TECHNICAL SCIENCES
Abbrev.: Techn. Sc., No 15(1), Y 2012

SECURE DOCKING STATION AND ITS PROTECTION AGAINST HARDWARE ATTACKS

***Apostolos P. Fournaris*[1,2], *Jacques Fournier*[3], *Daniel Hein*[4], *Guillaume Reymond*[3]**

[1] Electrical and Computer Engineering Dpt, University of Patras, Rio Campus, Greec
[2] KNOSSOSnet Research Group, Informatics and Mass Media Dpt, Technical Educational Institute of Patras, Greece
[3] CEA-LETI Minatec, Gardanne, France
[4] Institute of Applied Information Processing and Communications, Graz University of Technology, Graz, Austria

K e y w o r d s: Hardware, Security Module, Physical attacks, AES, RSA, security counter – measures.

A b s t r a c t

Security and Trust in communication systems where very sensitive information are exchanged is achieved and retained through hardware means. In the SECRICOM project where seamless, interoperable crisis management communication is required, we have developed a security and trust managements mechanism based on a smart card like hardware structure called Secure Docking Module (SDM). However, given the highly secure and hostile environment (emergency, crisis situation) where the SDM needs to function, this security module can be the subject of many attacks. While cryptanalytic attacks on the SDM security are impossible due to the employed strong cryptographic algorithms, attacks targeting the SDM implementation constitute a pragmatic threat that cannot be neglected. In this paper, we address possible hardware issues of the SDM chip and focus on the Hardware attack protection mechanisms especially on the SDM RSA and AES cryptographic accelerators. We present the research work that was done through the SECRICOM project on the above issues and analyze the basic concept behind the protected RSA-AES structures that complement the SDM architecture. Those hardware structures are fully compatible with the SDM protocols and offer strong protection against hardware power attacks and fault attacks while retaining high performance characteristics.

MODUŁ SECURE DOCKING STATION ORAZ JEGO OCHRONA PRZED ATAKAMI SPRZĘTOWYMI

***Apostolos P. Fournaris*[1,2], *Jacques Fournier*[3], *Daniel Hein*[4], *Guillaume Reymond*[3]**

[1] Electrical and Computer Engineering Dpt, University of Patras, Rio Campus, Greece
[2] KNOSSOSnet Research Group, Informatics and Mass Media Dpt, Technical Educational Institute of Patras, Greece
[3] CEA-LETI Minatec, Gardanne, France
[4] Institute of Applied Information Processing and Communications, Graz University of Technology, Graz, Austria

S ł o w a k l u c z o w e: mechanizmy bezpieczeństwa, Secure Docking Module (SDM), ataki sprzętowe.

Abstrakt

Bezpieczeństwo i zaufanie w systemach łączności, gdzie są przetwarzane informacje niejawne, jest zapewniane za pomocą rozwiązań sprzętowych. W projekcie SECRIOM, w którym jest wymagana interoperacyjna oraz „bezszwowa" łączność w zarządzaniu kryzysowym, wytworzono mechanizm zapewniania bezpieczeństwa oraz zaufania oparty na rozwiązaniu typu kart inteligentnych – Secure Docking Module (SDM). Biorąc jednak pod uwagę wysoki poziom zagrożenia środowiska łączności w sytuacjach kryzysowych, sam moduł SDM może być przedmiotem wielu ataków. Pomimo że ataki kryptoanalityczne na SDM są niemożliwe ze względu na zastosowane silne algorytmy kryptograficzne, zagrożenie wynikające z ataków na implementację SDM nie powinno być zaniedbywane. W artykule opisano możliwe problemy rozwiązań sprzętowych w chipie SDM oraz wyeksponowano mechanizmy zapobiegania atakom sprzętowym, szczególnie skierowanym na SDM RSA i akceleratory kryptograficzne AES. Zaprezentowano ponadto struktury RSA-AES, które uzupełniają architekturę SDM z punktu widzenia wzmocnienia ochrony. Te struktury sprzętowe są w pełni kompatybilne z protokołami w ramach SDM i oferują silną ochronę przed atakami fizycznymi, jednocześnie nie obniżają wysokich właściwości użytkowych.

Introduction

Trust in Information Systems constitutes a fundamental security issue in most sensitive data handling applications. However, achieving a high level of trust in the entities of such systems is not an easy task. There are some computer communication systems where the nature of the handled information is so sensitive that untrusted behaviors cannot be tolerated. In such systems, security and trust is ensured by hardware means.

SECRICOM European project is based on the efficient, seamless communication of civil emergency responders in situations of crisis (SECRICOM, 2008). The project's goal is to provide to emergency agencies a communication infrastructure that is fully interoperable regardless of what devices (mobile phone, smart phone, PC, Tablet, Push-to-Talk equipment e.t.c.) or communication system each agency uses (analog radio, GSM, 3G, TETRA, wifi, Internet e.t.c.).

