Invited Paper

DOI : 10.3745/JIPS.2009.5.4.187

Journal of Information Processing Systems, Vol.5, No.4, December 2009 187

Design of Cryptographic Hardware Architecture for Mobile Computing

Mooseop Kim*, Youngsae Kim* and Hyunsook Cho*

Abstract: This paper presents compact cryptographic hardware architecture suitable for the Mobile Trusted Module (MTM) that requires low-area and low-power characteristics. The built-in cryptographic engine in the MTM is one of the most important circuit blocks and contributes to the performance of the whole platform because it is used as the key primitive supporting digital signature, platform integrity and command authentication. Unlike personal computers, mobile platforms have very stringent limitations with respect to available power, physical circuit area, and cost. Therefore special architecture and design methods for a compact cryptographic hardware module are required. The proposed cryptographic hardware has a chip area of 38K gates for RSA and 12.4K gates for unified SHA-1 and SHA-256 respectively on a 0.25um CMOS process. The current consumption of the proposed cryptographic hardware consumes at most 3.96mA for RSA and 2.16mA for SHA computations under the 25MHz.

Keywords: *Trusted Computing, MTM, Cryptographic circuit, RSA, HASH, Mobile Computing*

1. Introduction

In the last decade, the significant increasing in processing power of the mobile device, coupled with pervasive network connectivity, has resulted in a large number of applications and services to be developed and deployed on these devices. These advances present new security problems, which cannot be satisfied with present security facilities available to current limited purpose mobile devices.

To solve these new problems, several organizations have proposed security requirements and specifications for the mobile phone security. Among those organizations, the most prominent group is Trusted Computing Group (TCG). TCG recently published specifications for Mobile Trusted Module (MTM) [1], which is a modified version of the Trusted Platform Module (TPM) that is the counterpart for PC platforms. TCG formed a dedicated Mobile Phone Working Group (MPWG) to address the security requirements for mobile devices. MPWG recently released a Trusted Mobile Phone Reference Architecture (TCG-MPRA) specification [2], which specifies a general architecture for mobile devices relying on the trusted services of MTMs.

The built-in hardware engine for cryptographic computation in a MTM is one of the most important circuit blocks and dominates the performance of the whole platform because it is used as a key primitive to support most MTM commands concerning to the platform integrity and the command authentication. TCG specification mandates the use of RSA public-key algorithm and SHA-1 algorithm for signature and hash computation in the TCG-MPRA respectively. Unlike the general TPM for personal computers, the MTM that is to be employed in mobile devices has very stringent limitations with respect to available power, physical circuit area and so on. Therefore, the MTM needs the spatially-optimized architecture and design method for the construction of a compact cryptographic hardware module. Therefore, it is important to design compact and power efficient cryptographic engine that supports both RSA and SHA hash algorithm at the same time according to the application purpose.

In this paper, we propose efficient hardware architecture of the RSA algorithm and the unified SHA-1 and SHA-256 algorithm for trusted mobile platforms. As a result, a compact cryptographic hardware implementation capable of supporting the digital signature and the integrity check of mobile platforms was developed and evaluated. The main contributions of this paper are summarized as follows.

First, we proposed compact hardware architecture of RSA and SHA algorithm for MTM, especially on the algorithmic and architectural level. As the result of applying the proposed architecture, we can design a unified SHA-1 and SHA-256 hash module and scalable RSA module that use the minimum hardware resources among all kinds of published designs. Unfortunately, there is no prescription or recommendation data with respect to the reasonable

Manuscript received August 26, 2009; accepted October 26, 2009.
Corresponding Author: Mooseop Kim
* Software & Content Research Laboratory, ETRI, Daejeon, Korea. (gomskim, vincent, hscho@etri.re.kr)

logic area and power consumption for the cryptographic engine of MTM. Therefore, we also present the reasonable guidelines for maximum allowable circuit area and consuming power of the cryptographic module for the MTM chip.

Second, to the best of our knowledge, the proposed design is the first effort to implement a unified SHA-1 and SHA-256 algorithm for mobile trusted computing that requires small area and low-power characteristics. The proposed SHA hardware has optimized structure which consumes very small power consumption.

Finally, the proposed architecture outperforms the cryptographic computing time of the existing commercial TPM chips. The proposed design is at least three times faster than any published TPM chips both in core level and system level computation.

2. Mobile Trusted Module

Mobile Trusted Module (MTM) guarantees the integrity of the mobile platform and is a new requirement in the process where the mobile device changes into the open platform and value-based application technology. As shown in Fig.1, mobile devices can improve the reliability and the security of the device using Mobile Trusted Module (MTM), which ensures that the device is running under the authorized software and hardware.

Security in a mobile platform relies heavily on cryptographic techniques. Cryptographic algorithms can be implemented in hardware or software. However, nCipher, a computer security company, proved that software-based security had fatal flaws in 2000 [3]. Therefore, to offer high-level security and cryptographic performance, MTM protect and accelerate computations of essential cryptographic operations using a dedicated hardware co-processor.

