

# Cryptographic processors : applications and attacks survey

MOEZ BEN MBARKA  
benmbark@labri.fr

May 25, 2008

## Abstract

The combination of cryptographic tools and tamper-resistant mechanisms appeared first in military applications to secure communication links using tamper-resistant cipher machines. The spread of ATM networks brought the technology into the commercial mainstream. During the last years, the use of embedded cryptographic processors has spread from low-cost cryptoprocessors, such as smart cards used for holding decryption keys, to more modern applications, such as electronic payment schemes, Digital Right Management and Trusted Computing Initiative (TCI). This survey, will summarize the main applications of the cryptographic processors and will insist on their use into PKI based systems. It will also address the main class of attacks which target the cryptographic processors.

**Keywords** : Cryptoprocessors, HSM, PKI, RSA, security API.

## 1 Introduction

A typical crypto-processor is a physically tamper-resistant embedded processor which can perform a predefined set of cryptographic operations using keys that are protected within the device. The fact that security functions are performed in hardware instead of software means that key materials can be better protected - both from physical attacks and network attacks - than if they were running inside a general-purpose server. Another advantage of hardware is that it can be optimized to perform security functions much faster and more effectively than its software counterpart. Such a device has active hardware protection [32] which, on detecting tampering attacks, will destroy the protected cryptographic materials.

During the last years, the use of embedded cryptographic processors [2] has spread from low-cost cryptoprocessors, such as smart cards used for holding decryption keys, to more modern applications such as electronic payment [13] schemes, Digital Right Management and Trusted Computing initiative [1].

Electronic payment systems use hardware modules to securely store the payment card authentication keys and to secure communications between banks, merchants and clients.

The Trusted Computing Initiative (TCI) aims to embed a special cryptoprocessor module in current computing platforms like desktop PCs and PDAs. The key idea is that a TC machine\* can be trusted to certify both a program and the platform on which it is executing. The main application of TCI is Digital Rights Management (DRM) [20]. Most of actual DRM systems are software based and can eventually be hacked. The use of an embedded cryptoprocessor can help to give vendors more ensurance that the protected content is executed by a trusted machine using a trusted program, and that the content remains under its control.

---

\*TC machine : is a machine with an embedded processor.

## 2 Cryptoprocessors attacks

Before deciding whether to use a low cost cryptoprocessor, a middle market component or a powerful tamper-resistant module, it is important to understand the basics of the technology of software and hardware attacks [2] [3]. The main class of security attacks are addressed in the next sections.

### 2.1 Invasive attacks

This class of attacks involves direct access to the internal components of the device [25]. It includes manual micro-probing, laser cutting, focused ion-beam manipulation and glitch attacks. The first step of any invasive attack is to make a hole in the passivation layer which leads to destroy the processor packaging. Most of modern semiconductors call for sophisticated and expensive probing technologies which in most cases exclude any potential invasive attack.

### 2.2 Local non-invasive attacks

This class of attacks involves close observations to the device operations. It includes power analysis [23], timing attacks [24] and eavesdropping [25].

For example, an RSA implementation involves many modular multiplications and the time taken to generate an RSA signature depends strongly on the input values. Thus, suitable timing analysis can compromise (even partially) the signing private key. Similar attacks can be conducted using differential power analysis [23] to guess partially the data being processed by analyzing the current drawn by the processor.

Defenses to this class of attacks include noise generators and randomization. An example, is to randomly include no operation instructions (NOP) in the device instruction stream.

Current cryptoprocessor design approach is design-time validation. The chip maker uses tools to simulate possible non-invasive attacks and then locate most of the devices weakness to be able to perform any needed improvements before the chip is fabricated.

### 2.3 Remote attacks

This class of attacks involves manipulation of the device normal output/input interfaces [2] [25]. While, to conduct invasive or local non-invasive attacks, the attacker needs to have physical access to the device, remote attacks can be conducted remotely. The attacker will only need access to the device input/output traffic.

