SEG – A Provably Secure Variant of El-Gamal

David Soldera
Trusted E-Services Laboratory
HP Laboratories Bristol
HPL-2001-149
June 21st , 2001*

E-mail: David_Soldera@hp.com

provable security, Zheng-Seberry, Cramer-Shoup

The Zheng-Seberry (ZS) [9] encryption scheme was published in 1993 and was one of the first practical schemes that was considered secure against an adaptively chosen ciphertext adversary. This paper shows that the semantic security of the one-way hash variant of the ZS scheme is actually insecure against an adaptively chosen ciphertext adversary. Attempts to modify the ZS scheme resulted in a variant of El-Gamal that is provably secure against an adaptively chosen ciphertext adversary using standard public-key cryptography assumptions i.e. not the random oracle model.

* Internal Accession Date Only
Approved for External Publication

SEG - A PROVABLY SECURE VARIANT OF EL-GAMAL

David Soldera*

Hewlett-Packard Labs, Bristol BS34 8QZ, England
David_Soldera@hp.com

Abstract. The Zheng-Seberry (ZS) [9] encryption scheme was published in 1993 and was one of the first practical schemes that was considered secure against an adaptively chosen ciphertext adversary. This paper shows that the semantic security of the one-way hash variant of the ZS scheme is actually insecure against an adaptively chosen ciphertext adversary. Attempts to modify the ZS scheme resulted in a variant of El-Gamal that is provably secure against an adaptively chosen ciphertext adversary using standard public-key cryptography assumptions i.e. not the random oracle model.

Keywords: Zheng-Seberry, provable security, Cramer-Shoup.

1 INTRODUCTION

In 1993 Zheng-Seberry presented a paper introducing three new public-key encryption schemes that were the first efficient schemes (considered) secure against an adaptively chosen ciphertext adversary, under some assumptions. The ZS paper has been widely referenced in literature [2, 4], even as recently as Eurocrypt 2000 [6]. This paper introduced a new notion called 'sole-samplable space', a precursor to the idea of message awareness. It was also one of the first papers to combine encryption and signatures in the one scheme. This would eventually lead to the new concept of signcryption, introduced by Zheng in [10].

In section 2.2 of this paper we show the ZS one-way hash scheme is not secure against an IND-CCA2 adversary. Lim and Lee [4] also discovered how ZS can be manipulated, but they appear to have failed to see how to use it to break ZS in the IND-CCA2 sense. Also presented is a simple fix for the ZS scheme. Actually the fix is one suggested by Zheng in a paper extending ZS for use in authenticated encryption [8]; however Zheng stresses the change is only needed for the authenticated encryption scheme. The fix of ZS says nothing about its security.

Since then much progress has made in the area of provable security for public-key cryptosystems, from those that use the Random Oracle (RO) model [2] to the scheme by Cramer-Shoup (CS) [4] that is provably secure using standard public key cryptography assumptions.

Using the RO model or standard assumptions for a proof of security, represent opposite ends of the provable security spectrum. The RO model yields extremely efficient [2] schemes yet practical implementations using hash functions cannot hope to achieve actual RO's. At the other end of the spectrum are standard assumptions, they give us much more confidence in

* This work was partially carried out during a Masters of Computer Science (Hons) with the Centre for Computer Security Research University of Wollongong, NSW 2522, Australia.

security, yet the schemes that are available are still too inefficient (at least compared to RO schemes) for the majority of practical implementations.

The new scheme SEG presented in this paper starts to bridge this gap between efficiency and assumptions. If we compare SEG to CS, SEG has half the communication overhead and has only 3 exponentiations in total, compared to 8 for CS, yet relies on the same assumption, the Diffie-Hellman Decision Problem. While SEG falls just short of being as efficient as some RO schemes, it is closer than any other scheme that enjoys provable security using standard assumptions.

The new SEG scheme was born out of studying the one-way hash (OWH) variant of the original ZS scheme. For SEG to be considered secure a proof of security needs to be provided and the best proof technique (in that it requires the least assumptions) is that by Cramer-Shoup. So Section 4 presents the new SEG scheme and a proof of security which borrows many parts of the CS proof.

2 ZS SCHEME

The ZS paper presented three variants of an El-Gamal like cryptosystem. The three variants were described as 'immunising' the cryptosystem against an adaptively chosen ciphertext adversary. The variants incorporated a one-way hash function (OWH), a universal hash function and a digital signature.

2.1 ZS-OWH

The OWH variant is presented below.

