Course Project

Robert Hildebrand
Cryptography - CS 227 - Professor Phillip Rogaway
March 13, 2012

Abstract

We will briefly describe the contents of the paper *Encryption Schemes Secure under Selective Opening Attack* by Mihir Bellare and Scott Yilek. The purpose of this summary is to highlight key points of the paper and put them into a language slightly more familiar and quickly readable for someone in our course.

1 Introduction

Consider a scenario where n sources are sending messages to a single receiver. We wish to have an encryption scheme that is secure even when an adversary has compromised a proper subset I of the senders. We call this type of security *Selective Opening Attack* (SOA) security. Surprisingly, no encryption schemes are known to be SOA secure. We will define SOA security in an indistinguishability sense and give an encryption scheme that is secure in this sense under the DDH assumption. The authors also describe a notion a SOA semantic security and show their encryption schemes are secure in this sense as well.

We first describe formally our notion of SOA security in the indistinguishability sense. We discuss *lossy encryption schemes* and give a simple example based on the DDH assumption. We then show that lossy encryption schemes are SOA secure. This shows that SOA security arises fairly easily from pre-existing encryption schemes.

2 Selective Opening Attach (SOA) Security

Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme, $n : \mathbb{Z}_+ \to \mathbb{Z}_+$ and $t : \mathbb{Z}_+ \to \mathbb{Z}_+$ be integer functions, $\mathcal{M}$ be a n-message sampler. We call (n, t) a valid SOA parameter pair if $t(\lambda) \leq n(\lambda)$ for all $\lambda \in \mathbb{Z}_+$. The adversary is allowed one call to an oracle **Corrupt**, described below. The advantage of ind-so-enc adversary A with respect to $\mathcal{M}, n, t$ is given by making the adversary play the game INDSO if the adversary wins, the game returns true, and if not, the game returns false. The advantage is thus given by

$$\mathbf{Adv}^{\text{ind-so-enc}}_{A,\Pi,\mathcal{M},n,t}(\lambda) = 2 \cdot Pr[\text{INDSO}^{A}_{\Pi,\mathcal{M},n,t}(\lambda) \Rightarrow \text{true}] - 1.$$

We describe the game here in words, as it may be easier to comprehend at first glance. The game begins by initializing keys $(pk, sk) \overset{\$}{\leftarrow} \mathcal{K}(1^\lambda)$ and a vector of messages $\mathbf{m}_0 \overset{\$}{\leftarrow} \mathcal{M}(1^\lambda)$. For each $i = 1, \ldots, n$, we store all random coins $\mathbf{r}[i]$ and the ciphertext $\mathbf{c}[i] = \mathcal{E}(pk, \mathbf{m}_0[i]; \mathbf{r}[i])$. The adversary A is given $(pk, \mathbf{c})$ and also the encryption algorithm $\mathcal{E}$.
Now the adversary is thrown into one of two worlds.

1. The adversary A choses a set $I \subset \{1, \ldots, n\}$ with $|I| = t(\lambda)$ and calls $\mathbf{Corrupt}(I)$, which returns $(\mathbf{r}[I], \mathbf{m}_0)$
2. The adversary A choses a set $I \subset \{1, \ldots, n\}$ with $|I| = t(\lambda)$and calls $\mathbf{Corrupt}(I)$, which returns $(\mathbf{r}[I], \mathbf{m}_1)$ where $\mathbf{m}_1[I] = \mathbf{m}_0[I]$, and $\mathbf{m}_1[\{1, \ldots, n\} \setminus I] \overset{\$}{\leftarrow} \mathcal{M}$.

If the adversary correctly distinguishes which world the adversary is in, the game returns true. Otherwise, the game returns false.
Note that when the adversary A is given the coins $\mathbf{r}[I]$ from **Corrupt**, the adversary can compute $\mathcal{E}(pk, \mathbf{m}[I]; \mathbf{r}[I]) = \mathbf{c}[I]$, that is, the adversary has the power to check if the messages given by corrupt actually correspond to the ciphertexts as promised. Handling the revealing of the internal coins has been the crux in the past for creating SOA secure encryption schemes.

