Single-chip implementation of a cryptosystem for financial applications

Nikolaus Lange
SICAN Braunschweig GmbH
Richard-Wagner-Str.1, D-38106 Braunschweig
nlange@sican-bs.de

Abstract

This article presents a hardware architecture called "GCD - General Crypto Device", realized as a single chip for system solutions in the EFT area. Special emphasis is put to this application area as the GCD supports all functions and security mechanisms commonly required by financial security systems (DES, RSA, key generation schemes).

The GCD mainly targets at the electronic financial area, like electronic funds transfer, Electronic Cash, Electronic Banking and Chipcard applications. Other typical applications of the GCD are network security (e.g. on ATM or ISDN), access control systems (ACS), or upcoming consumer cryptosystems like pay TV or pay radio.

Keyfeature of the devices new concept is an optimized processor containing instructions especially required by crpto functions. Additionally the single chip realisation reduces the required space and the accessibility of sensitive signals. Probing sensitive internal data like a generated session key or global master keys requires a very high level of technical skills (like microprobes) not to be expected to become commonly available in the near future

The ASIC is based on a 32-Bit RISC special processor with embedded high speed crypto functions. The major new achivement lies in the processor architecture, which includes a pipeline stage, designed for efficient long number arithmetic, like it is needed in the RSA cryptosystem [1]. The resulting overall performance of the device is significantly higher than that of existing realizations due to the used design concept, the performance is further enhanced due to the direct linking of the embedded components. The physical security combined with a high cryptographic flexibility at reasonable costs allows the usage of new cryptographic algorithms even in consumer market applications.

The paper is organized as follows: Section 1 gives an introduction. Section 2 pointes out the motiviation to design a device dedicated to the efficient realization of cryptosystems. Section 3 described the requirements and the architecture of typical cryptosystems used in the financial cryptography area. Section 4 presents the architecure of the General Crypto Device and its contents. While section 5 shows two examples, section 6 concludes the paper.

1 Introduction

Electronic information and communication techniques play an increasing role in business and private life. This development implies an increasing dependency of the society on the faultless function of communication systems. Only trust and confidence in their indisputable and proper performance will allow the acceptance of new systems like electronic banking or electronic money.

On the other hand one faces a growth of potential attacks to IT-systems due to the raising amount of data collected in distributed and open systems, which can no longer be secured only by organisational steps. Cryptographic methods can be applied to provide the appropriate security services for the named problems. To point out the usage of cryptography in financial applications, the example of an ATM (automatic teller machine) is roughly analysed:

During the CVM (cardholder verification method), the communication between user and ATM, particularly the exchange of the PIN (Personal Identification Number) is encrypted. This procedure shall avoid the eavesdropping of the PIN. Next, the communication of the ATM with the central banking unit (inquiry, amount, transaction) via public network is secured in terms of authenticy, privacy and integrity using cryptographic protocols and encryption algorithms. In latest electronic cash systems the communication between chipcard and terminal is also executed according to a cryptographic protocol, CAM (card authentication method). It can be seen from the example, that several cryptologic functions are executed in a cryptosystem for different partial tasks. These functions may base on the same algorithms, differing in the parametrisation.

2 Motivation

Many cryptologic algorithms have been reported to implement and optimise cryptosystems in the last years. Although there exists a variety of hardware platforms, in many cases time-critical parts of the algorithms are still performed in software. It is obvious, that there is a lack of powerful, universal hardware platforms dedicated to efficient implementation of cryptosystems, espescially in the dramatically increasing market of financial applications.