ZS-OWH
Preliminaries
Consider messages of length n, a one-way hash function H with output length k_0 and a PRNG G with output length $n + k_0$. Operations are modulo p and there is a generator g.
Key Generation
Private key is $x_R \in$ GF(p) and public key is $y_R = g^{x_R} \bmod p$.
Encryption
Encrypt message m 1) $x \in_R [1, p-1]$ 2) $z = G\left(y_R^x\right)_{[1..(n+k_0)]}$ 3) $t = \mathrm{H}(m)$ 4) $c_1 = g^x$ 5) $c_2 = z \oplus (m \| t)$ Ciphertext is (c_1, c_2)
Decryption
1) $z' = G\left(c_1^{x_R}\right)_{[1..(n+k_0)]}$ 2) $w = z' \oplus c_2$ 3) $m = w_{[1...n]}$ 4) $t' = w_{[(n+1)..(n+k_0)]}$ If $\mathrm{H}(m) = t'$ then output m else output $\varnothing$

2.2 Breaking ZS-OWH in IND-CCA2 Sense

It has become standard practice that the level of security required for a public-key cryptosystem is industinguishability of encryptions, IND, (equivalently semantic security or

non-malleability) against a chosen ciphertext adversary (CCA2). For formal definitions and notation see [1]. The basic idea behind an IND-CCA2 adversary is that they are given access to an encryption and decryption oracle, they then choose two messages, one of which gets encrypted (they do not know which). They are then presented with the ciphertext of the encrypted message and asked to determine which of the two messages was encrypted. A successful adversary succeeds with probability non-negligible better than ½. The only restriction is that the adversary may not query the decryption oracle with the challenge ciphertext.

To break ZS-OWH in the IND-CCA2 sense involves creating a new ciphertext from an existing ciphertext; however, this can only be done if the message corresponding to the existing ciphertext is known.

To see how this is achieved consider the last part of the ciphertext, $(m \,||\, \mathrm{H}(m))$, it only depends on the message, so if the message is known, this part of the ciphertext can be recreated. If the adversary wishes to replace the message m with another message m', this can be achieved via:

$$\begin{aligned} c_2' &= c_2 \oplus (m \,||\, \mathrm{H}(m)) \oplus (m' \,||\, \mathrm{H}(m')) \\ &= z \oplus (m \,||\, \mathrm{H}(m)) \oplus (m \,||\, \mathrm{H}(m)) \oplus (m' \,||\, \mathrm{H}(m')) \\ &= z \oplus [(m \,||\, \mathrm{H}(m)) \oplus (m \,||\, \mathrm{H}(m))] \oplus (m' \,||\, \mathrm{H}(m')) \qquad \text{(expression in [] is 0)} \\ &= z \oplus (m' \,||\, \mathrm{H}(m')) \end{aligned}$$

The new ciphertext is (c_1, c_2') and the adversary is successful in manipulating the cryptosystem.

This attack can be used by a CCA2 adversary to defeat IND and the adversary succeeds 100% of the time. In this situation the adversary does not know which of two messages, m_0 or m_1, has been encrypted, but he knows one of them has been. Let the encrypted message be m_b where $b \in [0,1]$. The adversary uses the above attack by setting $m = m_0$ and $m' = m_1$ and creates a new cryptogram via:

$$\begin{aligned} c_2' &= c_2 \oplus [\, m_0 \,||\, \mathrm{H}(m_0) \,] \oplus [\, m_1 \,||\, \mathrm{H}(m_1) \,] \\ &= z \oplus \left[m_{\bar{b}} \,||\, \mathrm{H}(m_{\bar{b}}) \right] \end{aligned}$$

Hence the adversary creates a new ciphertext (c_1, c_2'), which is a valid ciphertext for the message that was not encrypted in the challenge ciphertext. Since the adversary is a CCA2 adversary, and the new ciphertext is not the challenge ciphertext, he may query the decryption oracle with it. The decryption oracle will dutifully return the message that was not encrypted, $m_{\bar{b}}$, and the adversary makes their choice for b as corresponding to the message not returned by the decryption oracle.

The ZS-OWH scheme is largely of theoretical value to the cryptographic community, so while breaking the scheme does not have many practical implications, it is still of theoretical interest. This break highlights the importance of adding random information to the integrity check on the message.

This attack on ZS-OWH is a relatively trivial one and as could be expected a trivial change to the scheme thwarts this attack. By simply creating a new variable $r = y_R^x$ and changing $t = \mathrm{H}(m||r)$, then the attack no longer works. The change incorporates some randomness into the hash calculation and thus defeats the above attack as the adversary can no longer create the concatenation of message and hash. This is because the adversary does not know the random information. This change defeats the above attack, but of course this does not prove the security of the scheme.

