Rekeying Approach against Side Channel Attacks

Tran Song Dat Phuc*, Byoungjin Seok*, Changhoon Lee*†
*Dept. of Computer Science and Engineering, Seoul National University of Science and Technology,
datphuc_89@yahoo.com, sbj7534@gmail.com ,chlee@seoultech.ac.kr

Abstract

Side-channel attacks and in particular differential power analysis (DPA) attacks pose a serious threat to cryptographic implementations. One approach to counteract such attacks is cryptographic schemes based on fresh re-keying. In settings of pre-shared secret keys, such schemes render DPA attacks infeasible by deriving session keys and by ensuring that the attacker cannot collect side-channel leakage on the session key during cryptographic operations with different inputs. This paper present a study on rekeying approach against side channel attacks with current secure schemes and their rekeying functions.

1. Introduction

Passive side-channel attacks and in particular differential power analysis (DPA) pose a serious threat to the security of cryptographic implementations. These attacks allow learning information about the secret key that is processed in a device by observing physical properties, like the power consumption or the electromagnetic (EM) field.

DPA attacks are the most powerful passive side-channel attacks in practice. They accumulate information about a cryptographic key by observing multiple en-/decryptions of different inputs. The fact that different inputs are used allows statistical techniques, like Bayesian distinguishers or correlation techniques, to extract keys very efficiently.

The approach to counteract side-channel attacks is to change cryptographic protocols in such a way that certain types of side-channel attacks cannot be performed at all on the underlying cryptographic primitive. An example of such an approach of inherently preventing DPA attacks is fresh re-keying and leakage-resilient cryptography, which include encryption schemes, message authentication codes and authenticated encryption schemes.

Schemes with inherent protection against DPA attacks require a side-channel secure initialization in order to obtain a fresh session key for every cryptographic operation. This session key is typically derived from a pre-shared master key using a nonce. The purpose of the secure initialization is to ensure that cryptographic operations for different data inputs are always done using different keys. Hence, whenever a party encrypts or authenticates data, a new nonce has to be generated to derive a new session key.

In this paper, we give a study on rekeying approach against side channel attacks in combination with different secure rekeying functions in construction. An approach in rekeying function is also presented to apply to other schemes with more efficient performance.

2. Rekeying against Side Channel Attacks

Rekeying is a countermeasure to DPA that can be seen to work on protocol level. The idea of frequent re-keying is to prevent DPA on the cryptographic primitive by limiting the number of processed inputs per key. In other words, it limits the data complexity for each key by a small number q that renders DPA on the key infeasible. It is nowadays a common assumption that small data complexities have sufficiently small side-channel leakage and do not allow for successful key recovery from DPA attacks.

Rekeying was first proposed for protecting embedded devices such as RFID tags. On the encryption of every new plaintext P, the block cipher E is provided with a new session key K*. This session key K* is derived from a pre-shared master secret K and a nonce N that is randomly generated on the tag. This inherently prevents DPA on the session key K* of the block cipher E. However, for key derivation it requires a re-keying function g: $(K, N) \rightarrow K^*$ that is easy to protect against both SPA and DPA attacks.

One major problem of rekeying schemes is that the reader remains vulnerable to DPA attacks. For instance, such a re-keying scheme can successfully prevent DPA attacks on a device that solely performs encryption or authentication of messages, i.e., the sender of a message, but fails to protect a device performing decryptions or verifications, i.e., the receiver of a message. This is caused by the lack of control of a decryption device on the nonce N and allows attackers to send arbitrary messages to the decryption device using the same nonce N for all sent messages. This malicious procedure results in different messages which being decrypted using the same session key K^*. As a result, decryption is vulnerable to DPA, and more concretely, it is the multiple decryption with the same session key K^* that causes this DPA vulnerability.

In order to prevent this kind of DPA attacks, the receiver either needs to be protected by other means, or the receiver

* "이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임.(No. 2015R1C1A1A02036817)"
† 교신저자, chlee@seoultech.ac.kr(Coresponding author)

needs to be stateful in order to prevent decryption with the same session key twice, or all communication parties are required to contribute to the nonce that is used to derive the session key from a pre-shared master key.

