Strong Adaptive Chosen-Ciphertext Attacks with Memory Dump (Or: The Importance of the Order of Decryption and Validation)

[Published in B. Honary, Ed., *Cryptography and Coding*, vol. 2260 of *Lecture Notes in Computer Science*, pp. 114–127, Springer-Verlag, 2001.]

Seungjoo Kim[1], Jung Hee Cheon[2], Marc Joye[3], Seongan Lim[1],
Masahiro Mambo[4], Dongho Won[5], and Yuliang Zheng[6]

[1] KISA (Korea Information Security Agency),
A-4th FL., IT Venture Tower, 78 Garak-Dong, Songpa-Ku, Seoul, Korea 138-160
{skim, seongan}@kisa.or.kr
http://www.crypto.re.kr/ or http://www.crypto.ac/
[2] ICU (Information and Communications Univ.),
58-4 Hwaam-Dong, Yusong-Gu, Taejon 305-732, Korea
jhcheon@icu.ac.kr – http://vega.icu.ac.kr/~jhcheon/
[3] Gemplus Card International, Card Security Group,
Parc d'Activités de Gémenos, B.P. 100, 13881 Gémenos, France
marc.joye@gemplus.com – http://www.geocities.com/MarcJoye/
[4] Graduate School of Information Sciences, Tohoku University,
Kawauchi Aoba Sendai, 980-8576 Japan
mambo@icl.isc.tohoku.ac.jp
[5] Sungkyunkwan University,
300 Chunchun-dong, Suwon, Kyunggi-do, 440-746, Korea
dhwon@dosan.skku.ac.kr – http://dosan.skku.ac.kr/~dhwon/
[6] UNC Charlotte,
9201 University City Blvd, Charlotte, NC 28223
yzheng@uncc.edu – http://www.sis.uncc.edu/~yzheng/

Abstract. This paper presents a new type of powerful cryptanalytic attacks on public-key cryptosystems, extending the more commonly studied adaptive chosen-ciphertext attacks. In the new attacks, an adversary is not only allowed to submit to a decryption oracle (valid or invalid) ciphertexts of her choice, but also to emit a "dump query" prior to the completion of a decryption operation. The dump query returns intermediate results that have not been erased in the course of the decryption operation, whereby allowing the adversary to gain vital advantages in breaking the cryptosystem.

We believe that the new attack model approximates more closely existing security systems. We examine its power by demonstrating that most existing public-key cryptosystems, including OAEP-RSA, are vulnerable to our extended attacks.

Keywords. Encryption, provable security, chosen-ciphertext security, ciphertext validity, OAEP-RSA, ElGamal encryption.

1 Introduction

Provably security is gaining more and more popularity. Only known to theoreticians a decade ago, provable security becomes now a standard attribute of any cryptographic scheme.

A scheme is said provably secure if the insecurity of the scheme implies that some widely-believed intractable problem is solvable. The security of a scheme is measured as the ability to resist an adversarial goal in a given adversarial model. The standard security notion for public-key encryption schemes is *indistinguishability under adaptive chosen-ciphertext attacks* (IND-CCA2) [2].

This paper introduces a *stronger* attack. In addition to having access to a decryption oracle, the adversary can perform a "*memory dump*" of the decryption oracle at any time; the sole restriction is that secret data (e.g., private keys) are inaccessible. We believe that this new attack model approximates more closely existing security systems, many of which are built on such operating systems as Unix and Windows where a reasonably privileged user can interrupt the operation of a computing process and inspect its intermediate results at ease.

We show that many IND-CCA2 encryption schemes are vulnerable within our extended attack scenario. Examples include OAEP-RSA and several ElGamal variants. A problem in almost all those schemes is that the validity of the ciphertext cannot be checked until *after* all the decryption is completed. By emitting a dump query prior to the validity checking, the adversary may get access to some secret information resulting from the *raw* decryption of an (invalid) ciphertext and thereby may weaken the scheme.

We hope that in the future cryptographers will analyze the security of their schemes in our extended setting. We note that the scope of our attacks is very broad and may apply to other cryptographic primitives as well (e.g., digital signatures). In the same vein as in [2], it may also be interesting to see what are the different implications and separations implied by our attacks when considering various adversarial goals.

Organization. The rest of this paper is organized as follows. In the next section, we review current security notions for encryption schemes and introduce the new notion of *strong* chosen-ciphertext security. Applicative examples of our attack scenario are also provided. Next, in Section 3, we apply our attack model to the celebrated OAEP-RSA and show that it is insecure under our extended setting. Similar attacks against various ElGamal variants are also presented. Finally, we conclude in Section 4.

