed Crypto-Processor architecture passed all the above test vectors correctly. In addition, the VLSI synthesis results of the Crypto-Processor hardware implementation are given in the next Section 5.

## 5. Synthesis results and evaluation

The synthesis results of the proposed Crypto-Processor are shown in the Table 1.

In the FPGA proposed implementation RAM, blocks are used, for both keys and specified constants storage. Many FPGAs provide embedded RAM, while external RAM blocks can also be used, in the cases that internal RAM is not available. In such implementations, the switching time of the RAM is a factor that has to be considered in the total performance timing measurements. The proposed Crypto-Processor requirements for RAM blocks are: 128-bit for IDEA, 2048-bit for the Reconfigurable Authentication Unit (RSA, D.H.), 16 × 64-bit for DES and finally 72 × 32-bit for the Reconfigurable Integrity Unit (8 × 32-bit for SHA-1 and 4 × 16 × 32-bit for MD5). In the next Tables 2 and 3 performance comparisons of the proposed Crypto-Processor with other related works, are presented.

The proposed ciphers implementations are compared with the best implementations published in the technical literature. The introduced work of [7] achieves throughput

*Table 1.* Implementation synthesis analysis

| | FPGA DEVICE (Xilinx Virtex v2000ebg560) | | | |
|---|---|---|---|---|
| | Covered area | | | |
| Hardware device | CLBs | FGs | DFFs | F (MHz) |
| System component | | | | |
| IDEA Algorithm | 1852 | 3104 | 380 | 50 |
| DES Unit | 341 | 682 | 170 | 85 |
| Reconfigurable Integrity Unit (SHA-1, MD5) | 1653 | 2905 | 1049 | 70 |
| Reconfigurable Authentication Unit (RSA, D.H.) | 5421 | 8303 | 4056 | 41 |
| Bus Interface | 242 | 413 | 453 | 70 |
| Reconfigurable Logic Block | 117 | 546 | 387 | 70 |
| Data Registers | 1952 | 0 | 680 | 70 |
| Control Unit | 1576 | 3200 | 1200 | 70 |
| Crypto-Processor | 14154 | 20153 | 8375 | – |

D Flip-Flops (DFFs), Configurable Logic Blocks (CLBs), Function Generators (FGs), Frequency: F (MHz).

*Table 2*. Encryption algorithms performance comparison I

| | Implementations performance | | |
|---|---|---|---|
| Architectures | Area (CLBs) | F (MHz) | Through (Mbps) |
| IDEA [36] | 40.561 gates | 8 khz | – |
| IDEA [48] 1.2 $\mu$m | 108 mm$^2$ | 25 | 177 |
| IDEA [7] | 2444 | 82 | 1166 |
| | 2878 | 150 | 600 |
| IDEA proposed | 1852 | 50 | 711 |
| RSA [4] (512-bit) | 2555 | 45.6 | 1,4 |
| | 3413 | | 4,6 |
| RSA [6] 0.8 $\mu$m (512-bit) | 77988 Gates | 50 | 24 Kbps |
| RSA [18] 0.5 $\mu$m (1024-bit) | 105000 Gates | 40 | 20 Kbps |
| RSA [31] 2 $\mu$m (512-bit) | 23000 Opt Cost | 77 | 300 Kbps |
| RSA [29] 1 $\mu$m (512-bit) | 75000 Gates | 25 | 100 Kbps |
| Reconfigurable Authentication Unit (RSA, D.H.) | 5421 | 41 | 1,1 RSA<br>0,5 D.H. |
| DES [44] | 11.1 $\times$ 11.1mm$^2$ | 105 | 10 Gps |
| DES [13] | 50 k transistor | 250 | 1 Gbps |
| DES [5] | – | – | 11.6 |
| DES [21] | 262 | 25 | 99 |
| | 433 | 18 | 148 |
| | 741 | 11 | 184 |
| DES proposed | 341 | 85 | 245 |