In such a communication environment, strong security places a very important role. The communication channel where the all transactions are made should always remain secure and protected from eavesdropping and involved emergency responders must have trust to the communication they are engaged in and also trust that their device provides them with accurate, untampered data meaning that it has not been compromised. While there are several hardware means of achieving high security, very few solutions exist when it comes to offering trust. Trusted Computing Group's TPM chip is the most promising such solution however this Hardware Security Module (HSM) is not easily deployable in crisis situations where extreme conditions are at hand (portability of actors, various communication means, frequent disruptions of communication channels e.t.c.). TPM requires remote attestation

procedures in order to guarantee a high trust level. In the crisis management case, that cannot and may not be provided, it would be much better if trust attestation is provided locally. The above reasons stemmed the need for a local security and trust attestation mechanism so within the SECRICOM project we designed a passive smart card like hardware token, complementing the TPM functionality and acting as a local trusted third party, capable of storing security keys, credentials and attesting the trust level of devices connected to it.

The SDM is described as an SD, MMC card or usb token that is physically attached to a Host machine and upon request from its host, releases the keys related to this host only if the host provides sufficient credential that it is in a trusted state. The keys for each host along with the host's id and public key are stored in the SDM secure memory. The SDM has a unique id number and a set of cryptographic keys (public, private key pair) that should not be transmitted in any way through the communication channel. Apart from the above, the SDM holds a series of valid configuration states (PCR values) for each Host authorized to communicate with it in order to be able to verify the trust state of such host.

In a way, the SDM plays the role of a local trusted third party. The process of verifying the host's trust level is called local attestation since it is similar to remote attestation but do not require a network communication channel since the SDM is attached to the host device (the attestation is performed locally). In an SDM enabled environment, the various programs of the network are controlled by trusted servers and are cryptographically secured using specific keys.

Based on the above concept, the SDM is capable of validating the local software integrity of a Host platform through trust measurements and providing sufficient proof that the measurements are authentic, fresh and untampered. The SDM as an add-on structure on the user's communication device is associated to a specific user through a password mechanism and therefore can bind a user along with the device to the crisis management communication system. As such, the fundamental security principle of user non-repudiation is retained, the user is bonded to the SDM and cannot deny its actions.

The SDM secret information that are handled by its cryptographic algorithms (RSA and AES) cannot be deduced using traditional cryptanalysis since the above algorithms are considered highly secure especially for high bit length keys (in the SDM case, 2048 bit RSA and 256 bit AES is used). However, there exist a series of attacks that do not target the cryptographic algorithm itself but the algorithm's implementation that can be successful in deducing secret data. Even if secret information cannot be learned, attackers may be able to disrupt the SDM hardware or deny service leading to other kinds of failures in

the SECRICOM security system. Those Hardware attacks are powerful yet easy to mount and can be invasive, semi-invasive and non invasive. While invasive attacks require considerable expertise, chip depackaging and special equipment (laser cutter microscope, probes) in order to work, semi-invasive and non invasive attacks can be mounted by even inexperienced attackers following instructions or with cheap equipment. For the above reasons, in the SDM system design special care must be taken in order to include resistance against the above hardware attacks.

In this paper, we elaborate on the hardware structure of the SDM chip and focus on possible ways of attacking the SDM structure through hardware means especially by targeting the RSA and AES cryptographic accelerators. We present the research work that was done through the SECRICOM project on the above issues including Fault attack and side channel attack protection mechanisms capable of thwarting hardware attacks. The proposed hardware structures are fully compatible with the SDM protocols offering strong protection against power attacks and fault attacks while retaining high performance characteristics.

The remaining of the paper is organized as follows. In section 2, the SDM architecture is discussed briefly. Section 3 provides a general overview on Hardware attacks that can be mounted on the SDM and section 4 focuses explicitly on attacks on the SDM AES and RSA accelerators along with implemented countermeasures. Section 5 concludes the paper.

SDM Hardware structure

We envision the SDM as a synchronous System on Chip (SoC) device. The hardware structure of the SDM can be determined by the functions that it must fulfill. The SDM due to its potential connection with a TPM has a TPM-like structure and includes an RSA signature unit, a control processor unit, a non volatile memory unit for key storage, a true random number generator unit (TRNG), a SHA-1 hash function unit and a symmetric key encryption/decryption and key generation unit (AES algorithm).

The generic hardware structure of the SDM chip is presented in Figure 1. The system is structured around a data bus where all the data values are transferred for reading by or writing to a requesting unit of the SDM. There is also an address bus connected to the memory unit for a successful memory data reading and writing. An additional bus is also connected to all the units of the chip which is responsible for passing all the control signals to those units. Signals of this bus are in general managed by the processor.