2 Strong Adaptive Chosen-Ciphertext Security

2.1 Security notions

Indistinguishability of encryptions, defined by Goldwasser and Micali [14], captures the intuition that an adversary should not be able to obtain any partial information (but its length) about a message given its encryption. In their paper, Goldwasser and Micali study the case of completely *passive* adversaries, i.e., adversaries can only eavesdrop.[1] They do not consider adversaries injecting messages into a network or otherwise influencing the behavior of parties in the network.

To deal with active attacks, Naor and Yung [18] introduced security against *chosen-ciphertext attacks*. They model such an attack by allowing the adversary to get access to a "decryption oracle" and obtain decryptions of her choice. When the security goal is indistinguishability, the attack of [18] runs as follows. During the first stage, called *find stage*, the adversary (viewed as a polynomial-time machine) has access to a decryption oracle. At the end of the stage, the adversary generates two (equal-length) plaintexts m_0 and m_1. In the second stage, called *guess stage*, the adversary receives the encryption of m_b, say c_b, with b randomly drawn from $\{0, 1\}$. The attack is successful if the adversary recovers the value b or, equivalently, if the adversary distinguishes the encryption of m_0 from that of m_1.

Rackoff and Simon [21] later generalized this attack by allowing the adversary to have still access to the decryption oracle after having obtained the target ciphertext c_b. Since the adversary could simply submit the target ciphertext itself to the decryption oracle, Rackoff and Simon restrict the adversary's behavior by not allowing her to probe the decryption oracle with c_b.

In summary, chosen-ciphertext attacks can be classified in two categories:

- (*Static Chosen-Ciphertext Attacks* [18]) The adversary has access to the decryption oracle uniquely prior to obtaining the target ciphertext.[2]
- (*Adaptive Chosen-Ciphertext Attacks* [21]) Not only can the adversary get access to the decryption oracle during the find stage but also during the guess stage. The only restriction is not to submit the target ciphertext itself to the decryption oracle.

As explicitly pointed out in [16], we stress that the adversary may query the decryption oracle with invalid ciphertexts. Although seemingly useless, such attacks are not innocuous. The decryption oracle can for example be used to learn whether a chosen ciphertext is valid or not. From this single bit of information and by iterating the process, Bleichenbacher successfully attacked several implementations of protocols based on PKCS #1 v1.5 [5]. More recently, Manger pointed out the importance of preventing an attacker from distinguishing between rejections at the various steps of the decryption algorithm, say, using timing analysis [17]. The lesson is that implementors must ensure that the reasons for which a ciphertext is rejected are hidden from the outside world.

[1] We note, however, that in the public-key setting, an adversary can always mount a chosen-plaintext attack since encryption is public, by definition.

[2] In the past, this attack has also been called "lunch-time attack" or "midnight attack".

2.2 Chosen-ciphertext attacks with memory dump

Now we consider more powerful adversaries who not only can obtain the plaintexts corresponding to chosen ciphertexts, but can also invade a user's computer and read the contents of its memory (i.e., non-erased internal data). In [24], Shoup introduced the *strong adaptive corruption model*, i.e., a memory dump attack combined with forward secrecy in the context of key exchange protocols. We apply this notion to the security of encryption algorithms and consider strong adaptive chosen-ciphertext security, i.e., a memory dump attack combined with chosen-ciphertext security.[3]

Definition 1 (Strong Chosen-Ciphertext Query). *Let k be a security parameter that generates matching encryption/decryption keys (e, d) for each user in the system. A* strong chosen-ciphertext query *is a process which, on input 1^k and e, obtains either*

- *the plaintext (relatively to d) corresponding to a chosen ciphertext; or*
- *an indication that the chosen ciphertext is invalid; or*
- *non-erased internal states of the decryption oracle decrypting the submitted ciphertext, when a "*`dump`*" query is submitted.*

In the case the decryption oracle would like to prevent the exposure of an internal state, or part of it, it can take the operation

$$\texttt{atomic\_action}[\cdots]\ .$$

The term "*atomic*" means that the action must be executed without external interruptions (e.g., memory core dump, system crash, user signal to kill the transaction, and so on [26]). After the execution being successfully completed, the internal variables used by atomic process are erased.

Most cryptosystems proposed so far do not specify which steps in the decryption process need to be protected from external probes. Such an implementation matter can be actually viewed as a cryptographic design matter: the designer of a cryptosystem *should* express the steps in the decryption process that should be protected by an "atomic action". Of course, the best design is a cryptosystem wherein only a single operation (using the private key) is executed atomically, in the above sense.