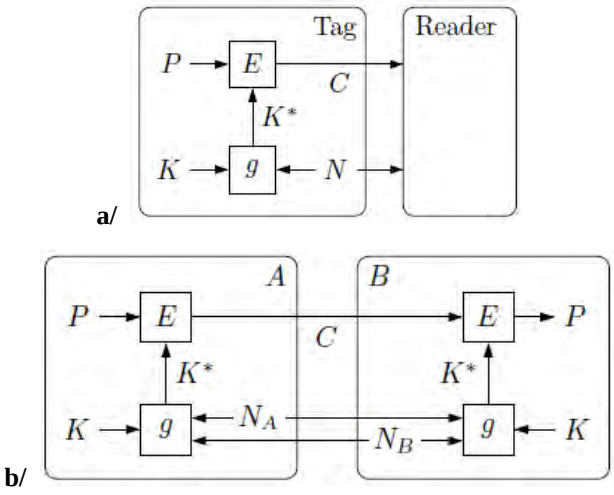

Fig.1. Rekeying scheme with **a/** one party and **b/** two parties.

3. Current Schemes with Secure Rekeying Function

3.1. Basic Rekeying Schemes: Fresh Rekeying

The scheme was proposed at AFRICACRYPT 2010, built from a block cipher BC and a rekeying function g. The rekeying function g(*k*, *r*) to derive new session keys, and the block cipher E(*k**, *m*) to encrypt message blocks. First, the rekeying function produces a session key *k** from the master key *k* and a random nonce r. Second, the plaintext *x* is encrypted by the fresh key *k** with a block cipher. It can easily turn into a hybrid rekeying by using a counter instead of the random nonce *r*.

However, the weakness of this scheme is that it only allows low cost side-channel countermeasures for one of the two communication parties. Another scheme was proposed at CARDIS 2011 which allows cheaper side-channel countermeasures for all parties, suitable for multi-party communication. In this scheme, both communication parties generate the session key *k** with a random nonce *r*.

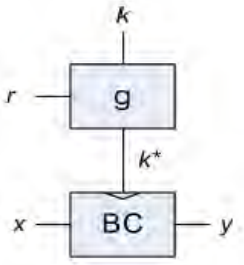

Fig. 2. Fresh rekeying scheme (AFRICACRYPT 2010 [1]).

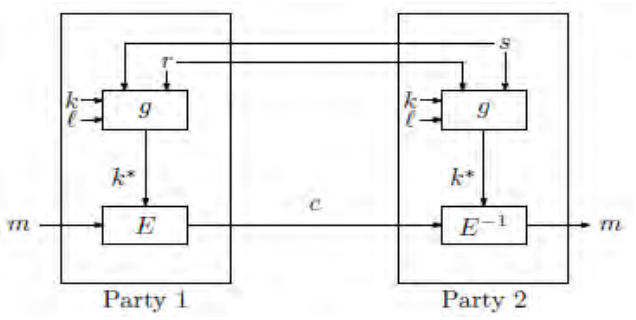

Fig. 3. Rekeying scheme for multi-party (CARDIS 2011 [2]).

3.2 Abdalla-Bellare Rekeying Scheme

This scheme [3] is based on a pseudo-random function (PRF) in combination with a hash function at instantiation step. It proves the security when combination of g with a well-chosen compression function PRF. The function g handles with side-channel protection, since the compression function prevent pre-image and collision attack. It also guarantees that an attacker cannot distinguish the output of F from a random sequence, which implies that he cannot recover the key *k* that generated this output. This construction is provably resistant against the collision-based key recovery attack. It can be seen as an extension of the basic rekeying scheme.

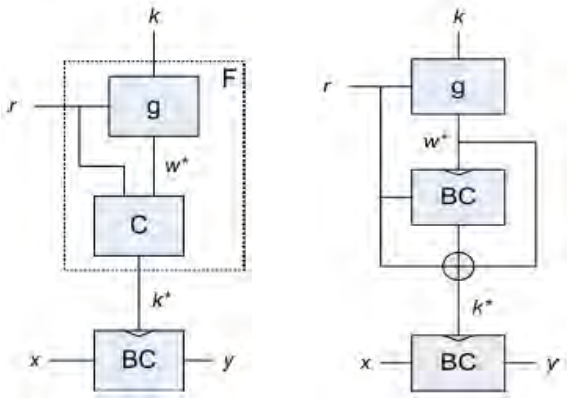

Fig. 4. Abdalla-Bellare rekeying scheme.

However, because of the additional compression function and block cipher, it leads to a large performance overhead for a single encryption. And, implementation is also more expensive than original fresh rekeying scheme.

