Exhaustive Key Search of the DES: Updates and Refinements

Jean-Jacques Quisquater & François-Xavier Standaert

UCL Crypto Group
Laboratoire de Microélectronique
Université Catholique de Louvain
Place du Levant, 3, B-1348 Louvain-La-Neuve, Belgium
quisquater,standaert@dice.ucl.ac.be

Abstract. Exhaustive key search is the simplest attack against a cryptosystem, but it is sometimes the most realistic. This is specially true for carefully designed block ciphers for which advanced cryptanalysis (*e.g.* linear, differential) is not applicable. In this paper, we first update the cost of an exhaustive key search of the Data Encryption Standard (DES) using Field Programmable Gate Arrays (FPGAs). Then we illustrate how a time-memory tradeoff attack can be mounted for a similar cost, with much more dramatic consequences.

1 History

People have questioned the security of the DES [9] for a long time and there has been much speculation on its design principles, *e.g.* for the cryptographic significance of the S-boxes. However, in practice, no other attack has been as intensively discussed (and demonstrated) as exhaustive key search. Even nowadays, the major concern about the security of the DES is its too short key length. In this section, we briefly review certain historical results of exhaustive key search machines.

The first exhaustive DES key search machine estimation was proposed by Diffie and Hellman in 1977 [4] and contained 10^6 DES chips, with an estimated cost of US$ 20M (Million) and a 12-hour expected search time. They argued that this was out of reach for almost everybody, excepted organizations like the National Security Agency (NSA), but that by the 1990s, the DES would be totally insecure. Meanwhile, hardware implementations of the DES slowly approached the million-encryption-per-second requirement of Diffie and Hellman's special-purpose machine. By 1987, chips performing 512 000 encryptions per second were being developed.

Subsequently, Quisquater and Delescaille studied the related question of collision search in the DES in [11], while Quisquater and Desmedt investigated the cost of a random key search machine compared to a systematic search in a table in [12]. They also suggested that distributed computing could be a solution for such computationally intensive problems.

In 1993, Wiener [18] provided a gate-level design for a US$ 1M machine using 57 600 DES chips with an expected success in 3.5 hours. Every chip contained 16 pipeline stages, running at a clock frequency of 50 Mhz, going on with the popular story of "DES crackers".

At the 1997 annual RSA Cryptographic Trade Show in San Francisco, a prize was announced for cracking a DES cryptogram. The prize was claimed in five months, by a loose consortium using computers scattered around the Internet. It was the most dramatic success so far for an approach earlier applied to factoring and to breaking cryptograms in systems with 40-bit keys. At the 1998 RSA show, the prize was offered again. This time, it was claimed in 39 days.

Finally, to prove the insecurity of the DES, the Electronic Frontier Foundation (EFF) built the first unclassified hardware for cracking DES. On Wednesday, July 17, 1998, the EFF DES Cracker [5], which was built for less than US$ 200 000, easily won the RSA Laboratories's "DES Challenge II " contest. It took the machine less than 3 days to complete the challenge, shattering the previous record of 39 days set by a massive network of ten thousand computers.

2 Cost updates

Due to its potential to greatly accelerate a wide variety of applications while maintaining good flexibility, reconfigurable computing has gained importance in the industrial development of digital signal processing applications. FPGAs notably allowed to develop extremely fast and compact encryption architectures of various block ciphers including the DES, *e.g.* in [14]. These designs have been used to perform the fastest-known experimental linear cryptanalysis of the DES in [15].

It is usually considered that the major problem of current reconfigurable devices is their high individual cost. Latest Xilinx Virtex-II® devices can presently cost up to US$ 1 000. However, regarding cheaper technologies, Xilinx® recently announced (October 6, 2003) breakthrough price points for its low cost Spartan-3® family. Table 1 compares some implementation results of a 37 pipeline stages unrolled DES for these two technologies.

Device	**Virtex-II®**	**Spartan-3®**
Nbr of LUTs	3775	3780
Nbr of registers	4387	4408
Nbr of slices	2965	2958
Clock frequency (Mhz)	333	180
Encryption rate (Gbits/sec)	21.3	11.5

Table 1. DES implementation results.

