Hiding Information Hiding

Adam Young[1] and Moti Yung[2]

[1] Cryptovirology Labs
aly@cryptovirology.com
[2] RSA Labs and Columbia University
moti@cs.columbia.edu

Abstract. In this paper we introduce a new tool that hides whether or not an "encryption" algorithm actually performs encryption or not. We call this a computational questionable encryption scheme and show how it can be used to devise mobile agents that conceal whether they encrypt or delete data prior to data transmission. Such agents may be useful in the honest-but-curious setting in which the author of the agent wishes to keep confidential whether or not the agent collects and transmits data while in transit. Informally, a questionable encryption scheme adds a "fake" key generation algorithm to a PKCS. The key generation algorithms of a computational questionable encryption scheme produce a "public key" y and a poly-sized witness x. Depending on which of the two key generation algorithms the user decides to use, y is real or fake. When the cipher is supplied with a real y then it produces decipherable ciphertexts and x proves this. When the cipher is supplied with a fake y then it produces *computationally* indecipherable ciphertexts (with respect to *everyone*) and x proves this. We call the former a witness of encryption and the latter a witness of non-encryption. We formally define the notion of a computational questionable encryption scheme and present a construction for it based on the ElGamal cryptosystem. We prove the security based on the Decision Diffie-Hellman problem and a reasonable new intractability assumption in the random oracle model. Finally, we show how a computational questionable encryption scheme is related yet different from all-or-nothing disclosure of secrets and related notions.

1 Introduction

The theory of information hiding is broad in scope and encompasses everything from steganography, to subliminal channels in cryptosystems, to covert channels in operating systems. In this paper we expand the scope of information hiding even further by considering how to hide the true nature of a particular class of functions that execute in the *honest-but-curious* model. In this model, functions are executed by an agent (e.g., mobile agent) in an environment that can be trusted not to interfere with the operation of the function (i.e., trusted not to introduce faults) but that is curious and may seek to log and analyze the agent as it executes.

J. Camenisch et al. (Eds.): IH 2006, LNCS 4437, pp. 161–171, 2007.

In particular we present a new scheme that hides whether or not a function (that appears to be an asymmetric encryption function) actually encrypts plaintext or effectively deletes the plaintext. This is called a *questionable encryption* scheme and it can be used to make mobile agents more robust in the aforementioned threat model.

To motivate the introduction of this scheme, consider the following scenario. A mobile agent is found that passes a value that appears to be a public key to an asymmetric encryption function (e.g., in an OS API call). It also passes plaintext that is taken from the host system to the encryption function. The agent transmits the resulting ciphertext outside the host system. Without understanding the subtleties of public key cryptography it may be easy to jump to the conclusion that encryption is taking place and hence that sensitive information is being sent outside of the host computer system.

This assumption is inherently flawed since in some cases the requisite algebraic structure of the public key may be incorrect, or perhaps (e.g., in ElGamal [9]) the public key was sampled randomly without knowing the pre-image. An improperly generated public key can effectively *erase* plaintext data rather than encrypt it. This is one of the properties that a questionable encryption scheme provides.

Questionable encryptions enable a two pronged application. The user deploys numerous mobile agents, each with a unique "public key." Some of the agents contain a real public key and the rest contain a fake public key. Only the agents with the real public keys will transmit data that is gathered from the host system. The rest will effectively *delete* the plaintext prior to transmission although they will appear to asymmetrically encrypt data (deletion occurs since decryption is provably intractable in a *computational* questionable encryption scheme). The user later reveals witnesses of non-encryption at his or her discretion.

This application ensures that no particular agent (that has not had it's witness revealed) can be known for certain to actually transmit data outside the host. We argue that this provides a useful level of robustness in the honest-but-curious threat model for agents that collect and transmit data.

The contributions of this paper are the following:

1. We provide the first formal and complete definition of a *computational questionable encryption scheme*.
2. A construction is given based on ElGamal and we prove that it is secure based on the Decision Diffie-Hellman problem and a reasonable intractability assumption.
3. We show how questionable encryptions differ from all-or-nothing disclosure, $(1, 2)$-oblivious transfer, and deniable encryptions (we relate questionable encryptions to these primitives in Appendix A).
4. We describe an application of computational questionable encryptions that helps hide the true functionality of mobile agents that collect host data.

We implemented our computational questionable encryption scheme and describe how we did so in Section 6. We show that the portion of the implementation that needs to reside in the agent is *trivial* to implement using Windows Cryptographic API calls (built-in DLL calls).