This change was borrowed from an authenticated-encryption version of ZS-OWH by Zheng [8], however Zheng stresses that the changes made are only needed for the new scheme proposed and that the original scheme is secure.

3 SEG

The attack and the repair of the original ZS-OWH leave a rather large question mark over its security. Securing the original ZS-OWH scheme led to a new El-Gamal variant. (Note, completely new notation is adopted for the rest of this paper)

SEG
Preliminaries
Consider messages of length $n - k_0$, a hash H with output length k_0. All operations are performed in the group G of order q (q is a large prime) in which there exists a generator g. There also exists some (invertible) deterministic method $p(\cdot)$ to encode a message as an element of G.
Key Generation
Private key is $z \in Z_q$ and public key is $h = g^z$.
Encryption
Encrypt message m 1) $r \in_R Z_q$ 2) $e = h^r$ 3) $t = H(m \| e)$ 4) $M = p(m \| t)$ 5) $u = g^r$ 6) $e = e \cdot M$ Ciphertext is $y = (u, e)$
Decryption
1) $e' = u^z$ 2) $M' = \frac{e}{e'}$ 3) $m \| t' = p^{-1}(M')$ If $H(m \| e') = t$¢ then output m else output ∅

If the group chosen were the set of quadratic residues a possible encoding method $p(\cdot)$ would be simple squaring (given $m\|t$ was interpreted as an element of Z_p modulo a large prime p of the form $2q + 1$). Then in step 2 of the decryption, if neither square root yields a correct hash then the output is also ∅.

4 PROOF OF SECURITY

4.1 DDHP

All the proofs for SEG rely on the difficulty of the Decision Diffie-Hellman Problem (DDHP), the definition of which, from [4], is given below.

Definition 1 – [4, pg. 16] Let G be a group of large prime order q, and consider the following two distributions:

- the distribution **R** of random quadruples $(g_1, g_2, u_1, u_2) \in G^4$;

- the distribution **D** of quadruples $(g_1, g_2, u_1, u_2) \in G^4$, where g_1, g_2 are random, and $u_1 = g_1^r$ and $u_2 = g_2^r$ for random $r \in Z_q$.

An algorithm that solves the DDHP is a statistical test that can effectively distinguish these two distributions. ?

4.2 SEG¢

We will prove the security of SEG by proving the security of an equivalent cryptosystem SEG′, presented below.

SEG′
Preliminaries
Consider messages of length $n - k_0$, a hash H with output length k_0. All operations are performed in the group G, of order q (q is a large prime) and there exists two generators g_1 and g_2. There also exists some (invertible) deterministic method $p(\cdot)$ to encode a message as an element of G.
Key Generation
Private key is $z_1, z_2 \in Z_q$ and public key is $h = g_1^{z_1} g_2^{z_2}$.
Encryption
Encrypt message m 1) $r \in_R Z_q$ 2) $e = h^r$ 3) $t = \mathrm{H}(m \,\|\, e)$ 4) $M = p(m \,\|\, t)$ 5) $u_1 = g_1^r$ 6) $u_2 = g_2^r$ 7) $e = e \cdot M$ Ciphertext is (u_1, u_2, e)
Decryption
1) $e' = u_1^{z_1} u_2^{z_2}$ 2) $M' = \frac{e}{e'}$ 3) $m \,\|\, t' = p^{-1}(M')$ If $\mathrm{H}(m \,\|\, e') = t$¢ then output m else output ∅

SEG′ can be converted to SEG by setting $z_2 = 0$, this makes u_2 completely redundant since it is no longer needed to decrypt.

4.3 Reducing SEG¢ to SEG

We show that the security of SEG′ implies the security of SEG. This will be done in two steps, first the security of SEG′ with $z_2 = 0$, call it $\mathrm{SEG}'_{z_2=0}$, will be shown, then u_2 will be removed. The two schemes are identical after these changes.

Let A be an IND-CCA2 adversary with an advantage in breaking $\mathrm{SEG}'_{z_2=0}$. We will use A to construct an IND-CCA2 adversary B with an advantage in breaking SEG′.

We now define adversary B. B can run in two stages, a ‘find’ stage and a ‘guess’ stage. The find stage is responsible for finding a pair of messages to distinguish (it will also output some

state information s) and the guess stage is responsible for distinguishing which message was encrypted in the challenge ciphertext. Let $D()$ be the decryption oracle that B has access too.