Regarding price constraints, the 3S1000 Spartan-3® device with 1 million system gates is available for under US$ 12. It contains 7 680 slices and can consequently embed 2 DES designs. Considering a realistic clock frequency of 134 Mhz, we may therefore assume an encryption rate of $2 \times 134.10^6 \simeq 2^{28}$ encryptions per second for US$ 12.

These predictions already suggest that the use of recent (low cost) reconfigurable devices can be relevant for exhaustive key search attacks against block ciphers. As a practical estimation, a US$ 12 000 machine could break DES in about 3 days, presently or in the near future. Although these estimations do not take the development costs into account and must therefore be relativized, they clearly underline that the cost of a DES cracker is presently reachable for most organizations, including universities. In the next section, we show how time-memory tradeoff attacks can be implemented for a similar cost.

3 Time-Memory Tradeoffs

A cryptanalytic time-memory tradeoff allows the cryptanalysis of any k-bit key cryptosystem in $O(K^{\frac{2}{3}})$ operations[1] with $O(K^{\frac{2}{3}})$ storage, if a precomputation of $O(K)$ operations was performed beforehand. Such tradeoffs were initially suggested by Hellman in 1980 [7]. In this section, we investigate a variant denoted as "time-memory tradeoff using distinguished points", originally suggested by Rivest in [3] and practically implemented using FPGAs in [16]. The technique was also discussed in [1]. As we only intend to discuss the implementation cost of such attacks, we refer to the original papers for further theoretical considerations.

3.1 Brief description

A time-memory tradeoff using distinguished points method is composed of a precomputation task and an online attack.

Precomputation task: A chain is formed by a number l of encryptions involving a chosen plaintext and l different keys. The chaining is obtained by using every block cipher output as the next stage key. A defined property holds for the first and last keys and we call them distinguished points (DP). During the precomputation, a number of chains are computed and start points, end points and the corresponding chain lengths are stored in a table.

[1] $K = 2^k$.

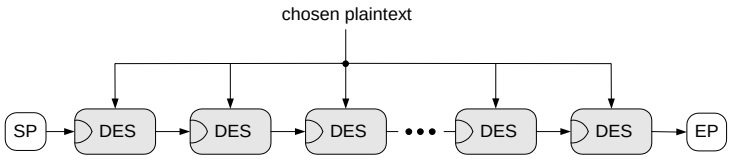

Fig. 1. Precomputation task.

Online attack: Let the chosen plaintext be encrypted with a secret key and intercepted by an attacker. During the attack, he can use the resulting ciphertext as a key and start a chain until he finds a DP. Then, he checks if this end point is in the precomputation table, takes the corresponding start point and restarts chaining until he finds the ciphertext again. The secret key is its predecessor in the computed chain.

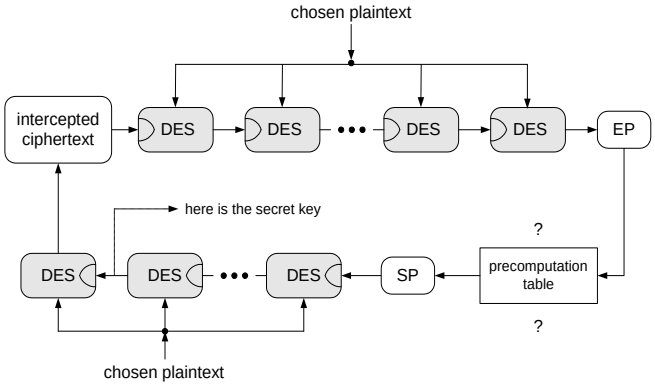

Fig. 2. Online attack.

From these descriptions, it is clear that the actual probability of success of the attack depends on how the precomputed chains cover the key space. Unfortunately, there is a chance that chains collide or merge. The larger is a table, the higher is the probability that a new chain merges with a previous one. Therefore, to obtain a higher probability of success, one generates multiple tables by using different mask function at the end of the encryption. A simple and efficient mask function is a XOR with a constant value (see [16]).

3.2 Implementation

As it is possible to use efficient FPGA implementations of block ciphers to perform exhaustive key search, we can adapt a pipeline design in order to carry out the precomputation part of the time-memory tradeoff attack. In this section, we present a generic design for precomputing encryption chains with distinguished points.