To fix the ideas, suppose that an adversary attacks the plain RSA algorithm, as originally described in [22]. Let n denote the RSA modulus and let (e, d) denote the pair of encryption/decryption keys. If the (plain) RSA decryption, $m = c^d \bmod n$, is not performed atomically then an appropriate "`dump`" query may reveal the values of m, c, d and n as they are, during the course of the decryption, available somewhere in the memory.

[3] Similarly, for digital signature schemes, stronger security notions can be defined such as, for example, *existential unforgeability* under *adaptive chosen-message attacks with memory dump.*

However, by using the operation $\texttt{atomic\_action}[m \leftarrow c^d \bmod n]$, which is made up of the following six unit operations

$$\texttt{atomic\_action}\left[\begin{array}{ll} 1: & \texttt{MEM[1]} \leftarrow n \\ 2: & \texttt{MEM[2]} \leftarrow c \\ 3: & \texttt{MEM[3]} \leftarrow d \\ 4: & \texttt{MEM[4]} \leftarrow \texttt{MEM[2]}^{\texttt{MEM[3]}} \bmod \texttt{MEM[1]} \\ 5: & m \leftarrow \texttt{MEM[4]} \\ 6: & \texttt{erase internal memory states MEM[1], MEM[2],} \\ & \quad \texttt{MEM[3]} \text{ and } \texttt{MEM[4]} \end{array}\right],$$

the oracle has only information on n and c (which are publicly known), and on m from a "`dump`" query. In particular, remark that the value d (`MEM[3]`) is inaccessible since it is erased from the memory (cf. Step 6).

In our model, oracle's operations using private key are always executed atomically. This restriction is the weakest possible: allowing the adversary to have access to user's private key (e.g., d in the above example) has the same effect as allowing the adversary to submit the target ciphertext itself to the decryption oracle. Note here that the power of a strong adaptive chosen-ciphertext attack deeply depends on the amount of (unit) operations contained inside a atomic action, so we can consider this amount of operations as one of the security evaluation criteria of a given cryptosystem.

Definition 2 (Strong [Static/Adaptive] Chosen-Ciphertext Attack). *A* strong static chosen-ciphertext attack *consists of the following scenario:*

1. *On input a security parameter k, the key generation algorithm $\mathcal{K}$ is run, generating a public key and a private key for the encryption algorithm $\mathcal{E}$. The adversary of course obtains the public key, but the private key is kept secret.*
2. *[Find stage] The adversary makes polynomially (in k) many strong chosen-ciphertext queries (as in Definition 1) to a decryption oracle.*
   (The adversary is free to construct the ciphertexts in an arbitrary way —it is certainly not required to compute them using the encryption algorithm.)
3. *The adversary prepares two messages m_0, m_1 and gives these to an encryption oracle. The encryption oracle chooses $b \in_R \{0, 1\}$ at random, encrypts m_b, and gives the resulting "target ciphertext" c' to the adversary. The adversary is free to choose m_0 and m_1 in an arbitrary way, except that they must be of the same length.*
4. *[Guess stage] The adversary outputs $b' \in \{0, 1\}$, representing its "guess" on b.*

In a strong adaptive chosen-ciphertext attack*, the adversary has still access to the decryption oracle after having received the target ciphertext: a second series of polynomially (in k) many strong chosen-ciphertext queries may be run. The unique restriction is not to probe the decryption oracle with the target ciphertext c'.*

The success probability in the previous attack scenario is defined as

$$\Pr[b' = b] \ .$$

Here, the probability is taken over coin tosses of the adversary, the key generation algorithm $\mathcal{K}$ and the encryption algorithm $\mathcal{E}$, and $(m_0, m_1) \in M^2$ where M is the domain of the encryption algorithm $\mathcal{E}$.

Definition 3 (Strong [Static/Adaptive] Chosen-Ciphertext Security). *An encryption scheme is secure if every* strong *(static/adaptive) chosen-ciphertext attack (as in Definition 2) succeeds with probability at most negligibly greater than 1/2.*

2.3 Real-world applications

When our strong adaptive chosen-ciphertext attack is mounted, the adversary may learn (i.e., "`memory core-dump`") the entire internal state of the decryption oracle, excluding data that has been explicitly erased.

It may be useful to illustrate the definition of security with a simple example. Consider the case of a security-enhanced electronic mail system where a public-key cryptosystem is used to encrypt messages passed among users. It is a common practice for an electronic mail user to include the original message s/he received into a reply to the message. This practice provides an avenue for chosen-ciphertext attacks, as an adversary can send a ciphertext to a target user and expect the user to send back the corresponding plaintext as part of the reply. For instance, a reply to a message may be as follows [28].

```
  (original message)
> ......
> Hi, is Yum-Cha still on tonight ?
> ......

  (reply to the message)
  ......
  Yes, it's still on. I've already made the bookings.
  ......
```

Now suppose one step further that, via computer viruses, the adversary can modify the target user's electronic mail system to core-dump (i.e., automatically write the exact contents of the memory to a file) and send back the core-dump file in a stealthy way. This is a concrete example of our attack model. Another concrete example is when user's computer is crashed suddenly, internal secret information may then remain unerased.

