ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 12, December 2016

Key Updating For Leakage Resiliency with Advanced Encryption Standard

M.Revathi Bai[1], B.Sowmya[2]

M.Tech Student, Dept. of CSE, INTELL Engineering College, Affiliated to JNTUA, Andhra Pradesh, India [1]

Assistant Professor, Dept. of CSE, INTELL Engineering College, Affiliated to JNTUA, Andhra Pradesh, India[2]

ABSTRACT: The leakage resilient cryptography is used to overcome SCA attacks, were external details such as power consumption, time taken to generate the key can be acquired by the attacker in order to get the parts of the secret key and then compute the leakage function to aggregate the secret key. This can be overcome using the concept of key updating. Thus key updating makes it impossible for the attacker to guess the exact key that is being generated. This paper is focused on the actual risk of SCA lies within the ability to mount attacks over small components of the important thing and to aggregate understanding over different encryptions. The risk of SCA may also be thwarted by altering the secret key at every run. Certainly, many contributions in the domain of leakage resilient cryptography tried to acquire this intention. However, the proposed options had been computationally intensive and weren't designed to clear up the drawback of the current cryptographic schemes. In this paper, we advise a established framework of lightweight key updating that can preserve the present cryptographic requisites and overview the minimal requisites for heuristic SCA-protection. Then, we endorse a whole approach to safeguard the implementation of any general mode of advanced Encryption standard. Our resolution maintains the same degree of SCA-safety (and commonly better) as the state of the art, at a negligible subject overhead even as doubling the throughput of the first-class earlier work.

KEYWORDS: Side Channel Assaults, Cryptographic schemes, Key Distribution Center, Encryption and Decryption.

I. INTRODUCTION

Side-channel analysis (SCA) is an implementation attack that targets recovering the key of cryptographic modules by monitoring side-channel outputs which include, but are not limited to, electromagnetic radiation, execution time, acoustic waves, photonic emissions and many more. The real threat of SCA is that the adversary (Eve) can mount attacks over small parts of the key, and to aggregate the information leakage over different runs to recover the full secret.

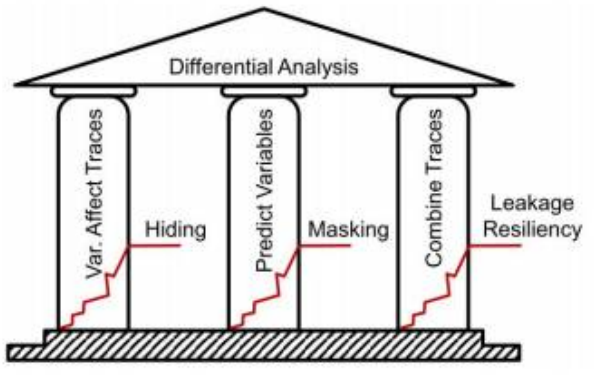

Fig 1 Pillars of SCA attacks.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 12, December 2016

SCA attacks are commonly based on three pillars, as showninFig.1:
1) Sensitive variables affect leakage traces.
2) Eve can calculate hypothetical sensitive variables.
3) She can combine information from different traces.
The design of countermeasures against SCA attacks is a vast research field. Contributions in this regard fall into three categories: Hiding, Masking and Leakage Resiliency.
Hiding depends on breaking the link between intermediate variables and the observable leakage by minimizing the signal-to-noise ratio within the trace. This can be achieved using balanced circuits and/or noise generators. Unfortunately, cryptographic modules with hiding require more than double the area.

Masking depends on breaking Eve's ability to calculate
hypothetical intermediate variables, by splitting the useful information into n shares based on random variable(s). The random variables are generated on-the-fly and discarded afterwards. Each share is processed independently. The final outputs (of each share) are combined to retrieve the original output. Similarly, cryptographic modules supported with masking require more than double the area.
Leakage resiliency depends on using a fresh key for every execution of the cryptographic module hence, prevents aggregating information about any secret. Leakage resiliency is achieved by utilizing a key-updating mechanism (aka re-keying or key-rolling). Although leakage resilient primitives can be implemented using unprotected cores, the overall performance is at least halved.