3.3 Kocher's Rekeying Scheme

Kocher's rekeying scheme [5] proposed a different way to produce session key. The session key is not derived from a static secret master key and a random nonce. Instead, it uses a tree structure concept to update and assign the secret key as session key. The number of usable session keys k^*_i is determined by the depth of the tree.

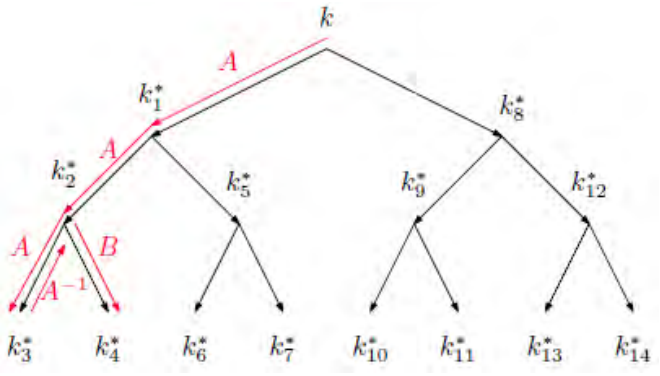

Fig. 5. Kocher's rekeying scheme.

The root of the tree is the secret master key k and the other vertices represent session keys k^*_i. To traverse through the tree, the functions A, B, A^{-1}, and B^{-1} are used, where A^{-1}, and B^{-1} are the inverse functions of A, and B. For instance, $k^*_1 = A(k)$, and $k^*_8 = B(k)$.

4. Rekeying Scheme with Block Cipher

This rekeying approach uses a rekeying function in combination with a tweakable block cipher to produce session key. Similarly, the rekeying function g is regarding to side-channel resistance, while the tweakable block prevents the collision attack.

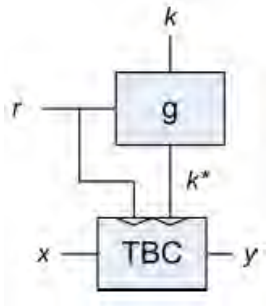

Fig. 6. Rekeying scheme with tweakable block.

In this scheme, we get always a different session key k^* for different nonce r. For every different value of the tweak, we have different and independent block cipher instances. So, basically, we just use different block ciphers with different keys, and none of them is used with multiple keys, which makes the collision attack impossible to apply. It increases the size of the list, since an attacker trying to perform the first step of the attack and he need to pre-calculating a list for a set of pairs $(r_i; k^*_i)$, not for a set of k^*_i.

5. Conclusion

In this paper, we present a study on rekeying approach against side channel attacks with an overview of current secure schemes with their rekeying function. A combination between rekeying function with tweakable block in rekeying construction is also considered with more efficient performance and security results in side channel attacks as well as collision attack. It is expected to be applied to more other rekeying schemes for a better implementation and analysis results.

References

[1] Medwed, M., Standaert, F.X., Großschädl, J., Regazzoni, F.; "Fresh Re-keying: Security against Side-Channel and Fault Attacks for Low-Cost Devices". In: Bernstein, D.J., Lange, T. (eds.) AFRICACRYPT, LNCS, vol. 6055, pp. 279-296, Springer (2010).

[2] Medwed, M., Petit, C., Regazzoni, F., Renauld, M., Standaert, F.; "Fresh re-keying II: Securing Multiple Parties against Side-Channel and Fault Attacks". In: Prouff, E. (ed.) CARDIS 2011, LNCS, vol. 7079, pp. 115-132, Springer (2011).

[3] Abdalla, M., Bellare, M.; "Increasing the Lifetime of a Key: A Comparative Analysis of the Security of Re-keying Techniques. In: Okamoto, T. (ed.) ASIACRYPT 2000, LNCS, vol. 1976, pp. 546-559, Springer (2000).

[4] Stefan Dziembowski, Sebastian Faust, Gottfried Herold, Anthony Journault, Daniel Masny, and François-Xavier Standaert; "Towards Sound Fresh Re-keying with Hard (Physical) Learning Problems. In Matthew Robshaw and Jonathan Katz, editors, CRYPTO 2016, vol. 9815 of LNCS, pages 272–301, Springer **(**2016).

[5] Kocher, P.C.: Leak-Resistant Cryptographic Indexed Key Update, US Patent 6,539,092, (2003).

[6] Belaid, S., Grosso, V., Standaert, F.: "Masking and leakage-resilient primitives: One, the other(s) or both?", Cryptography and Communications 7(1), 163{184 (2015).