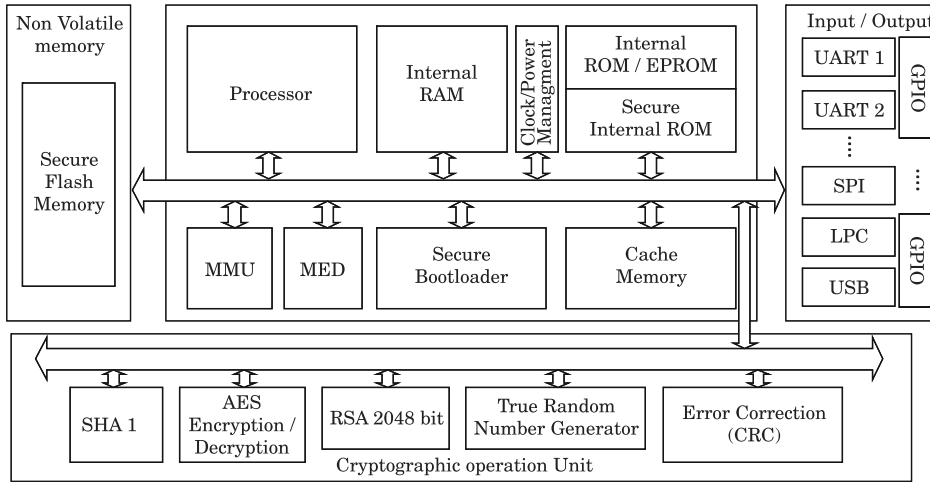

Fig. 1. The SDM hardware structure

The processor unit is responsible for controlling the whole SDM system and realizing the SDM functionality by enabling-controlling (chip select, CS, signal) SDM component units and performing operations that don't require the involvement of other units (i.e. comparisons or memory search). For this task, the processor has stored in its ROM memory a firmware program implementing the various SDM commands and a series of data required by those commands.

There are three memory modules included in the SDM hardware structure. The first module is a RAM unit that is employed for temporary value storage during a single SDM to Host protocol session. The second module is a ROM – EEPROM unit that is mainly used for storing and updating the SDM firmware realizing the SDM functionality. The third module is an NVRAM (flash memory) unit which constitutes the main storage area for all the sensitive information involved in the AAT should be SDM transactions. The three memory units are protected by special hardware structures that deter attackers from deciphering memory data.

The SHA-1 unit is implementing the SHA-1 hash function and the RSA encryption – decryption unit is responsible for performing the arithmetic operation of modular exponentiation ($m^e mod N$) as defined in RSA public key scheme. The SHA-1 unit has a data input/output and a control signal indicating the beginning of a hash function operation. The RSA unit has as inputs the modulus N value (part of the RSA public key), the message m to be encrypted-decrypted and the public or private key e along with a control signal indicating the beginning of a modular exponentiation encryption or decryption.

The AES encryption/decryption unit is responsible for the key generation, encryption and decryption of an established session's data that are transmitted to and from the SDM. It has control signals that indicate an encryption, decryption and key generation operation. The AES generated session key is only saved within the AES unit and is changed when is replaced by a newly generated session key.

The TRNG unit is connected to the data path through its data output and has a 9 bit control signal that determines the bit length of the generated random number. The NVRAM has a read/write control signal while is connected to the data and address bus in order to read the address and use it to write or read the data values in or out of it. The system has a clock generator unit for managing the SDM different clocks and a reset signal along with chip select (CS) signals to enable or disable each SDM unit. The system works in a synchronous way.

Note, that in order to ensure a high security level, the RSA keys used in the SDM have a bit length of 2048 bits. As a result, the data related to the RSA encryption and decryption will have similar bit length. However, there is no feasible processing system able to operate with buses of such bits. Therefore, the 2048 bit values are broken into several blocks (to match the bus bit length) and reconstructed inside the SDM units (in example the RSA encryption-decryption unit). The same problem exists with the SHA-1 unit that handles 160 bit values and is solved in a similar way.

Hardware Attacks on SDM

The SDM can be considered as an embedded system that operates in hostile environment from security perspective. It can be stolen and manipulated in order to give out the sensitive information that are stored or processed in it. While the SDM system is protected from cryptanalytic attacks by the use of strong security schemes (RSA and AES), it can still be compromised when an adversary applies a hardware attack, an attack on the implementation itself. Three types of such attacks can be discriminated.

1. Invasive attacks. These attacks aim at physically disrupting the correct operation of an embedded chip. They involve removing chip packaging, micro probing of the chip's activity (memory, registers, buses) and physically tampering – interfering with the chip functionality. Invasive attack techniques begins by chip depackaging, removal of on-chip protection layers (depassivation) and modifying with probing tools the executed code or change values in Registers or simply observing the behavior of static/dynamic RAM blocks after power off since such units tend to "remember" values long after no power is applied to

them. Invasive attacks are not easily mounted, require attackers with considerable on chip expertise and expensive, specialize equipment like laser cuter microscope, micro probes e.t.c. Such attacks are considered difficult to be mounted on the SDM chip.

2. Semi invasive Attacks. These attacks aim at observing the behavior of the embedded system chip after an attacker specialized triggering. Like invasive attacks, they require depackaging the chip in order to get access to its surface. However, the passivation layer of the chip remains intact, as semi-invasive methods do not require depassivation or creating contacts to the internal lines. The goal is to induce a fault in the computation flow of the chip during a cryptographic operation and observe the cryptographic result as the fault propagates. The attacker can deduce sensitive information from such result on an unprotected embedded chip. In these attacks, also called Fault attacks, faults are injected using power or clock glitches, extreme variations in temperature, UV radiation or even optical laser beam induction. Depending on the attacker's equipment and expertise, the fault attacks are moderately difficult to mount. Fault attacks constitute a real danger for the SDM security.

