Side Channel Attack on various Cryptographic Algorithms

Shraddha More, Prof.Rajesh Bansode

Abstract— Information security is the protection of information system against unconstitutional access to or amendment of information, whether in storage, processing or transit and against the denial of service to authorized users. In today's environment, security has become an important threat. Good cryptography techniques can help to mitigate this security threat. Advanced Encryption Standard (AES) is a symmetric key encryption standard widely used to secure data where data confidentiality is a critical issue. NIST exhausted several years to analyzing the Rijndael algorithm for vulnerabilities against all known breeds of attacks and finally declared it to be a secure algorithm. In 2005 Daniel J. Bernstein stated that the software implementation of AES is susceptible to side channel attacks. Cache timing attack is a type of side channel attack where the leaking timing information due to the cache performance of a cryptosystem is used by an attacker to break the system. Over the years many researchers have proposed a number of countermeasures against Cache Timing Attack but there is no substantiation to date of any investigation carried out to determine their effectiveness and efficiency. This paper focused on implementing timing attack on different cryptographic algorithms and investigating the countermeasures against the remote cache timing attack with positive outcomes, in which research work establish a secured environment for the cryptography.

Index Terms— AES, Caesar cipher, Cryptography, Countermeasures, Decryption, Encryption, DES, RSA, Side Channel Attack, Timing attack.

1 INTRODUCTION

Information security plays a foremost role in every automated system today. Information of a particular custom should be stored and conveyance securely without allowing unconstitutional parties to access or modify them. At this occurrence, the concept of cryptography comes in to picture. Today cryptography holds an important role since a large amount of military and financial data are being transferred through the Internet. Symmetric key cryptosystems are rendered with two public algorithms known as the encryption algorithm and the decryption algorithm. Encryption is a invertible process, in which mathematical operation performed on the sensitive message with the assistance of a public encryption algorithm and a secret-key which is portioned amongst authorized parties. When the encrypted message comes to a sanctioned person who has the knowledge of the secret-key, they can reverse the encrypted message with the help of a public decryption algorithm and the secret-key to get the meaningful erogenous message [1].

Advanced Encryption Standard (AES) is an encryption algorithm. It has become the US Federal Standard for information security after Data Encryption Standard (DES) which has become crumbly. AES is an evolved version of the Rijndael algorithm developed by John Daemen and Vincent Rijmen [2]. AES uses a fixed block size of 128- bit (16 bytes) and a key of size 128-bit, 192-bit or 256-bit.

- *Shraddha More: Master of Engineering in Information technology, TCET, Mumbai University, India. Email:moreshraddha30@gmail.com.*
- *Rajesh Bansode: Associate professor in Department of Information technology, TCET, Mumbai University, India. Email:rajesh.bansode@thakureducation.org.*

The number of rounds of encryption is varied according to the size of the key. Key size for 10, 12 or 14 rounds are 128-bit, 192-bit or 256-bit key respectively.

In each round except the final round, four operations are taking part. They are Sub Bytes, Shift Rows, Mix Columns and Add Round Key. Byte arrays of size 4x4 (16 bytes) are used for each of these operations. After a particular number of rounds according to the size of the key, the plain text is converted into the cipher text [3]. AES is an iterated block cipher, which uses a fixed block size of 128 bits and a key which is 128, 192 or 256 bits in length. Different transformations operate on the intermediate results, which is called as called states. After an initial round key addition, the state array is transformed by implementing a round function 10, 12, or 14 times depending on the length of the key. SubBytes, ShiftRows, MixColumns and AddRoundKey, these four stages are included in each round except the last round. Two of these stages involve transformations over Galois Field (GF - 28). Generally in software implementations, the multiplicative inverse over GF (28) is precomputed and stored in memory in a table named SBOX. To speed up execution of the cipher, software implementations may further combine the SubBytes and ShiftRows with MixColumns, transforming them into a sequence of table lookups. These tables store pre-computed values avoiding time consuming computations [4].

