Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Public key encryption resilient to leakage and tampering attacks

Shi-Feng Sun [a,b], Dawu Gu [a,*], Udaya Parampalli [b], Yu Yu [a], Baodong Qin [c]

[a] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*
[b] *Department of Computing and Information Systems, The University of Melbourne, Victoria 3010, Australia*
[c] *School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 250100, China*

ARTICLE INFO

Article history:
Received 14 November 2015
Received in revised form 13 September 2016
Accepted 1 March 2017
Available online xxxx

Keywords:
Public key encryption
Hash proof system
Chosen-ciphertext security
Key-leakage attack
Tampering attack

ABSTRACT

In this work, we investigate how to protect public key encryption from both key-leakage attacks and tampering attacks. First, we formalize the notions of chosen ciphertext (CCA) security against key-leakage and tampering attacks. To this goal, we then introduce the concept of key-homomorphic hash proof systems and present a generic construction of public key encryption based on this new primitive. Our construction, compared with previous works, realizes leakage-resilience and tampering-resilience simultaneously but completely independently, so it can tolerate a larger amount of bounded-memory leakage and be instantiated with more flexibility. Moreover, it allows for an unbounded number of affine-tampering queries, even after the challenge phase. With slight adaptations, our construction also achieves CCA security against subexponentially hard auxiliary-input leakage attacks and a polynomial of affine-tampering attacks. Thus, to the best of our knowledge, we get the first public key encryption scheme secure against both auxiliary-input leakage attacks and tampering attacks.

1. Introduction

A crucial ingredient for a successful provable security is a correct definition of its security model. If the security model fails to capture the power of all real potential attacks, a cryptographic system proven secure in such a model may still be vulnerable in practice. So ultimate objective of cryptography ought to be presenting efficient cryptographic systems that could be proven secure against potential attacks whose size is as large as possible. Along this line, much progress has been made recently for providing stronger security guarantees in practice [1–7].

Leakage-resilient cryptography [1,2] was initiated by the cryptographic community with the increasing popularity of real-world physical attacks [8–10]. The goal of this new framework is to construct various cryptographic primitives that can be proven secure against such adversaries that can learn partial information about secret states by observing physical characteristics of executions of a cryptographic device, such as timing, power consumption, etc.

In general, leakage-resilience of cryptographic primitives is formulated by requiring the original security notion hold even if the attackers are provided with an additional oracle called leakage-oracle. In such an oracle, the attackers are allowed to specify an arbitrary and efficient leakage function f and get the result of applying f to the secret state. Obviously, the

[*] Corresponding author.
E-mail addresses: crypto99@sjtu.edu.cn (S.-F. Sun), dwgu@sjtu.edu.cn (D. Gu), udaya@unimelb.edu.au (U. Parampalli), yuyu@cs.sjtu.edu.cn (Y. Yu), baodong.qin@gmail.com (B. Qin).

http://dx.doi.org/10.1016/j.jcss.2017.03.004

leakage function f should be restricted so that it is infeasible for the attackers to obtain the entire secret key from the leaked information, as otherwise no cryptographic primitives could be proven secure.

Motivated by cold-boot attacks presented by Halderman et al. [10], an important line of work on *memory leakage model* was initiated by Akavia et al. [3] and Naor et al. [4]. In contrast to the initially introduced "only computation leaks information" model [1,2], this is a relatively more general model, where the attacker is allowed to learn information about the whole memory state. In terms of the constrains on the leakage function f, the memory leakage model can be generally sorted into: (1) *bounded-memory leakage model* [3], where f is an arbitrary and efficiently computable function with output length $\lambda < |sk|$; (2) *auxiliary-input leakage model* [11], where f is stipulated to be sufficiently hard to invert for any efficient algorithm but with no bound on its output length. In fact, the latter further generalizes the former, and it replaces the *information-theoretical restriction* on leakage with a *computational restriction*, which greatly enlarges the set of leakage functions and hence models a larger class of side-channel attacks.

Memory leakage model, as a general and important leakage model, has been well-studied and a large number of works have been presented in this model so far, such as [3,4,11–23], which includes public key encryption schemes, signatures, identity-based encryptions, etc.

