Proposal Report

Decryption Algorithm Characterization

Ravi Mohan Boggula, Rajesh Reddy Bandala

Introduction

Computer World has taken an amazing revolution with introduction of Internet, which serves as a communication medium among various users throughout the world. Maintaining privacy in network communications is something everyone desires. “Encryption&Decryption” is a means to achieve that kind of privacy. It was invented for that very purpose.

In simple and common layman terms “Encryption” could be regarded as a process of making a readable form into unreadable form. And the reverse process could be regarded i.e. conversion from unreadable form into readable form (original form) as “Decryption”.

Encryption&Decryption is the process of scrambling a message so that only the intended recipient can read it. The actual cryptographic (methods or schemes available for Encryption/Decryption) process is generally a complicated mathematical formulation, the more complex -- the more difficult to break. A key is supplied to the recipient so that they can then decipher the message. Keys for encryption algorithms are described in terms of the number of bits. The higher the number of bits - the more difficult that cryptosystem would be to break.

Encryption&Decryption can provide a means of securing information. As more and more information is stored on computers or communicated via computers, the need to insure that this information is invulnerable to snooping and/or tampering becomes more relevant. Any thoughts with respect to your own personal information (i.e. medical records, tax records, credit history, employment history, etc.) may bring to mind an area in which you DO want, need or expect privacy.

Data confidentiality concerns who can read and understand a message that is exchanged between sender and receiver. Confidentiality is achieved through the use of cryptosystems, that is, companion encryption and decryption algorithms that respectively lock and unlock the contents of a message. Prior to encryption, the message is said to be in plaintext form. After encryption, it is usually called ciphertext.

As the number of Intrusions in to Network systems increasing year by year, the Encryption and Decryption Algorithms play an important and specific role in maintaining the data privacy and secrecy. Now further In order to characterize these algorithms we must evaluate certain performance characteristics of algorithms with respect to time of transmission.
Previous Work

In order to provide a secure communication for users in Network Communications NIST (National Institute of Federal Standards) has devised an important standard named DES (Data Encryption Standard), in this scheme data is encrypted and decrypted in 64-bit blocks, using a 64-bit key (although the effective key strength is only 56 bits, as the 8 bits are solely used for parity). It takes a 64-bit block of plaintext as input and outputs a 64-bit block of ciphertext. Since it always operates on blocks of equal size and it uses both permutations and substitutions in the algorithm, DES is both a block cipher and a product cipher.

Fiestel Structure of DES:

DES has 16 rounds, meaning the main algorithm is repeated 16 times to produce the ciphertext. It has been found that the number of rounds is exponentially proportional to the amount of time required to find a key using a brute-force attack. So as the number of rounds increases, the security of the algorithm increases exponentially.

The below figure explains how the data is converted from readable text (Plain Text) to unreadable text (Cipher text). In this there are 16 identical stages of processing termed as rounds, IP which is Initial Permutation and Final Permutation, the symbol XOR signifies XOR operation. Before the main rounds, the block size is divided into two equal halves (32 bit). The function F scrambles half a block together with some key, the output from the function F is then combined with the other half of the block, and these halves are swapped before the next round and after the final round or stage, the halves are not swapped. This is operation of Fiestal Structure.

The Fiestel structure ensures that decryption and encryption are very similar processes, the only difference being the keys that are applied in the reverse process while Decryption, the rest of the algorithm is identical.
Disadvantages of DES:

DES is now considered to be insecure for many applications. This is chiefly due to the 56-bit key size being too small; DES keys have been broken in less than 24 hours. There are also some analytical results which demonstrate theoretical weaknesses in the cipher, although they are infeasible to mount in practice. The algorithm is believed to be practically secure in the form of Triple DES, although there are theoretical attacks. In recent years, the cipher has been superseded by the Advanced Encryption Standard (AES).
**Present Work**

NIST proposed the much advanced version (AES) implemented by different algorithms namely