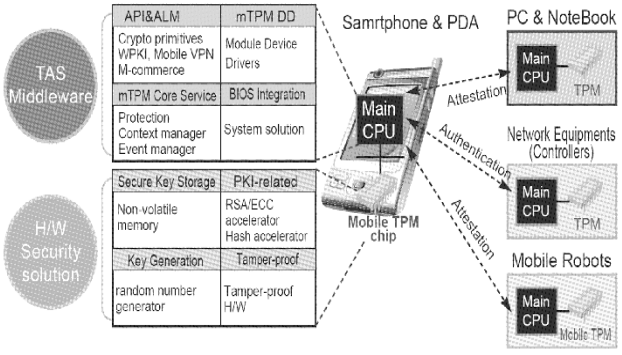

Fig. 1. MTM for mobile devices

2.1 Security engine for MTM

The TCG requires mandatory hardware cryptographic functions, which are usually provided by a cryptographic co-processor inside the MTM. According to the TCG standard, the MTM requires an RSA public-key signature, encryption and decryption to enable secure storage of data and digital secrets. MTM also requires the storage of hashing (SHA-1) and keyed hashing (HMAC) that enable verifiable attestation of the platform configuration when it is booted. A functional block diagram of a minimal capability for MTM and the simplified functional block diagram of the security engine are shown in Fig.2.

The built-in cryptographic engine in the MTM is one of the most important circuit blocks because it is used as a key primitive supporting digital signature and integrity verification used in the most of commands for authentication in the mobile platform. Therefore, the efficient hardware design of the cryptographic module is one of the most important factors influencing on the overall performance of a platform. As depicted in Fig. 2, the cryptographic engine contains the various cryptographic primitives used by the MTM as well as the random number generator vital to these components.

RSA is an algorithm for public-key cryptography. It is the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public key cryptography. RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations. According to the MTM specification, MTM contains a hardware engine to perform up to 2048 bit RSA encryption and decryption. MTM uses its built-in RSA engine during digital signing and key wrapping operations.

The integrity hardware block of the MTM contains both a SHA-1 engine and a dedicated HMAC engine. The HMAC module is used to provide information in two cases. The first is to provide proof of the authorization data and the second is to provide proof that an arriving request is authorized and has not been modified. SHA-1 is used as the primary hash function in the MTM. The SHA-1 hash

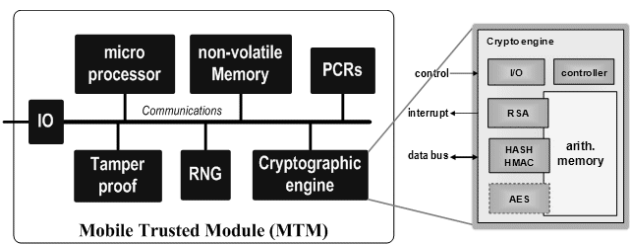

Fig. 2. Block diagram of MTM and Cryptographic engine

module allows users outside the MTM to support measurement during boot and to give environment with limited resources to the hash capabilities. Therefore, the built-in SHA-1 engine in the MTM is one of the most important circuit blocks and dominates the performance of the whole platform because it is used as a key primitive to support most MTM commands concerning to the platform integrity and the command authentication.

Other cryptographic algorithms such as Advanced Encryption Standard (AES), Data Encryption Standard (DES) and other hashing algorithms (MD5, SHA variants) can be added to a MTM in optionally but there is no guarantee that other MTMs will understand and decode the encrypted data.

3. Proposed Architecture of the Crypto-engine

An actual MTM chip is comprised of many components such as a micro-processor, memory blocks, random number generator (RNG), and peripheral modules. Therefore, the available circuit area for the cryptographic engine in the MTM is very small. There may be many design methods but in this whole work we will focus on methods minimizing the hardware resources because our work is concerned with the mobile system. After reviewing the design constraints of the cryptographic hardware for a MTM chip, we will estimate the reasonable circuit area and the maximum allowable consuming power for the cryptographic engine. We then propose an optimized hardware architecture that meets our design goals.

3.1 Requirements for Cryptographic Engine

Unfortunately, there are no prescriptions or recommendations concerning to the preferred power consumption of the MTM chip integrated into a mobile device. Therefore, there is no data about the reasonable and recommended consuming power of the crypto engine in the MTM. As an indirect method, therefore, we suggest a reasonable level of consuming power of the cryptographic hardware in the MTM chip suitable for a mobile device.

For the estimation of the reasonable consuming power, we refer to the power consumption of the Universal Subscriber Identity Module (USIM) embedded in the mobile phone. According to the USIM specification [4], the maximum allowable current for Class B type is about 50mA at the 5MHz. Since both MTM and USIM chips provide a similar quality of value-added security functions for mobile devices, it is reasonable and practical that the recommended maximum current, which the MTM chip can consume in mobile device, is derived from the maximum current value of USIM chip. However, the operating frequency of the MTM differs from that of USIM. So, the maximum current value of USIM is not directly used for MTM. The dynamic power dissipation of a CMOS circuit is given by [5]:

$$P = \alpha_{0\to1} \cdot C_L \cdot V_{DD}^2 \cdot f_{CLK} \tag{1}$$

where, $\alpha_{0\to1}$ represents the probability of the logic gate output to change from 0 to 1, which is the switching activity, C_L is the load capacitance, and f_{CLK} is the operating frequency. From the above equation, dynamic power dissipation of a circuit depends on the operating frequency. Therefore, the maximum current of Class B type of USIM is calculated as 12.5mA at the 20MHz. So we assume that this value is a reasonable maximum current, which the MTM chip can consume in a mobile device. Moreover, for the more strict operation environment, we assume that the cryptographic module consumes only 40% of the whole available power of the MTM. So, according to our strict assumption, the maximum allowable current that the cryptographic module of the MTM can consume is about 5.5mA. In the MTM, there is no case that the integrity block and the RSA module operate at the same time. Therefore, 5.5mA could be assumed the maximum allowable current for the cryptographic engine of the MTM.