From cryptologists point of view it is often desirable to realise a cryptosystem only based on public-key schemes. Most public-key schemes are based on the one-way functions performed in terms of modular exponentiation in finite fields $GF(2^n)$, with typically n = 512..2048. For example the RSA cryptosystem performs encryption and decryption by means of modular exponentiation, whereas the DSS (Digital Signature Scheme [2]) used modular exponentiation for the generation of the signature. The computational complexity of modular exponentiation leads in most realisations to a compromise in form of a hybrid

solutions, like found in [3]. In a hybrid cryptosystem, session key generation and key exchange protocols are performed according to a public-key scheme, whereas the high speed data encryption is executed with a session key based on a block cipher algorithm. Some private key cryptosystems, like DES [4], are based on confusion algorithms, which require specific hardware. Other block cipher algorithms, as the International Data Encryption Algorithm (IDEA [5]), use mixed different group operations. Typically, their encryption rate is still about factor 10^3 higher than that of public key cryptosystems.

In most implementations the above described functions of exponentiation and confusion are realised in different ICs, resulting from the high computational complexity of the algorithms. Furthermore, there lies obviously a strong risk in attackable interfaces between theses components. Consequently, a main target of a universal cryptosystem is to increase the degree of integration, raising security. However, integrating all mentioned functions in one device gives no direct yield in performance.

Another aim is to increase the flexibility of a cryptosystem, allowing the implementation of proprietary or future algorithms. This will allow to upgrade a cryptosystem when the current state no longer provides sufficient security. This might be the case, when the algorithm or the keys are examined , or when key lengths have to be increased.

The idea of the GCD is the realize a high performance special processor, which is extended by embedded crypto functions, resulting in a processor core, which is optimized for cryptographic algorithms used in the financial data security area.

3 Cryptosystem architecture

A cryptosystem shall provide the required security by using appropriate methods, i.e. protocols, functions and algorithms. Thus a cryptosystem must provide sufficient resources to execute the required methods. In EFT applications, this needs the following features:

- flexible programmability to realise different protocols and front-end I/O
- hardware support for specific crypto functions which require instructions, not efficiently executed by multi-purpose hardware (e.g. crypto coprocessor or specific hardware to support feistel ciphers like DES or modular exponationation for RSA)
- true random number generation, for generating a secret key
- random access memory and non-volatile memory to store programs, data, intermediate results, masterkeys, sessionkeys
- communication interface

To illustrate the main features of a cryptosystem, an example scenario from the electronic funds transfer area is given:

1. two communication partners: host (central unit), terminal (POS-Terminal = Electronic Cash)
2. partial key generation at each partner (typically random number of 512 bit)
3. performance of key exchange protocol (typical micro controller task)
4. session key generation (e.g. RSA: exponentiation of 512 bit integers)
5. block encrypted data transmission (e.g. DES: 64 bit block cipher)

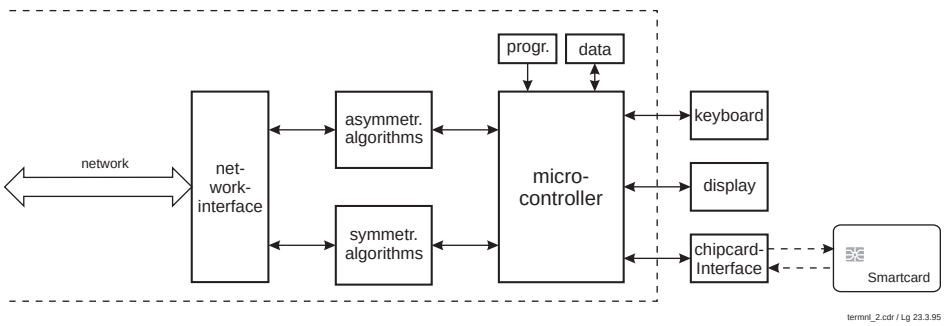

Fig. 1. Electronic Funds Transfer Example

The following components are found in current cryptosystem implementations (Fig. 1):

- microcontroller (program control) incl. program and data memory for multi-purpose tasks
- front-end / user-interface (periphery)
- back-end / communication interface (network)
- a unit to support asymmetric cryptoalgorithms (long number arithmetic)
- a unit to support symmetric cryptoalgorithms (blockcipher)

There are different theoretical possiblities to realize such a system. The classical solutionused in Point-of-sales terminals is a system based on three chips. One chip performing arithmetic functions is used for modular multiplication resp. exponentiation. This is either a digital signal processor (DSP) or a specific RSA chip [6]. Furthermore a specific DES chip [7] for the block cipher algorithm is employed, since permutations cannot be realized efficiently in software neither on a multi-purpose hardware nor on a DSP. Finally a standard microcontroller is necessary to realize an application interface and to link all functions together without losing overall performance. It is obvious, that an increase in performance of one module not automatically increases the overall performance if the modules are designed individually without respect of their inter-operation and communication.