3 Lossy Encryption

A lossy encryption scheme $\Pi = (\mathcal{K}, \mathcal{K}_\ell, \mathcal{E}, \mathcal{D})$ is a tuple of polynomial time algorithms similar to a regular encryption scheme. The parameters $\mathcal{K}, \mathcal{E}, \mathcal{D}$, act as usual. The key generation algorithm $\mathcal{K}$ takes in the security parameter 1^λ and outputs a pair of keys (pk, sk). The scheme $\mathcal{E}$ takes pk and a message m and outputs a ciphertext c, and the decryption $\mathcal{D}$ takes sk and c and correctly outputs m (or $\perp$ if c is not a valid ciphertext). The lossy key generation algorithm $\mathcal{K}_\ell$ takes 1^λ as an input and outputs *lossy public and secret keys* (pk, sk) from which dummy ciphertexts can be created. The decryption $\mathcal{D}$ is not required to properly decrypt when given lossy keys.
A lossy encryption scheme Π must also have good advantage in the following two senses.

1. *Indistinguishability of real keys from lossy keys.* No polynomial-time adversary can distinguish between the first outputs of $\mathcal{K}$ and $\mathcal{K}_\ell$. We write this security as

$$\mathbf{Adv}^{\text{los-key}}_{A,\Pi}(\lambda) = Pr[k \overset{\$}{\leftarrow} \mathcal{K} : A^k \Rightarrow 1] - Pr[k \overset{\$}{\leftarrow} \mathcal{K}_\ell : A^k \Rightarrow 1].$$

2. *Lossiness of encryption with lossy keys.* For any $(pk, sk) \overset{\$}{\leftarrow} \mathcal{K}_\ell$ and two distinct messages m_0, m_1, the encryptions $\mathcal{E}(pk, m_0)$ and $\mathcal{E}(pk, m_1)$ must be *statistically close*, by which we mean $|Pr[\mathcal{E}(pk, m_0) = c] - Pr[\mathcal{E}(pk, m_1) = c]| < \delta$ for some small value δ. The los-ind advantage of A is then

$$\mathbf{Adv}^{\text{los-ind}}_{A,\Pi}(\lambda) = Pr[c \leftarrow \mathcal{E}(pk, m_0) : A^c \Rightarrow 1] - Pr[c \leftarrow \mathcal{E}(pk, m_1) : A^c \Rightarrow 1].$$

We will further assume that there exists a (possibly unbounded) algorithm **Opener** that given a cipher text $c = \mathcal{E}(pk_\ell, m)$ for some lossy public key pk_ℓ and a message m, returns r, which is the internal coins of $\mathcal{E}$, and hence we can compute $\mathcal{E}(pk_\ell, m; r) = c$.

4 Encryption using DDH

We now describe a scheme $\Pi = (\mathcal{K}, \mathcal{K}_\ell, \mathcal{E}, \mathcal{D})$ that satisfies the definition of a lossy encryption scheme. Let G be any cyclic group of prime order p and let G^* be the set of generators of this cyclic group. The structure of the following algorithm is reminiscent of ElGamal encryption.

- **Algorithm** $\mathcal{K}(1^\lambda)$:

  ```
  $g \overset{\$}{\leftarrow} G^*; x, r \overset{\$}{\leftarrow} \mathbb{Z}_p; pk \leftarrow (g, g^r, g^x, g^{rx}); sk \leftarrow x$
  Return $(pk, sk)$
  ```

- **Algorithm** $\mathcal{K}(1^\lambda)$:

  ```
  $g \overset{\$}{\leftarrow} G^*; x \neq y, r \overset{\$}{\leftarrow} \mathbb{Z}_p; pk \leftarrow (g, g^r, g^x, g^{ry}); sk \leftarrow \perp$
  Return $(pk, sk)$
  ```

- **Subroutine** $Rand(g, h, \tilde{g}, \tilde{h})$

  ```
  $s, t \overset{\$}{\leftarrow} \mathbb{Z}_p, u \leftarrow g^s h^t; v \leftarrow \tilde{g}^s \tilde{h}^t$
  Return $(u, v)$
  ```

- **Algorithm** $\mathcal{E}(pk, m)$

  ```
  $(g, h, \tilde{g}, \tilde{h}) \leftarrow pk, (u, v) \overset{\$}{\leftarrow} Rand(g, h, \tilde{g}, \tilde{h})$
  Return $(u, v \cdot m)$
  ```

- **Algorithm** $\mathcal{D}(sk, c)$

  ```
  $(c_0, c_1) \leftarrow c$
  Return $c_1 / c_0^{sk}$
  ```

This is a very simple algorithm based on the DDH assumption. The authors give two other encryption schemes: one based on *lossy trapdoor functions*, and one based on the Goldwasser-Micali Probabilistic encryption scheme - this last one conveniently has an efficient **Opener** algorithm which is useful for proving semantic security in the SOA sense.