Algorithm B(find, g_1, g_2, h, q, G)
    Run A(find, g_1, g_2, h, q, G)
        When A makes a decryption query, $y¢$, respond with
            $m \leftarrow D(y¢)$
    A returns (m_0, m_1, s)
B returns (m_0, m_1, s)

Algorithm B(guess, m_0, m_1, s, y)
    Run A(guess, m_0, m_1, s, y)
        When A makes a decryption query, $y¢$, respond with
            $m \leftarrow D(y¢)$
    A returns $b¢$
B returns $b¢$

Any valid ciphertext that A produces will be of the form $\left(u_1, u_2, g_1^{z_1} g_2^{z_2} M\right)$ since A encrypts with public key $h = g_1^{z_1} g_2^{z_2}$, hence any valid ciphertexts can be passed to $D()$ and will be correctly decrypted. It follows that if A has an advantage then so does B.

By simple inspection of $\mathrm{SEG}'_{z_2=0}$ we see that u_2 now serves no purpose in the decryption algorithm and so can be removed from the scheme leaving us with SEG.

4.4 IND-CPA security of SEG¢

We will show SEG′ is secure against an IND-CPA adversary. Proving the IND-CPA security of SEG′ is important as it will be needed to prove the IND-CCA2 security.

Theorem 1 – *If the DDHP is hard in the group G, then SEG¢ is secure in the sense of IND-CPA.*

Proof.
We assume there exists an adversary, A, that has an advantage in breaking SEG′ in the IND-CPA sense. We will use A to construct an adversary B with an advantage in breaking El-Gamal in the IND-CPA sense. El-Gamal encrypts a message m as $(g^r, h^r m)$ where g is a generator, $h = g^z$ is the public key with z the secret key and r is random. This will complete the proof as the IND-CPA security of El-Gamal has been shown to be equivalent to DDHP [7].

Let the number of bits used to represent a group element in El-Gamal be l. Let the number of bits used to represent the message in SEG′ be $l¢$. Then $l = l¢ + |\mathrm{H}|$, where $|\mathrm{H}|$ is the size in bits of the output of the hash function.

Algorithm B(find, g, h, q, G)
    Let $g_1 = g$
    Choose $w \in_R Z_q$ such that $g_2 = g_1^w$ is a generator
    $(m_0¢, m_1¢, s) \leftarrow A$(find, g_1, g_2, h, q, G)
    Choose random elements $c_0, c_1 \leftarrow Z_{|\mathrm{H}|}$
    $m_0 = m_0¢ \,||c_0$, $m_1 = m_1¢ \,||c_1$
B returns (m_0, m_1, s)

Algorithm B(guess, m_0, m_1, s, y)

Parse y as (u_1, e)
$y\phi = (u_1, u_1^w, e)$
$m_0\phi = m_{0[1..l\phi]}$, $m_1\phi = m_{1[1..l\phi]}$
$b\phi \leftarrow A(\text{guess}, m_0\phi, m_1\phi, s, y\phi)$
B returns $b\phi$

We assume that the same encoding of the message to a group element is used in both schemes. Note, in SEG′ the encoding is of the message hash concatenation.

Let $Advantage^{IND-CPA}_{A,SEG'}(k) = a$ where $Advantage^{IND-CPA}_{A,SEG'}(k)$ is the advantage of the IND-CPA adversary A against SEG′ on input security parameter k. We will show that if this advantage is non-negligible then $Advantage^{IND-CPA}_{B,El-Gamal}(k) = a - 2^{-l'}$.

First we argue the construction of B is valid. In the guess stage, B has to choose $w \in_R Z_q$ such that $g_2 = g_1^w$ is a generator, there are $f(q)$ generators in the group, so algorithm B_2 can do this in polynomial time. The value $h^r M$ has the form $g^{zr}M$ for El-Gamal and $g_1^{r(z_1+wz_2)}M$ in SEG′, but the El-Gamal form is submitted to the SEG′ adversary, this is not important however since DDHP ensures no adversary can tell g^{zr} and $g_1^{r(z_1+wz_2)}$ apart for random r. Also, without access to a decryption oracle, the value of the secret key is not important (nor the value of the hash).

The find stage of A creates and returns m_0 and m_1, both have length $l\phi$. This means that A cannot be used to distinguish between all the message pairs in the message space for El-Gamal, as El-Gamal messages are of length l. But A will distinguish between all message pairs where the first (counting from the most significant bit) $l\phi$ bits of both messages differ in at least 1 bit.