To meet the requirements of such a system, an integration of these modules would be the first approach. Unfortunally a pure integration of such an architecture does not solve the structural limitations. The whole system is limited through the microcontroller as the program control unit, thus an increase of the encryption or exponentation rate does not increase the system performance. Furthermore, the realizable algorithms are limited due to specific, unflexible hardware.

To overcome these restrictions an optimized implementation of all named functions on one chip is worked out. This maximizes the overall performance as well as it offers a higher grade of flexibility, providing the possibility to implement a wide range of cryptographic solutions. The idea of melting functions together avoids as a side effect bottlenecks in communication.

The approach presented in this article puts an emphasis on an efficient system architecture. From the experience, that a specific hardware for fast multiplications is necessary, a DSP architecture is chosen as the heart of the device. In opposite to standard DSPs, which are dedicated for signal processing, this one is optimized for long number arithmetic. This means, that the execution unit of the processor is extended with specific hardware to support easy and fast construction of those modular arithmetic based algorithms, which are commonly used in cryptosystems. So the instruction set allows the simple construction of an algorithm computing $M = A^E \bmod P$ in software. In fact, this is done by applying the square-and-multiply algorithm [8] to long number multiplication operations. Furthermore, the standard instruction set of the processor is extended for an optimized crypto algorithm implementation., e.g. with instructions for linear feedback shift registers.

Permutation functions cannot be efficiently be realized with standard processors. This is the reason for the further extention of the DSP with substitution specific hardware, allowing a high-speed implementation of DES-like algorithms on the processor core.

In general, the concept aims at a high performance processor, which is extended with those functions, which give an immediate yield in performance, leaving all other functions as flexible as possible. Within the ASIC, a second processor is integrated, which can purely be used for application purposes. This proposal leads to a hardware complexity, which is still integrateble on a single die in commonly available semiconductor technologies.

4 Architecture of the General Crypto Device

In this section the architecture of the ASIC implementation GCD is presented. The GCD (General Crypto Device) fulfils many of the requirements of a universal cryptosystem. The cryptospecific parts are state-of-the-art design, especially a 32-Bit special processor with embedded crypto functions.

With this ASIC for the first time all relevant functions of a cryptosystem are combined in one piece of silicon, thus increasing the security significantly. Special internal security features, optimised for banking applications, provide additional security against attack scenarios. For example the chip contains 128 bit memory area, dedicated to store the

master keys. This area is asynchronously erased (within 10 ns) after triggering an external sensor (e.g. light, movement or temperature).