However, leakage models only capture the power of *passive physical attacks* (i.e., leakage attacks) in practice. Recent research has shown that the attacker may also be able to obtain secret information through launching *active physical attacks* (i.e., tampering attacks) such as heating, electromagnetic radiation and fault injection attacks [24–26]. In such scenarios, the attackers usually can tamper with the secret state and learn additional information about the secret state by observing the output of a cryptographic execution on a tampered/transformed secret state.

Similar to leakage-resilient cryptography, tampering attacks are captured by a class of efficient functions $\mathcal{T} = \{f_t : \mathcal{SK} \to \mathcal{SK}\}$ named tampering functions, and tampering-resilience is formulated by requiring a cryptographic primitive meet its original security notion even for the attackers having access to a tampered-functionality oracle. For example, in a signature scheme Sig with signing algorithm $\mathsf{Sign}(sk, \cdot)$ the tampered-signing oracle allows the attacker to ask for signing queries of the form (f_t, m) and get the corresponding signature $\mathsf{Sign}(f_t(sk), m)$ with respect to (w.r.t.) the tampered signing key $f_t(sk)$.

The security under such tampering attacks was initially formalized by Bellare et al. [5] in the context of symmetric cryptographic primitives. Following this work, there is a vast body of literature that considers tampering attacks for various cryptographic primitives [6,7,27–31]. One concerned line of research work is to enlarge the family $\mathcal{T}$ of tampering functions and hence to encompass a larger class of tampering attacks.

In view of the fact that physical attacks in practice are not limited to passive attacks but also include active attacks, in this work we focus on designing cryptographic systems that could be simultaneously resilient to leakage attacks and tampering attacks. Thus, stronger security guarantees could be provided for the deployments of cryptographic primitives in practice, e.g., in smart cards or sensors of IoT where the stored decryption keys may suffer from various physical side-channel attacks.

1.1. Related work

It is not easy to construct leakage-resilient or tampering-resilient cryptosystems especially for a large class of leakage functions or tampering functions, let alone design cryptographic algorithms resilient to both leakage and tampering attacks. To our best knowledge, there are few works considering how to achieve leakage and tampering resilience at the same time.

In 2011, Kalai et al. [32] initially took into account both leakage and tampering attacks, and presented the first feasibility results in the so-called continuous leakage and tampering (CLT) model. In this model, the attacker is permitted to continuously ask for both leakage queries and tampering queries. Thus their construction achieved a very strong security guarantee, although they didn't consider chosen-ciphertext (CCA) security. As indicated in [28], however, their scheme is rather inefficient and relies on non-standard assumptions.

Later, with the goal of enlarging the class of tampering functions, Damgård et al. [28] connected the leakage realm with tampering realm and managed to achieve tampering-resilience through leakage-resilience. Particularly, they introduced the notion of bounded leakage and tampering (BLT) security, where the attacker is only allowed to ask for a bounded number of tampering queries. Under this notion, they also presented a BLT-resilient CCA secure public key encryption scheme on the basis of the leakage-resilience of BHHO-construction [33]. Through this novel approach, their proposed scheme could achieve tampering-resilience against arbitrary key-relations, but at the cost of largely weakening its leakage-resilience. In addition, the approach leads to the dependence of tampering-resilience on leakage-resilience and thus requires the underling encryption scheme to tolerate a larger amount of leakage than the public key size, which greatly hinders the instantiation of the BLT security.

Another relevant line of work is (leakage-resilient) non-malleable codes [34–37,31] which may provide a generic method for protecting against physical attacks. Recently, Liu et al. [38] investigated how to generally protect cryptographic devices from continuous leakage and tampering attacks, and put forward a general cryptographic functionality compiler based on leakage-resilient non-malleable code and robust non-interactive zero knowledge proof system. While these methods could offer surprisingly strong security guarantees, they all require certain hardware assumptions such as the split-state model, where the memory has to be split into at least two compartments and the attacker is only allowed to leak information about the memory compartments or tamper with them separately.

Please cite this article in press as: S.-F. Sun et al., Public key encryption resilient to leakage and tampering attacks, J. Comput. Syst. Sci. (2017), http://dx.doi.org/10.1016/j.jcss.2017.03.004