So, we argue that since there is a significant set of message pairs that differ in the first $l\phi$ bits (for suitable large l, $l\phi$ and |H|), then the adversary against El-Gamal will have an advantage in the *average* case (that is, choose m_0 and m_1 uniformly).

$$Pr[m_{0[1..l']} = m_{1[1..l']}] = 2^{-(l - |H|)} = 2^{-l\phi}$$

Hence for suitable large $l\phi$, the adversary against El-Gamal will have an advantage.

$$Advantage^{IND-CPA}_{B,El-Gamal}(k) = a - 2^{-l'}$$

4.5 The Hash function

The nature of the hash function has not been described yet as it was unimportant for IND-CPA security. The hash function could be a weakly collision-free one-way hash function (a weak universal one-way hash function or a weak one-way hash function would be appropriate). This however, is probably a stronger assumption than is necessary.

Weak collision freeness requires some target x and the problem is to find a y such that H(x) = H(y). Consider the input to the hash for SEG′, it is the message and ephemeral key (h^r), now a set of possible messages may be known but the DDHP ensures the ephemeral key cannot be recreated. This means that the input value of the hash cannot be recreated, so we have no target x to find a collision for. One-wayness assumes you have some hash value H(x) and the problem is to find x, but IND-CPA and the DDHP guarantees that not one bit of the coded message (M) is leaked so the ciphertext perfectly hides the value of the hash.

Considering these facts for an adversary trying to attack a challenge ciphertext show the adversary's options are severely limited. Even knowing a collision related to m_0 or m_1 is of no use if the ephemeral key is not known. Being able to invert a one-way hash function is only useful if you know the hash value. It is not even clear how the birthday attack would useful in this situation. These arguments of course do not rule out some ingenious attack, but they are compelling. Of course if we assume collision freeness and one-wayness then we can be assured of security.

Certainly standard hash functions like SHA-1 are fine for use in any implementation of the scheme.

4.6 Sketch of proof for IND-CCA2 security of SEG¢

Now we show SEG′ is secure against an IND-CCA2 adversary. First we give the construction of the proof. It is assumed there exists an adversary A that can break the cryptosystem in the IND-CCA2 sense and then it is shown how this adversary can unwittingly be used to help solve what is considered a computationally unfeasible problem, in this case the DDHP. The construction of the proof can be seen in Figure 1.

The input to the proof are quadruples coming from either **D** or **R** (but not both). These go to a constructed simulator, which is responsible for, the creation of keys, simulation of an encryption oracle and simulation of a decryption oracle. The adversary receives all its information, including oracle queries, from the simulator.

The proof runs as follows. A quadruple is input and the simulator creates a valid secret key and public key. The simulator runs the find stage of A, and A returns two messages, m_0 and m_1. The simulator then runs the simulated encryption oracle which chooses a random bit $b \in [0, 1]$, encrypts m_b and outputs the challenge ciphertext. The adversary cannot see the simulator's choice for b.

The simulator then runs the guess stage of the A on input the challenge ciphertext and A outputs its guess, b', for the random bit. Both the simulator and the adversary pass b and b' respectively to a distinguisher that outputs 1 if $b = b'$ otherwise 0.

Consider the case when the input comes from **R**, the simulator is unable to create a valid ciphertext (as the relation that quadruples from **D** have, are not present in quadruples from **R**). This fact will be crucial in showing the adversary cannot succeed in guessing b with any advantage. Alternatively, when the input comes from **D**, then the simulator creates a perfectly valid ciphertext and the adversary can guess the bit b with an advantage.