II. LITERATURE SURVEY

1) Leakage-resilient pseudorandom functions and side-channel attacks on Feistel networks
AUTHORS: Y. Dodis and K. Pietrzak
A cryptographic primitive is leakage-resilient, if it remains secure even if an adversary can learn a bounded amount of arbitrary information about the computation with every invocation. As a consequence, the physical implementation of a leakage-resilient primitive is secure against every side-channel as long as the amount of information leaked per invocation is bounded.
In this paper we prove positive and negative results about the feasibility of constructing leakage-resilient pseudorandom functions and permutations (i.e. block-ciphers). Our results are three fold:
1. We construct (from any standard PRF) a PRF which satisfies a relaxed notion of leakage-resilience where (1) the leakage function is fixed (and not adaptively chosen with each query.) and (2) the computation is split into several steps which leak individually (a "step" will be the invocation of the underlying PRF.)
2. We prove that a Feistel network with a super-logarithmic number of rounds, each instantiated with a leakage-resilient PRF, is a leakage resilient PRP. This reduction also holds for the non-adaptive notion just discussed, we thus get a block-cipher which is leakage-resilient (against non-adaptive leakage).
3. We propose generic side-channel attacks against Feistel networks. The attacks are generic in the sense that they work for any round functions (e.g. uniformly random functions) and only require some simple leakage from the inputs to the round functions. For example we show how to invert anr round Feistel network over 2n bits making $4 \cdot (n+1)^{r-2}$ forward queries, if with each query we are also given as leakage the Hamming weight of the inputs to the r round functions. This complements the result from the previous item showing that a super-constant number of rounds is necessary.

2) Towards fresh re-keying with leakage-resilient PRFs: Cipher design principles and analysis
AUTHORS: S. Belaïd et al
Leakage-resilient cryptography aims at developing new algorithms for which physical security against side-channel attacks can be formally analyzed. Following the work of Dziembowski and Pietrzak at FOCS 2008, several symmetric cryptographic primitives have been investigated in this setting. Most of them can be instantiated with a block cipher as underlying component. Such an approach naturally raises the question whether certain block ciphers are better suited for this purpose. In order to answer this question, we consider a leakage-resilient re-keying function, and evaluate its

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 12, December 2016

security at different abstraction levels. That is, we study possible attacks exploiting specific features of the algorithmic description, hardware architecture and physical implementation of this construction. These evaluations lead to two main outcomes. First, we complement previous works on leakage-resilient cryptography and further specify the conditions under which they actually provide physical security. Second, we take advantage of our analysis to extract new design principles for block ciphers to be used in leakage-resilient primitives. While our investigations focus on side-channel attacks in the first place, we hope these new design principles will trigger the interest of symmetric cryptographers to design new block ciphers combining good properties for secure implementations and security against black box (mathematical) cryptanalysis.

III. EXISTING SYSTEM

The threat considered in this paper is that Eve recovers the secret key of a hardware implementation of AES. Classical cryptography assumes that Eve can choose the input plaintext and the output ciphertext. SCA further assumes that Eve knows the underlying implementation and can capture the instantaneous power consumption. In the domain of leakage resiliency, it is also assumed that Eve can run any polynomial-time function (called leakage function) on the power consumption to recover some bits of the secret key.

The two categories of key-updating are stateless and stateful. One mechanism or the other is sufficient for a limited set of applications. However, the two mechanisms are both required for a complete and generic solution.

Stateless key-updating assumes that the two communicating parties share only the secret key and a public variable (nonce) i.e. there is no shared secret state between them. This updating mechanism is required whenever there is no synchronization between the two communicating parties e.g. during initialization of a secret channel. Stateless key-updating provides a complete solution for applications with single cryptographic execution e.g. challenge response protocols.

Stateful key-updating assumes that the two communicating parties share a common secret state (other than the key). They both can update the secret key into a new key without requiring any external variables. This scheme can provide a complete solution for synchronized applications e.g. key-fobs.

III. PROPOSED WORK

The proposed solution at the system level works as follows. We assume that an application on Device A needs to send secure data to an application on Device B. Both devices share a secret key, which we name master key. They can initiate the channel by exchanging a public nonce, and send the secure data using any cryptographic primitive (AES) running in a mode of operation. Although the black-box security of these modes is guaranteed by the cryptographic primitive, security is not guaranteed if Eve can monitor Device A. Here, we target protecting the master key against any SCA attack. Device A starts with a stateless key-updating mechanism to compute a pseudorandom secret state out of the master key and the nonce. Then, the stateful key-updating is executed, to compute running keys. Finally, the actual cryptographic mode is called using the input data and the same previously used nonce. Our solution honors the tree structure for the stateless key-updating. Each step of the tree involves processing a single bit of the nonce through a lightweight whitening function (Wt: whitening in the tree). The tree starts from the master key, and ends with a pseudorandom secret state. For the stateful key-updating, we use a chain of whitening functions (Wc: whitening in the chain). Every execution of the whitening function generates a new running key.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 12, December 2016

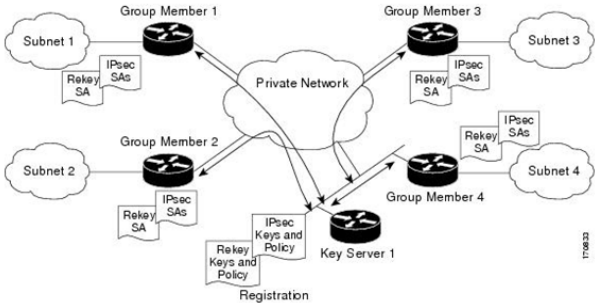

Fig 2: Architecture of system

Implementation:

Module description:

1. Network Setup

This is a versatile peer who needs to get into the information that put away at the storage peer (e.g., an officer). In the event that a peer has an arrangement of characteristics fulfilling the access policy of the encoded information characterized by the sender, and is not disavowed in any of the qualities, then he will have the capacity to decode the ciphertext and acquire the information. The communications between members go through authenticated and public channels and the number of users to be created. Their protocol is based on secret sharing and is considerably efficient, albeit it cannot revoke users.

2. Leakage Resiliency

The challenge arises from the requirement that the set of qualified users can change in each broadcast emission, and therefore revocation of individual users or user groups should be possible using broadcast transmissions, only, and without affecting any remaining users. As efficient revocation is the primary objective of broadcast encryption, solutions are also referred to as revocation schemes

3. Key Generation

Key Generation is the process of creating keys using protection parameters to generate the secret key for the peers. Key generation (KG) is an encryption process in which multiple parties contribute to the calculation of a shared public and private key set. Unlike most public key encryption models, distributed key generation does not rely on Trusted Third Parties. Instead, the participation of a threshold of honest parties determines whether a key pair can be computed successfully

4. Encryption/ Decryption:

The translation of data into a ciphertext is known as encryption. Encryption is the most effective way to prevent data from leakage resiliency. To read an encrypted file, you must have access to a secret key or password that enables you to decrypt it. It is also known as the Cipher text. Encryption is the process of transforming information so it is unintelligible to anyone but the intended recipient. Decryption is the process of transforming encrypted information so that it is intelligible again. A cryptographic algorithm, also called a cipher, is a mathematical function used for encryption or decryption. In most cases, two related functions are employed, one for encryption and the other for decryption. Decryption is the process of taking encoded or encrypted text or other data and converting it back into text that you or the computer can read and understand. This term could be used to describe a method of un-encrypting the data manually or with un-encrypting the data using the proper codes or keys

5. Provable Security

Security proofs give the guarantee that the assumption is enough for secrecy. To prove the security of a cryptographic

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 12, December 2016

scheme, one has to make precise.
• If an adversary can break the secrecy
• One can break the assumption
• The algorithmic assumptions
• The security notions to be guaranteed.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we focused on achieving a sound security at the smallest implementation cost (area and performance). To achieve this goal, a generic framework for lightweight key-updating and evaluate the minimum requirements for SCA-security is developed. Then, we propose a solution that maintains the same level of SCA-security (and sometimes better) as the state of the art, at a negligible area overhead while doubling the throughput of the best previous work.

REFERENCES

[1] K. Tiri et al., "Prototype IC with WDDL and differential routing—DPA resistance assessment," in Cryptographic Hardware and Embedded Systems. Berlin, Germany: Springer-Verlag, 2005, pp. 354–365.
[2] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: A very compact and a threshold implementation of AES," in Advances in Cryptology. Berlin, Germany: Springer-Verlag, 2011, pp. 69–88.
[3] F.-X. Standaert, O. Pereira, Y. Yu, J.-J. Quisquater, M. Yung, and E. Oswald, "Leakage resilient cryptography in practice," in Towards Hardware-Intrinsic Security. Berlin, Germany: Springer-Verlag, 2010, pp. 99–134.
[4] Y. Dodis and K. Pietrzak, "Leakage-resilient pseudorandom functions and side-channel attacks on Feistel networks," in Proc. 30th CRYPTO, 2010, pp. 21–40.
[5] S. Faust, K. Pietrzak, and J. Schipper, "Practical leakage-resilient symmetric cryptography," in Cryptographic Hardware and Embedded Systems. Berlin, Germany: Springer-Verlag, 2012, pp. 213–232.
[6] S. Dziembowski and K. Pietrzak, "Leakage-resilient cryptography," in Proc. IEEE 49th Annu. IEEE Symp. Found. Comput. Sci. (FOCS), Oct. 2008, pp. 293–302.
[7] D. Martin, E. Oswald, and M. Stam, "A leakage resilient MAC," Dept. Comput. Sci., Univ. Bristol, Bristol, U.K., Tech. Rep. 2013/292, 2013. [Online]. Available: http://eprint.iacr.org/
[8] M. Medwed, F.-X. Standaert, J. Großschädl, and F. Regazzoni, "Fresh re-keying: Security against side-channel and fault attacks for low-cost devices," in Progress in Cryptology. Berlin, Germany: Springer-Verlag, 2010, pp. 279–296.