Implementation of cryptographic standards and cryptanalysis using FPGA's: extended abstract

Gaël Rouvroy, François-Xavier Standaert, UCL Crypto Group, Laboratoire de Microélectronique, Département d'Electricité, Faculté des Sciences Appliquées, Université Catholique de Louvain

Abstract

This paper briefly summarizes the results of our final paper, rewarded for the "Research and Development Award" of the SRBE-KBVE in 2002. The main purpose of this work was to underline the relevance of hardware implementations in cryptography as well as the risk generated by cryptanalytic attacks using this computation power. We focus our attention on the description of the hardware used and its forthcoming perspectives, the purposes of cipher functions and the results of two cryptanalytic attacks attempted against the cipher DES (Data Encryption Standard).

Résumé

Cet article résume brièvement les résultats de notre travail de fin d'études, récompensé par le «Prix Recherche et Développement » de la SRBE en 2002. L'objectif principal de ce travail est de démontrer l'intérêt des implémentations matérielles en cryptographie ainsi que les risques générés par des attaques cryptanalytiques utilisant cette puissance de calcul. Nous focalisons ici notre attention sur la description du matériel utilisé et ses perspectives d'avenir, les objectifs des fonctions de chiffrement et les résultats de deux attaques réussies contre le chiffrement DES (Data Encryption Standard).

Samenvatting

Het artikel vat de resultaten van ons eindwerk samen, dewelke bekroond werd met de prijs „Onderzoek en Ontwikkeling" van de KBVE in 2002. Ons werk beoogde voornamelijk het belang van hardware implementaties in cryptographie aan te tonen, evenals de bestaande gevaren voor cryptanalytische aanvallen die ermee gepaard gaan te beklemtonen We hebben ons in het bijzonder gericht op de beschrijving van de gebruikte hardware en zijn toekomstperspectieven, de doelstellingen van cijfer functies en de resultaten van twee geslaagde cryptanalytische aanvallen op DES (Data Encryption Standard) encryptie.

Introduction

Cryptography is the science of secret writing. Cryptanalysis is the science of breaking codes. Cryptology encompasses both subjects and is a key technology in secure electronic systems. Modern cryptographic techniques have many uses, such as to digitally sign documents, for access control, to implement electronic money, and for copyright protection. Because of these important uses, it is necessary that users be able to estimate the efficiency and security of cryptographic techniques.

Microelectronics has been the enabling technology for the development of hardware and software technology in the recent decades. The continuously increasing level of integration of electronic devices led to very large integrated circuits.

This work focus on the importance of hardware implementations in cryptography in terms of performance and security provided. We implemented two encryption standards and evaluated two cryptanalytic techniques attempted against DES (Data Encryption Standard). The resulting designs are deployed on relatively expensive hardware and significantly improved existing results in terms of effectiveness.

Cryptography

One objective of cryptography is to solve the confidentiality problem that can be explained as follows. Suppose that someone wants to send a message to a receiver, and wants to be sure that no-one else can read the message. However, there is the possibility that someone else opens the letter or hears the electronic communication. In cryptographic terminology, the message is called a plaintext. Encoding the message in such a way that it hides its contents from outsiders is called encryption. The encrypted message is called ciphertext. The process of retrieving plaintext from ciphertext is called decryption. Encryption and decryption usually make use of a key, and the coding method is such that decryption can be performed only by knowing the proper key.

Our work focused on the confidentiality problem and we studied how a class of functions called block ciphers can solve this problem. In a block cipher, a key is shared between two users and it is used to encrypt and decrypt the messages. No block cipher is ideally suited for all applications, even one offering a high level of security. However, two standards have been chosen and actually result of a tradeoff between security and performance: DES and AES (Advanced Encryption Standard). DES and AES are typical iterated block ciphers, which involves the sequential repetition of an internal function called a round function. Parameters include the number of rounds *r*, the block bitsize *n*, and the key bit-size *k*. The basic idea of these ciphers is to build a complex encryption function by composing several simple operations. However, the internal structure of both standards is different: DES is a Feistel cipher whereas AES is a substitution-permutation network. The main difference is that the round function of a Feistel cipher need not to be invertible, only the combinations of the rounds is.

The Data Encryption Standard is an encryption/decryption algorithm developed in the mid-1970s. It was turned into a standard by the US National Institute of Standards and Technology and was also adopted by several other governments worldwide. It was and still is widely used in the financial industry, even though DES is supposed to be replaced by AES for the next decade. DES uses a 56-bit key which makes it susceptible to exhaustive key search with modern computers and special-purpose hardware.