In addition to the aspect of consuming power, there is limitation in the physical circuit area in the mobile device. An actual mobile device is assembled with many components on a compact PCB. Therefore, minimizing circuit size is one of the most important factors for the MTM design. The reasonable die size of the MTM mounted on the mobile phone is about 4mm×4mm. This means that the actual silicon area for the MTM should not exceed a logical area of 400K gates when the circuit area for pad part of chip is excluded. The estimation of the circuit area ratio in accordance with the major components for the MTM is shown in Fig.3.

The largest circuit element of the MTM chip is the memory. The memory of the MTM is mainly used for the purpose of ROM for operating system and of the data storage for the program execution. At least 128K byte of flash memory is required for the efficient program execution in the MTM which covers about 250K gates. As shown in Fig.2, the MTM consists of many other components such as microprocessor, DMA, hardware timers, interface logics, controllers and so on. Therefore, estimating the circuit area of all the other components, only 15% of logic area is remaining for the cryptographic engine which should not

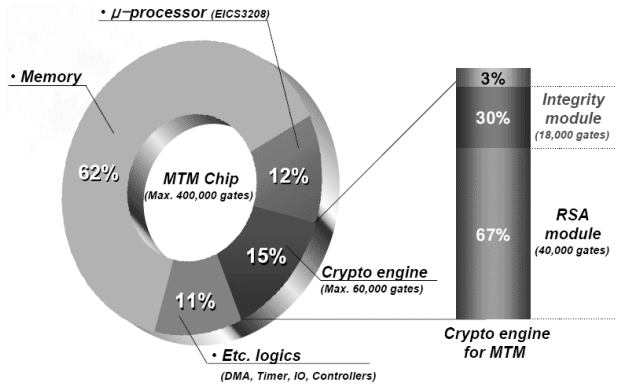

Fig. 3. Circuit area ratio for MTM components

exceed a chip area of 60K gates.

As shown in Fig.2, the cryptographic engine for the MTM is comprised of several components. RSA module is the largest component in the cryptographic engine for the MTM, which requires at least 40K gates in the circuit area. Therefore, the maximum logic area for the integrity module should not exceed 18K gates when considering the area for tamper proofing logic and glue logic for cryptographic engine. According to the reports of previous works for RSA and SHA implementations, this logic area is only suitable for the stand-alone type of RSA and SHA hardware implementation.

Therefore, design methodologies at different abstraction levels, such as systems, architectures, logic design, basic cells as well as layout, must take into account to design the compact cryptographic hardware for the MTM chip.

3.2 Unified SHA-1 and SHA-256 Hardware Module

TCG specification mandates the use of SHA-1 algorithm for hash computation in the TCG-MPRA. Recently, however, some research are announced the security weakness of SHA-1 algorithm. Furthermore, the National Security Agency (NSA) of US announced Suite B cryptography [6] that specifies a new security set of encryption, signature and hash algorithm in 2005. According to Suite B cryptography, SHA-256 is recommended for secure level usages. This cryptographic trend to move over Suite B cryptography is realized in the trusted mobile computing. TCG announced that TPM must support SHA-256 algorithm in the revised specification, TPM-NEXT [7]. Although the SHA-1 algorithm is replaced with SHA-256, SHA-1 is still used for HMAC computation in the MTM because the critical HMAC property is that finding the key used to produce a certain digest must be computationally infeasible. Therefore, an efficient hardware module supporting both SHA-1 and SHA-256 algorithm is required in the MTM used for mobile platform.

The efficiency of the general SHA hardware in terms of circuit area, power consumption and throughput is mainly determined by the data path structure of the message scheduler and the message compression units. To devise a compact and unified SHA architecture for a MTM chip that has strict limitations with respect to circuit area and available power, we adopt a folded architecture to design the message scheduler and the message compression units.

The folded architecture that is based on the loop rolling architecture executes each round of computation over several clock cycles to reduce hardware resources. We adopt a 32-bit data bus for an efficient design of the unified SHA hardware, since all the operations of SHA-1 and SHA-256 algorithm and the corresponding variables need 32-bit data as a basic execution unit. Although the smaller bus size may require the less registers, it invokes the more data selectors with the restricted resource sharing, resulting in an inefficient implementation.

The compact and unified SHA hardware architecture proposed in this paper uses just one adder to compute the whole round operation of both SHA-1 and SHA-256 algorithms. Adders are the most spacious part in the message compression unit. In addition, we use only one 32-bit register and one adder for the computation of message scheduler unit. Since the number of adders and registers determines the overall size of any SHA-1 and SHA-256 hardware circuit, this approach employed for implementing both message compression and message scheduler is crucial for an efficient SHA hardware implementation. Fig. 4 shows the data path architecture of the unified SHA circuit. The message compression performs the actual hashing operation over the stream of message-dependant words out of the message scheduler.

