Cryptology ePrint Archive: Report 2011/675

Basing Obfuscation on Simple Tamper-Proof Hardware Assumptions

Nico Döttling and Thilo Mie and Jörn Müller-Quade and Tobias Nilges

Abstract: Code obfuscation is one of the most powerful concepts in cryptography. It could yield functional encryption, digital rights managemen and maybe even secure cloud computing. However, general code obfuscation has been proven impossible and the research then focused on obfuscating very specific functions, studying weaker security definitions for obfuscation, and using tamper-proof hardware tokens to achieve gen code obfuscation. Following this last line this work presents the first scheme which bases general code obfuscation of multiple programs on one s stateless hardware token. Our construction is proven secure in the UC-framework and proceeds in three steps: 1. We construct an obfuscation scheme based on fully homomorphic encryption (FHE) and a hybrid functionality conditional decrypt, which decrypts the result of a homomorph computation given a proof that the computation was performed as intended. One difficulty of the first step are possible decryptions errors in the FHE. These decryption errors can occur whenever the randomness for the encryption is chosen maliciously by the receiver of the obfuscated co Such decryption errors then could make a real obfuscated computation distinguishable from a black box use of the non-obfuscated program. 2. Given two common reference strings (CRS) we construct a UC-protocol realizing the functionality conditional decrypt with a stateless hardware token. As the token is stateless it is resettable by a dishonest receiver and the proofs given to the token must be resettably sound. One additiona difficulty occurs when the issuer of the token can be corrupted. A malicious token can be stateful and it cannot be prevented that it aborts after a hardwired number of invocations. To prevent adaptive behavior of a malicious token the data of the receiver has to be hidden from the token an proofs given to the token must even hide the size of the program and the length of the computation. 3. Last we construct a protocol constructing CRS with a stateless hardware token. Care has to be taken here to not let the token learn anything about the resulting CRS which could not be simulated, because the very same token will later be used in a protocol based on the security of this CRS.

Category / Keywords: Obfuscation, Stateless Tamper-Proof hardware, Universal Composability, Universal Arguments, Fully-Homomorphic-Encryption

Date: received 13 Dec 2011, last revised 16 Jan 2012

Contact author: nico doettling at kit edu

Available formats: PDF | BibTeX Citation

Version: 20120116:101326 (All versions of this report)

Discussion forum: Show discussion | Start new discussion

[Cryptology ePrint archive]