We quote the following three articles as a basis of thinking about the relevance of security against strong adaptive chosen-ciphertext attack in the real life.

How your privacy is caught in the Net by Duncan Campbell [7]:

"(...) Hackers and government agencies are hard at work designing information stealing viruses. Six months ago, two of them popped up in the same week. "*Caligula*" was aimed at users who installed a privacy-protection system called PGP. Once it infected a computer, it looked for a file holding the secret keys to PGP. Then, automatically and silently, it transmitted the file to the hacker's Internet site. *Caligula* could have taken any information it wanted. Another virus called "*Picture*" was aimed at the America On-line (AOL) Internet service. *Picture* collected AOL users' passwords and log-in data, and sent them to a web site in China. Three months later, "*Melissa*" appeared. It automatically read users' address books, and used the information to mail itself to all their friends and contacts.

According to Roger Thompson, director of anti-virus security consulting firm ICSA, information theft is a price to be paid for the advent of the Internet. (...)

According to former ASIO deputy director Gerald Walsh, who proposed the new powers: *The introduction of other commands, such as diversion, copy, send, (or) dump memory to a specified site, would greatly enhance criminal investigations.* (...)".

Phone.com takes aim at WAP security hole in eWEEK [11]:

"(...) In current WAP transmissions, data must use two security protocols —WTLS during the wireless part of the journey and SSL once the data hits the wires. There is a split second when the data must decrypt and re-encrypt to switch from one protocol to the other. *A security flaw could occur if someone was able to crash the machine in the split second between decryption and re-encryption, causing a memory dump to the disk.* (...)".

Memory reconstruction attack in RSA Official Guide to Cryptography [6]:

"(...) Often, sensitive material is not stored on hard drives but does appear in a computer's memory. For example, when the program you're running allocates some of the computer's memory, the OS tags that area of memory as unavailable, and no one else can use it or see it. When you're finished with that area of memory, though, many operating systems and programs simply "free" it —marking it as available— without overwriting it. This means that anything you put into that memory area, even if you later "deleted" it, is still there. *A memory reconstruction attack involves trying to examine all possible areas of memory. The attacker simply allocates the memory you just freed and sees what's left there.*

A similar problem is related to what is called "virtual memory." The memory managers in many operating systems use the hard drive as virtual memory, temporarily copying to the hard drive any data from memory that has been allocated but is momentarily not being used. When that information is needed again, the memory manager swaps the current virtual memory for the real memory. *In August 1997, The New York Times published a report about* *an individual using simple tools to scan his hard drive. In the swap space, he found the password he used for a popular security application.* (...)".

3 On the Power of Strong Adaptive Chosen-Ciphertext Attacks

For the last few years, many new schemes have been proposed with provable security against chosen-ciphertext attacks. Before 1994, only theoretical (i.e., not very practical) schemes were proposed. Then Bellare and Rogaway [4] came up with the *random oracle model* [3] and subsequently designed in [4] a generic padding, called OAEP (Optimal Asymmetric Encryption Padding), to transform a one-way (partially) trapdoor *permutation* into a chosen-ciphertext secure cryptosystem. Other generic paddings, all validated in the random oracle model, were later given by Fujisaki and Okamoto [12] (improved in [13]), by Pointcheval [20], and by Okamoto and Pointcheval [19]. The first practical cryptosystem with provable security in the *standard model* is due to Cramer and Shoup [8]. They present an extended ElGamal encryption provably secure under the *decisional* Diffie-Hellman problem.

In this section, we demonstrate that most of "decrypt-then-validate"-type cryptosystems with provable security (including OAEP-RSA and most ElGamal variants) can be broken under our strong adaptive chosen-ciphertext attacks.

3.1 OAEP-RSA

We give here a brief overview of OAEP-RSA and refer the reader to [4] for details. The decryption phase of OAEP-RSA is divided into three parts: (i) RSA decryption, (ii) validation, and (iii) output.

Let $n = pq$ denote an RSA modulus, which is the product of two large primes p and q. Furthermore, let e and d, satisfying $ed \equiv 1 \pmod{\mathrm{lcm}(p-1, q-1)}$, respectively denote the public encryption exponent and the private decryption exponent. We assume a hash function $H : \{0,1\}^{k_m+k_1} \to \{0,1\}^{k_0}$ and a "generator" function $G : \{0,1\}^{k_0} \to \{0,1\}^{k_m+k_1}$, where $k_m + k_0 + k_1$ is the bit-length of n. The public parameters are $\{n, e, G, H\}$ and the secret parameters are $\{d, p, q\}$.

