Questionable Encryption and Its Applications

Adam Young[1] and Moti Yung[2]

[1] LECG LLC*
ayoung@mitre.org
[2] RSA Labs and Columbia University
moti@cs.columbia.edu

Abstract. In this paper we investigate a primitive called a questionable encryption that is related to oblivious transfer. We consider a mobile agent that asymmetrically encrypts plaintext data from the host machine that it resides on and then broadcasts the resulting ciphertext so that it can be obtained by the creator of the agent. We formally define the notion of a *questionable encryption* scheme that can be used to perform this operation. The user of a questionable encryption scheme chooses to generate a real or fake public key. The choice is conveyed to the key generation algorithm which then outputs a poly-sized witness and either a real or fake key pair. If the public key is 'real' then it produces decipherable encryptions and the poly-sized witness proves this. If the key is generated to be 'fake' then it produces indecipherable encryptions (even with the private key) and the poly-sized witness proves this. Without knowledge of the witness it is intractable to distinguish between the two types of public keys. We present a construction for a questionable encryption scheme based on the Paillier cryptosystem. We prove the security of the scheme based on the difficulty of deciding n^{th} degree composite residuosity. When applied to this application, the creator of the agent retains the exclusive ability to reveal whether or not the agent in fact transmits plaintexts. Our results show that agents that appear to compute asymmetric encryptions may in fact not (in a provable sense). We present other applications of questionable encryptions as well.

Keywords: Public key cryptosystem, Paillier cryptosystem, composite residuosity problem, decision composite residuosity problem, semantic security, questionable encryption, deniable encryption, oblivious transfer.

1 Introduction

Mobile agents have been an active area of research and in this paper we investigate a new tool that can be used to enhance the privacy of such agents. Typically, one of two threat models are used when designing modible agents. The first is the *honest but curious* model in which the host machines that the agent traverses are honest enough not to interfere with the operation of the agent, but

* Author is now at MITRE Corporation.

E. Dawson and S. Vaudenay (Eds.): Mycrypt 2005, LNCS 3715, pp. 210–221, 2005.

are "curious" about the data it contains and the results of the computations of the agent. The other threat model allows the hosts to be active adversaries that may introduce faults into the computations of the mobile agent.

In this paper we operate under the former threat model and address the following issue. *Does a mobile agent that appears to asymmetrically encrypt data really do so?* One can envision a scenario in which the system operator of a host jumps to the conclusion that the agent encrypts data since it passes a value that appears to be a public key to an asymmetric encryption function.

By dissecting the agent it can be determined that it uses a correct asymmetric cipher implementation. The question then becomes whether or not the public key is properly formed and whether or not the creator of the agent is in possession of the corresponding private key. The creator is not likely to include non-interactive zero-knowledge proofs in the agent for the benefit of proving that the requisite algebraic properties of the public key hold. In this paper we observe that for a variety of public key cryptosystems, the "public key" cannot be immediately construed as such. This has immediate consequences for proving whether or not the agent even transmits host data.

The integrity of public keys gives rise to the following cryptographic problem. Can we devise a plug-in for a well known asymmetric cryptosystem that accepts normal-looking public keys but that produces ciphertexts that provably cannot be decrypted by *anyone*? The answer to this is yes. We formally define a *questionable encryption scheme* that accomplishes this and present an instantiation (plug-in) for the Paillier cryptosystem.

The key generation algorithm of a questionable encryption scheme generates a poly-sized witness and either a real or fake key pair. The user chooses whether to create a real key pair or a fake key pair. If the public key is 'real' then it produces decipherable asymmetric encryptions and the witness proves their decipherability. If the key is 'fake' then it produces indecipherable encryptions and the witness proves that no such ciphertext can be deciphered. Without knowledge of the witness it is intractable to distinguish between the two types of 'public keys.' We call these witnesses of encryption and non-encryption, respectively.

When a mobile agent outputs a questionable encryption it is intractable to determine whether it outputs a valid asymmetric ciphertext or a value that to all intents and purposes is random. This holds even when the actions of the agent are recorded immediately after deployment, and even if the agent is *reverse-engineered* by the system administrator of a host.

We present applications of questionable encryptions in Section 7. Questionable encryptions are related to $(1, 2)$-oblivious transfer, all-or-nothing disclosure of secrets, and deniable encryptions. Differences between questionable encryptions and these primitives are given in Appendix A.

2 Questionable Encryptions

In this section we cover basic notation and definitions. Let PTM denote a probabilistic poly-time Turing machine. We let $a \mid b$ denote that integer b is evenly