3. Non-invasive Attacks. Such attacks, also called side channel attacks, exploit an embedded system's hardware characteristics leakage (power dissipation, computation time, electromagnetic emission e.t.c) to extract information about the processed data and use them to deduce sensitive information (cryptographic keys, messages e.t.c). An attacker does not tamper with the chip in any way and needs only make appropriate observations to mount a successful attack. Side channel attacks can be mounted very easily, cheaply, using a PC, a digital oscilloscope and some probes. Therefore, they can be mounted to an embedded system device by even the most inexperienced attacker. This ease of use makes SCA very potent. Some of the most widely used side channel attacks are the following:

– Power Attacks. These attacks involve physical measurement of the power dissipation emitted from the chip during cryptographic operations. Simple power signal analysis can reveal what mathematical operation is performed in the chip (e.g. modular multiplication or squaring during an RSA operation) and since in most cases the operation is related to the secret/private keys this action can reveal the key itself. Even if the chip is protected against simple power analysis, it is not fully resistant against power attacks since differential power analysis can still lead to compromise. In differential power analysis, the attacker perform guesses about a secret/private key bit, collects the related to this hypothesis power signal and correlates it with the actual power signal. The strongest correlation between the hypothesis and the actual measurement is the correct guess. Taking enough power samples and correlations the secret/private key can be revealed.

– Electromagnetic attacks. These attacks use the electromagnetic radiation emitted from the embedded system chip for simple analysis or differential analysis in a similar way as power attacks. In electromagnetic attacks, specific chip areas can be targeted and no physical access to the chip is strictly required (they can be mounted from afar). On the other hand, the existence of physical noise, RF interference or measurement error limits the attack's effectiveness.

Countermeasures

Designing countermeasures for Hardware attacks is not an easy task. Each security and cryptographic algorithm has its own vulnerabilities when implemented to hardware and therefore a ubiquitous approach toward Hardware attack resistance is impossible. In general, two approaches for countermeasures are used in practice, algorithmic based countermeasures and circuit based countermeasures. Algorithmic countermeasures aim at modifying the cryptographic algorithm and associated computer algebra operations so that when implemented it will leak to an attacked as less information as possible. Such countermeasures are more focused to semi-invasive and non-invasive attacks. Circuit countermeasures are hardware structures added to a cryptographic algorithm's hardware architecture, implementation or packaging, capable of detecting or thwarting a hardware attack. Such countermeasures can be used to protect an embedded system against all kinds of hardware attacks.

Invasive attack resistance is achieved by designing special structures during chip packaging and assembling in order to provide tamper evidence, detection and resistance. This may include mesh sensors implemented in the metal layer after packaging consisting of serpentine patterns of ground and power lines that are shortcircuited if attempts on depackaging or depassivation are done thus destroying the chip. Also, the on chip silicon layers can be designed in such a way that visual chip surface analysis through microscope is very difficult. Adding multiple layers with metal layers in between is such a technique applied during chip fabrication. All the invasive attack countermeasures are circuit based countermeasures.

Semi-Invasive attack resistance can be achieved by using some of the countermeasures for thwarting chip depackaging attempts (used in invasive attack resistance), however, usually such countermeasures are not enough or are too expensive. So, semi-invasive attack countermeasures are focused on detecting fault injection during cryptographic algorithm execution. One approach toward this end is to modify the cryptographic algorithm so as to

support infective computation. The basic concept of infective computation is that any computational errors introduced by a fault will propagate throughout the cryptographic computation, thus ensuring that the final result appears random and useless to the attacker in the end. Another approach that can be combined with infective computation is the design of specialized fault detection units in the cryptographic algorithm hardware architecture capable of detecting single or multiple faults. Such units involve elegant circuit design as well as modifications in the cryptography algorithmic flow to include specific conditions between intermediate values that the fault detection unit must detect after the computations are concluded but before the cryptographic result is released. When faults are detected then a random number or zero value is released thus denying an attacker any useful information about secret/private keys.

There is a wide variety of non-invasive countermeasures depending on what side channel attack they thwart. The basic goal of all countermeasures is implementing the cryptographic architecture in such a way that the implementation's characteristics like power consumption, timing or electromagnetic radiation, leaks as little as possible of the secret keys or data. This can be achieved either by scrambling the leakage signal in such a way that is unrelated to the secret information that it is computed in the cryptographic unit or by minimizing the leakage as a whole so that it is very difficult for an attacker use it for a side channel attack. The first approach is related to algorithmic countermeasures that aim at inserting randomization through the cryptographic algorithm computation flow by providing Boolean, or arithmetic (multiplicative or additive) masking of the secret information (multiplication or addition with a random number). These countermeasures, also known as blinding, is very useful against Differential attacks since they aim at decorrelation of the secret data with the leakage itself. The second approach is related to algorithmic and mostly circuit countermeasures. Through special circuitry, like double rail technique, power rebalancing or additional dummy operations (redundancy), we aim at normalizing the leaked signals so that they remain unchanged during cryptographic operations. In general, it should be mentioned however, that protection against side channel attacks is never expected to be absolute: a determined attacker with a vast amount of resources can eventually, given enough time and effort compromise an implementation. The goal from cryptographic engineering perspective is to realize in the cryptographic accelerator enough side channel attack countermeasures so that an attack on the system becomes too expensive in effort or cost to be interesting (MANGARD et al. 2007).

