Hardware Implementation of DES Encryption Cracker

Bilal Shahid, Haris Tauqeer, Muhammad Saqib Ilyas
Department of Computer and Information Systems Engineering
NED University of Engineering and Technology, Karachi, Pakistan
haris_tauqeer@hotmail.com, {bilal_shahid, msaqib}@ieee.org

Abstract

As the security of the e-property is becoming a major concern, the field of cryptography has gained much importance. It is becoming essential to ensure that only those intended to see the information should see it. Cryptography enables you to manipulate your message using different encryption algorithms in such a way that only authorized people can view it. The robustness of these algorithms is usually determined by putting them under numerous attacks known as Encryption Cracking and the surviving one is considered the securest. One of these algorithms is DES (Data Encryption Standard). In this paper we discuss the design and implementation of DES Encryption Cracker, proving the power of the dedicated hardware, using Brute Force attack.

Keywords: *Brute Force, Unicity Distance, CT (cipher Text), DEC (DES Encryption Cracker)*

1. INTRODUCTION

The history of Cryptography, the science of encrypting and decrypting information, dates as far back as 1900 BC. The basic idea invented at that time is still being used by scientists and mathematicians of modern age in more organized form to create complex encryption algorithms. The idea is as simple as to change the order of the secret message (means ciphering the text), making it unreadable, in a particular manner (using some key) .That manner is known only to the desired recipient who can rearrange it (deciphering the cipher text) to make it readable. The advancements in the field of cryptography are the result of rapid growth of cryptanalysts (Encryption Crackers).

DES [5] known as data encryption standard was developed by IBM in 1977.This cipher was then widely adopted by the industry for use in security products. It is no longer secure in the original form but in the adapted form it is still being used.
It was demonstrated in [1] that DES keys can be extracted within a matter of minutes from a banking ATM system using a low cost off the shelf FPGA hardware, exploiting weaknesses in the randomness of some of the S-boxes.

The work in [2] discussed an implementation of a DES cracker on a field programmable hardware trying 800 million keys/second which would be able to complete in 1040 days. The work in [3] cracked DES encryption using brute force utilizing collaborating computers on the Internet. Also, [6] discussed a similar attack on an RFID based vehicle immobilizer.

Keeping in view the advances in VLSI hardware power, we set about to implement a brute force DES cracker of our own using a low cost FPGA board. We managed to demonstrate a successful implementation.

2. DES (DATA ENCRYPTION STANDARD)

In DES, the plaintext is encrypted in blocks of 64 bits, yielding 64 bits of cipher text by using 56 bit key. DES has 19 distinct stages. The first stage is the key independent transposition on the 64-bit plaintext. The last stage is the exact inverse of this transposition. The stage prior to the last one exchanges the leftmost 32 bits with the right most 32 bits. The remaining 16 stages are functionally identical but are parameterized by different functions of key. The algorithm has been designed to allow decryption to be done with the same key as encryption. The steps are just run in the reverse order.

3. BRUTE FORCE ATTACK

As implied by the name, the algorithm simply cycles through all possible keys and attempts to perform the decryption each time, PT = DES (K, CT). Since one of the keys is the correct one, eventually the algorithm will succeed. It is also known as exhaustive search .It is the most brutal of all the attacks. Now the point arises here, what

is the point of encryption if you can crack it?

The answer is to make the cracking process take an infeasible amount of time to succeed. With a key of n bits, there are 2^n possible keys, so provided you use enough bits for the exponentiation to really kick in, then your key will be secure. Encryption systems work by making sure that the number of possible keys is so large that even with many years worth of computation, the attack will not succeed. Here comes the solution to shorten this many years computations to just the matter of hours by using the power of dedicated hardware, that is, using an FPGA board to implement the attack.

4. CALCULATIONS

In implementing brute force attack timing is the biggest constraint. Therefore, we did the following timing analysis before designing the DEC.

- Clock Frequency = 33 MHz (Available clock we will be using)
- Time Period = 30.3 ns
- DES completes its iterations in 16 clock cycles
- For 1 clock cycle we have 30.3ns
- For 2^35 combinations we have = (2^35 * 30.3ns*16) =4.52 Hours Approximately.

But according to human psychology, if we apply letters and numbers first, omitting the special characters this will reduce the time drastically. For letters, numbers and space we have 64^5 combinations instead of 128^5 so the required time is 2^30 * 30.3ns * 16 = 8.67 minutes.

If we Implement multiple DES crackers in the FPGA and coordinate their operation so as to crack the CT faster, since the FPGA is much bigger than a single DES module, we will be able to reduce this time to 8 minutes for the whole exhaustive search and 16.26 seconds if key consists of letters, numbers and space. Cracking time is a function of the number of parallel modules.

According to the same theory if we use 1 GHz processor we can break the whole DES 56 bit encrypted in just 9 Hours for exhaustive search and 18 minutes for alphabets.

5. ARCHITECTURE

DEC is designed to take 64 byte of cipher text as input encrypted with 35 bit key and produces a 64 byte plaintext. DEC mainly consists of 4 main modules, namely:

- Block Selector Module
- Plain Text Validating Module
- Key Generator Module
- Storage Module
- DES core used as a black box