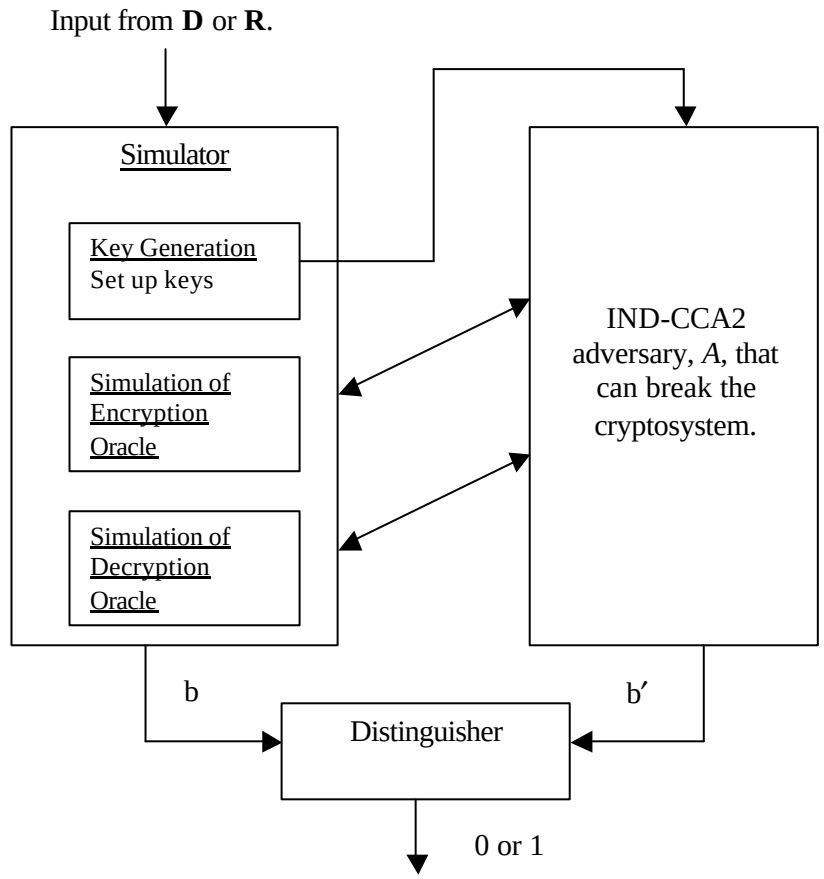

Figure 1 – Graphical representation for the construction of the SEG′ proof.

Hence by observing the distribution of 0's and 1's that are output by the distinguisher, it can be determined which distribution the quadruples are coming from. If the quadruples are coming from **R** then 1's will occur with probability ½ and 0's with probability ½. The adversary will only be correct half the time, as it has no advantage. If the quadruples come from **D** then the adversary has an advantage and 1's will occur with probability ½ + α (where α is the adversary's non-negligible advantage) and 0's with probability ½ - α.

Hence, by observation of the output distribution, one has a statistical test for the DDHP.

The construction of the proof is relatively simple, however there are several properties that must hold for the proof to be valid.

- The simulator must create a valid ciphertext if the quadruple comes from **D** and an invalid ciphertext if the quadruple comes from **R**.
- When the quadruple comes from **D** the joint distribution of the adversary's view and the random bit *b* must be statistically indistinguishable from that in an actual attack
- When the quadruple comes from **R** the distribution of the random bit *b* must be (essentially) independent from the adversary's view.

4.7 IND-CCA2 security for SEG¢

Theorem 2 – *If the Diffie-Hellman Decision Problem is hard in the group G, then SEG¢ is secure against an adaptive chosen ciphertext attack.*

First the simulator is described. On input the quadruple (g_1, g_2, u_1, u_2) the simulator generates random private keys $z_1, z_2 \in_R Z_q$ and outputs the public key as $h = g_1^{z_1} g_2^{z_2}$.

The simulator simulates the encryption oracle as follows. On input two messages m_0 and m_1 it selects a random bit $b \in [0, 1]$, a random number $j \in_R Z_q$ and computes:

$$e = u_1^{z_1} u_2^{z_2}$$
$$e = e \cdot p(m_b \,||\, \mathrm{H}(j))$$

The simulated encryption oracle outputs (u_1, u_2, e), where u_1 and u_2 come from the input quadruple to the simulator. Note the change from the normal encryption algorithm, the hash of m_b and e is replaced with the hash of a random number j. The reason for this will explained in Lemma 2. It is important that this change does not affect the adversary's advantage, and this will be shown in Lemma 1.

The simulated decryption oracle works in exactly the same way as the decryption algorithm, and is just given for completeness. On input (u_1, u_2, e) it computes:

$$e' = u_1^{z_1} u_2^{z_2}$$
$$M' = \frac{e}{e'}$$
$$m \,||\, t' = p^{-1}(M')$$

If $H(m \,||\, e') = t'$ the simulated decryption oracle outputs m, else it outputs $\varnothing$.

The aim now is to show that when the input comes from **D** the simulator simulates the encryption and decryption oracles perfectly (probabilistically) and the advantage of the adversary is apparent at the distinguisher. Alternatively, if the input comes from **R** then the output of the simulated encryption oracle will not be a valid ciphertext in the sense that $\log_{g_1} u_1 \neq \log_{g_2} u_2$ and the adversary can have no advantage in guessing b.

The theorem follows from the following two lemmas.

Lemma 1 – *When the simulator's input comes from* ***D****, the joint distribution of the adversary's view and the hidden bit b is statistically indistinguishable from that in the actual attack.*

In this case it is clear the output of the simulated encryption oracle has the right distribution as $u_1^{z_1} u_2^{z_2} = g_1^{rz_1} g_2^{rz_2} = \left(g_1^{z_1} g_2^{z_2}\right)^r = h^r$, which gives the same distribution as the output of the actual encryption oracle due to the ephemeral key being the same.