The Advanced Encryption Standard is due to a worldwide competition to develop a new encryption scheme. Researchers from different countries worked on developing advanced encryption techniques. NIST invited the worldwide cryptographic community to attack the encryption formulas in an effort to break the codes. After narrowing the field down to 5 final formulas, attacks were intensified on the finalists. Finally, RIJNDAEL was chosen as the future AES on October 2000. It was the best combination of security, performance, efficiency and flexibility. Rijndael uses a 128, 192 or 256-bit key.

FPGA

All our experiments were carried out on a FPGA (Field Programmable Gate Array). Basically, a FPGA is a programmable device that can be used to perform some computation tasks (like cryptographic encryption) at very high frequencies. It is divided into a certain number of logic blocks containing logic and storage elements. In our application, we used a Xilinx Virtex1000BG560 FPGA board developed by DICE (Microelectronics Laboratory at UCL). The board is composed of a control FPGA (FLEX10K), a Virtex1000 FPGA, processors (PIC and ARM) and several fast access memories. Although all our implementations are FPGA-based, the board was used to configure the FPGA and to recover the results on a PC.

Cryptanalysis

Generally speaking, a block cipher allows to encrypt a *n*-bit plaintext, using a *k*-bit key in order to produce a *n*-bit ciphertext. Cryptanalysis is equivalent to the searching problem of finding the correct secret key *K* in a set of 2^k possible keys and allows two extreme solutions: exhaustive key search and precomputation table. In exhaustive key search, the ciphertext can be deciphered under each key and the result compared with the known plaintext. If they are equal, the key tried is probably correct. Occasional false alarms can be rejected by additional tests. In precomputation table, the cryptanalyst first enciphers some fixed plaintext *P* under all possible keys to produce 2^k ciphertexts. These are sorted and stored in a table with their associated key. When a user chooses a new key, he provides (in a chosen plaintext attack) the cryptanalyst with the encipherment of *P*:

$$C = E_K(P)$$

Where $E_K(P)$ denotes the enciphering operation under key *K*. Because the table is sorted by ciphertexts, the cryptanalyst can find *C* and its associated key in at most $\log_2 N$ operations using a binary search. The 2^k operations required to compute the table are not counted here because they constitute a precomputation task that can be performed beforehand. However, we must ensure that the precomputation and the memory needed are not excessive.

Besides these two brute force attacks, mathematical descriptions of the encryption algorithm are studied in order to find possible lacks that can be used for cryptanalytic purposes. The most common examples are linear and differential cryptanalysis. The objective of linear or differential cryptanalysis is to recover some key-bits in less operations than an exhaustive key search over all possible keys. Linear cryptanalysis takes advantage of possible input-output correlations over a few rounds of an algorithm whereas differential cryptanalysis is possible if there are predictable difference propagation over a few rounds of an algorithm.

Finally, the aim of a time-memory tradeoff is to mount an attack which has a lower processing complexity than exhaustive key search and a lower memory complexity than a precomputation table. In this work, we propose a first hardware implementation of the linear cryptanalysis and a time-memory tradeoff using distinguished points.

Our implementations

DES is a block cipher with 64-bit block size and 56-bit keys. As the understanding of some parts of the algorithm are necessary in the next sections, we give a short description of it:

– the given plaintext *P* is divided in two parts of 32 bits according to an initial permutation *IP*: $IP(P) = L_0R_0$;
– 16 iterations of a round function are computed and sixteen keys, each bit strings of length 48 are derived from the key *K*;
– the inverse permutation IP^{-1} is applied to the bit string $L_{16}R_{16}$, obtaining the ciphertext *C*;

The key point of the algorithm is the round function illustrated by Fig. 1, where *F* is a non-linear function and the symbol ⊕ is the bitwise XOR operation. The non-linear function is divided into permutations of the bits, addition of the round key and non-linear substitutions represented by substitution boxes.

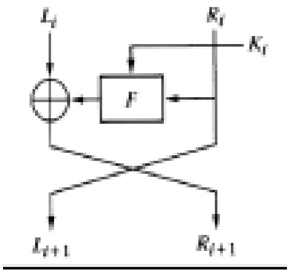

Fig. 1: The DES round

Our hardware implementation of DES is fully pipelined which means that we introduced registers between every round of the algorithm. It allows to encrypt one plaintext in one clock step. Therefore, the encryption rate only depends on the work frequency. We reached a work frequency of 84 MHz. Moreover, we could fit 5 DES blocks on one FPGA. It means a final encryption rate of 27Gbits/sec.