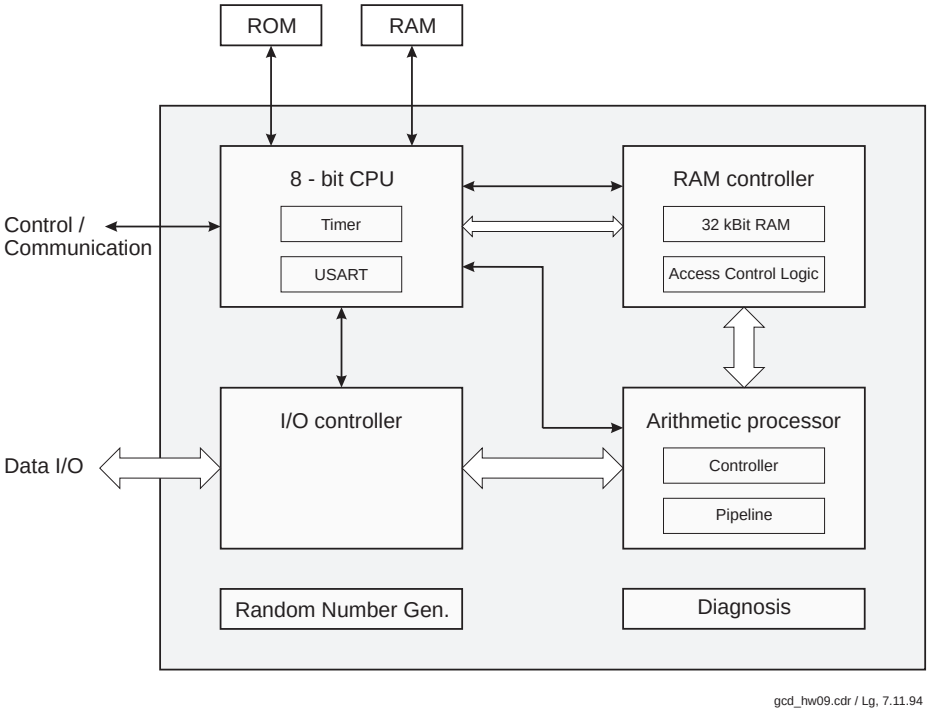

Fig. 2.: GCD architecture

Figure 2 shows the GCD architecture. It contains the following modules:

- a powerful 32 bit special processor for crypto algorithms, called Arithmetic Processor (AP)
- an 8-bit standard microcontroller for the application (8051 compatible, extended by additional ports, additional interrupts, wait states and remote program download functionality)
- 32 kbit of safe internal data, program and key memory, accessible by both processors, including special security functions (emergency erase, hypersecure memory, soft-locks)
- True random number generation by on-chip noise-source
- flexible IO controller to interface different communication networks and peripheral interfaces

These components allows a secured remote-download using the microcontroller and the security mechanisms. The device also allows secure procedures for key storage, key generation and key exchange. The algorithms executed by the Arithmetic Processor are scalable, thus allowing modular arithmetic for long numbers up to 4096 bit. The Arithme-

tic Processsor offers hardware support for most classes of cryptoalgorithms, including high flexibility by wide-scale programmability. In general, the GCD offers high speed and high degree of integration compared to existing implementations.

The GCD may be characterised as the integration of symmetric and asymmetric crypto modules with controller, random number generation, memory and security mechanisms and has in this shape not yet be realised before.

Fig. 3 shows the structure of the General Crypto Device. An 8-bit microcontroller is