Side Channel Attacks (SCA) [5] are a form of cryptanalysis that engrossments not on breaking the underlying cipher directly, but on exploiting weaknesses found in certain implementations of a cipher. A Side Channel Attack (SCA) is any attack based on "side channel information", the information which can be gained from encryption device, which we cannot consider as the plaintext to be encrypted or the cipher text that results from the encryption procedure. The main lineament of side channel attacks is that they do not focus on change of integrity of the attacks based on side-channel information gained through timing information [7], radiation of various

sorts [8], power consumption statistics [9], cache contents [10], etc. This research work examines the relevance of side channel attack and have implemented a number of countermeasures and have evaluated their performance and solidness.

The rest of the paper is organized as follows: Section II describes related work. In Section III work discuss implemention of attack on cryptography algorithm. Section IV is on countermeasures against attack. Section V projects on conclusion.

2 Related Work

In 2005 Daniel Bernstein demonstrated a remote cache timing attack against AES [11]. Bernstein performed the attack successfully by using the OpenSSL 0.9.7a AES implementation on an 850MHz Pentium III Desktop Computer, running FreeBSD-4.8 as a network server. In this research work, the complete AES key is extracted using a client machine and pointed out that the same technique can be performed on more complicated servers with additional timing information. Work has also tested an AMD Athlon, an Intel Pentium III, an IBM PowerPC RS64 IV and a Sun UltraSPARC III processor with positive results.

Kocher has performed timing attacks on implementations of Diffie-Hellman, RSA, DSS and other cryptosystems [5]. Work stated that timing attacks are centred on measuring the time it takes for a unit to perform operations, where it track to information about secret keys and break the cryptosystem. Research also stated that the attack is computationally not much difficult and most of the time only known cipher text is required and also work has presented some techniques for preventing the attacks.

Felten et al. [13] have done timing attacks on web privacy. They have delineated an attacks that can compromise the privacy of user's web-browsing histories. The attacks license a malicious web site to collect information on users' browsing activities. By assessing the time the user's browser requires to perform certain operations, it can be determined. According to research, the time required for operations depends on the user's browsing history and this time variations bear enough information to contain the user's privacy because of the various forms of caching performed by browsers. According to research work these attacks can be carried out without the victim's knowledge. They pointed out that simple countermeasures cannot prevent these types of attacks.

In 2011, Alawatugoda et al. [14] have implemented possible countermeasures against remote cache timing attacks. In order to prevent cache timing attacks, research have followed the approach of masking leaking timing information. They have added several code fragments into the AES implementation. They have been able to do it without changing its semantics and also without severely reducing the ability of it. The software based countermeasures work have tested, which involve adding randomness and few actions on T-tables such as prefetching table values and cache partitioning where cache locations are allocated to load T-tables.

In 2012, Jayasinghe et al. have presented constant time encryption as a countermeasure against remote cache timing attacks [15]. Most of the software based countermeasures are vulnerable to statistical analysis, though they are flexible and easily organized. By observing that difficulty, they have tested a countermeasure that is safe against statistical analysis. Research method rescheduling the instructions of AES algorithm where the encryption rounds will consume constant time, regardless of the cache hits and misses. They have done so in major three steps which are decomposing the code into smaller bitwise operations, adding each and every bitwise instruction set to queues and processing each queue. Work have shown that the countermeasures have eliminated the side channel vulnerability.

3 Attack on Cryptography Algorithm

The research work includes implementing side channel attack on various cryptography algorithms. Work executed attack on AES, DES, RSA and Caeser cipher cryptography algorithms.

3.1 Attack on AES implementation

After successful implementation of the AES algorithm, this research executed attack on the encrypted file in such a way that at the time of decryption, receiver could not get the original file instead the user gets the file which is in human non-readable format as shown in the Fig. 1.