The presence of the hash of j does not affect the advantage of any passive attack the adversary might try because IND-CPA guarantees that no information about M is leaked and so a ciphertext that contains $\mathrm{H}(j)$ is indistinguishable from a ciphertext containing $\mathrm{H}(m_b||e)$. It is easy to see if there exists some algorithm C that could distinguish ciphertext $c_0 = e_0.p(m_b||\ \mathrm{H}(m_b||e_0))$ from $c_1 = e_1.p(m_b||\ \mathrm{H}(j))$ with some advantage a, then we could construct an algorithm B to break SEG′ in the IND-CPA sense with advantage $a/2$. Algorithm B would just run its find stage and output two messages. B would pass these to the encryption oracle and receive the challenge ciphertext c. B then chooses a random bit $b¢$ and a random number j, and constructs $c¢ = e¢.p(m_{b¢}\,||\mathrm{H}(j))$ and runs $C(c, c¢)$. Clearly only ½ of the time, when $b = b¢$, will C have an advantage, making B's advantage $a/2$.

We also need to show there is no adaptive attack the adversary can use that relies on the hash in the challenge ciphertext being correct. The only attack that would need the correct hash value would be for the adversary to create some new ciphertext using the challenge ciphertext, such that this new ciphertext uses the same hash as the hash in the challenge ciphertext. However, this requires the same message and e to be used as these are the

corresponding inputs to the hash (we have assumed a collision cannot be found). This makes it impossible for the new ciphertext to be different from the challenge ciphertext, since the secret key is fixed and there is only one r that yields e.

If the simulated encryption oracle produces an indistinguishable output, then the entire simulation is indistinguishable (from the actual oracles to the adversary) if the simulated decryption oracle behaves in the exactly same way as the actual decryption oracle. Since the quadruple comes from **D** and the simulated decryption oracle is identical in its computations to the actual decryption oracle, the simulated decryption oracle will be indistinguishable from the actual decryption oracle.

Lemma 2 – *When the simulator's input comes from **R**, the distribution of the hidden bit is (essentially) independent from the adversary's view.*

When the quadruple comes from **R** we have $u_1 = g_1^{r_1}$ and $u_2 = g_2^{r_2}$ where there is only a negligible chance that $r_1 = r_2$. We will show that the adversary's view is independent of the hidden bit b by showing that if no information about the secret keys is leaked, then the challenge ciphertext is equally likely to be the encryption of m_0 or m_1, or in fact any message.

Assuming the simulated decryption oracle only decrypts valid ciphertexts, we now show that no information about the secret keys is leaked by a valid ciphertext. Consider the following equations from the public key and a valid ciphertext.

$$\log h = z_1 + wz_2$$
$$\log e = r \log h = rz_1 + rwz_2$$

Where $g_2 = g_1^{\,w}$ and log refers to $\log_{g_1}$. Clearly they are linearly dependant and leak no information about z_1 or z_2.

Now consider the output of the simulated encryption oracle, here we derive the following equation.

$$\log e = r_1 z_1 + r_2 wz_2$$

This is clearly linearly independent with $\log h = z_1 + wz_2$. If we consider the solutions to these two equations, they are all the pairs of z_1 and z_2 that satisfy $\log h = z_1 + wz_2$, but all these pairs cause e to take on every value (i.e. a permutation) of G. This means e perfectly hides M_b, as for every possible M_b there is an e consistent with e (e is fixed), and that e can be constructed from a pair of secret keys z_1 and z_2 that are consistent with the public key. The fact that e hides M_b perfectly makes it equivalent to a one-time pad.

If no other information about z_1 and z_2 is available (that is the simulated decryption rejects all invalid ciphertexts and a valid ciphertext leaks no information about z_1 and z_2), then clearly determining which solution is correct is impossible, as it varies uniformly. We are showing that when the quadruple comes from **R**, e is equally likely to be the encryption of m_0 or m_1, or any message. This is why we do not hash the message in the challenge ciphertext, as this would constrain e to being the encryption of m_b, and rule out it being the encryption of $m_{\bar{b}}$. Since the adversary cannot determine the correct solution, the adversary can only guess b, meaning the adversary has no advantage. Hence, the bit b is independent from the adversary's view.