*Table 3*. Encryption algorithms performance comparision II

| | Implementations performance | | |
|---|---|---|---|
| Architectures | Area (CLBs) | F (MHz) | Throughput (Mbps) |
| SHA-1 [12] | 1004 | 43 | 119 |
| MD5 [41] | – | 300 | 256 |
| MD5 [12] | 1004 | 43 | 146 |
| MD5 [10] | 880 | 21 | 165 |
| | 4763 | 71.4 | 354 |
| Reconfigurable Integrity Unit (SHA-1, MD5) | 1653 | 70 | 442 SHA-1<br>551 MD5 |

value almost equal to the proposed IDEA design. The basic drawback of this system is the doubled covered area, compared with the proposed FPGA implementation. The presented work in [36] operates with very low frequency compared with the proposed. Although, no other information of the system throughput, and the needed clock cycles for the encryption/decryption process is given in [36]. These omissions in the reported synthesis results of [36] do not ensure a detailed comparison of this work with the proposed and the other conventional IDEA architectures [7, 36, 48]. In the work [48], the round keys are generated internally by using the basic transformation round architecture.

It has to be stated that according to the architecture of the transformation round [48], 4 round keys can be generated during one clock cycle at maximum. The generation of the 104 specified sub-keys demands $104/4 = 26$ extra clock cycles during initialization and $104 \times 16 = 1664$-bit of allocated RAM blocks are needed, for key storage. This time delay has to be considered in the total system performance [48], in addition to the achieved 177 Mbps throughput. If the used initial key is refreshed $N$ times, as WAP specifies, the extra needed time is equal to $N \times 26$ clock cycles. This time delay decreases dramatically the performance of such an implementation [48]. In the proposed IDEA implementation, the Key Expansion Unit has also been integrated separately, and supports the on the fly key generation. Only 128-bit used RAM blocks for the initial key storage are used for keys storage reasons. In addition, the integrated Key Expansion Unit of the proposed IDEA architecture supports the dynamic specified key refreshing of WTLS with no delay penalty at all.

The proposed Reconfigurable Authentication Unit has operating frequency equal to 57 MHz, for both operation modes (RSA, D.H.). The RSA algorithm performance is in general data dependent. The performance values are illustrated in Table 2 and have been measured for 512-bit key and plaintext blocks specified by the WTLS. Different test vectors were used in order to measure the average value of the performance. This was done due to the fact that RSA performance is depended on the number of logic ones that the input key (exponent) may has. The proposed Reconfigurable Authentication Unit has almost the same performance compared with the introduced work of [4]. Although, in the proposed implementation the same array multiplier with [4] is used, our Reconfigurable Authentication Unit allocates 18% more area resources compared with the conventional. This extra area is allocated for the control logic and needed registers, in order the proposed unit to perform efficiently for the two different modes (RSA, D.H.). The other compared architectures [6, 18, 29, 31] have worst performance than the proposed, although this result is somehow unfair. The implementation technology libraries of these works are somehow dated (2, 1, 0.6 $\mu$m), making difficult the comparison aspects. It is possible, that these works [18, 29, 31] would achieve better performance, if they upgraded to currently available CMOS. On the other side, the D.H. performance it is estimated to be the half in the term of throughput compared with the RSA. This is due to the fact, that D.H. operation needs doubled number of performed multiplications, based on the two used exponents, compared with the RSA. Since not many hardware implementations of D.H. have been published until now [47], the comparative study of D.H. performance is a hard process. In the only well-know work of D.H. implementation [47] it is claimed that by using GF multipliers, the performance of this cipher could be increased at about 33%. Although, no detail synthesis results about the implementation operating frequency and covered area is given in this work [47].

For DES hardware implementation, different works have been proposed, which focus different implementation aspects [5, 13, 21, 31, 44]. The goals of these implementations vary from research in the key expansion process strength, to programmable and parallelism designs. The proposed DES implementation is 400% faster than the best implementation presented in [21]. Nevertheless, this work [21] is a universal Key-Search machine based on fast DES architectures. The achieved results related to the key-search on the DES key expansion process in [21], are very good, but the used DES implementations in this work are inferior to performance. A case study of exploiting parallelism

in hardware implementation of DES is introduced by the work [5], but with low performance. The proposed DES implementation has better performance at about 430 to 880% compared with [5]. The major goal of the designs [13, 44] is the high performance. The throughput of the implementation [44], still remains the best reported until now in the technical literature. The work [13] employs a novel methodology to the design of GaAs architectures and has operating frequency 3 times better than the proposed implementation.