In general, if the targeted block cipher is implemented as a p-stage pipeline design, we will deal with p different chains in parallel. A simple solution to manage this task is to store the different start points (denoted as $K_1, K_2, \ldots, K_p$) in a p-stage shift register as represented in Figure 3. In this architecture, the

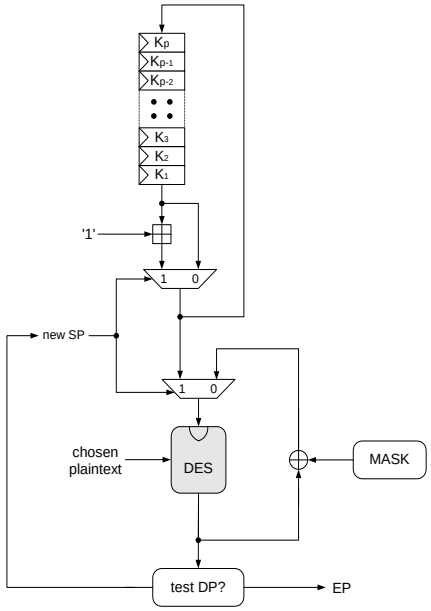

Fig. 3. FPGA precomputation design.

block cipher output always corresponds to an intermediate point in the chain for which the start point is available at the shift register output. Every time a distinguished point is detected, we output the resulting end point and increment the corresponding start point in order to initialize a new chain. In addition, a different mask function (represented as a bitwise $\oplus$ with a constant Boolean vector in the figure) can be applied to every computed chain. This design can obviously be parallelized for different mask functions. Finally, p additional counters are required in order to compute the different chain lengths, although it is not represented in Figure 3.

The major advantage of this scheme is that it only requires an efficient block cipher implementation and a few additional resources (*e.g.* counters). Moreover, due to the "shift register-pipeline" structure, it does not imply any complex interface to output the results and does not reduce the block cipher clock frequency. As a consequence, it allows the precomputation tables for time-memory tradeoffs to be computed at roughly the same cost and throughput as an exhaustive key search.

3.3 Practical attacks

The previous design suggests that the precomputation task of a time-memory tradeoff attack can be performed as efficiently and for a similar cost as exhaustive key search, *i.e.* around US$ 12 000. Let us now investigate the consequences if such precomputations were made public.

Say, for example, that precomputations against the DES are performed with average chain length of 2^{19} and 2^{19} different mask functions (these are the typical parameters suggested by Hellman). We also assume that every table contains around 2^{18} start points so that the key space is sufficiently covered by the precomputed chains[2] According to these parameters, the processing complexity of the online attack equals 2^{38}. Assuming a computational power of 2^{28}, provided by a single US$ 12 FPGA, the DES would be broken in half an hour, with high probability.

Before to conclude, remark that, from a practical point of view, time-memory tradeoff attacks have recently attracted significant attention within the cryptographic community because of two major results against "real-world" devices. These facts probably ended the "DES crackers" success story.

First, in 2002, Clayton and Bond [2] described experiments to attack the IBM 4758 CCA® device, used in retail banking to protect the ATM infrastructure. This practical scheme collected the necessary data in a single 10-minute session.

Secondly in 2003, a Swiss team from the EPFL[3] used a time-memory tradeoff device to implement an attack on MS-Windows® password hashes. Using 1.4 GB of data, they were able to crack 99.9% of all the alphanumerical passwords hashes (from a set of 2^{37} items) in 13.6 seconds. In addition, Oechslin [10] described an alternative solution to reduce the number of table lookups during the online attack, using rainbow tables.

4 Conclusion

This paper shortly updated the cost of an exhaustive key search attack against the DES, using recent and low cost FPGA devices. In addition, we suggested that the precomputation cost of a time-memory tradeoff attack is comparable to exhaustive key search, while the consequences of such attacks are by far more dramatic. In practice, such precomputations could be performed in less than one week for US$ 12 000 and the resulting online attack would break a DES key with high probability, in half an hour, using a single US$ 12 FPGA.

[2] Still, the precomputation is not perfect due to collisions, as explained in [16].
[3] EPFL: Ecole Polytechnique Federale de Lausanne.