A k_m-bit plaintext message m is encrypted through OAEP-RSA as

$$c = (s\|t)^e \bmod n \quad \text{with} \quad s = m0^{k_1} \oplus G(r) \text{ and } t = r \oplus H(s)$$

for a random $r \in \{0,1\}^{k_0}$. Given a ciphertext c, m is recovered as:

(i) $\boxed{\textit{RSA decryption}}$

- $s\|t = \texttt{atomic\_action}[c^d \bmod n]$;
- $z = s \oplus G(t \oplus H(s))$.

(ii) *Validation*

If the last k_1 bits of z are not 0 then

- Erase the internal data; and
- Return "`Invalid Ciphertext`".

(iii) *Output*

- Return the first k_m bits of z.

The attack. The problem with OAEP-RSA resides in that the validity test cannot be performed (i.e., the adversary cannot be detected) until *after* the RSA decryption is completed [25]. Thus an attacker can freely mount a strong adaptive chosen-ciphertext attack and extract the partial information from internal data in the decryption oracle's memory.

For example, consider the following game played by a strong chosen-ciphertext adversary. Let $c = (s \| t)^e \bmod n$ denote the target ciphertext. First, the adversary chooses a random r and forms the (invalid) ciphertext $c' = c \cdot r^e \bmod n$. Then she submits c' to the decryption oracle. Just after the decryption oracle computes $w' := \texttt{atomic\_action}[c'^d \bmod n]$ (note that $w' \equiv c^d \cdot r \pmod{n}$) but before it detects and rejects the invalid ciphertext c', the adversary submits a "`dump`" query, and then obtains the internal data w'. From w', she computes $s \| t = \frac{w'}{r} \bmod n$ and $z = s \oplus G(t \oplus H(s))$. Finally, the adversary recovers m as the first k_m bits of z, so it is easy to find a correct guess by using the recovered m.

3.2 Other provably secure cryptosystems

Similarly to the plain RSA cryptosystem, the plain ElGamal cryptosystem [10] is malleable. We recall the attack hereafter.

Let $\mathcal{G} = \langle g \rangle$ be the cyclic group generated by g. The public encryption key is $X = g^x$ and the private decryption key is x. A message m, considered as an element of $\mathcal{G}$, is encrypted as $(c_1, c_2) = (g^y, X^y \cdot m)$ for some random integer y. Given (c_1, c_2), plaintext message m is recovered as $m = {c_1}^{-x} \cdot c_2$.

If (c_1, c_2) is the target ciphertext, then an adversary can compute the ciphertext $(c'_1, c'_2) = (g^r \cdot c_1, X^r \cdot c_2)$ for some random r. Next, by submitting (c'_1, c'_2) to the decryption oracle, she recovers $m = m' = \texttt{atomic\_action}[{c'_1}^{-x}] \cdot c'_2$ from a "`dump`" query.

Several variants of the basic ElGamal scheme were proposed in order to make it secure against chosen-ciphertexts attacks. We review some of them in the chronological order of their appearance and analyze their resistance under our strong chosen-ciphertext attacks. Here, for simplicity, the same notation G, H are used for several schemes, but functions G and H may have different domain and image from those used in OAEP-RSA of Section 3.1.

Zheng and Seberry I [28]

- encryption: $(c_1, c_2) = (g^y, G(X^y) \oplus (m\|H(m)))$
- decryption: 1) $m\|t = G(\texttt{atomic\_action}[{c_1}^x]) \oplus c_2$
  2) if $t = H(m)$ then output m else output `reject`
- attack: 1) set $(c'_1, c'_2) = (c_1, c_2 \oplus r)$ for a random r
  2) recover m from $(m'\| \cdots) \oplus r = m\| \cdots$

Zheng and Seberry II [28]

- encryption: $(c_1, c_2, c_3) = (g^y, H_s(m), z \oplus m)$ with $z\|s = G(X^y)$
- decryption: 1) $z\|s = G(\texttt{atomic\_action}[{c_1}^x])$
  2) $m = z \oplus c_3$
  3) if $c_2 = H_s(m)$ then output m else output `reject`
- attack: 1) set $(c'_1, c'_2, c'_3) = (c_1, c'_2, c_3)$ with $c'_2 \neq c_2$
  2) recover $m = m'$

Zheng and Seberry III [28]