Designing Protection Measures against SDM Side channel Attacks and Fault Attacks for AES and RSA

AES Accelerator countermeasures

AES accelerator implementation of the SDM can be the target of several fault and side channel attacks. Side channel attacks of special interest are differential attacks like *Differential Power or EM Analysis* (KOCHER et al. 1999, GANDOLFI et al. 2001, QUISQUATER et al. 2001), *Correlation Power Analysis* or *Mutual Information Analysis* (BRIER et al. 2004, GIERLICHS et al. 2008). Very potent AES implementation fault attack is *Differential Fault Analysis* (DFA) (BIHAM, SHAMIR 1997, PIRET et al. 2003, GIRAUD 2005).

Specific countermeasures on AES resistance against side channel attacks consist in either adding noise to blur the measurements or reducing the information-rich signal. "Balancing" consists in rendering the Hamming Weight (HW) or Hamming Distance (HD) of sensitive internal data constant by using "dual-rail" (each bit is encoded onto two wires) with, say, a "Return-to-Zero" (RTZ) (TIRI et al. 2003, SOARES et al. 2008, AMBROSE et al. 2011, CHEN et al. 2010). The propagation of the encoded values between the different parts of the circuit can also be physically balanced by using ad hoc Place and Route (P&R) techniques (GUILLEY et al. 2005). Moreover, noise can be added by randomizing the order of the instructions, by adding dummy operations or by masking the internal computations (AKKAR et al. 2001, TOKUNAGA et al. 2009).

As already stated in the previous subsection, countermeasures against fault attacks consist either in detecting errors during the computation and then taking actions to protect data or in making the circuit less sensitive to fault injections. The detection of error is mainly based on information redundancy either in space or in time (BERTONI et al. 2002, KARRI et al. 2003, KARPOVSKY et al. 2004) and can be further enhanced by placing several sensors in order to detect abnormal modifications of the chip's environment (voltage, temperature, clock frequency, light, etc.). Once a fault has been detected, reactions may consist in stopping the communication with the outside and/or resetting parts of the running software or deleting the sensitive data etc.

However, existing countermeasures do not address both fault and side channel attacks. In the AES SDM accelerator we realize countermeasures that are meant to thwart both classes of attacks. To achieve this, we designed an architecture based on duplicated-complemented (also called "dual") data paths applied to the AES algorithm. The dual data paths balance the data HW and are also used to detect faults.

The AES protection against DFA consists of detecting faults and reacting on them. We use spatial duplication in order to achieve that, as suggested in

(MALKIN et al. 2005) and implement two instances of the algorithm working in parallel. By checking the consistency between the results of the two instances we can determine if a fault have been injected (the outcome of the two instances won't match). If a fault injection is detected then the AES architecture reacts by returning an error value instead of the correct result. This value is extracted by blurring the erroneous ciphertext with the scrambled value of the detected fault. More on the fault detection mechanism can be found in (JOYE et al. 2007).

The AES protection against side channel attacks consists on designing the two parallel instances of the algorithm in such a way that when a bit of each intermediate value is computed in one instance, the other instance computes the complementary value. This approach effectively scrambles the power signal and disassociates power dissipation with the AES processed information (DOULCIER-VERDIER et al. 2011).

RSA Accelerator countermeasures

In the RSA cryptographic scheme, three n-bit numbers are used, the public modulus N, the public key e and the private key d. Let $N = p \cdot q$, where p, q are secret prime numbers. Let also $e \cdot d = 1 \bmod (p-1)(q-1)$. Assuming that m is the message to be encrypted (plaintext), the RSA encrypted outcome (ciphertext) is $c = m^e \bmod N$ and decrypted outcome is $m = c^d \bmod N$. CRT is usually used during RSA decryption since the bit length of the private key d is bound to be long. In CRT RSA, we compute $S_p = c^{dp} \bmod p$ and $S_q = c^{dq} \bmod q$, where $d^p = d \bmod (p-1)$ and $d^q = d \bmod (q-1)$. Then, the final result is computed following Gauss's combination algorithm, meaning

$$S = \mathrm{CRT}(S_p, S_q) = (S_p \cdot q \cdot q_i) + (S_q \cdot p \cdot p_i) \bmod N \tag{1}$$

Where:
$q_i = q^{-1} \bmod p$, $p_i = p^{-1} \bmod q$.