Comparable results were reached with AES, and allowed us to evaluate the performances of both standards. Our main conclusion was that the additional security provided by AES is expensive in terms of hardware resources. Notably the substitution boxes used in Rijndael are large and do not allow very efficient implementations on FPGA's. However, a final encryption rate of 11 Gbits/sec was reached using the same hardware resources as for DES.

Our cryptanalysis

Finally, we present here two attacks attempted against DES. Both of them improved existing results in terms of effectiveness due to our fast hardware implementations. Linear cryptanalysis results from Matsui's work [2] but could not be applied as such and had to be modified to face hardware constraints. We broke a key in about 14 hours on one single FPGA becoming the fastest implementation to our knowledge. In parallel, we evaluated the possibility of a time-memory tradeoff using distinguished points. The original idea from Hellman [3] has never been implemented against practical ciphers. We performed first experimental results and designed a machine that can break a 40-bit DES in about 10 seconds, with a high success rate (72 %), using one PC. An exhaustive key search on the same PC would have taken about 50 days.

Linear cryptanalysis

Linear cryptanalysis is a cryptanalytic technique that takes advantage of eventual input-output correlations over a few rounds of an algorithm. DES presents this interesting property and therefore is susceptible to be broken by a linear cryptanalysis attack. In its basic version, linear cryptanalysis is a known plaintext attack. The purpose of this method is to obtain a linear approximation of the DES.

To find this expression, we looked for boolean relations between inputs and outputs of substitution boxes with a probability as different as possible from 1/2. The approach chosen by Matsui was to investigate the probability that a XOR relation between input bits coincides with a XOR relation between output bits, for every substitution box. After an empiric search of all the possible linear relations, Matsui found the best relation that holds with probability 12/64. Other relations were found with worse probabilities.

The next step was to extend these relations to one round and finally to combine them in order to get a linear approximation of DES, involving plaintext bits, key bits and ciphertext bits, with the probability that this approximation holds. According to Matsui, this equation holds with probability $1/2 + 1.19 \times 2^{-21}$.

In order to recover the 12 key bits involved in this equation, we have to compute it for all the 4096 possible keys and a large number of plaintexts. Knowing that only one of these 4096 keys is correct, there is one of the 4096 equations that will hold more significantly, corresponding to the correct key.

The success rate of the algorithm depends on the number of plaintext tried and the probability that the linear approximation holds. Approximately, if the probability that the linear approximation hold is $1/2 + 1.19 \times 2^{-21}$, we will need $(2^{21})^2 = 2^{42}$ plaintexts to recover the key bits.

The hardware requirements of this algorithm depend on the number of times the linear approximation has to be computed, corresponding to a number of counters. It was practically impossible to fit 4096 counters on our FPGA and therefore, we had to modify the method to face hardware constraints. The alternative solution allowed us to recover 6 key bits but we implemented it on a single FPGA.

Table 1: Success rate

N	SR
38	4 %
39	8 %
40	14 %
41	33 %
42	77 %

The resulting experimental results are summarized in Table 1. We performed tests with 50 different keys and express the success rate in terms of a number of plaintexts tried (N is the number of plaintexts tried in a log 2 scale and SR the success rate).

Once these 6 key bits of information have been obtained, the question becomes: "How can they be exploited to obtain the complete key?". Since the 6 bits recovered belong to the 12 involved in the linear approximation, we can now implement it with the remaining 6 key bits. Applying the same treatment to the dual equation (where we just invert the way we cover the rounds) would provide another 12 bits. Therefore, we could find 24 bits in about 12 hours. The exhaustive search of the remaining 32 bits would take about 20 seconds.

To conclude, we implemented a first FPGA implementation of linear cryptanalysis. Due to hardware constraints, the attack had to be adapted to make it less memory-consuming. The resulting design is deployed on reasonably expensive hardware (3500 $) and is capable to break a full DES key in 12-15 hours, including a final exhaustive search.