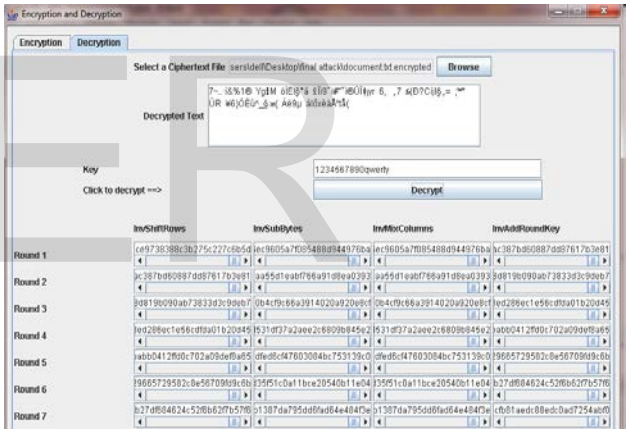

Fig. 1. Attack on AES implementation

Decryption time required for receiver to decrypt the AES implementation which is altered by the attacker is 1.9678 milliseconds.

3.2 Attack on DES implementation

After successful implementation of the DES algorithm, this research executed attack on the encrypted file in such a way that at the time of decryption, receiver could not get the original file instead the user gets the file which is in human non-readable format as shown in the Fig. 2.

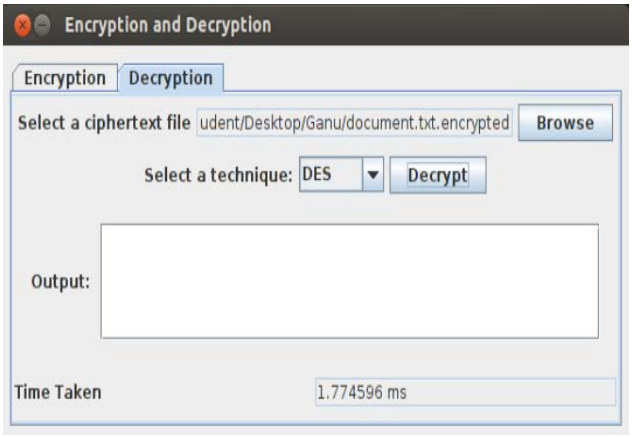

Fig. 2. Attack on DES implementation

Decryption time required for receiver to decrypt the DES implementation which is altered by the attacker is 1.774596 milliseconds.

3.3 Attack on RSA implementation

After successful implementation of the RSA algorithm, this research executed attack on the encrypted file in such a way that at the time of decryption, receiver could not get the original file instead the user gets the file which is in human non-readable format as shown in the Fig. 3.

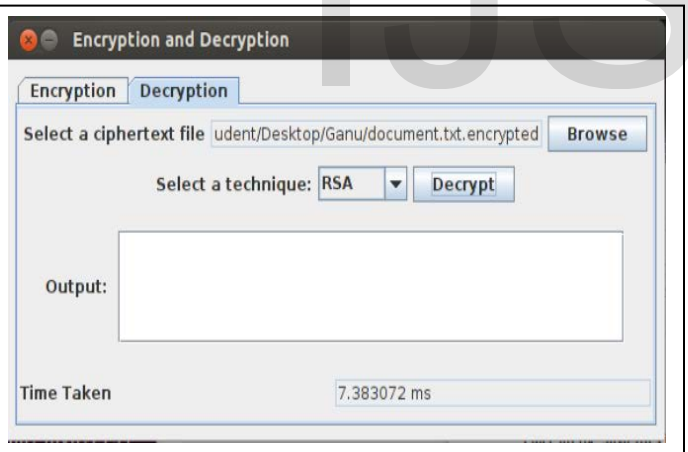

Fig. 3. Attack on RSA implementation

Decryption time required for receiver to decrypt the RSA implementation which is altered by the attacker is 7.383072 milliseconds.

3.4 Attack on Caesar cipher implementation

After successful implementation of the Caesar cipher algorithm, this research executed attack on the encrypted file in such a way that at the time of decryption, receiver could not get the original file instead the user gets the file which is in human non-readable format as shown in the Fig. 4.