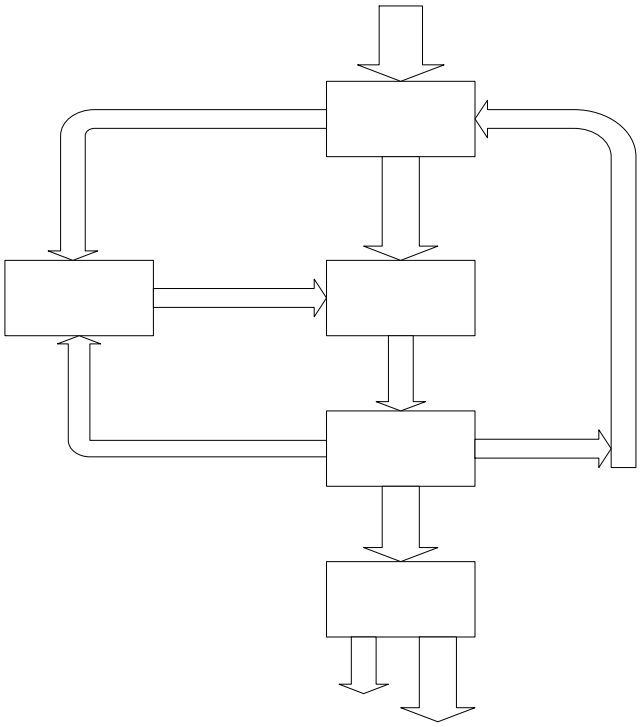

Fig. 1: DES Encryption Cracker Block Diagram

5.1 Block Selector Module

As DES algorithm takes 8 byte of input at a time, so as to process a 64 byte cipher text, it is divided into blocks of 8 byte each. This is the primary function of this module.

5.2 Intelligent Key Generator Module

This module is designed to generate appropriate 56 bit keys. This is basically the brute forcing module producing numerous keys. Keeping in mind the human psychology, this module first applies keys consisting of alphabets, numbers and spaces, thus reducing the number of keys drastically and saving time. If unsuccessful it will fallback to the exhaustive search and would

apply all combinations.

5.3 Plaintext Validating Module

The tricky part in applying brute force attack to DES is to recognize the correct plaintext out of streams of results produced.

5.3.1 How to recognize the plaintext. The machine knows that it found the plaintext because it looks like plaintext. [4] discusses this problem in detail.

Plaintext tends to "look" like plaintext. It's an English-language message, or a data file from a computer application (programs like Microsoft Word have large known headers; even PK-ZIP files have known headers), or a database in a reasonable format. When you look at a decrypted file, it looks like something understandable. When you look at a cipher text file, or a file decrypted with the wrong key, it looks like gibberish.

Another thing that is most important for brute forcing any encryption algorithm is unicity distance.

5.3.2 Unicity Distance. In the 1940s, Claude Shannon invented a concept called the unicity distance. Among other things, the unicity distance measures the amount of cipher text required such that there is only one reasonable plaintext. This number depends both on the characteristics of the encryption algorithm and its key length. The unicity distance for DES is 8.2 Bytes

The PVM module scans every character of the result produced by the DES core and checks its validity as a plaintext (numbers, letters and space) and rejects the gibberish.

5.4 Storage Module

The storage module stores all the 64 byte of the valid plaintext. To make the design simpler the attack is applied on only 8 byte of the cipher text and the cracked key thus produced is used for cracking the remaining 56 bytes

6. TECHNICAL ISSUES

The power of the cracking machine is evaluated on the basis of its timing. Speed is the major consideration in the design of this project. One of the challenges faced during the design was the integration of the separate modules and their timing. Due to the high speed requirement, wastage of even a single clock cycle was not acceptable.

Individual testing prior to the integration is one of the most important jobs not to be overlooked. Modules must be thoroughly tested individually.

The mode of the algorithm is another very important factor that needs to be considered in the design.

This DEC is designed for the area-optimized core of DES operating in ECB (Electronic Code Book mode).

7. FUTURE PROPOSITIONS

As there is always room for improvement, we propose some techniques, that we learnt by the design and implementation of this project, which can be implemented to boost up the performance of this cracker.

Key Generating algorithm can be ameliorated by exacting the keys with the use of probabilistic techniques, random algorithms and removing the most inadequate keys thus minimizing the number of combinations.

8. CONCLUSION

In this paper we have tried to document every issue relating to the management, design and implementation of this project. This project not only proves the power of the hardware but also expose the vulnerability of the DES encryption to the high speed VLSI chips. The idea behind this cracking with some improvement can be utilized to attack more advance encryption algorithms. In our view attacking the Encryption Algorithms is one of the best methods to check the robustness of the algorithm.

We can also speed up the cracking process by introducing additional optimizations such as, pipelining the DES core, exiting each test as soon as a non-plaintext block is encountered operation so as to crack the CT faster.

References

[1] Richard Clayton, Mike Bond, "Experience Using a Low-Cost FPGA Design to Crack DES

Keys," Cryptographic Hardware and Embedded Systems - CHES 2002: 4th International Workshop Redwood Shores, CA, USA, pp. 579-592, August 13-15, 2002

[2] Ivan Hamer and Paul Chow, "DES Cracking on the Transmogrifier 2a," Cryptographic Hardware and Embedded Systems: First International Workshop, CHES'99, Worcester, MA, USA, pp. 13, August 1999

[3] Curtin, M., and Jolske, J., "A Brute Force Search of DES Keyspace," login - the Newsletter of the USENIX Association, May 1998

[4] Crypto-Gram Newsletter, December 15 1998, by Bruce Schneier, President Counterpane Systems

[5] Federal Information, Processing Standards Publication 81, 1980 December 2. Announcing the standards for "DES modes of operation".

[6] Steve Bono, Mathew Green, "Analysis of the Texas Instrument DST RFID"

About the authors

Bilal Shahid and Haris Tauqeer are undergraduate senior students at Department of Computer and Information Systems Engineering. Their senior project deals with the design of a Linux based Embedded System using an FPGA board. Their interests lie in the area of ASIC/VLSI design.

Muhammad Saqib Ilyas is an Assistant Professor at Department of Computer and Information Systems Engineering. He holds a Masters degree in Electrical Engineering from Wichita State University, Wichita, KS, USA. His areas of interest include computer networks, computer network security, and connected systems.