An implementation attack against the EPOC-2 public-key cryptosystem

Alexander W. Dent

January 23, 2002

Abstract

We present a chosen ciphertext attack against an implementation of EPOC-2 in which it is possible to tell for what reason the decryption of a given ciphertext fails.

1 Introduction

The EPOC-2 cryptosystem [4] is a new public key encryption scheme based on the Okamoto-Uchiyama public-key cryptosystem [5] and the Fujisaki-Okamoto hybrid encryption system [1]. The algorithm has been submitted to the NESSIE project [3] and is described in Figure 1.

2 The Attack

We present a chosen ciphertext attack against this system in a manner similar to the attack against RSA-OAEP by James Manger in [2]. We assume that we can differentiate between errors generated during step 4 of the decryption (OU errors) and errors generated during step 5 of the decryption (integrity errors). We use the following property:

Lemma 1 *Suppose* $C_1 = g^z$ *in* $\mathbb{Z}_n^*$ *for some* z *and let* $0 < z' \leq p$ *be such that* $z \equiv z' \bmod p$*. Suppose further that* C_2 *is some appropriately sized bit string. If* $z' \geq 2^{k-1}$ *then the decryption of* (C_1, C_2) *will fail due to an OU-error but if* $z' < 2^{k-1}$ *then the decryption of* (C_1, C_2) *will either be successfully completed or will fail due to an integrity error.*

Let $C_1 = g^{2^{k-1}+2^{k-2}}$ and let C_2 be a randomly generated, appropriately sized binary string. We ask for the decryption of the ciphertext (C_1, C_2). With high probability this ciphertext will not be decrypted however if the decryption fails due to an OU error then we know that $p > 2^{k-1} + 2^{k-2}$ i.e. the second most significant bit of p is a one. Otherwise $p < 2^{k-1} + 2^{k-2}$ and the second most significant bit of p is a zero.

Now suppose that we know the first i bits of p are $1a_2a_3\ldots a_i$ and we want to find the $(i+1)^{th}$ bit. Let

$$C_1 = g^{2^{k-1}+a_2 2^{k-2}+\ldots+a_i 2^{k-i}+2^{k-i-1}}$$

and ask for the decryption of (C_1, C_2) where C_2 is as before. Again the decryption of this ciphertext will fail with high probability however if the decryption fails due to an OU error then we know that $(i+1)^{th}$ bit of p is a one, otherwise the $(i+1)^{th}$ bit is a zero. We may continue this process until we find all the bits of p.

It is worth noting there are many ways in which an attacker might be able to determine which error caused the decryption to abort, see [2] for more details.

3 Conclusion

There is a practical chosen ciphertext attack against a poor implementation of EPOC-2 that recovers the secret key.

References

[1] E. Fujisaki and T. Okamoto, 'Secure Integration of Asymmetric and Symmetric Encryption Schemes'. *Advances in Cryptology - CRYPTO '99.*

[2] J. Manger, 'A Chosen Ciphertext Attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as Standardized in PKCS #1 v2.0'. *Advances in Cryptology - CRYPTO 2001.*

[3] New European Scheme for Signatures, Integrity and Encryption (NESSIE). `http://www.cryptonessie.org/`

[4] NTT Corporation, 'EPOC-2 Specifications'. Available from `http://www.cryptonessie.org/`

[5] T. Okamoto and S. Uchiyama, 'A New Public-Key Cryptosystem as Secure as Factroing'. *Advances in Cryptology - EuroCRYPT '98.*

Key Generation

Inputs	k, a security parameter
Step 1	Generate two k bit primes p and q. Let $n = p^2 q$.
Step 2	Choose an element $g \in \mathbb{Z}_n^*$ such that g^{p-1} has order p in $\mathbb{Z}_{p^2}^*$ and set $h = g^n$.
Step 3	Let the public key be $PK = (n, g, h, k)$ and the private key be $SK = (p, q)$.
Step 4	Output PK and SK.

Encryption

Inputs	m, a message. PK, a public key
Step 1	Pick an integer $0 < r < 2^{k-1}$ uniformly at random.
Step 2	Let $C_2 = KDF(r) \oplus m$.
Step 3	Let $M = MGF(m\|\|r\|\|C_2)$.
Step 4	Let $C_1 = g^r h^M \bmod n$.
Step 5	Output (C_1, C_2).

Decryption

Inputs	(C_1, C_2), a ciphertext PK, a public key SK, a private key
Step 1	Let $g_p = g^{p-1} \bmod p^2$ and $w = \frac{g_p - 1}{p} \bmod p$.
Step 2	Let $C_p = C^{p-1} \bmod p^2$ and $w' = \frac{C_p - 1}{p} \bmod p$.
Step 3	Let $r' = w'/w \bmod p$.
Step 4	If $r' \geq 2^{k-1}$ then output 'ERROR' and abort.
Step 5	Let $m' = C_2 \oplus KDF(r')$.
Step 6	Let $g' = g \bmod q$, $h' = h \bmod q$ and $M' = MGF(m'\|\|r'\|\|C_2) \bmod q - 1$.
Step 7	Calculate $C_1' = g'^{r'} h'^{M'} \bmod q$. If $C_1' = C_1 \bmod q$ then output m' else output 'ERROR'.

where $KDF()$ and $MGF()$ are respectively appropriately sized Key Derivation Functions and Mask Generation Functions.

Figure 1: The EPOC-2 Public Key Cryptosystem