The message compression for SHA-1 requires four

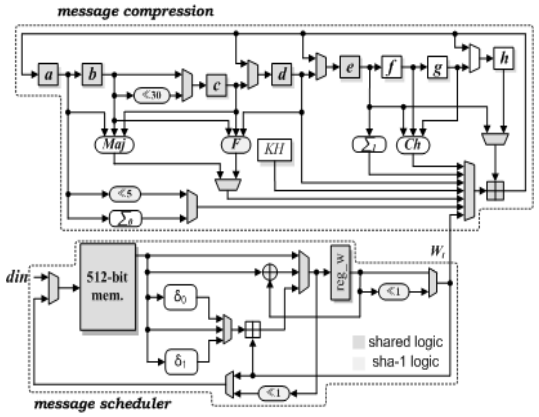

Fig. 4. Proposed architecture for unified SHA hardware

modulo 2^{32} additions. In the SHA-1 computation, four out of five words (b…e) remain almost unchanged by a single round. These words are only shifted by one position down in each round. The round computation of SHA-256 is similar to that of SHA-1. The primary differences are as follows: The number of words processed by each round is 8, and the longest path is equivalent to the addition of seven operands.

Therefore, we can design a compact and low-power architecture for the unified message compression unit, which reduces the numbers of adder chains. In the SHA-1 computation, register *e* is used for the storage of the temporary addition values $T= S^5(a)+ f_t(b, c, d)+e+W_t +K_t$ for the iterative computation of *a*. This computation requires four modulo 2^{32} additions. Therefore, four clock cycles are required for one round operation of SHA-1 algorithm. On the contrary, SHA-256 requires more comprehensive round operations. For SHA-256, the iterative computation of $T_1=h+ \Sigma_1(e)+Ch(e, f, g)+K_t+W_t$ *and* $T_2= \Sigma_0(a)+Maj(a, b, c)$, register *h* and *d* are used for the storage of the temporary addition values. Seven modulo 2^{32} additions are necessary for a single round operation. Therefore, seven clock cycles are required for one round operation of SHA-256.

Another important part of the compact and unified SHA-1 and SHA-256 hardware is a message scheduler that generates message dependant words W_t used as an input data for each step of round operation of the message compression unit. The message scheduler is the most expensive part of SHA hardware in terms of hardware resources. Therefore, its space-efficient implementation forms a critical part in the compact design of the compact and unified SHA-1 and SHA-256 circuit. In the previous works, the message scheduler for SHA-1 and SHA-256 hardware module has been implemented as a chain of sixteen 32-bit shift registers that store the intermediate message schedules of W_t. The message scheduler operations for SHA-1 and SHA-256 are almost identical except that three 32-bit XORs for SHA-1 are replaced by three 32-bit adders and new 32-bit operators $\Sigma_0()$ and $\Sigma_1()$ are appended in SHA-256. The proposed message scheduler performs the cryptographic computation and was formulated as follows.

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) & SHA-1 \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16} & SHA-256 \end{cases} \quad (2)$$

It can be easily seen from the above formula that the input message stored in the memory is used only to compute the first sixteen round operations. In other words, the memory is not used any more after sixteenth round calculations. Therefore, in the aspect of resource efficiency, this method is inefficient and wastes too many hardware resources.

As an alternative method for compact design, we propose an optimized architecture of the message scheduler unit, which enhances the resource sharing of the aforementioned memory and reduces the number of adder used for SHA-256 algorithm. Four values (M_{t-3}, M_{t-8}, M_{t-14}, M_{t-16} for SHA-1 and M_{t-2}, M_{t-7}, M_{t-15}, M_{t-16} for SHA-256 respectively) of the memory data have to be read and applied to compute the 32-bit message dependent word W_t. This process takes four clock cycles because our approach uses only one 32-bit register (*reg_w* in Fig.4) and 32-bit adder to implement the message scheduler.

The computation of the message schedule could be calculated simultaneously during the computation of message compression, which consumes 4 and 7 clock cycles for SHA-1 and SHA-256 respectively. Since the functional steps for message scheduling are shorter than that of message compression, no additional clock cycle is required for the computation of W_t in the unified message scheduler unit.

The new result of W_t, completed at the end of the fourth clock, is fed to the message compression unit by switching the output data using the 32-bit 2:1 multiplexer at the far right-hand side of Fig.4. It is also written back to the memory in each round for the calculation of the remaining rounds using the multiplexer at the far left-hand side. In these processes, the memory address is defined as $t \bmod 2^4$ when the current round is *t*. A dedicated hard-wired logic and counter in the controller are used to compute the necessary address for memory access.

3.3 Compact and Scalable RSA Hardware Module

RSA is an algorithm for public-key cryptography. It is the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public-key cryptography. RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations. According to the specification for TCG-MPRA, MTM must contain a hardware engine to perform up to 2048 bit RSA encryption/decryption.

In terms of the data encryption processing in RSA, The computation for modular exponentiation using repeated modular multiplications is the main processing workload. However, the demand on high-performance RSA systems requires very large bit-length operands making efficient hardware implementation difficult especially in an embed-

ded system like MTM. Therefore, it is important to employ efficient hardware design techniques to improve the overall performance of the RSA system in MTM.

The core operation of RSA cryptographic system is modular exponentiation. The modulus N is the product of two large primes p and q and a public key is obtained by $e = d^{-1} \bmod (p-1)(q-1)$. The encryption operation is performed using the public key e, as $C=M^e \bmod N$. Where M is the plaintext such that $0<M<N$, and C is the cipher text which can be decrypted using the secret key d, as $M=C^d \bmod N$.