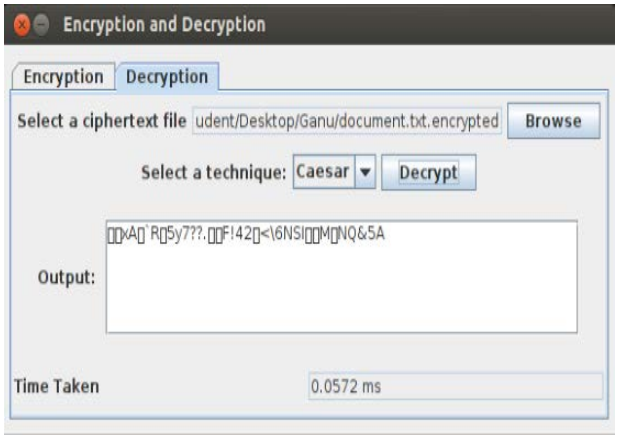

Fig. 4. Attack on Caesar cipher implementation

Decryption time required for receiver to decrypt the Caesar cipher implementation which is altered by the attacker is 0.0572 milliseconds.

4 Result Analysis

In this section, the work presented result graph of encryption and decryption time taken by performing attack on various cryptographic algorithm.

4.1 Result graph for Encryption time

In Fig. 5. The graph shows the encryption time taken for performing attack on AES, DES, RSA and Caesar cipher algorithm.

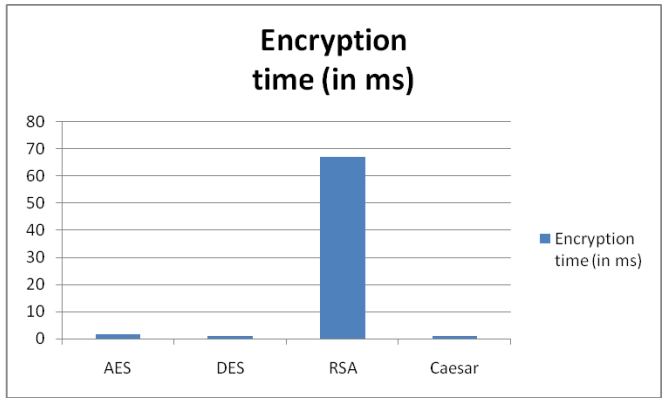

Fig. 5. Encryption time taken for performing attack

4.2 Result graph for Decryption time

In Fig. 6. The graph shows the decryption time taken for performing attack on AES, DES, RSA and Caesar cipher algorithm.

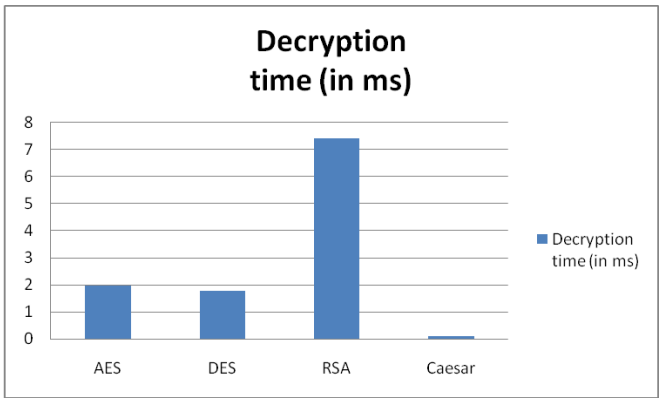

Fig. 6. Decryption time taken for performing attack

Above experimental result shows that RSA is most powerful cryptography algorithm which perform efficient side channel attack on cryptographic environment.

5 Conclusion

Any information on an intermediate value in a cryptographic computation gathered via a side-channel can be exploited in a successful attack when the intermediate value is a relatively simple function of known information and unknown key. In this paper, work we have presented a novel framework for performing side-channel attacks on cryptographic algorithm. Research work executed side channel attack on AES, DES, RSA and Caesar cipher cryptographic algorithms. Also work has tested effectiveness by measuring time complexity as a performance parameter. Future work includes investigating efficient countermeasures against side channel attack.

Acknowledgment

I would like to thank my honourable guide, Prof. Rajesh Bansode. He rendered his valuable guidance with a touch of inspiration and motivation. He guided me through quite a lot of substantial hurdles by giving plenty of early ideas which finally resulted in the present fine work.