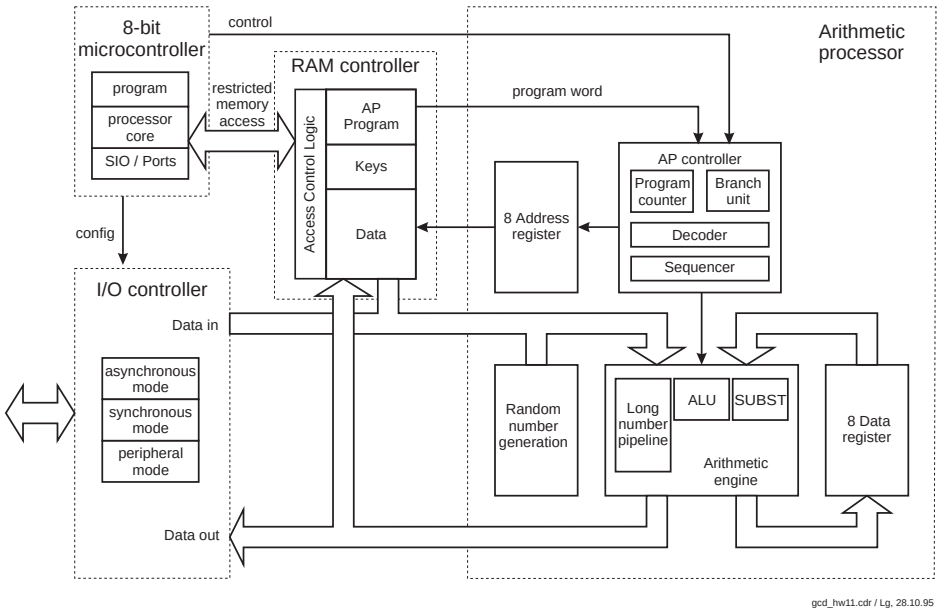

Fig. 3: Structure of the General Crypto Device

integrated in order to build the user interface for e.g. a keypad or display control or a smartcard interface. A flexible IO interface allows to exchange data at high rates (up to 160 Mbit/s) with a host according to many industrial standards.

The heart of the device is an application-specific 32-bit RISC core, called the Arithmetic Processor. It has the characteristic of a standard DSP, which incorporates e.g. the parallel operation on register, memory and pointers. The processor realizes the execution of up to four memory accesses per cycle at a clock frequency of 25 MHz. The RISC core guarantees through the pipelined architecture high performance, especially for modular exponentiation functions and arithmetic functions.

One main difference compared to the crypto coprocessor concept is an integrated multi-processor system, allowing a host processor directly accessing the embedded crypto macro functions provided by the special processor. Both processors share the same memory, thus

minimizing all communication overhead. This gives a software designer a powerful standard hardware platform extended with a set of crypto-specific special functions. The structure allows to implement most cryptographic algorithms with a high efficiency.

The ressources for the 32-bit Arithmetic Processor (AP) are 4 kByte of chip-internal memory, 8 data registers, 8 address registers, 8 special registers and 4 masterkey registers. Masterkey registers as well as the 4 kByte of RAM are battery backuped, allowing a download of an algorithm or a device key in a secure environment, which stays resident in the field. To illustrate its capability, an example of the instruction set is given: The long number operation MLSAC performs Lng(Ax) := Lng(Ax) << 32 + Lng(Ay) * (Az), where Lng(Ax) and Lng(Ay) are pointers to long numbers of length m*32 (m = 2 .. 64), (Az) is a 32 bit slice of Lng(Az).

The security concept is based on the assumption, that the integrated microcontroller is insecure, since its programm resided externally. Thus the AP can selectively suppress the 8-bit microcontrollers access to internal memory. This is done by an AP controlled access control logic, individually representing the read/write permissions in slices of 32 words. In this way areas dedicated to store keys or intermediate results remain inaccessible for the microcontroller, whereas open memory areas can be used for data or program exchange between both processors.

5 Application examples

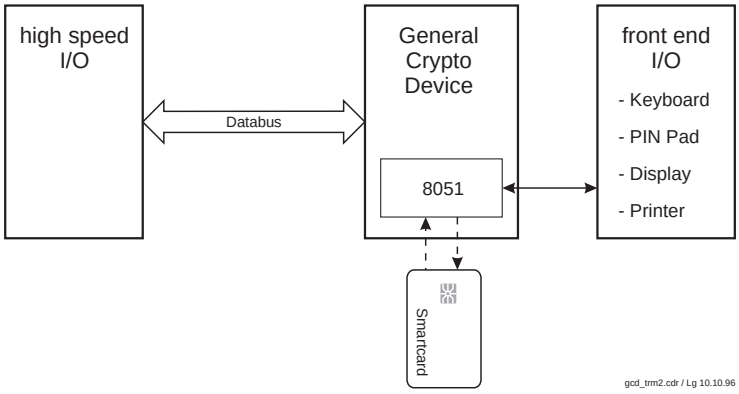

Fig. 4. Application example Electronic Cash terminal

Figure 4 shows an application example of the GCD from the Electronic Cash area, an POS (Point of Sale) or ATM terminal: The user interface consists of a pinpad and the access via a smart card. The communication with the host via the back-end interface is performed authorised and secured according to the chosen protocols and algorithms.