Given two N-residues A and B, and extra factor r such that $r=2^n$, Montgomery modular multiplication scheme computes the modular product $R=ABr^{-1} \bmod N$. The computation of the word-based Montgomery modular multiplication is depicted in Algorithm 1.

Where A, B, N, and N0p are n-bit numbers but all the numbers are handled in ether w-bit word sizes. This is the simplest way of adapting Montgomery algorithm to large operand sizes. Hence it would be to just replace every arithmetic operation by its multi-precision equivalent.

A modified algorithm to make Montgomery multiplication scalable to any input operands precision and flexible to any requirements is given as Algorithm 2, where shift_data and m represent w-bit numbers and c_pre and c_next means 1-bit carry information respectively.

The proposed algorithm provides some advantages in the implementation of modular multiplication. The major feature of the modified word-based modular multiplication is that it can be operated without the Montgomery correction factor N0p. It makes the high-precision and the high-radix multiplication easy to implement. In addition, the extra bits α ($\alpha>1$) can remove the additional operation for final comparison and subtraction in the modular exponentiation. The architecture for the proposed Montgomery multiplier depicted in algorithm 2, is shown in the Fig. 5. There are three main functional blocks: multiplier core, c-RAM (crypto-RAM), and IO block.

The IO block provides the interface among system bus,

Algorithm 1. Word-based Montgomery modular multiplication

Inputs : A, B, N(modulus), N0p(N[0] * N0p = -1 mod 2^w)
Begin
  Step 1 : R = 0
  Step 2 : for i = 0 to p-1 do {
    Step 2a : R = R + A*B[i]
    Step 2b : m = (R[0]*N0p) mod 2^w
    Step 2c : R = R + N*m
    Step 2d : R = R/2^w }
  Step 3 : if(R > N) R = R – N
  Step 4 : return (R)
End
Output : R = ABr^{-1} mod N, 0≤ R < N, r = 2^n

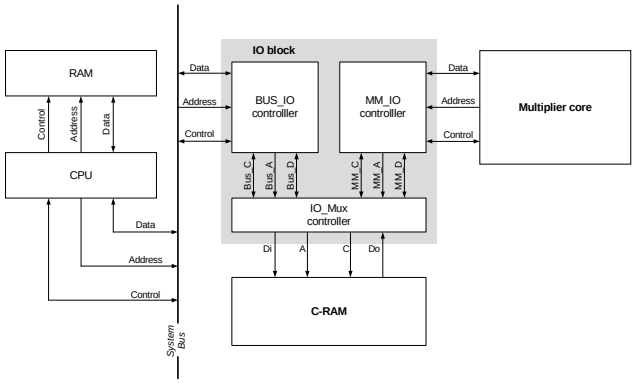

Fig. 5. Proposed architecture for Montgomery modular multiplier

multiplier core, and c-RAM. In RSA system implementation, the IO delay between system bus and modular multiplier is an additional redundancy. It causes the performance to decrease. Therefore, avoiding the redundant delay, the multiplier should access the data memory directly rather than through system bus. Using this IO block and internal c-RAM, the IO delay can be removed without changing the main system architecture. As a result, the modular multiplier can be isolated from system bus except the initialization and the termination. Therefore, it can be reused as a modular multiplier for other systems without modifying the architecture. In addition, the IO block makes the multiplier scalable with bounding the iteration time of modular multiplier core.

The multiplier core block is the functional block implementing the proposed Montgomery algorithm summarized in Algorithm 2. It receives w-bit words of B, S, and some

Algorithm 2. Proposed Montgomery Modular Multiplication

Inputs : A, B, N(modulus)
Begin
  Step 1 : R = 0, S = 0
  Step 2 : for i = 0 to p-1 do {
    Step 2a : c_pre = 0, c_next = 0
    Step 2b : for j = 0 to p-1 do {
      Step 2ba : if(i==0) R[j] = 0, else R[j] = S[j]
      Step 2bb : for b = 0 to w-1 do {
        Step 2bba : R[j] = R[j] + A[j]$_b$*B[i]
        Step 2bbb : if(j==0) m_b = R[j]$_0$, else m_b = m_b
        Step 2bbc : R[j] = R[j] + m_b*N[j]
        Step 2bbd : shift_data$_b$ = R[j]$_0$
        Step 2bbe : R[j] = R[j]/2
      } end for step 2bb
      Step 2bc : (c_pre,S[j]) = R[j] + c_pre
      Step 2bd : (c_next,S[j-1]) = S[j-1] + shift_data + c_next
    } end for step 2b
  } end for step 2
  Step 3 : return (S)
End
Output : S = ABr^{-1} mod N, 0≤ S < 2N, r = $2^{n+\alpha}$