References

[1] U. Herath, J. Alawatugoda and R. Ragel, "Software Implementation Level Countermeasures against the Cache Timing Attack on Advanced Encryption Standard," in *IEEE 8th International Conference on Industrial and Information Systems (ICIIS)*, Aug. 2013, Sri Lanka, pp. 18-20.

[2] (Wikipedia, (2011, May). *Advanced Encryption Standard* [Online]. Avalible: http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[3] J. Alawatugoda, R. Ragel and D. Jayasinghe, "Countermeasures Against Bernstein's Remote Cache Timing Attack," in *IEEE 6th International Conference on Industrial and Information Systems (ICIIS)*, Aug. 2011, Kandy, W.-K. Chen, *Linear Networks and Systems*. Belmont, C Sri Lanka, pp. 43 - 48.

[4] D. Jayasinghe, J. Fernando, R. Herath, and R. Ragel, "Remote Cache Timing Attack on Advanced Encryption Standard and Countermeasures," in *IEEE 5th International Conference on Information and Automation for Sustainability (ICIAfS)*, Dec. 2010, Colombo, Sri Lanka, pp. 177-282.

[5] Paul Kocher. "Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems", Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 1996), pp. 104-113, Springer- Verlag London, UK, 1996.

[6] Wikipedia, (2011, May). *Side Channel Attacks* [Online]. Available: http://en.wikipedia.org/wiki/Side_channel_attack.

[7] D. Brumley and D. Boneh. Remote Timing Attacks are Practical, in *USENIX*, August 2003.

[8] J.-J. Quisquater and D. Samyde. Electromagnetic Analysis (EMA): Measures and counter-measures for smart cards. In *E-smart*, pages 200–210, 2001.

[9] S. Mangard. A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion. In ICISC *2002*, volume 2587, pages 343–358.

[10] Dag Arne Osvik Adi Shamir and Eran Tromer Cache attacks an countermeasures: The Case of AES, Lecture Notes in Computer Science, Volume 3860/2006, pages 1-20 December 2005

[11] Daniel J. Bernstein, "Cache Timing Attacks on AES", April 2005.

[12] Edward W. Felten and Michael A. Schneider. "Timing Attacks on Web Privacy", Secure Internet Programming Laboratory, Univ. Princeton, Princeton, NJ 08544 USA

[13] Janaka Alawatugoda, Roshan Ragel and Darshana Jayasinghe; "Countermeasures Against Bernstein's Remote Cache Timing Attack", in Proceedings of the 6th IEEE International Conference on Industrial and Information Systems (ICIIS2011), Kandy, Sri Lanka, August 2011.

[14] J. Daemen and V. Rijmen, AES Proposal: Rijndael (Version 2).

[15] P.C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems," in *16th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO)*, 1996, Springer- Verlag London UK, pp. 104-113.

[16] Z. Xinjie, W. Tao, M. Dong , Z. Yuanyuan, L. Zhaoyang, "Robust First Two Rounds Access Driven Cache Timing Attack on AES," in *IEEE InternationalConference on Computer Science and Software Engineering* , Dec.2008, Wuhan, Hubei, China, pp. 785 – 788.

[17] W. Stallings, Cryptography and network security.

[18] O. Aciic¸mez, W. Schindler, and C.K. Koc¸, "Cache Based Remote Timing Attack on the AES," *CT-RSA, ser. Lecture Notes in Computer Science, M. Abe, Ed.*, vol. 4377, pp. 271-286, 2007.

[19] K.L.Baishnab, A. Nag, FATalukdar, "Cache-Timing Attacks on AES and Remedies," in *IEEE International Conference on Emerging Trends in Electronic and Photonic Devices & Systems (ELECTRO)*,pp. 218-221, Varanasi, Dec. 2009.

[20] J. Kong, O. Aciic¸mez, J. Seifert, and H.Zhou, "Architecting against Software Cache-Based Side-Channel Attacks," *IEEE Transactions on Computers*, vol. 62, no. 7, pp. 1276 - 128, Apr. 2012.