- encryption: $(c_1, c_2, c_3, c_4) = (g^y, g^k, \frac{H(m)-yr}{k} \pmod{\#\mathcal{G}}, z \oplus m)$
  with $r = X^{y+k}$ and $z = G(r)$
- decryption: 1) $r = \texttt{atomic\_action}[(c_1 c_2)^x]$
  2) $m = G(r) \oplus c_4$
  3) if $g^{H(m)} = {c_1}^r \cdot {c_2}^{c_3}$ then output m else output `reject`
- attack: 1) set $(c'_1, c'_2, c'_3, c'_4) = (c_1, c_2, c'_3, c_4)$ with $c'_3 \neq c_3$
  2) recover $m = m'$

Tsiounis and Yung [27]

- encryption: $(c_1, c_2, c_3, c_4) = (g^y, X^y \cdot m, g^k, y \cdot H(g, c_1, c_2, c_3) + k)$
- decryption: 1) if $g^{c_4} \neq {c_1}^{y \cdot H(g, c_1, c_2, c_3)} c_3$ then output `reject`
  2) output $m = \texttt{atomic\_action}[{c_1}^{-x}] \cdot c_2$

Cramer and Shoup [8]

- encryption: $(c_1, c_2, c_3, c_4) = ({g_1}^s, {g_2}^s, {X_1}^s \cdot m, {X_2}^s \cdot {X_3}^{s \cdot H(c_1, c_2, c_3)})$
  with $X_1 = {g_1}^z$, $X_2 = {g_1}^{x_1} \cdot {g_2}^{x_2}$ and $X_3 = {g_1}^{y_1} \cdot {g_2}^{y_2}$
- decryption: 1) $\alpha = H(c_1, c_2, c_3)$
  2) $v = \texttt{atomic\_action}[{c_1}^{x_1 + y_1 \alpha} \cdot {c_2}^{x_2 + y_2 \alpha}]$
  3) if $c_4 = v$ then output $m = \texttt{atomic\_action}[{c_1}^{-z}] \cdot c_3$
  else output `reject`

Fujisaki and Okamoto [12]

- encryption: $(c_1, c_2) = (g^{H(m\|s)}, (m\|s) \oplus X^{H(m\|s)})$
- decryption: 1) $m\|s = \texttt{atomic\_action}[{c_1}^x] \oplus c_2$
  2) if $c_1 = g^{H(m\|s)}$ then output m else output `reject`
- attack: 1) set $(c'_1, c'_2) = (c_1, c_2 \oplus r)$ for a random r
  2) recover m from $(m'\| \cdots) \oplus r = m\| \cdots$

Fujisaki and Okamoto [13]

- encryption: $(c_1, c_2, c_3) = (g^{H(s,m)}, X^{H(s,m)} \cdot s, \mathcal{E}^{\mathsf{sym}}_{G(s)}(m))$
- decryption: 1) $s = \texttt{atomic\_action}[{c_1}^{-x}] \cdot c_2$
  2) $m = \mathcal{D}^{\mathsf{sym}}_{G(s)}(c_3)$
  3) if $c_2 = X^{H(s,m)} \cdot s$ then output m else output `reject`
- attack: 1) set $(c'_1, c'_2, c'_3) = (g^r \cdot c_1, X^r \cdot c_2, c_3)$ for a random r
  2) recover $m = m'$

Pointcheval [20]

- encryption: $(c_1, c_2, c_3) = (g^{H(m\|s)}, X^{H(m\|s)} \cdot k, (m\|s) \oplus G(k))$
- decryption: 1) $m\|s = G(\texttt{atomic\_action}[{c_1}^{-x}] \cdot c_2) \oplus c_3$
  2) if $c_1 = g^{H(m\|s)}$ then output m else output `reject`
- attack: 1) set $(c'_1, c'_2, c'_3) = (c_1, c_2, c_3 \oplus r)$ for a random r
  2) recover m from $(m'\| \cdots) \oplus r = m\| \cdots$

Baek, Lee, and Kim [1]

- encryption: $(c_1, c_2) = (g^{H(m\|s)}, (m\|s) \oplus G(X^{H(m\|s)}))$
- decryption: 1) $m\|s = G(\texttt{atomic\_action}[{c_1}^{x}]) \oplus c_2$
  2) if $c_1 = g^{H(m\|s)}$ then output m else output `reject`
- attack: 1) set $(c'_1, c'_2) = (c_1, c_2 \oplus r)$ for a random r
  2) recover m from $(m'\| \cdots) \oplus r = m\| \cdots$

Schnorr and Jakobsson [23]

- encryption: $(c_1, c_2, c_3, c_4) = (g^y, G(X^y) + m, H(g^s, c_1, c_2), s + c_3 \cdot y)$
- decryption: 1) if $c_3 \neq H(g^{c_4} \cdot {c_1}^{-c_3}, c_1, c_2)$ then output `reject`
  2) output $m = c_2 - G(\texttt{atomic\_action}[{c_1}^{x}])$