The above argument relies on the simulated decryption oracle rejecting all invalid ciphertexts; otherwise information about z_1 and z_2 may be leaked. A valid ciphertext is (u_1, u_2, e), an

invalid one is (u_1', u_2', e'), and $t\phi = \mathrm{H}(m\phi \,||e\phi)$. We consider possibly ciphertexts submitted to the simulated decryption oracle.

1) (u_1', u_2', e). The adversary will choose u_1' and u_2' to create an $e\phi$ such that either $e\phi = e$ or $e\phi \neq e$. There are q pairs of $r_1\phi$ and $r_2\phi$ such that $e' = u_1'^{z_1} u_2'^{z_2} = g_1^{r_1' z_1} g_2^{r_2' z_2} = u_1^{z_1} u_2^{z_2} = e$ but without knowledge of z_1 and z_2 the adversary can only guess from the set of size q^2 of all $r_1\phi$ and $r_2\phi$ pairs, which means he only succeeds with probability $1/q$. If $e\phi \neq e$ then let $t^* = p^{-1}\left(\frac{e}{e'}\right)_{[n+1\ldots n+k_0]}$ and $m^* = p^{-1}\left(\frac{e}{e'}\right)_{[0\ldots n]}$ and we claim that $t^* \neq \mathrm{H}(m^* \,||e')$ except with a probability equivalent to that of finding a random collision for the hash function. In both cases the simulated decryption oracle will reject the ciphertexts with overwhelming probability.

2) (u_1, u_2, e'). With $e \neq e'$, then let $t^* = p^{-1}\left(\frac{e'}{e}\right)_{[n+1\ldots n+k_0]}$ and $m^* = p^{-1}\left(\frac{e'}{e}\right)_{[0\ldots n]}$ and we claim that $t^* \neq \mathrm{H}(m^* \,||e)$ except with a probability equivalent to that of finding a random collision for the hash function.

Thus, the simulated decryption oracle will reject all invalid ciphertexts, except with negligible probability. ?

Hence if the DDHP is a computationally unfeasible problem then an IND-CCA2 attacker for SEG′ cannot exist.

5 CONCLUSION

This paper has shown that the one-way hash variant of the scheme by Zheng-Seberry [9] is insecure in the sense of IND against an adaptively chosen ciphertext adversary.

A new scheme was created called SEG, which was shown to be provably secure against an IND-CCA2 adversary. The advantage of this new scheme is its efficiency (compared to CS), and that its proof relies only on standard assumptions (it does not require the RO assumption).

6 ACKNOWLEDGMENTS

The break of the Zheng-Seberry scheme was discovered during discussions with Associate Professor Josef Piepzyck. I would also like to thank Steven Galbraith for useful discussions about the proof of security for SEG′.

7 REFERENCES

1. Bellare, M., Desai, A., Pointcheval, D., and Rogaway, P. "Relations among notions of security for public-key encryption schemes" in *CRYPTO'98*. LNCS 1462, pg 26-45. Springer-Verlag, California, 1998.
2. Bellare, M. and Rogaway, P. "Optimal asymmetric encryption - how to encrypt with RSA" in *EUROCRYTP'94*. LNCS 950, pg 92-111. Springer-Verlag, 1994.
3. Boneh, D. "The decision Diffie-Hellman problem" in *Third Algorithmic Number Theory Symposium (ANTS)*. LNCS 1423, Springer-Verlag, 1998.
4. Cramer, R. and Shoup, V. "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack" in *CRYPTO'98*. LNCS 1462, pg 13-25. Springer-Verlag, California, 1998.
5. Lim and Lee, "Another method for attaining security against adaptively chosen ciphertext attacks", in CRYPTO'93, LNCS 773, pg 420-434. Springer-Verlag 1993.

6. Shoup, V. "Using hash functions as a hedge against chosen ciphertext attack" in *EUROCRYPT'00*. LNCS 1807, pg 275-288. Springer-Verlag, 2000.
7. Tsiounis, Y. and Yung, M. "On the security of El-Gamal based encryption" in *PKC'98*. LNCS 1431, Spinger-Verlag, Japan, 1998.
8. Zheng, Y., "Improved public key cryptosystems secure against chosen ciphertext attacks", Technical Report 94-1, *University of Wollongong*, 1994.
9. Zheng, Y. and Seberry, J., "Immunizing public key cryptosystems against chosen ciphertext attacks". *IEEE Journal on Selected Areas in Communications*, 1993. 11(5): p. 715-724.
10. Zheng, Y. "Digital signcryption or how to achieve Cost(Signature & Encryption) << Cost(Signature) + Cost(Encryption)" in *CRYPTO'97*. LNCS 1294, pg 165-179. Springer-Verlag, California, 1997.