This class includes cryptanalysis, protocol analysis and API analysis. The former two attacks are not specific to cryptographic processors. They involve exploitation of design flaws in cryptographic primitives (such encryption or hash algorithms) and protocols using these primitives. There is a large literature [9] [29] [28] on these types of attacks.

The API analysis attacks [7] [4] [6] are specific to cryptographic processors and target the device security API [8]. This API sits on the boundary between the trusted (the device internal operations) and untrusted (external servers and users) environments. The security API is composed of a cryptographic API (for example PKCS<sup>†</sup> #11) and a security policy. The policy defines which cryptographic functionalities an external user can call and how the protected cryptographic materials can be accessed. The idea of an API attack was born as an unexpected sequence of transactions which would trick a security module into revealing a secret in spite of the device security policy [2]. An example of API attack targetting the Visa Security Module was first described by Anderson [7]. Designing a secure and robust API is a fundamental challenge, which has until recently been overlooked by both formal methods [27] [5] and software engineering research.

---

<sup>†</sup>PKCS : Public Key Cryptographic Standard.

Another approach is to look for evaluations by third parties. The next section will survey the two main certification schemes : FIPS <sup>‡</sup> 140 [32] (defined by the US National Institute of Standards and Technology) and Common Criteria [11] (defined by many member countries).

## 2.4 Evaluation and Certification

There are two main schemes under which cryptoprocessors can be validated and certified : FIPS 140 and Common Criteria.

The Common Criteria (CC) and FIPS 140 are different in the abstractness and the focus of tests. FIPS 140 testing targets a defined cryptographic module and provides a suite of conformance tests to four security levels. The lowest level (Level 1), imposes very little security requirements. The evaluated product should have protections against most of egregious kinds of attacks. The highest level (Level 4) makes the required security properties more stringent. Mainly, it adds more requirements for physical tamper resistance and robustness against environmental attacks.

CC is an evaluation against a protection profile (PP) or security target (ST). Typically, a PP covers a broad range of products and formalizes the security properties that the device is supposed to fulfill.

## 3 Public key cryptoprocessors

Modern business depends on trust which relies on PKI (Public Key Infrastructure) [19]. A PKI based system can provide the main key elements of trust in any commercial transaction : authenticity, confidentiality, integrity and non-repudiation. The trust of the PKI depends on the technologies used to manage the security of the infrastructure. The main role of a PKI is to generate, validate and store certificates. A transaction between business parties can be trusted if a chain of trust can be established from each party certificate and a trusted certificate authority (CA). If the CA is compromised, then there is no trust. Thus, the main role of a PKI security module is to ensure the integrity and the confidentiality of the CA private key used to sign the issued certificates.

A security module can be a Software Security Module (SSM) or a Hardware Security Module (HSM). The major difference, is that a SSM is running on a general purpose machine, while a HSM is dedicated computer designed to have a security role and can be adapted to speed up cryptographic functions [33]. For example, one of the processes at the heart of certificate creation is the generation of random numbers. HSM can provide dedicated hardware specifically designed to generate number with greater randomness than its software counterparts.

However, since most public key cryptography algorithms need a large computing power, low cost cryptoprocessors based on cheap micro-controllers are not appropriate for PKI systems. Modular exponentiation, presents the computational bottleneck for most public-key algorithms like RSA [22] and can not be performed in a reasonable time by low cost hardware modules. The modern generation of HSMs has accelerator cards for public key cryptography, eventually including dedicated hardware acceleration for modular arithmetic.

A lot of research work is done to define models and protocols to outsource a large part of the cryptographic computation to external servers [30] [21] [14] [12] [17] [10], thus, allowing performing public key cryptography into computationally limited devices (typically a smart card). Some of these protocols are addressed in the next section.