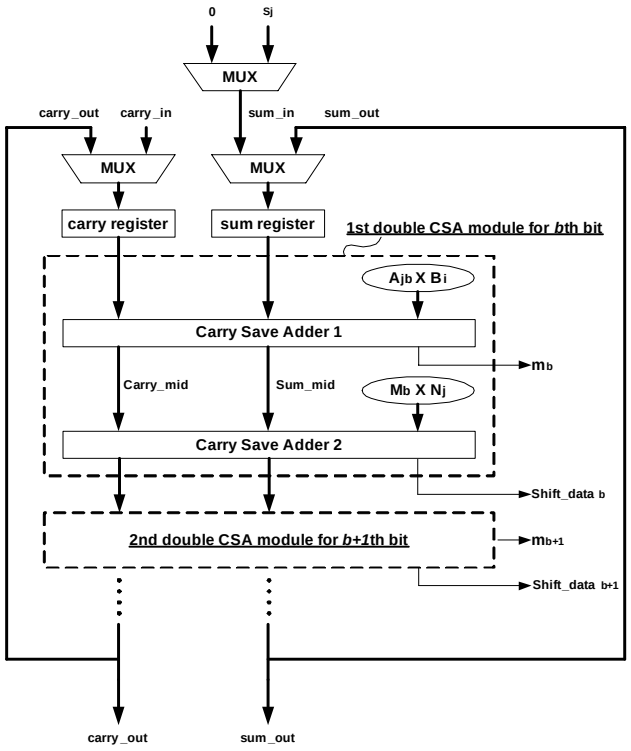

Fig. 6. Detailed architecture for modular multiplier core

bits of A as input data according to the radix size. In hardware implementation, every step in our algorithm is designed by the double carry save adder (CSA) module. The detailed architecture for the multiplier core is shown in Figure 6. It shows that the high-radix multiplier core can be designed easily based on the CSA architecture. Therefore, we can apply n double CSA modules to the multiplier core to implement the radix-2^n modular multiplication.

4. Implementation Results

In general, hardware implementations for cryptographic algorithm often target FPGAs, which allow for rapid prototyping of designs and for the implementation to be varied based on the change of the architecture. ASIC implementations are more useful in systems that require high performance and low-power.

The described architectures of our design have been implemented in VHDL and their operations are verified through functional simulation using ModelSim, from Mentor Graphics. In order to evaluate our design, we used Synopsys synthesis flows on Sun Solaris platform. The target technology is Samsung Electronics' STD110 0.25um CMOS standard cell library featuring 2.5V core voltage. Synopsys Power Compiler was then used to calculate the overall power dissipation of our design. The activity of the netlist was verified using various test messages including the official test vectors so they could be considered as reasonable values.

We would like to emphasize that our design is on the algorithmic and architectural level. Implementing our designs using a low power ASIC library or a full custom design will enable higher energy and power savings.

The synthesis results of the proposed cryptographic hardware are summarized in the Table 1. The number of gates in the table for estimated circuit area is derived from counting the number of 2-input NAND gates used as the basic element in the implementation for each blocks and logic cells. Considering the essential elements like memory, registers for working variables, which cover more than 60% of total area, Table 1 shows that the resource sharing of logic blocks is fully used to design the proposed cryptographic hardware.

We introduced the maximum allowable circuit area for RSA and unified SHA hardware is less than 40K gates and 18K gates respectively. From Table 1, we can see that the unified SHA and RSA hardware cover reasonable circuit area.

The average value of consuming power of the RSA and unified SHA hardware at 25MHz is 9.9mW and 5.4mW respectively. Although the maximum operating frequency obtained using timing analysis for unified SHA and RSA is 137MHz and 50MHz respectively, we used 25 MHz as the operating frequency to evaluate the consuming power of our circuit because the system clock of most cellular phones is about 20MHz. We already assumed that the maximum allowable current is 5.5mA.

Notice that the proposed RSA and unified SHA architecture consumes 3.96mA and 2.16mA respectively at the 2.5V supplying voltage, which meets the maximum allowable current of the MTM chip.

Table 2 shows the comparison with power-aware de-

Table 1. Synthesis results of cryptographic engine based-on the functional blocks

crypto functions	logic blocks	circuit area		consuming power	
		gates	per[%]	㎼	per[%]
unified SHA	IO interface	106	0.85	190	3.52
	message schedule	4,462	35.99	3,192.95	59.14
	message compress	7,444	60.03	1993.6	36.88
	controller	388	3.13	24.61	0.46
	SHA total	12,400	100%	5.4 ㎽	100%
RSA	IO block	4,576	12.04	542.5	5.48
	multiplier core	33,424	87.96	9,357.5	94.52
	RSA total	38,000	100%	9.9 ㎽	100%

Table 2. Comparison between the proposed design and the previous works of SHA-1 and SHA-256 function based on circuit area, consuming power and performance

Ref.	func.	Platf. (㎛)	circuit area	f_{max} (㎒)	clock cycles	Th.put (Mbps)	power (㎼)
this work	SHA-1	0.25	12,400	137	355	197.6	5,400@25MHz 21.6@100KHz
	SHA-256				490	143.2	
[8]	SHA-1	0.18	8,120	--	1,274	--	35.24@100KHz
[9]	SHA-256		10,868	50	1,128	22.5	52.37@100KHz
[10]	SHA-1	0.25	10,641	--	330	--	19.5@100KHz

signs of SHA-1 or SHA-256 for secure RFID system, which has much more fierce implementation constraints than MTM chips. These implementations have features of very small and power-efficient SHA-1 or SHA-256 design. However, these results come from stand-alone type of design. Although our unified SHA hardware is a little bit larger than the design of [8-10], it has an advantage of supporting SHA-1 and SHA-256 hash function on a single data path.

The design of [8-10] consumes a very small amount of power for the computation. However, it is difficult to compare the consuming power with the design of [8-10], because they utilize the different technology and operating frequency. The consuming power depends on the operating frequency as shown in equation (1). When we analogize the consuming power indirectly using equation (1), our design consumes about $21.6uW$ at the operating frequency of 100 KHz. This result means that the consuming power of our architecture is as small as that of the RFID systems.