The main computation and side channel attack security bottleneck of the RSA structure is the modular exponentiation operation. This operation is realized as a iterative process of modular multiplications that by themselves have considerable computation cost. Also, the exponent in modular exponentiation, which is usually sensitive information (private key), determines the computation flow during the operation's execution sequence and needs to be fast enough in order to support the SDM functional and non-functional specifications (speed, hardware resources e.t.c.).

We propose optimizing and enhancing the modular multiplication operation, constituting the heart of modular exponentiation and mounting hardware and algorithmic SCA countermeasures on the modular exponentiation operation itself. To achieve the first goal, an optimized version of Montgomery modular multiplier, described in (FOURNARIS 2010), is adopted. This structure employs Carry-Save logic in all its inputs, outputs and intermediate results as well as constant value precomputation during the multiplication algorithmic flow, managing to reduce considerably both the time and space complexity of modular multiplication.

To achieve the second goal, we focus our counter measures on popular side channel attacks that can easily be mounted on RSA. Such attacks are Power attacks (especially simple power attacks) and fault attacks. RSA Side Channel Attack countermeasures can be generic, on circuit level, (more effective against power attacks) (BHATTACHARYA et al. 2008), (TIRI et al. 2006) or specialized, focused on specific cryptographic algorithms, on an algorithmic level. The second approach can be more effective since it utilizes techniques that better negate the RSA cryptoalgorithm's specialized SCA weaknesses. Our design adopts the second approach since it makes the proposed architecture Hardware agnostic, meaning that it can guarantee Side Channel Attack resistance regardless of the Hardware implementation technology.

In general, the target of RSA Fault and Power attacks is the modular exponentiation unit. The use of CRT, increases RSA vulnerability to such attacks, so strong countermeasures are needed for modular exponentiation protection. Simple Power attack (SPA) resistance is achieved by making the arithmetic operations during the exponentiation algorithm execution undiscriminated from an external observer (JOYE, YEN 2003). RSA Fault attack countermeasures are based on techniques of detecting single fault injection and blocking further processing thus prohibiting the release of secret information (AUMULLER et al. 2002).

Assuming that a fault is introduced during the first exponentiation (with modulus p), then the faulty output would be S'_p and the CRT reconstruction in (1) would be $\bar{S} = \mathrm{CRT}(S_p,S_q) = (S'_p \cdot q \cdot q_i) + S_q \cdot p \cdot p_i) \mathrm{mod} N$.

Knowing a legitimate CRT-RSA outcome S and a faulty one $\bar{S}$, one can find the secret prime q by calculating $q = gcd((S - \bar{S}), N)$. The deliberate insertion of a fault in the computation flow of one of the exponentiations constitute a fault attack, originally proposed by BONEH et al. (1997) and enhanced by LENSTRA (1996) where no legitimate outcome is also needed, as can be observed by $q = gcd((\bar{S}^e = m) \mathrm{mod} N, N)$.

To thwart the above attack, we employ Girauld's technique of detecting faults (GIRAUD 2006). This approach uses Montgomery power ladder as an

iterative modular exponentiation algorithm thus apart from fault detection it offers resistance against SPA. Two Montgomery power ladder calculations are performed, in parallel in each modular exponentiation round with different initial inputs each. The initial input of the second Montgomery power ladder is the first round's result of the first Montgomery power ladder. At the end of each round, two values are calculated, S and S_1, that have a known mathematical connection between them ($S_o = {}^{O}m \cdot S_1 \text{mod}(p \cdot q)$). This connection exists due to appropriate initialization at the beginning of the algorithmic flow. Fault detection is performed in the end of a modular exponentiation after CRT reconstruction by checking if the connection between S_o and S_1 is true. If this test fails then a fault attack is detected and the cryptographic processes are canceled. Thorough analysis revealed that the above technique is not completely fault attack resistant as observed by KIM and QUISQUATER (2007), since a carefully injected fault after the final Montgomery power ladder round and before the fault detection operation theoretically can damage the whole protection mechanism without being detected. So, to thwart this attack, the final Montgomery power ladder round result must be masked by adding a random number a that is to be removed after CRT reconstruction and fault detection. In that case, the correct (unmasked) RSA result is not revealed nor stored during the whole RSA computations only after fault detection test is