5 SOA Security from Lossy Encryption

The main theorem proved in this paper is that SOA security implies los-key and los-ind security. That is, the advantage in the ind-so-enc sense is bounded above by a linear combination of the advantages in the los-key and los-ind senses. This is summarized in the following theorem.

Theorem 1 (Lossy Encryption implies IND-SO-ENC security)**.** *Let λ be a security parameter, $\Pi = (\mathcal{K}, \mathcal{K}_\ell, \mathcal{E}, \mathcal{D})$ be any lossy public-key encryption scheme, (n, t) any valid SOA parameters, $\mathcal{M}$ any efficient n-message sampler that supports efficient resampling, and A any efficient ind-so-adversary. Then, there exists an unbounded los-ind adversary c and an efficient los-key adversary B such that*

$$\boldsymbol{Adv}^{\text{\it ind-so-enc}}_{A,\Pi,\mathcal{M},n,t}(\lambda) \leq 2\left[\boldsymbol{Adv}^{\text{\it los-key}}_{B,\Pi}(\lambda) + n \cdot \boldsymbol{Adv}^{\text{\it los-ind}}_{C,\Pi}(\lambda)\right]$$

Proof Sketch. We assume, without loss of generality, that the adversary's call to **Corrupt** never returns $\perp$. Recall that

$$\mathbf{Adv}^{\text{ind-so-enc}}_{A,\Pi,\mathcal{M},n,t}(\lambda) = 2 \cdot Pr[\text{INDSO}^{A}_{\Pi,\mathcal{M},n,t}(\lambda) \Rightarrow \text{true}] - 1.$$

We define a sequence of hybrid games (not explicitly here in this sketch), and bound $Pr[\text{INDSO}^{A}_{\Pi,\mathcal{M},n,t}(\lambda) \Rightarrow \text{true}]$.
In the first game, we change the initial key generation to be done with lossy keys, thus creating lossy keys and lossy cipher texts to give to the adversary. This creates a difference bounded by $\mathbf{Adv}^{\text{los-key}}_{B,\Pi}(\lambda)$.
For the next n games, we consecutively change the original messages given to be dummy messages, each change resulting in a difference between games of $\mathbf{Adv}^{\text{los-ind}}_{C,\Pi}(\lambda)$.
Lastly, we make some rearrangements in how things are handed to the adversary. In this last game, the adversary must distinguish between two completely random things, resulting of an advantage of exactly 1/2.
Hence

$$Pr[\text{INDSO}^{A}_{\Pi,\mathcal{M},n,t}(\lambda) \Rightarrow \text{true}] \leq \mathbf{Adv}^{\text{los-key}}_{B,\Pi}(\lambda) + n \cdot \mathbf{Adv}^{\text{los-ind}}_{C,\Pi}(\lambda) + 1/2.$$

Plugging this into the original advantage that we are interested in, we see

$$\begin{aligned} \mathbf{Adv}^{\text{ind-so-enc}}_{A,\Pi,\mathcal{M},n,t}(\lambda) &\leq 2\left[\mathbf{Adv}^{\text{los-key}}_{B,\Pi}(\lambda) + n \cdot \mathbf{Adv}^{\text{los-ind}}_{C,\Pi}(\lambda) + 1/2\right] - 1 \\ &= 2\left[\mathbf{Adv}^{\text{los-key}}_{B,\Pi}(\lambda) + n \cdot \mathbf{Adv}^{\text{los-ind}}_{C,\Pi}(\lambda)\right] \end{aligned}$$

□

6 Remarks

The paper brings to light positive results for SOA encryption that suggest that some of the encryption schemes we are already working with are SOA secure. This is encouraging and suggests that ind-so-enc security is not much harder to accomplish than ind-cpa security. We note that this paper written in the context of having one receiver and multiple senders of messages. As the authors note, their result is in stark contrast to previous strong negative results regarding the same problem phrased with one sender and multiple receivers. On the other hand, maybe this positive result is not so surprising. The authors mention that the difficulty of working with SOA security is that the adversary is allowed to uncover certain random coins involved in the encryption scheme. Knowing these coins should not be very helpful as the coins are supposedly random and therefore knowing a few should give you little to no advantage of learning other coins or anything else about the system. Defining lossy encryption helps us formalize this intuition and indeed shows that this intuition is correct.