As the Table 3 presents, the SHA-1 operation mode of the proposed Reconfigurable Integrity Unit is better at about 260% in the term of throughput compared with [12]. The proposed Reconfigurable Unit area resources are at about 1.6 times more than in the same work [12]. The proposed Reconfigurable Integrity Unit, in the case of MD5 operation mode, has better throughput compared with the conventional implementations [10, 12, 41], by about 260, 115 and 55% respectively. It has to be mentioned that the work [41] provides a performance estimation of a theoretical MD5 hardware implementation and does not report implementation results in detail. Of course the estimations of [41] are still important for the readers/researchers. The first implementation of [10] allocates less covered area than the proposed. This is a physical result of this design rather than a disadvantage of the proposed Reconfigurable Integrity Unit. In this work [10] the specified MD5 processes of both padding and data manipulation have not been integrated. The proposed Reconfigurable Integrity Unit supports both these fundamental units of MD5 specifications. Of course the integration of them has consult to a low increase of the covered area resources. The second implementation of [10] uses full-step architecture. Although this design approach achieves doubled performance compared with the first implementation of the same work [10], the covered area resources are increased by a factor equal to 6. The proposed Reconfigurable Integrity Unit has better area-delay product compared with the two implementations of [10]. The introduced architecture in [12] has been designed like a typical digital processor with data and address buses. This work requires 206 and 255 clock cycles to perform the 64 rounds of MD5 and the 80 rounds of SHA-1, respectively, with 59 MHz clock frequency. Our proposed unit, based on the full rolling (feedback) technique, requires for SHA-1 and MD5 operation modes, 81 and 65 clock cycles respectively, with clock frequency up to70 MHz. The shared used arithmetic units in [12], supported by data and address buses, are a design technique with not very good performance, compared with the applied technique in the proposed Reconfigurable Integrity Unit. This is due to the fact that the architecture of [12] requires much clock cycles. Although, in [12] components can be added and remove from the system easily (scalability) and the system performance can be increased by using more arithmetic units (exploiting parallelism). In the case of WAP Integrity Unit, the proposed full rolling loop architecture is a design with better performance, for both MD5 and SHA-1 operation modes.

The major advantage of both Reconfigurable Authentication Unit and Reconfigurable Integrity Unit is that each one ensures the operation of two ciphers, RSA-D.H. and MD5-SHA-1 respectively, but they allocate at about 40–60% minimized area resources compared with two separate implementations of each pair of ciphers. This a major issue in mobile communications where many limitations exist in the area resources and the available memory also. In addition, these reconfigurable units have high operation frequency. Both authentication and integrity units achieves throughput compatible and

in many cases better than the other separate conventional implementations. In the case of bulk encryption the specified by the WAP ciphers has no commonality at all in their architecture. This reason makes inefficient every design approach for a reconfigurable bulk encryption unit. That's why ciphers for bulk encryption have been designed as separate cores.

The main scope of the design of this proposed Crypto-Processor architecture is to achieve the best balance as possible between the implementation parameters such as bandwidth, allocated area, energy etc. In the previous sections, the design criteria of each separate unit of the proposed architecture are analyzed in detailed. It has to be mentioned that the achieved performance is superior to today WTLS specifications and it is estimated that could satisfy efficiently future upgrades. Concerning area and energy issues, optimizations and better synthesis results could be achieved by excluding security features, or possible one of the integrated units such as authentication unit, in the cases that the applications demand. Of course such an approach would result in reducing the supported security level of the proposed Crypto-Processor, and it is not recommended from security aspects.

The proposed architecture can also be used as a powerful security core, in wireless communication networks of any kind, supporting bulk encryption, authentication and data integrity. This means that wireless networks with no specific security requirements could adopt the powerful WTLS security layer, as an alternative flexible Crypto-Processor.

## 6. Conclusion and outlook

An efficient architecture for the WAP security layer (WTLS) implementation is proposed in this paper. All the WTLS specified encryption units are supported by the introduced system, which guarantees high level of security strength at the same time. The proposed architecture performs efficiently for a great set of ciphers: IDEA, RSA, D.H., DES, MD5 and SHA-1, integrated in the same hardware module. In addition, an Authorized User Verification has also been implemented. The proposed architecture operation is mainly based in two reconfigurable designed units. With this applied technique the allocated area resources have been minimized by a great factor, compared with other conventional implementations. The introduced system has been integrated in an FPGA hardware device and has been tested in real time conditions, by using an FPGA board. The synthesis results prove that the system has compatible (for RSA and D.H. operation modes) and better performance (for IDEA, DES, MD5, and SHA-1), compared with previous published works. IDEA proposed architecture is based on a modified transformation round which minimizes the allocated area resources by about 40%. DES implementation performs better, with a range from 200 to 400%, compared with the conventional works. The Reconfigurable Integrity Unit has better performance at about 50 to 300% compared with the other conventional architectures for both MD5 and SHA-1 operation modes.