1. Rijndael (which supports 128,192,256 bits)
2. Serpent (which supports 128,192,256 bits)
3. Two Fish (which supports 128,192,256 bits)
4. RC6 (which supports 128,192,256 bits)
5. Mars (which supports 128,192,256 bits)
6. Krypton (which supports 128,192,256 bits)

All these algorithms support a key size, block size of value ranging from 128 to 256 bits ($2^{128}$ to $2^{256}$ bits) which takes millions of years to break. So thus it has edge over DES (which supports 56 bits i.e. $2^{56}$ bits) standard which was broke by Brute Force attack within 24 hours.

Further in October 2000, NIST released its report on the development of an Advanced Encryption Standard which compared the five Round 2 algorithms in a number of categories. The table below summarizes the relative scores of the five candidates (1=low, 3=high):

<table>
<thead>
<tr>
<th></th>
<th>MARS</th>
<th>RC6</th>
<th>Rijndael</th>
<th>Serpent</th>
<th>Twofish</th>
</tr>
</thead>
<tbody>
<tr>
<td>General security</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Implementation of security</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Software performance</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Smart card performance</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Hardware performance</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Design features</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

NIST recommended adoption of the Rijndael algorithm, and released a draft Federal Information Processing Standard (FIPS) AES Specification for public review and comment in February 2000. The draft standard is under review prior to being released later this year (2001).

(Material source from: [http://www.cescomm.co.nz/about/aes.html](http://www.cescomm.co.nz/about/aes.html))
The purpose of this project mainly deals with evaluating the processing performance with respect to Energy and Power Consumed; this could be achieved by taking an analysis or using the tool called Wattch.

About Wattch Tool

“WATTCH” is an architectural simulator that estimates CPU power consumption. The power estimates are based on a suite of parametrical power models for different hardware structures and on per-cycle resource usage counts generated through cycle level simulation. The power models have been integrated into the “Simple Scalar” architectural simulator.

A modified version of Simple Scalar’s sim-outorder is used to collect results by WATTCH. SimpleScalar provides a simulation environment for modern out-of-order processors with 5-stage pipelines: fetch, decode, issue, writeback and commit. Speculative execution is also supported. Separate banks of 32 integer and floating point registers make up the architected register file and are only written on commit. The power oriented modifications provided by WATTCH (whose modules are integrated within SimpleScalar) track which units are accessed on each cycle and how and compute the power values associated with those units accordingly.

These power modules have been verified against industrial circuits and have been found to be within 10% for low level capacitance estimates.
Expected Results:

Our approach in this project is to evaluate the energy and power consumption for different AES candidates mentioned above by using the Wattch simulator, and then evaluating the performance by plotting graphs with respect to file size with time and further with respect to Energy and Power Consumed i.e. during the file transmission. We will tabulate the obtained values in a chart and then plot a graph using the tabulated values and further analyze the performance of the AES candidates.

The first graph is plotted with respect to time taking an ideal assumption that irrespective of whatever the size may be the Time factor is equal and the second one with respect to power/energy taking a linear condition that as file size increases the energy and power increase linearly.
Project Summary:

Encryption and Decryption play a specific role in data secrecy and integrity in Internet where Hackers wait eagerly to modify the data. In order to optimize further characterizations of the Decryption algorithms we have to make a thorough study of the different Encryption and Decryption standards. Now our purpose of the project is to study and understand the various AES candidates (algorithms) simulate them in real time environment using Watch simulator, evaluate their performance with respect to Energy requirement and Power Consumption which will enable to analyze the Processing cost in real time environment. So our goal is mainly to characterize the Decryption algorithms specified by the AES standards in terms of performance while processing the files.

References:

http://herongyang.com/crypto
http://fp.gladman.plus.com/cryptography_technology/index.htm
http://www.cl.cam.ac.uk/~rja14/Papers/serpent.pdf