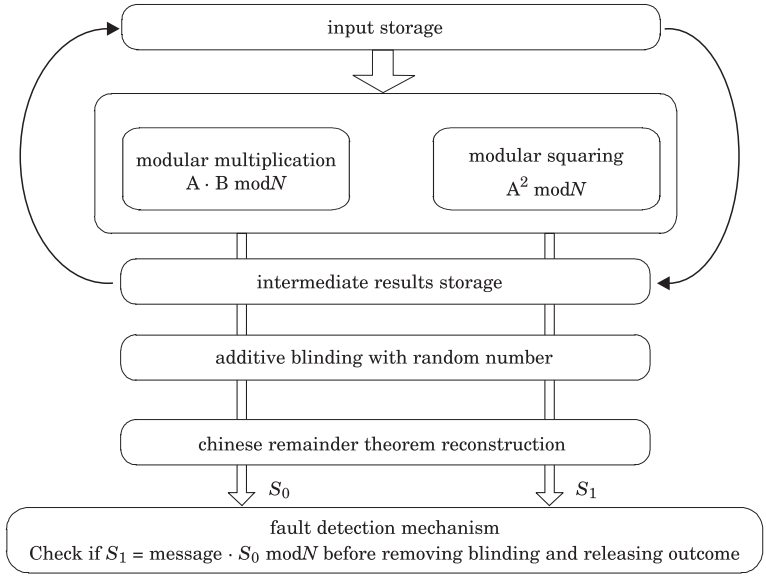

Fig. 2. The SDM RSA accelerator concept and Hardware attack protection mechanism

passed. More on the SDM RSA accelerator can be found in (FOURNARIS 2010, FOURNARIS, HEIN 2011, FOURNARIS, KOUFOPAVLOU 2011). The concept behind the above approach is presented in Figure 2.

Conclusions

In this paper, the SECRICOM project hardware security and trust module (SDM) structure was analyzed and possible hardware attacks on its structure where mapped. The cryptographic accelerator structures (RSA and AES) of the SDM were identified as potential targets of such attacks and specific fault and side channel attacks on these structures were presented. Specific countermeasures both for side channel and fault attacks were presented for SDM AES and RSA implementations based on spatial duplication (for AES) and modified Montgomery Ladder technique (for RSA). The presented methodologies manage to protect the SDM structure from the most popular side channel and fault attacks and since they are based on open architectures (not proprietary), they can be expanded in the future to include resistance against attacks yet to appear.

Acknowledgment

This work was funded by the SECRICOM project (EC FP7-SEC-2007 grant 218123).

Translated by AUTHORS

Accepted for print 30.06.2012

References

AKKAR M.-L., GIRAUD C. 2001. *An Implementation of DES and AES, Secure against Some Attacks.* In: *Proceedings of CHES'01*. Edited by Çetin Koç, D. Naccache, C. Paar. LNCS, 2162: 309–318, Springer-Verlag, Paris, France.

AMBROSE J., RAGEL R., PARAMESWARAN S., IGNJATOVIC A. 2011. *Multiprocessor information concealment architecture to prevent power analysis-based side channel attacks.* Computers Digital Techniques, IET, 5(1): 1–15.

AUMULLER C., BIER P., FISCHER W., HOFREITER P., SEIFERT J.-P. 2002. *Fault Attacks on RSA with CRT: Concrete Results and Practical Countermeasures*, In: *Revised Papers from the 4th* International Workshop on Cryptographic Hardware and Embedded Systems (CHES '02), Eds. B.S. Kaliski Jr., C.K. Koç, Ch. Paar. Springer-Verlag, London, UK, pp. 260–275.

BERTONI G., BREVEGLIERI L., KOREN I., MAISTRI P., PIURI V. 2002. *A parity code based fault detection for an implementation of the advanced encryption standard.* In: *Proceedings of DFT'02*. IEEE Computer Society, Washington, DC, USA, pp. 51–59.

BHATTACHARYA K., RANGANATHAN N. 2008. *A linear programming formulation for security-aware gate sizing.* In: *GLSVLSI '08*. Proceedings of the 18th ACM Great Lakes symposium on VLSI. 1em plus 0.5em minus 0.4em New York, NY, USA: ACM, pp. 273–278.

BIHAM E., SHAMIR A. 1997. *Differential Fault Analysis of Secret Key Cryptosystems.* In: Proceedings of CRYPTO '97, LNCS, 1294: 513–525.

BONEH D., DEMILLO R.A., LIPTON R.J. 1997. *On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract).* In Proc. EUROCRYPT, pp.37–51.

BRIER E., CLAVIER C., OLIVIER F. 2004. *Correlation Power Analysis with a leakage model.* In: Proceedings of CHES 2004. Edited by M. Joye and J.-J. Quisquater, Lecture Notes in Computer Science, 3156: 16–29, Springer-Verlag.

CHEN Z., SINHA A., SCHAUMONT P. 2010. *Implementing virtual secure circuit using a custom-instruction approach.* In: Proceedings of CASES '10, ACM, New York, NY, USA, pp. 57–66.

DOULCIER-VERDIER M., DUTERTRE J-M., FOURNIER J., RIGAUD J-B., ROBISSON B., TRIA A. 2011. *A Side-Channel and Fault Attack Resistant AES circuit working on duplicated complemented values.* In: *Solid State Circuits Conference – Digest of technical papers, 2011 (ISSCC 2011).* Page 15.6, IEEE International.

FOURNARIS A.P. 2010. *Fault and Simple Power Attack Resistant RSA using Montgomery Modular Multiplication*. Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS 2010) IEEE, pp. 1875–1878).

FOURNARIS A.P., HEIN D.M. 2011. *Trust Management Through Hardware Means: Design Concerns and Optimizations*. In: Eds., N. Voros, A. Mukherjee, N. Sklavos, K. Masselos, M. Huebner, Symposium VLSI 2010 Annual, vol. 105, pp. 31–45. Springer Netherlands.