References

1. J. Borst, B. Preneel, J. Vandewalle, *On the Time-Memory Tradeoff Between exhaustive key search and table precomputation*, in the proceedings of the 19th Symposium in Information Theory in the Benelux, pp 111-118, Veldhoven, Netherlands, 1998.
2. R. Clayton, M. Bond, *Experience using a Low-Cost FPGA Design to Crack DES Keys*, in the proceedings of CHES 2002, Lecture Notes in Computer Sciences, vol 2523, pp 579-592, Redwood City, USA, August 2002, Springer-Verlag.
3. D. Denning, *Cryptography and Data Security*, pp 100, Addison-Wesley, 1982.
4. W. Diffie, M. Hellman, *Exhaustive Cryptanalysis of the NBS Data Encryption Standard*, Computer, vol 10, pp 74-84, 1977.
5. Electronic Frontier Foundation, *Cracking DES*, O'Reilly & Associates, 1998.
6. A. Fiat, M. Naor, *Rigorous Time/Space Tradeoffs for Inverting Functions*, in the proceedings of STOC 1991, pp 534-541, New Orleans, Louisiana, USA, May 1991.
7. M. Hellman, *A Cryptanalytic Time-Memory Tradeoff*, IEEE Transactions on Information Theory, Vol 26, num 4, pp 401-406, 1980.
8. K. Kusuda, T. Matsumoto, *Optimization of Time-Memory Tradeoff Cryptanalysis and its Applications to DES, FEAL-32 and Skipjack*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science, vol 79-A, pp 35-48, January 1996.
9. National Bureau of Standards, *FIPS PUB 46, The Data Encryption Standard*, Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, Jan 1977.
10. P. Oechslin, *Making a faster Cryptanalytic Time-Memory Trade-Off*, in the proceedings of Crypto 2003, Lecture Notes in Computer Sciences, vol 2729, pp 617-630, Santa Barbara, California, USA, August 2003, Springer-Verlag.
11. J.-J. Quisquater, J.P. Delescaille, *How easy is collision search? Application to DES*, in the proceedings of Eurocrypt 1989, Lecture Notes in Computer Sciences, vol 434, pp 429-434, Houthalen, Belgium, August 1989, Springer-Verlag.
12. J.-J. Quisquater, Y.G. Desmedt, *Chinese Lotto as an Exhaustive Code-Breaking Machine*, in Computer, vol 24, pp 14-22, 1991.
13. J.-J. Quisquater, F.-X. Standaert, G. Rouvroy, J.-D. David, J.-D. Legat, *A Cryptanalytic Time-Memory Tradeoff: First FPGA Implementation*, in the proceedings of FPL 2002, Lecture Notes in Computer Sciences, vol 2438, pp 780-789, Montpellier, France, September 2002, Springer-Verlag.
14. G. Rouvroy, F.-X. Standaert, J.-J. Quisquater, J.-D. Legat, *Design Strategies and Modifed Descriptions to Optimize Cipher FPGA Implementations: Fast and Compact Results for DES and Triple-DES*, in the proceedings of FPL 2003, Lecture Notes in Computer Science, vol 2778, pp 181-193, Lisbon, Portugal, September 2003, Springer-Verlag.
15. G. Rouvroy, F.-X. Standaert, J.-J. Quisquater, J.-D. Legat, *Efficient Uses of FPGAs for Implementations of the DES and its Experimental Linear Cryptanalysis*, IEEE Transactions on Computers, vol 52, num 4, April 2003.
16. F.-X. Standaert, G. Rouvroy, J.-J. Quisquater, J.-D. Legat, *A Time-Memory Tradeoff Using Distinguished Points: New Analysis and FPGA Results*, in the proceedings of CHES 2002, Lecture Notes in Computer Sciences, vol 2523, pp 593-609, Redwood City, USA, August 2002, Springer-Verlag.
17. P.C. van Oorschot, M.J. Wiener, *Parallel collision search with cryptanalytic applications*, Journal of Cryptology, vol 12, num 1, pp 1-28, Winter 1999, Springer-Verlag.
18. M.J. Wiener, *Efficient DES Key Search*, Technical Report TR-244, School of Computer Science, Carleton University, Ottawa, 1994, *Presented at the rump session of Crypto'93.*