Table 3 summarizes the comparison with previous low-power RSA implementations features power-efficient architectures. A direct quantitative comparison with previous implementations is not practical because previous implementations differ in terms of their underlying technology and estimate the consuming power under the different frequency. Moreover, [12, 13] just shows the maximum frequency and does not open the operating frequency used for their power estimation. However, it is clear from the Table 3 that the proposed RSA architecture trades speed for a compact area and our implementation uses the minimum hardware resources among all kinds of published designs for power-aware RSA designs.

Table 3. Comparison between the proposed RSA design and the previous low-power 2048-bit RSA implementations

Ref.	Platf. (㎛)	max. N-len.	circuit area	f_{max} (MHz)	Th.put (kbps)	power (mW)
this work	0.25	2,048	38,000	50	18.4	9.9@25MHz
[11]	0.18	2,048	98,500	50	--	61.5@40MHz
[12]	0.18	2,048	61,000	200	107.5	32.5@--
[13]	0.18	1,024	5.76mm^2	460	586	830@--

Although [12, 13] does not provide exact power consumption measurement frequency, we believe that the proposed architecture spends the energy less than that of [12, 13], since the proposed design is optimized mainly to minimize the number of gates and their implementation area, assuring that the power consumption rate of the proposed design would be no greater than that of these previous works.

There exist several commercial TPM chips, which support RSA and SHA-1 algorithm [14, 15]. However, there has been no commercial chip that support SHA-256 algorithm in the TPM module. As far as we know, the proposed architecture of unified SHA hardware and its implementation is the first effort to implement the unified SHA-1 and SHA-256 for TCG-MPRA specifications. Therefore, it is difficult to directly compare these chips with our architecture since these chips usually do not provide the sufficient performance measurement results. For example, AT97SC3203 [14] opens just the computing time for RSA sign and single 512-bit message block of SHA-1 core and, on the other hand, SSX35A [15] provides computing time for both RSA sign and RSA verify. However, it only announces the system level performance of SHA-1 for the 1 M-bit data length, but both of them do not open the measurement data of the consuming power.

We showed that the proposed cryptographic hardware is compact and power efficient. However, there is still doubt whether the cryptographic hardware is a suitable and practical solution for a real MTM system because the performance improvement on the core-level does not always mean that the whole performance on the system level will show improvement.

In order to evaluate the experimental testing of our design, we designed an evaluation system, which is called as STPM Evaluation System Assembly (SESA) board. Fig. 7 shows the logical architecture of the evaluation system supporting the same design environment of MTM chip.

We used two Xilinx's Virtex2-pro xc2vp20 FPGA chips for fast development and easy test. FPGA1 contains microprocessor, dedicated program memory, and dedicated RAM. As a microprocessor, we employed EISC3208H core module from ADchips Corp [16].

The user program and test code are developed in the EISC Studio and downloaded to the microprocessor through EISC Serial Flash Programmer (ESFP). FPGA2 is used for dedicated cryptographic modules. The developed cryptographic module is downloaded into FPGA2 and con-

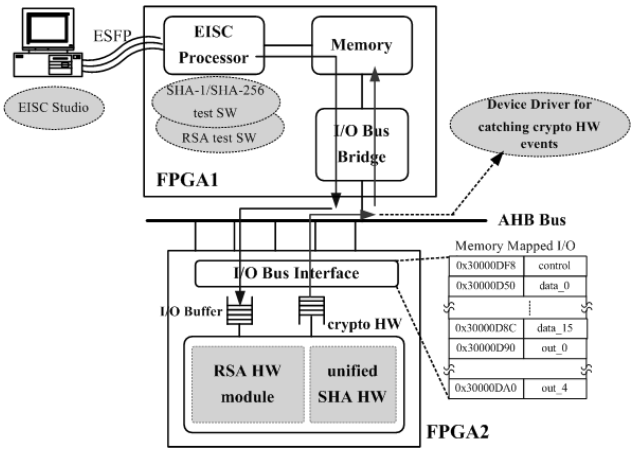

Fig. 7. Architecture of evaluating system

nected to an AHB slave bus system. Polling was used to ensure reliable communication between microprocessor and cryptographic hardware.

The microprocessor signals the cryptographic module to begin its computation after the former fills the input message into the latter and polls FPGA2 to find out the ending of the cryptographic operation. Then, the microprocessor starts the new SHA-1 invocation cycle.

For the performance evaluation of the actually implemented cryptographic circuit, we measured the data transmission time between the cryptographic hardware in FPGA2 and an application program in FPGA1, in addition to RSA or SHA hardware module's calculation time itself. We measured the elapsed time using the hardware clock timer located in the EISC3208H processor.

Table 4 shows the comparison results measured on the system level test with the most representative commercial TPM chips, AT97SC3203 and SSX35A manufactured by Atmel and Sinosun respectively.

From the Table 4, we can see that the proposed architecture is at least two or three times faster than commercial TPM chips in SHA and RSA computation speed at the system level test. Although these commercial TPM chips do not provide their power consumption measurement data, we believe that the proposed architecture spends the energy less than that of these chips, since the proposed design is optimized mainly to minimize the number of gates and their implementation area, assuring that the power consumption rate of the proposed design would be no greater than that of AT97SC3203 [14] and SSX35A [15].