FOURNARIS A.P., KOUFOPAVLOU O. 2011. *Efficient CRT RSA with SCA countermeasures.* In: Proceedings of 14th Euromicro DSD '11. Oulu, Finland, pp. 593–599.

GANDOLFI K., MOURTEL C., OLIVIER F. 2001. *Electromagnetic Analyis: Concrete Results.* In: *Proceedings of CHES'01*, Eds. Ç. Koç, D. Naccache, C. Paar, *LNCS*, 2162: 251–261, Springer-Verlag, Paris, France.

GIERLICHS B., BATINA L, TUYLS P., PRENEEL B. 2008. *Mutual Information Analysis – A Generic Side-Channel Distinguisher.* In: *Proceedings of CHES'08.* Eds. E. Oswald, P. Rohatgi, Lecture Notes in Computer Science, 5154: 426–442, Springer-Verlag, Washington DC,US.

GIRAUD C. 2006. *An RSA implementation resistant to fault attacks and to simple power analysis.* IEEE Transactions on Computers, 55(9): 1116–1120.

GIRAUD C. 2005. *DFA on AES.* In: *Advanced Encryption Standard – AES.* Eds. H. Dobbertin, V. Rijmen, A. Sowa, Lecture Notes in Computer Science, 3373: 27–41, Springer Berlin / Heidelberg.

GUILLEY S., HOOGVORS T.P., MATHIEU Y., PACALET R. 2005. *The backend duplication method.* In: Proceedings of CHES'05, pp. 383–397.

JOYE M., YEN S.-M. 2003. *The Montgomery powering ladder.* In: CHES '02: Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems. 1em plus 0.5em minus 0.4em London, UK: Springer-Verlag, pp. 291–302.

JOYE M., MANET P., RIGAUD J.-B. 2007. *Strengthening Hardware AES Implementations against Fault Attack.* IET Information Security, 1: 106–110.

KARPOVSKY M. G., KULIKOWSKI K. J., TAUBIN A. 2004. *Robust protection against fault injection attacks on smart cards implementing the Advanced Encryption Standard.* In: Proceedings of DSN 2004, pp. 93–101, IEEE Computer Society.

KARRI R., KUZNETSOV G., GOESSEL M. 2003. *Parity-Based Concurrent Error Detection of Substitution-Permutation Network Block Ciphers.* In: Proceedings of CHES'03, LNCS, 2779: 113–124, Springer--Verlag, Cologne, Germany.

KIM C. H., QUISQUATER J.-J. 2007. *Fault attacks for CRT based RSA: New attacks, new results, and new countermeasures.* In: *WISTP*, Eds. D. Sauveron, C. Markantonakis, A. Bilas, J.-J. Quisquater. Lecture Notes in Computer Science, 4462. pp. 215–228, 1em plus 0.5em minus 0.4em Springer.

KOCHER P., JAFFE J., JUN B. 1999. *Differential Power* Analysis. Advances in Cryptology Proceedings of Crypto 1999, pp. 388–397. Springer-Verlag.

LENSTRA A. K. 1996. *Memo on RSA signature generation in the presence of faults*.

MALKIN T. G., STANDAERT F.-X., YUNG M. 2005./ *A comparative cost/security analysis of fault attack countermeasures.* In: Proceedings of FDTC'05, pp. 109–123, Edinburgh, UK.

MANGARD S., OSWALD E., POPP T. 2007. *Power Analysis Attacks: Revealing the Secrets of Smart Cards.* Springer.

PIRET G., QUISQUATER J.-J. 2003. *A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD.* In Proceedings of CHES'03, 2 LNCS, 779: pp. 77–88, Springer-Verlag.

QUISQUATER J.-J., SAMYDE D. 2001. *Electromagnetic Analysis (EMA): Measures and coutermeasures for smart cards.* In: e-smart 2001, LNCS, 2140: 200–210.

SECRICOM. 2008. *Seamless communication for crisis management.* http://www.secricom.eu/menu-objectives.

SOARES R., CALAZANS N., LOMNÉ V., MAURINE P., TORRES L., ROBERT M. 2008. *Evaluating the robustness of secure triple track logic through prototyping.* In: Proceedings of SBCCI'08, pp. 193–198, ACM, New York, NY, USA.

TIRI K., VERBAUWHEDE I. 2003. *Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology.* In: Proceedings of CHES'03, LNCS, 2779: 125–136, Springer-Verlag, Cologne, Germany, 2003.

TIRI K., VERBAUWHEDE I. 2006. *A digital design flow for secure integrated circuits*, IEEE Trans. on CAD of Integrated Circuits and Systems, 25(7): 1197–1208.

TOKUNAGA C., BLAAUW D. 2009. *Secure AES engine with a local switched-capacitor current equalizer.* In: *Digest of Technical Papers of ISSCC 2009.* IEEE International, pp. 64–65, San Francisco, USA.