Okamoto and Pointcheval [19]

- encryption: $(c_1, c_2, c_3, c_4) = (g^y, X^y \oplus R, \mathcal{E}^{\mathsf{sym}}_{G(R)}(m), H(R, m, c_1, c_2, c_3))$
- decryption: 1) $R = \texttt{atomic\_action}[{c_1}^{x}] \oplus c_2$
  2) $m = \mathcal{D}^{\mathsf{sym}}_{G(R)}(c_3)$
  3) if $c_4 = H(R, m, c_1, c_2, c_3)$ then output m
  else output `reject`
- attack: 1) set $(c'_1, c'_2, c'_3, c'_4) = (c_1, c_2, c_3, c'_4)$ with $c'_4 \neq c_4$
  2) recover $m = m'$

Table 1. Analysis of several ElGamal variants.

ElGamal variant	Type	Attack
Zheng and Seberry I, II, III [28]	Decrypt-then-validate	Yes
Tsiounis and Yung [27]	Validate-then-decrypt	No
Cramer and Shoup [8]	Validate-then-decrypt	No
Fujisaki and Okamoto [12]	Decrypt-then-validate	Yes
Fujisaki and Okamoto [13]	Decrypt-then-validate	Yes
Pointcheval [20]	Decrypt-then-validate	Yes
Baek, Lee, and Kim [1]	Decrypt-then-validate	Yes
Schnorr and Jakobsson [23]	Validate-then-decrypt	No
Okamoto and Pointcheval [19]	Decrypt-then-validate	Yes

Table 1 summarizes the cryptographic characteristics of the previously described schemes. According to the table, the "decrypt-then-validate"-type schemes are all susceptible to our extended attacks. This is certainly the case when a component c_i in the ciphertext is especially dedicated to the validity test (e.g., as in [28, II and III] or [19]); in that case, it suffices to probe the decryption oracle with the target ciphertext where component c_i is replaced by an arbitrary component $c_i' \neq c_i$.

Remark that the attacks we described are very powerful. Even if the *whole* decryption operation (excluding the validation checking) is performed through an `atomic_action`, our attacks are still successful. This, however, does not mean that it is impossible to construct a "decrypt-then-validate"-type scheme secure against strong adaptive chosen-ciphertext attacks. For example, we were not able to find an attack on the following modification of Baek, Lee, and Kim scheme wherein the $\oplus$ operator is replaced by a pair of symmetric encryption/decryption algorithms $(\mathcal{E}_K^{\mathsf{sym}}, \mathcal{D}_K^{\mathsf{sym}})$ like DES.

- encryption: $(c_1, c_2) = (g^{H(m\|s)}, \mathcal{E}^{\mathsf{sym}}_{G(X^{H(m\|s)})}(m\|s))$
- decryption: 1) $m\|s = \texttt{atomic\_action}[\mathcal{D}^{\mathsf{sym}}_{G({c_1}^x)}(c_2)]$
  2) if $c_1 = g^{H(m\|s)}$ then output m else output `reject`

(Note that we did not prove the security of the scheme.)
This modified scheme is nevertheless less satisfactory than a scheme wherein only the private-key operations in the decryption process are run "atomically" such as in $m\|s = \mathcal{D}^{\mathsf{sym}}_{G(\texttt{atomic\_action}[{c_1}^x])}(c_2)$. Unfortunately, this latter scheme is insecure. Submitting $(c_1', c_2') = (c_1, c_2')$ with $c_2' \neq c_2$ to the decryption oracle, the adversary obtains the value of $R := \texttt{atomic\_action}[{c_1}^x]$ from a "`dump`" query and therefore recovers m as the most significant $|m|$-bits of $\mathcal{D}^{\mathsf{sym}}_{G(R)}(c_2)$.

The ElGamal variants by Tsiounis and Yung [27] and by Schnorr and Jakobsson [23] are actually *signed* encryption schemes. The security proofs can be extended to the "strong adaptive chosen-ciphertext attack" scenario since the validity is checked prior to and separately from the decryption operation itself.

For the scheme by Cramer and Shoup [8], things are slightly different. Some secret data are involved in the validity test. As a consequence, not only the operations using the private decryption-key (i.e., z) but also the operations using the private "validation-keys" (i.e., (x_1, x_2) and (y_1, y_2)) need to be "wrapped up" in an `atomic_action`. Thus, from our evaluation criteria, we can say that this latter scheme needs more protection than the schemes of Tsiounis and Yung, and of Schnorr and Jakobsson.