Table 4. Comparison with commercial TPM chips about cryptographic computations on the system level

Parameter	This work	AT97SC3203[14]	SSX35A[15]
Operating frequency	20MHz	33MHz	33MHz
RSA-2048 Sign/Verify	585/45ms	500ms/--	300/40ms
RSA-1024 Sign/Verify	190/17ms	100ms/--	120/15ms
SHA-1 (64 bytes)	18us	50us	--
SHA-1/-256(1M bits)	106/114ms	--	258ms/--

5. Conclusion

This paper presents a compact architecture for a cryptographic engine possible on a mobile platform, which has very stringent limitations with respect to the circuit area and the consuming power. The presented architecture is a highly effective architecture that can implement the scalable RSA and unified SHA algorithms with a minimum resource usage.

The presented cryptographic hardware has a chip area of 38K gates for RSA and 12.4K gates for unified SHA respectively. The current consumption of the proposed cryptographic hardware consumes at most 3.96mA for RSA and 2.1mA for SHA computing under the 25MHz.

Compared to the other academic implementations and some commercial TPM chips supporting RSA and SHA-1 hardware modules, the proposed design demonstrated the smallest area in terms of logic gates. Furthermore, according to the implementation result of system level test, the computation speed of the proposed design is at least 200% faster than that of commercial TPM chip supporting RSA and SHA-1 circuit, while using lower operating frequency.

In summary, the combined performance results of circuit area, power efficiency, throughput, and functionality strongly indicate that the proposed architecture for cryptographic hardware is suitable for mobile computing systems and other low-end embedded systems that urge for high performance and small-sized solutions.

References

[1] Trusted Mobile Platform NTT DoCoMo, IBM, Intel. *Trusted Mobile Platform: Hardware Architecture Description Rev1.0.* Trusted Computing Group, 2004.

[2] Trusted Computing Group, *TCG mobile reference architecture specification, version 1.0*, June. 2007. https://www.trustedcomputinggroup.org

[3] Roger L. Kay, "How Hardware Security Will Become Nearly Ubiquitous as a Rock Solid Solution to Safeguarding Connected Computing," 2006. http://www.ndpta.com/TPMForecast.html

[4] ETSI TS 102.221: "UICC-Terminal Interface; Physical and Logical Characteristics".

[5] K.Shimohigashi and K.Seki, "Low-Voltage ULSI

Design," *IEEE Journal of Solid State Circuits*, 28(4), pp.408-413, 1993.

[6] NSA, Fact Sheet Suite B Cryptography, http://www.nsa.gov/ia/industry/crypto_suite_b.cfm

[7] Trusted Computing Group, *Trusted Module Library: Commands and Structures*, Specification version 0.7, Level 1 Revision 030, 28 NOV., 2007.

[8] M.Feldhofer and C.Rechberger, "A Case Against Used Hash Functions in RFID Protocols," *OTM Workshops 2006, LNCS 4277*, pp. 372-381, 2006.

[9] Y.Choi et al, "Low power implementation of SHA-1 algorithm for RFID system," *Proc. of ISCE 2006*, pp.1-5, 2006.

[10] M.Feldhofer and J.Wolkerstorfer, "Strong Crypto for RFID Tags -A Comparison of Low-Power Hardware Implementations," *Proc. of ISCS 2007*, pp.1839-1842, 2007.

[11] Toru Hisakad et al, "61.5mW 2048-bit RSA Cryptographic Co-processor LSI based on N bit-wised Modular Multiplier," *IEEE VLSI-DAT*, pp. 1-4, 2006.

[12] Xinjian Zheng, Zexiang Liu, and Bo Peng, "Design and Implementation of an Ultra Low Power RSA Coprocessor," *IEEE WiCOM'08*, pp. 1-5, 2008.

[13] Chingwei Yeh, En-Feng Hsu, "An 830mW, 586kbps 1024-bit RSA Chip Design", *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 24-29, 2006.

[14] AT97SC3203 Advance Information Summary, Atmel, http://www.atmel.com/dyn/products/product_card.asp?part_id=3736.

[15] SSX35A, Sinosun, available at: http://www.sinosun.com.cn/eng/product/index.asp

[16] EISC3208, ADChip Inc., http://www.adc.co.kr/.

Mooseop Kim

He received his B.S. in Electrical Engineering from Kumoh National Institute of Technology, Korea, a M.S. in Electrical Engineering from Kyungpook National University, Korea, and a Ph.D degree in Computer Science and Engineering from Chungnam National University, Korea, in 1995, 1998, and 2009, respectively. He was a research engineer in the Organic LED (OLED) group in the Device and Materials Laboratory, LG Electronics Institute of Technology (LG Elite), Seoul, Korea from 1998 to 1999. He has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, since 1999 and currently he is a senior research engineer. His current research interests include low-power circuit design, reconfigurable computing, cryptographic circuit, and embedded systems.

Youngsae Kim

He received the BS and MS degrees in Electrical Engineering from Kyungpook National University, Korea, in 1999 and 2001. He has been with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, since 2001. He is currently a senior member of engineering staff. His research interests include cryptographic circuit and embedded SoC designs.

Cho, Hyun Sook

She received her MS and Ph.D. degree in Electrical and Computer Engineering from Chungbuk National University in 1991 and 2001 respectively. Currently, she is the Director of Knowledge-based Information Security & Safety Research Department of Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. Her research interests include information security, network security and convergence security.