The proposed architecture is a flexible solution for WAP security layer implementation. The introduced system can be applied efficiently in both servers and mobile devices of WTLS wireless networks. The implementation of the proposed architecture achieves high-speed performance and minimized area resources, supporting six ciphers operation.

## References


1. Advanced Encryption Standard. http://csrc.nist.gov/CryptoToolkit/aes/, 2003.
2. T. Blum. Modular exponentiation on reconfigurable hardware. Master thesis, *Electrical and Computer Eng. Dept.*, Worcester Polytechnic Inst., May 1999.
3. T. Blum and C. Paar. Montgomery modular exponentiation on reconfigurable hardware. In *Proc. 14th Symp. Computer Arithmetic*, pp. 70–77, 1999.
4. Th. Blum and Chr. Paar. High-Radix montgomery modular exponentiation on reconfigurable hardware. *IEEE Transactions on Computers*, 50(7), 2001.
5. A. G. Broscius and J. M. Smith. Exploiting parallelism in hardware implementation of the DES. *Advances in Cryptology: CRYPTO-91 Proceedings*, Springer-Verlag, pp. 367–376, 1992.
6. P. S. Chen, S. A. Hwang, and C. W. Wu. A systolic RSA public key cryptosystem. In *Proceedings of International Symposium of Circuit and System (ISCAS'96)*, vol. 4, pp. 408–411, 1996.
7. O. Y. H. Cheung, K. H. Tsoi, P. H. W. Leong, and M. P. Leong. Tradeoffs in parallel and serial implementations of the international data encryption algorithm. In *Proceedings of CHES 2001*, LNCS 2162, Springer-Verlag, pp. 333–337, 2001.
8. A. Curiger, H. Bonnenberg, and H. Kaeslin. Regular VLSI architectures for multiplication modulo $(2n+1)$. *IEEE Solid-State Circuits*, 26(7):990–994, 1991.
9. Data Encryption Standard, Federal Information Processing Standard (FIPS) 46, National Bureau of Standards, 1977.
10. J. Deepakumara, H. M. Heys, and R. Venkatesan. FPGA implementation of MD5 hash algorithm. In *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2001)*, Toronto, Ontario, May 2001.
11. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644–654, 1976.
12. S. Dominikus. A hardware implementation of MD4-Family hash algorithms. In *Proceedings of IEEE International Conference on Electronics Circuits and Systems (ICECS'02)*, Croatia, vol. III, pp. 1143–1146, Sept. 15–18, 2002.
13. H. Eberle. A high-speed des implementation for network applications. In *Proceedings of 12th Annual International Cryptology Conference, CRYPTO '92*, Santa Barbara, August 16–20, 1992.
14. I. Goldberg and D. Wagner. Architectural considerations for cryptanalytic hardware. Chapter 10 of *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*, O'Reilly, July 1998.
15. F. T. Gramp and R. H. Morris. UNIX operation system security. *AT&T Bell Laboratories Technical Journal*, 63(8 part 2), 1984.
16. G. J. Hwang, J. C. R. Tseng, and Y. S. Huang. I-WAP: An intelligent wap site management system. *IEEE Transactions on Mobile Computing*, 1(2), 2002.
17. IEEE P1363. Standard specifications for public-key cryptography. Draft Version 8, October 1998.
18. S. Ishii, K. Ohyama, and K. Yamanaka. A single-chip RSA processor implemented in a 0.5 $\mu$m rule gate array. in *Proceedings of 7th Annual IEEE International ASIC Conference Exhibit*, pp. 433–436, 1994.
19. S. Jormalainen and J. Laine. *Security in WTLS*. http://www.hut.fi/~jtlaine2/wtls/, 2002.
20. B. S. Kaliski Jr. and Y. L. Yin. On the security of the RC5 Encryption algorithm. RSA Laboratories Technical Report TR-602, Sept. 1998.
21. J. Kaps and Chr. Paar. Fast DES implementations for FPGAs and its application to a universal key-search machine. *5th Annual Workshop on Selected Areas in Cryptography (SAC '98)*, August 17–18, Ontario, Canada, 1998.
22. D. E. Knuth. *The Art of Computer Programming*, Vol. 2: *Seminumerical Algorithms.* Addison-Wesley, Reading, Massachusetts, 2nd edition, 1981.
23. N. Leavitt. Will WAP deliver the wireless internet? *Proceedings of IEEE Computer*, 16–20, 2000.
24. X. Lai and J. L. Massey. A proposal for a new Block Encryption Standard. In *Proceedings of Eurocrypt'90*, Aarhus, Denmark, pp. 389–404, May 21–24, 1990.
25. T. Lewis. Why WAP may never get off the ground. *Proceedings of IEEE Computer*, 110–112, 2000.
26. A. Menezes, P. van Oorchot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Inc, October 1997.