---

<sup>‡</sup>FISP : Federal Information Processing Standard

### 3.1 Secure cryptographic computation outsourcing

The key idea of outsourcing is to delegate some parts of the computation from a trusted device to external helpers which are computationally more powerful. External, means the helpers are located outside the trusted environment and thus can not be trusted. Depending on the nature of the outsourced computation, many security assumptions have to be made on the helpers. The main assumption is that the helpers can potentially be under the control of adversary parties (eventually the same party). For example, if an enemy helper can guess during a signature generation the used exponent, then he can determine the signer private key if he can later intercept the signature. Therefore, the first property of any outsourcing scheme must be privacy. Although most of the computation is carried out by external devices, the scheme must hide as much information as possible about the actual computation from the helpers.

The second property is robustness: the trusted device must be able to detect any failure during the outsourced computation. Namely, the trusted device must detect if one or more helpers try to corrupt the result of the delegated computation. There are two types of malicious failures: intelligent and unintelligent failures. An intelligent failure occurs when the decision to deviate from the expected behavior is based on the input of the computation. For example, a malicious helper may return an incorrect result when he can guess from the input that a competitive vendor key is used. By contrast, unintelligent failures may occur independently of the computation input.

Many outsourcing schemes and protocols are already defined. Most of them aim to use untrusted resources to help a computationally limited device to perform RSA exponentiation without revealing the secret key [10] [26] [18]. This may be used to make reasonable the generation of RSA signatures into smart cards.

## 4 Conclusion

We have surveyed main cryptographic processors applications. Areas of low-cost and mid-cost devices are the most rapidly developing ones, with the “Trusted Computing” likely to bring them into many mass-market platform with new, but rapidly developing, business models. The main types of attacks that a tamper-resistant cryptographic processor have to expect range from invasive attacks which usually lead to destroy the device to non-invasive attacks. Non-invasive attacks can be either local (the attacker has physical access to the device) or remote (the attacker needs only access to the communication layer).

We have also introduced the use of tamper-resistant cryptographic processors into PKI systems. Although, low-cost devices are often computationally limited and thus are not suitable to be used for PKI algorithms, many schemes and protocols have been defined to outsource the main part of the computation to more powerful but less trusted helpers.

On the other hand, most of modern chips expected to be used into a PKI trust platform have dedicated cryptographic processor [16] [15] [31] and thus provide both powerful and secure environment for private key operations. In this case, outsourcing the PKI algorithm computation is not needed. However, although the processor is very fast when executing any pre-defined cryptographic function, it is not so fast when dealing with other computationally expensive processing. Therefore, outsourcing may be still needed for some applications where powerful no-cryptographic functions are expected to be executed in the same trust environment and with the same security assumptions as cryptographic functions. Currently, there is no important research results in this direction. It would be interesting to see if outsourcing methods surveyed at 3.1 can be extended for such specific use-cases.