Time-memory tradeoff

In 1980, Hellman introduced the concept of cryptanalytic time-memory tradeoff, which allows the cryptanalysis of any N key symmetric cryptosystem in $O(N^{2/3})$ operations with $O(N^{2/3})$ storage, provided a precomputation of $O(N)$ is performed beforehand. This idea never led to realistic implementations. This section refers to Rivest [4] who introduced the idea of a time-memory

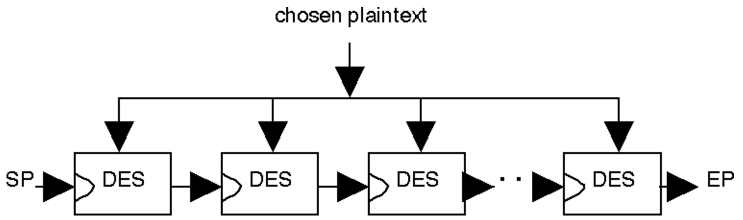

Fig. 2: Precomputation

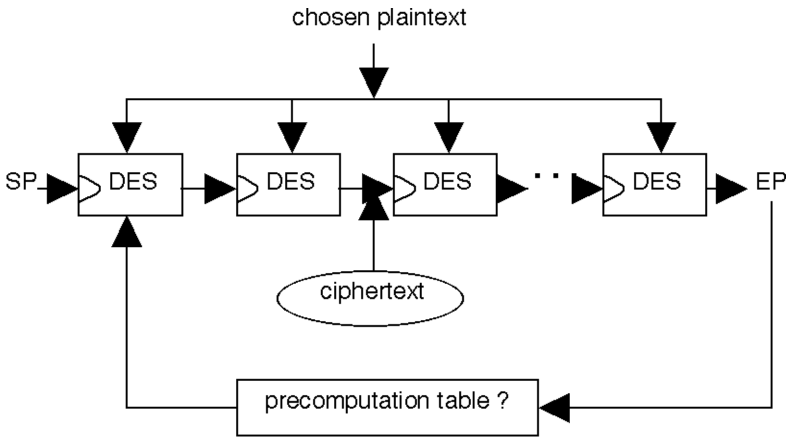

Fig. 3: Online attack

tradeoff using distinguished points. We present the first experimental results of a cryptanalytic time-memory tradeoff using distinguished points in the context of a chosen-plaintext attack against a 40-bit DES.

The time-memory tradeoff method for breaking ciphers is composed of a precomputation task and an online attack. We briefly introduce these steps with two intuitive schemes:

1. A chain is formed by a number *l* of encryptions using a chosen plaintext and l different keys. A defined property holds for the first and last keys and we call them distinguished points (DP). During the precomputation, we compute a number of chains and store start points (SP), end points (EP) and the corresponding length of chains in a table.

2. Let the chosen plaintext be encrypted with a secret key. During the attack, we can use the resulting ciphertext as a key and start a chain until finding a distinguished point. Then, we check if this end point is in our table, take the corresponding start point and restart chaining until we find the ciphertext again. The secret key is its predecessor in the computed chain.

This basic scheme illustrates that the success rate of the attack depends on how well the computed chains "cover" the key space. In order to improve this cover, we made use of different mask function that mixes the output bits of DES, allowing different covers of the key space. In the tradeoff method, one tries to store information about as many different keys as possible by taking as long chains as possible. However, different critical overlap situations can appear during the precomputation and add constraints to the tradeoff.

1. A chain can cycle. This is the case where we find an already computed key of the chain before finding a distinguished point.

2. Two chains computed with the same mask function can merge. This is the case where two start points give two different chains with the same image. This means that from the moment of the merge until the end of at least one chain, both chains contains the same keys.

3. A chain can collide with another chain computed with a different mask function. This is the situation where two chains computed with different mask functions have some common points between SP and EP. It means that some keys are stored several times, which is not efficient.

We dealt with cycles by choosing an adequate maximum chain length and the mergers were rejected after the precomputation by keeping the longest of two merging chains. Collisions only appear in the final success rate. Consequently, the effectiveness of the tradeoff highly depends on the choice of its parameters and our experimental results underlined theoretical lacks in existing models.

The implementation choices were enforced by precomputation task and online attack. Obviously, the online attack had to be efficient on every PC and therefore is being dealt with in the software part. On the other hand, we performed the precomputation task with an optimal usage of the FPGA considering its limited size. It led us to carry out some parts of the precomputation by software like the sort on end points. We also wanted our hardware circuit to be parametric in order to change the tradeoff parameters by software. Therefore, some tasks are hardware implemented with a software control (SWC). Table 2 summarizes the hardware vs software design decisions.

The DES encrypts a 64-bit plaintext using a 56-bit secret key. We defined a 40-bit DES by fixing 16 key bits to arbitrary values and propose a chosen-plaintext attack to recover the 40 bits of the secret key.