27. M. Metter and R. Colomb. WAP enabling existing HTML applications. In *Proc. First Australian User Interface Conf.*, 49–57, 2000.
28. P. Montgomery. Modular multiplication with trial division. *Math. of Computation*, 44:519–521, 1985.
29. H. Orup. A 100 kbits/s single chip modular exponentiation processor. In *HOT chips VI, Symposioum Record,* pp. 53–59, 1994.
30. J. M. Rabaey. *Digital Integrated Circuits*. Prentice Hall, 1996.
31. S. S. Raghuram and C. Chakrabarti. A programmable processor for cryptography. In *Proceedings of IEEE International Symposium on Circuits & Systems (ISCAS'00)*, May 28–31, Switzerland, 2000.
32. R. Rajsuman. *System-on-a-Chip*. Design and Test, Artech House, 2002.
33. R. L. Rivest. The RC5 encryption algorithm. In *Proceedings of the 1994 Leuven Workshop on Fast Software Encryption*, Springer, pp. 86–96, 1995.
34. R. L. Rivest. *The MD5 Message Digest Algorithm*. RFC 1321, MIT LCS & RSA data Security, Inc., April 1992.
35. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Comm. ACM*, 21:120–126, 1976.
36. W. Sachs and S. Wolter. Specification and implementation of a crypto-coprocessor for ISDN. In *Proceedings of IEEE International Symposium on Circuits & Systems (ISCAS'00)*, Switzerland, vol. I, pp. 275–278, May 28–31, 2000.
37. B. Schneier. *Applied Cryptography—Protocols, Algorithms and Source Code in C*, Second edition. John Wiley and Sons, New York, 1996.
38. SHA-1 Standard National Institute of Standards and Technology (NIST), Secure Hash Standard, FIPS PUB 180-1, www.itl.nist.gov/fipspubs/fip180-1.htm, 2001.
39. *SSL Protocol Specifications*, www.netscape.com/eng/ssl3, 2002.
40. D. R. Stinson. *Cryptography: Theory and Practice*. CRC Press LLC, 1995.
41. J. D. Touch. Performance analysis of MD5. In *Proceedings of ACM SIGCOMM'95*, Cambridge, Massachusetts, 1995.
42. C. D. Walter. Systolic modular multiplication. *IEEE Transactions on Computers*, 42(3):376, 1993.
43. WAP Forum. Wireless Application Protocol Architecture Specification and Wireless Transport Layer Security, www.wapforum.org, 2002.
44. D. C. Wilcox, L.G. Pierson, P.J. Robertson, E.L. Witzke, and C. Gass. A DES ASIC suitable for network encryption at 10 GPS and Beyond. In *Proceedings of CHESS'99*, LNCS 1717, pp. 37–48, 1999.
45. Xilinx Inc., San Jose, California, Virtex, 2.5 V Field Programmable Gate Arrays, 2002.
46. R. Zhang and K. Chen. Improvements on the WTLS protocol to avoid denial of service attacks. *Computers and Security, Elsevier Science*, 24(1):76–82, 2005.
47. C. N. Zhang, M. Deng, and R. Mason. Two improved algorithms and hardware implementations for key distributing using extended programmable cellular automate. *14th Annual Computer Security Applications Conference*, Phoenix, Arizona, 1998.
48. R. Zimmermann, A. Curiger, H. Bonnenberg, H. Kaeslin, N. Felber, and W. Fichtner, A 177 Mb/s VLSI implementation of the international data encryption algorithm. *IEEE Journal of Solid State Circuits*, 29(3), 1994.