## References

- [1] Trusted computing group. <http://www.trustedcomputinggroup.org/home>.
- [2] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov. Cryptographic processors-a survey. *Proceedings of the IEEE*, 94:357–369, 2006.
- [3] R. J. Anderson and M. G. Kuhn. Low cost attacks on tamper resistant devices. In *Proceedings of the 5th International Workshop on Security Protocols*, pages 125–136. Springer-Verlag, 1998.
- [4] R. J. Anderson and M. G. Kuhn. Low cost attacks on tamper resistant devices. In *Proceedings of the 5th International Workshop on Security Protocols*, pages 125–136. Springer-Verlag, 1998.
- [5] M. Bond. A chosen key difference attack on control vectors.
- [6] M. Bond. Attacks on cryptoprocessor transaction sets. In *CHES '01: Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, pages 220–234. Springer-Verlag, 2001.
- [7] M. Bond and R. Anderson. Api-level attacks on embedded systems. *Computer*, 34(10):67–75, 2001.
- [8] M. Bond and J. Clulow. *Understanding Security APIs*. PhD thesis, Computer Laboratory, University of Cambridge, UK, 2006.
- [9] E. Brickel and A. Odlyzko. Cryptanalysis: A survey of recent results.
- [10] J. Burns and C. J. Mitchell. Parameter selection for server-aided rsa computation schemes. *IEEE Trans. Comput.*, 43(2):163–174, 1994.
- [11] Common Criteria. *Common Criteria Evaluation Scheme*.
- [12] M. Dijk, D. Clarke, B. Gassend, G. E. Suh, and S. Devadas. Speeding up exponentiation using an untrusted computational resource. *Des. Codes Cryptography*, 39(2):253–273, 2006.
- [13] EMV. *Integrated Circuit Card Specifications for Payment Systems*, 2004.
- [14] A.-M. Ernvall and K. Nyberg. On server-aided computation for RSA protocols with private key splitting. In S. Knapskog, editor, *Proceedings of Nordsec 2003*. Department of Telematics, NTNU, 2003.
- [15] A. Fuchsberger, D. Gollmann, P. Lothian, K. G. Paterson, and A. Sidiropoulos. Public-key cryptography on smart cards. In *Proceedings of the International Conference on Cryptography: Policy and Algorithms*, pages 250–269. Springer-Verlag, 1995.
- [16] H. Handschuh and P. Paillier. Smart card crypto-coprocessors for public-key cryptography. In *CARDIS '98: Proceedings of the The International Conference on Smart Card Research and Applications*, pages 372–379. Springer-Verlag, 2000.
- [17] S. Hohenberger and A. Lysyanskaya. *How to securely outsource cryptographic computations*, volume 3378/2005 of *Lecture Notes in Computer Science*, pages 264–282. Springer Berlin / Heidelberg, 2005.
- [18] S.-M. Hong, J.-B. Shin, H. Lee-Kwang, and H. Yoon. A new approach to server-aided secret computation. In *Information Security and Cryptology*, pages 33–45, 1998.

- [19] R. Housley, W. Ford, W. Polk, and D. Solo. Internet x.509 public key infrastructure : Certificate and crl profile. Technical Report RFC 2459, IETF, 1999.
- [20] IBM. *IBM Enhanced Media Management System*. <http://www-306.ibm.com/software/data/emms/>.
- [21] M. Jakobsson and S. Wetzel. Secure server-aided signature generation. In *PKC '01: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography*, pages 383–401. Springer-Verlag, 2001.
- [22] N. Kobitz and A. J. Menezes. A survey of public-key cryptosystems. *SIAM Rev.*, 46(4), 2004.
- [23] P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. *Lecture Notes in Computer Science*, 1666:388–397, 1999.
- [24] P. C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, pages 104–113. Springer-Verlag, 1996.
- [25] O. Kommerling and M. G. Kuhn. Design principles for tamper-resistant smartcard processors. In *WOST'99: Proceedings of the USENIX Workshop on Smartcard Technology on USENIX Workshop on Smartcard Technology*, pages 2–2. USENIX Association, 1999.
- [26] C. H. Lim and P. J. Lee. Security and performance of server-aided rsa computation protocols. In *CRYPTO '95: Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, pages 70–83. Springer-Verlag, 1995.
- [27] D. Longley and S. Rigby. An automatic search for security flaws in key management schemes. *Comput. Secur.*, 11(1):75–89, 1992.
- [28] M. Matsui. The first experimental cryptanalysis of the data encryption standard. In *CRYPTO '94: Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, pages 1–11. Springer-Verlag, 1994.
- [29] M. Matsui. Linear cryptanalysis method for des cipher. In *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 386–397, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.
- [30] T. Matsumoto, K. Kato, and H. Imai. Speeding up secret computations with insecure auxiliary devices. In *CRYPTO '88: Proceedings on Advances in cryptology*, pages 497–506. Springer-Verlag, 1990.
- [31] D. Naccache and D. M’Ra. Arithmetic co-processors for public key cryptography: The state of the art, 1996.
- [32] Nationale Institute of Standards and Technology. *Security Requirements for cryptographic modules. Fips 140-2*, 2001.
- [33] S. W. Smith, E. R. Palmer, and S. Weingart. Using a high-performance, programmable secure coprocessor. volume 1465, pages 73–89. Second International Conference, FC'98, 1998.