4 Conclusion

Many security problems are often viewed as "implementation errors". We believe that those could be more fruitfully viewed as cryptographic design errors. In this paper we presented a new security model for encryption algorithms and analyzed the security of several algorithms provably secure against (standard) adaptive chosen-ciphertext attacks. Amongst other things, we showed that, in view of higher-level application programs, "validate-then-decrypt"-type schemes generally better behave facing our extended attacks.

References

1. J. Baek, B. Lee, and K. Kim, "Secure length-saving ElGamal encryption under the computational Diffie-Hellman assumption", *Information Security and Privacy (ACISP 2000)*, volume 1841 of *Lecture Notes in Computer Science*, pages 49–58, Springer-Verlag, 2000.
2. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes", *Advances in Cryptology – CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45, Springer-Verlag, 1998.
3. M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols", *First ACM Conference on Computer and Communications Security*, pages 62–73, ACM Press, 1993.
4. M. Bellare and P. Rogaway, "Optimal asymmetric encryption", *Advances in Cryptology – EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111, Springer-Verlag, 1995.
5. D. Bleichenbacher, "A chosen ciphertext attack against protocols based on the RSA encryption standard PKCS #1", *Advances in Cryptology – CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 1–12, Springer-Verlag, 1998.
6. S. Burnett and S. Paine, "RSA Security's official guide to cryptography", *RSA Press*, 2001.
7. D. Campbell "How your privacy is caught in the Net", http://www.theage.com.au/daily/990808/news/specials/news1.html, 8 August 1999.
8. R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack", *Advances in Cryptology – CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25, Springer-Verlag, 1998.
9. O. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography", *23rd ACM Annual Symposium on the Theory of Computing*, pages 542–552, ACM Press, 1991.
10. T. ElGamal, "A public key cryptosystems and a signature schemes based on discrete logarithms", *IEEE Transactions on Information Theory*, **IT-31**(4):469–472, 1985.
11. eWEEK, "Phone.com takes aim at WAP security hole", http://news.zdnet.co.uk/story/0,,s2081576,00.html, 23rd September 2000.
12. E. Fujisaki and T. Okamoto, "How to enhance the security of public-key encryption at minimum cost", *Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68, Springer-Verlag, 1999.
13. E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes", *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–544, Springer-Verlag, 1999.
14. S. Goldwasser and S. Micali, "Probabilistic encryption", *Journal of Computer and System Sciences*, **28**:270–299, 1984.
15. G. Itkis and L. Reyzin, "Forward-secure signatures with optimal signing and verifying", *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 332–354, Springer-Verlag, 2001.
16. M. Joye, J.-J. Quisquater, and M. Yung, "On the power of misbehaving adversaries", *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 208-222, Springer-Verlag, 2001.
17. J. Manger, "A chosen ciphertext attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as standardized in PKCS #1", *Advances in Cryptology –*

CRYPTO 2001, volume 2139 of *Lecture Notes in Computer Science*, pages 230–238, Springer-Verlag, 2001.

18. M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks", *22nd Annual ACM Symposium on Theory of Computing*, pages 427-437, ACM Press, 1990.
19. T. Okamoto and D. Pointcheval, "REACT: Rapid enhanced-security asymmetric cryptosystem transform", *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 159–175, Springer-Verlag, 2001.
20. D. Pointcheval, "Chosen-ciphertext security for any one-way cryptosystem", *Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 129–146, Springer-Verlag, 2000.
21. C. Rackoff and D. Simon, "Noninteractive zero-knowledge proof of knowledge and chosen ciphertext attack", *Advances in Cryptology – CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444, Springer-Verlag, 1992.
22. R.L. Rivest, A. Shamir, and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, **21**(2):120–126, 1978.
23. C.P. Schnorr and M. Jakobsson, "Security of Signed ElGamal Encryption", *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 73–89, Springer-Verlag, 2000.
24. V. Shoup, "On formal models for secure key exchange", version 4, *Revision of IBM Research Report RZ 3120 (April 1999)*, November 15, 1999.
25. V. Shoup and R. Gennaro, "Securing threshold cryptosystems against chosen ciphertext attack", *Advances in Cryptology – EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 1–16, Springer-Verlag, 1998.
26. A. Silberschatz, J. Peterson, and P. Galvin, *Operating system concepts*, Third edition, Addison-Wesley Publishing Company.
27. Y. Tsiounis and M. Yung, "On the security of ElGamal-based encryption", *Public Key Cryptography*, volume 1431 of *Lecture Notes in Computer Science*, pages 117–134, Springer-Verlag, 1998.
28. Y. Zheng and J. Seberry, "Immunizing public key cryptosystems against chosen ciphertext attacks", *IEEE Journal on Selected Area in Communications*, **11**(5):715–724, 1993.