Using our hardware implementation, we performed the precomputation task in about one week, using one single FPGA. The resulting chains were stored on 16 CDROM's representing the memory requirements of the online attack. Then, we implemented the online attack that can be applied using any PC with a sufficient memory (roughly 8 Gbits). We experimentally tried the resulting attack for 2000 different keys and we recovered one key in about 10 seconds with a success rate of 72 %. An exhaustive key search of the key on the same PC would have taken 50 days.

Table 2: Design decisions

Task	HW	SW	SWC
SP generation	X		
DES chaining	X		
Mask functions	X		X
Chains rejection	X		X
DP detection	X		
Triples storage		X	
Sort on EP		X	
Mergers rejection		X	
Online attack		X	

To conclude, we performed a first implementation of a time-memory tradeoff using distinguished points and presented experimental results that confirm their effectiveness in cryptanalytic contexts. The resulting chosen-plaintext attack significantly improves all existing complete cryptanalytic techniques attempted against DES in terms of speed. The method is general and could be applied to other block ciphers or one-way functions. Note that time-memory tradeoff attacks can be dangerous even when the key space is too large to be exhaustively precomputed (say 2^{80}). Consider an application where immediate inversion of a single cipher can be disastrous (e.g. an online bank transfer), then, constructing tables that would cover “only” 2^{60} keys would allow online inversion with probability 2^{-20}, which is not negligible.

Conclusion

In cryptographic applications, FPGA’s can be used as dedicated computers in order to perform some computations at very high frequencies. It allows to reach high encryption rates for different block ciphers. A consequence is that these implementations can speed up well-known attacks and improve their effectiveness, with some additional constraints that can be solved either by modifying algorithms or by hardware/software co-designs.

Practically, we implemented DES and AES and performed two cryptanalytic attacks. Both improved existing results. We performed the fastest implementation of the linear cryptanalysis and the first experimental results of a time-memory tradeoff using distinguished points.

Finally, future devices combined with high speed processors should reduce the hardware constraints and permit the implementation of a variety of cryptographic algorithms and cryptanalytic attacks on FPGA’s. This could replace distributed computations in the coming years.

References

[1] Mitsuru Matsui: Linear Cryptanalysis Method for DES Cipher, EUROCRYPT93 : 386-397.

[2] Mitsuru Matsui: The First Experimental Cryptanalysis of the DES, CRYPTO94 : 1-11.

[3] M.Hellman: A Cryptanalytic Time-Memory Tradeoff, IEEE transactions on Information Theory, Vol 26, 1980, pp.401-406.

[4] D.Denning: Cryptography and Data Security, p.100, Addison-Wesley, 1982, Out of Print.

[5] Douglas R. Stinson: Cryptography, Theory and Practice, CRC press, 1995.

[6] Xilinx: Virtex 2.5V Field Programmable Gate Arrays Data Sheet, http://www.xilinx.com.

[7] Rouvroy G, Standaert FX: Implementation of Cryptographic Standards and Cryptanalysis using FPGA’s, Master thesis, UCL, 2002.

Opto-Electronics review
An International Journal

Opto-Electronics review (O-ER) was established in 1993 for publication of scientific papers concerning optoelectronic materials, systems and signal processing. This journals covers the whole field in theory, experiments, techniques and instrumentation and brings together, within one journal, contributions from a wide range of optoelectronics related desciplines. Papers covering novel topics extending the frontiers of knowledge in optoelectronics are very encouraged.

Articles are published in O-ER in the following categories:

– invited reviews presenting the current state of knowledge,
– refereed research contributions reporting on original scientific or technological achievements,
– conference papers printed either in O-ER Special Issues, serving as Conference Proceedings, or in regular issues as invited or contributed papers, with special annotations indicating the conference where the paper has been presented,
– short news informing on recent research trends and technical achievements, dealing with all aspects of scientific, technological, technical and industrial works concerning generation, transmission, transformation, detection and application of light and other forms of radiative energy of which photon is the quatum unit,
– miscellany on various subjects, including conference reports and annoucements, presentations of research laboratories, advertisements, etc...

Opto-Electronics review is sponsored by the State Committee for Scientific Research (KBN), the Polish ministry of Science. Opto-Electronics review is published quarterly as a journal of the Association of Polish Electrical Engineers (SEP) by the Centre for training and Publications of SEP, under the auspices of the Polish Optoelectronics Committee of SEP and the Polish Chapter of the Society of Photo-Optical Instrumentation Engineers). Only subscription guarantees receiving the journal. You may subscribe Opto-Electronics reviewstarting from any future issue. Back issues of all previously published volumes in hard copies are available directly from the Editorial Office. The annual rate for 2002 is 40 USD.