iCryptoTracer: Dynamic Analysis on Misuse of Cryptography Functions in iOS Applications

Yong Li, Yuanyuan Zhang, Juanru Li, and Dawu Gu

Dept. of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, China
yyjess@sjtu.edu.cn

Abstract. Cryptography is the common means to achieve strong data protection in mobile applications. However, cryptographic misuse is becoming one of the most common issues in development. Attackers usually make use of those flaws in implementation such as non-random key/IV to forge exploits and recover the valuable secrets. For the application developers who may lack knowledge of cryptography, it is urgent to provide an efficient and effective approach to assess whether the application can fulfill the security goal by the use of cryptographic functions. In this work, we design a cryptography diagnosis system *iCryptoTracer*. Combined with static and dynamic analyses, it traces the iOS application's usage of cryptographic APIs, extracts the trace log and judges whether the application complies with the generic cryptographic rules along with real-world implementation concerns. We test *iCryptoTracer* using real devices with various version of iOS. We diagnose 98 applications from Apple App Store and find that 64 of which contain various degrees of security flaws caused by cryptographic misuse. To provide the proof-of-concept, we launch ethical attacks on two applications respectively. The encrypted secret information can be easily revealed and the encryption keys can also be restored.

1 Introduction

Mobile devices such as smartphones and tablet computers are becoming the vessel of personal information such as contact list, physical location, social information and even banking service, online payment. As the popularity of such devices grows, the malicious software have the increasing impact on personal privacy. Current mobile OSes(mainly Android and iOS) use layered security strategies to endow the dominance to the end-users to control the access of the sensitive data. Aiming at providing high security assurance, iOS is designed with various security features. At system level, full-disk encryption, ASLR [1], sandboxing profile and privilege assignment are adopted to fulfil access control policy [2]. At applications level, the Apple App Store scrutinizing on the applications also reduces the risk of malicious behaviors in the apps as a beneficial supplement. Besides for those built-in security features, third-party iOS developers resort

M.H. Au et al. (Eds.): NSS 2014, LNCS 8792, pp. 349–362, 2014.

to modern cryptographic algorithms to provide stronger protection on sensitive data.

It is possible that the emphasis on cryptographic techniques for protecting information mitigates the attention to the issue of cryptographic usage. The security of the primitives are provided by intellectual properties or industrial standards. There is a tendency to focus on problems that are mathematically interesting to the exclusion of implementing problems which must be solved in order to actually increase operational security. We've seen lots of security applications contradicting to some basic cryptography applying rules caused by developer's ignorance of general cryptographic usage guidelines, or sometimes the ambiguous documentation misleading to defective implementations. Both facts could result in software vulnerability or privacy leaks.

The well-known *Citibank* iOS application [3] and *Starbucks* application [4], for instance, storing the customer's privacy information such as payment passcode, bank account number, etc. The Verge has reported that Starbucks' iPhone application stores user passwords in plaintext. By connecting iOS device through iTunes, an attacker can easily retrieve the password and payment records. Therefore, it's crucial to evaluate the correctness of cryptographic usage inside the emerging third-party iOS applications.

As a contrast to the open-source Android system, iOS is a proprietary operating system and is relatively close. Developing a third-party security analysis extension for iOS system requires essential work and is difficult for lacking details of the operating system. Recent studies on iOS application mainly apply static analysis to detect security vulnerability such as privacy leak. Egele et al. proposed PiOS [7] based on static analysis using a control-flow graph to identify from where the sensitive data leaks. However, static analysis tends to be less accurate due to the dynamic messaging mechanism of iOS applications, which are primarily developed with Objective-C. Most iOS applications are heavily based on event-driven schemes. Simply analyzing an application with static analysis is not feasible because the dynamic events can not be predicted, the inputs can not be constructed either, parameters may be generated while executing, and the return value is unforeseeable. Such dynamic characteristic determines that many information can only be monitored accurately at runtime and in this situation dynamic analysis would be a better choice.

Dynamic analysis of iOS applications is facing lots of challenges. One challenge is that encryption is input-related, so that some data should be provided. iOS Applications are GUI-rich, and most of input areas are of *UITextField* component, and sometimes files should be provided as input, so manual work is inevitable during test. To study iOS kernel and Objective-C runtime to dynamically observe the application running in iOS, we have to resort to instrumentation and API hooking techniques. To the best of our knowledge, no previous dynamic analysis on cryptographic usage has been proposed on mobile system so far.

An approach to diagnose the implementation code to assure the proper validity of cryptographic usage is in demand. We present *iCryptoTracer* to fulfill such purpose. As a cryptographic usage vetting system (we use *crypto-vetting*