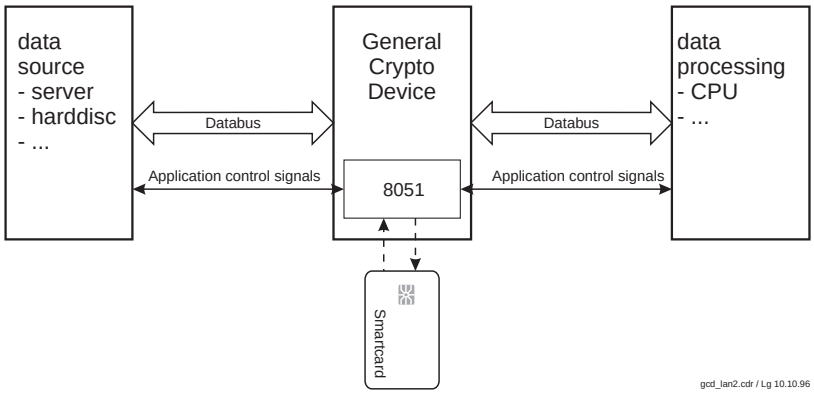

Fig. 5. Application example network security

Figure 5 shows an network or PC security application: The GCD is integrated in the communication bus, thus the transmission and/or storage of data is encrypted, whereas the (authorised) user accesses transparent data.

6 Performance

The presented ASIC is produced in 0.6 mm standardcell technology operating at 3.3 V. Its complexity is about 400,000 transistors. The ASIC operates at a moderate clock rate of 25 MHz.

To give some performance ideas, the DES algorithm in the cipher modes ECB, CBC, CBF, OFB, MAC can be executed in 100 MBit/s. The International Data Encryption Algorithm can be implemented with a data rate of 16 MBit/s. A typical RSA encryption of 512 bit can be performed in the standard method with an encryption rate of 20 kBit/s using the straight-forward approach. The modular exponentiation of 1024 bit is performed in less than 150 ms, 2048 bit parameters will lead to a execution time below 1 second. This can be improved by implementing CRT (Chinese-Remainer Theorem) or Montgomery method [9].

7 Conclusion

The major benefit of the presented structure is the availability of all cryptosystem functions on one silicon die, allowing the implementation of RSA, DES and additional security features with reasonable performance at a high level of security. Its usage is not only restricted to standard algorithms, it is also an dedicated platform for new or future algorithms like elliptic curves [10] or for proprietary crypto systems.

The ASIC implementation GCD allows the introduction of new cryptographic algorithms in cost-sensitive areas, allowing besides DES also the usage of other high speed block ciphers. The device is also appropriate for development tasks, since it offers a multi-purpose crypto hardware platform, supporting the evaluation of new algorithms. The ASIC is nevertheless limited to algorithms without high memory requirements due to the limited on-chip memory of 4 Kbytes.

References

[1] R.L.Rivest, A. Shamir, L. Adleman, "A method for obtaining digitial signatures and public-key cryptosystems", Communication of the ACM, 21 (2), Feb. 1978, pp. 120-128

[2] A proposed Federal information processing standard for digital signature standard (DSS), Federal Register 56 (169), Aug. 1991, pp. 42980-42982

[3] A. Glade, H. Reimer, B. Struif, "Digitale Signatur & Sicherheitssensitive Anwendungen", Vieweg TeleTrusT Publications, 1995, pp. 66-88

[4] National Bureau of Standards, "Data Encryption Standard", FIPS pub. 46, 1977

[5] X. Lai, J. Massey, "A proposal for a new block encryption standard", Proc. Eurocrypt '90, 1990

[6] E. F. Brickel, "A Survey of Hardware Implementation of RSA", Advances in Cryptology - CRYPTO '89, 1989, pp. 368-370

[7] B. Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C", Wiley, 1994

[8] K.E. Knuth, "The art of computer programming", Addison-Wesley, 1981

[9] P. L. Montgomery, "Modular Multiplication Without Trial Division", Mathematics of Computation, Vol. 44, 170, April 1985, pp. 519-521

[10] N. Koblitz, "Elliptic curve Cryptosystems ", Methematics of Computation, Vol. 48, 177, 1987, pp. 203-209