Abuses of Probabilistic Encryption Schemes

Yongge Wang*

Abstract

We will demonstrate a method to abuse any probabilistic encryption schemes.

1 Introduction

In a key escrow cryptosystem, the private-key is broken up into pieces and distributed to different authorities. The authorities can get together and reconstruct the private-key. Key escrow guarantees that the police can eavesdrop on all conversations or personal data files even though they are encrypted. Of course, a cryptographic user gains nothing from key escrow systems at all. He has to trust the escrow agents' security procedures, as well as the integrity of the people involved. However, we will show that even if a user surrenders his private-key for a probabilistic cryptosystem to key escrow agents, he still can abuse the cryptosystem and communicate under the government's very nose without having to worry that it would be detected.

2 Abuses of probabilistic cryptosystems

In the past, several probabilistic public-key cryptosystems have been introduced by several authors, for example, the Blum-Goldwasser cryptosystem (1985), the Ajtai-Dwork cryptosystem (1997), and the Goldreich-Goldwasser-Halevi cryptosystem (1997). Here is a formal mathematical definition of this concept.

Definition 2.1 *A probabilistic public-key cryptosystem is defined to be a six-tuple* $(\mathbf{P}, \mathbf{C}, \mathbf{K}, \mathbf{E}, \mathbf{D}, \mathbf{R})$*, where* **P, C, K, E, D, R** *are the set of plaintexts, the set of ciphertexts, the keyspace, the set of encryption rules, the set of decryption rules, and the set of randomizers respectively. The following properties should be satisfied. For each key* $K \in \mathbf{K}$*, there is an encryption rule* $e_K : \mathbf{P} \times \mathbf{R} \to \mathbf{C}$ *and a decryption rule* $d_K : \mathbf{C} \to \mathbf{P}$ *such that* $d_K(e_K(b, r)) = b$ *for every plaintext* $b \in \mathbf{P}$ *and every* $r \in \mathbf{R}$*.*

Let us first describe the following situation: Alice and Bob work for a criminal or terrorist organization, and they want to exchange some secrets using a probabilistic cryptosystem. But according to some laws, they must surrender their private-keys to some key escrow agents. Thus, at some later time, if their keys have been revealed, all their communications will be decrypted and they will be put into prison. However, for a probabilistic cryptosystem, they do not need to worry about this. They can abuse the system and communicate secrets without being detected.

Now we demonstrate how Alice sends a secret $s_1 s_2 \dots s_n$ $(s_i = 0, 1)$ to Bob through a probabilistic cryptosystem without being detected even though their keys have been escrowed. In general the protocol looks like this.

1. Alice and Bob first agree on an "abuse-key" $k_a = k_1 k_2 \dots k_n$ $(k_i = 0, 1)$.
2. Alice puts $c_i = s_i \oplus k_i$ for $i \leq n$, and generates an innocuous message $x_1 x_2 \dots x_n$.

*Department of EE & CS, University of Wisconsin-Milwaukee, P.O. Box 784, Milwaukee, WI 53201, wang@cs.uwm.edu

3. For each $i \leq n$, Alice uses Bob's public-key K to check the random encryptions $e_K(x_i, r)$ of x_i until finding an encryption e_i such that $\#(e_i) = c_i \pmod 2$ where $\#(e_i)$ denotes the number of 1's in e_i. Note that the odds of an encryption of x_i having the above property is 1 in 2 according to the properties of probabilistic cryptosystems. Therefore Alice can find e_i by trying 2 encryptions of x_i in average.

4. After Bob gets the ciphertext. He reconstructs the secret s simply by counting the numbers of 1's in the ciphertexts of x_i's and XORing them with k_i's.

It is clear that the above abuses of a probabilistic cryptosystem cannot be detected by a warden. This attack shows that a user of a probabilistic cryptosystem does not need to worry about the key escrow policy. She can still have her own privacy even though his private-key has been escrowed and her communications might be eavesdropped by distrustful key escrow agents. However a manufacturer (we will call him Mallory) of a probabilistic cryptosystem implementers can also use this abuse to leak his customer's private-keys.

Mallory puts his implementation of a probabilistic cryptosystem in a tamperproof VLSI chip, so that no one can examine its inner workings. He chooses an abuse-key k_a and embeds the abuse mentioned above in his implementation. Then she distributes the chips to his customers, e.g., Alice, Bob, and everyone else who wants them. Every time when Alice sends a message to Bob, the chip reads Alice's private-key k_A, and encrypts the message in such a way that the key k_A can be reconstructed by Mallory after Mallory eavesdrops the communications between Alice and Bob.

However, this attack by (Mallory) can be easily overcome because Alice can save her private-key in a safe place and does not let the chip read it when she encrypts a message for others. Of course, the chip can be so designed that it will remember Alice's private-key when decrypting some encrypted messages received by Alice and leak Alice's private-key in the next time when encrypting a message for Bob. But this attack can also be overcome by using different chips for encryptions and decryptions.

Note that our attack against probabilistic cryptosystems is similar (though different) to the subliminal channels discovered by Simmons in 1983 (see, e.g., Desmedt [1]). The subliminal channel consists in hiding a message in the authenticator through the authentication scheme such that the warden cannot detect its use nor read the hidden part. Several methods have been proposed in the literature for overcoming subliminal channel attacks against cryptosystems. It is not clear whether there is a general way to overcome our attack against probabilistic cryptosystems. For the Ajtai-Dwork cryptosystem, we have found a solution to avoiding the abuses. Roughly speaking, in order to prevent Alice from abusing the Ajtai-Dwork cryptosystem, every time when Alice wants to send a message to Bob, she must send the encrypted message to the censoring warden first. Then the censoring warden perturbs the encrypted message a little and sends the resulting encryption to Bob. Due to the space limit, the details are omitted.

References

[1] Y. Desmedt. Abuses in cryptography and how to fight them. In: *Advances in Cryptology, Proc. of Crypto '88*, pp. 375–389, Springer Verlag, 1988.