A new security notion for asymmetric encryption Draft #16

Muhammad Rezal Kamel Ariffin[1,2]

[1] Al-Kindi Cryptography Research Laboratory,
Institute for Mathematical Research,
[2] Department of Mathematics, Faculty of Science,
Universiti Putra Malaysia (UPM), Selangor, Malaysia
rezal@upm.edu.my

Abstract. A new practical asymmetric design is produced with desirable characteristics especially for environments with low memory, computing power and power source.

KEYWORDS: asymmetric security, diophantine equations

1 Key Generation

1. Generate random primes $p, t_1, t_2, t_3, t_4, g_1, g_2, r$ n-bits.
2. Generate e where $gcd(e, p-1) = 1$. Compute $d_0 \equiv e^{-1} \pmod{p-1}$.
3. Compute $A = t_1 + t_3, B = t_2 + t_4$ and $C \equiv g_1^A - g_2^B \pmod{p}$.
4. Compute $e_1 \equiv g_1^{t_1} \pmod{p}$.
5. Compute $e_2 \equiv g_2^{t_2} \pmod{p}$.
6. Compute $w \equiv g_1^{t_3} g_2^{t_2} - g_1^{t_1} g_2^{t_4} \pmod{p}$.
7. Compute $a_1 \equiv \frac{-re_1}{w} \pmod{p}$ and $a_2 \equiv \frac{re_2 + a_1 C}{w} \pmod{p}$.
8. Compute $e_3 \equiv a_1 g_1^{t_3} + a_2 g_2^{t_4} \pmod{p}$.
9. Compute $e_4 \equiv a_2 C \pmod{p}$.
10. Compute $d_1 \equiv a_1 g_2^{t_4} + a_2 g_1^{t_3} \pmod{p}$
11. Publish (e, e_1, e_2, e_3, e_4) as public keys.
12. Publish (d_1, r, p) as private keys and keep $(a_1, a_2, t_1, t_2, t_3, t_4, g_1, g_2, w, d_0)$ secret.

Remark 1. We can have the congruence relation:

1. $e_1 e_3 + e_2(r - d_1) \equiv 0 \pmod{p}$
2. $(a_2 e_1 e_3 - a_1 e_2 e_3) - (a_1 e_4 + a_2 e_1 e_3 - g_1^{t_3}(a_1 a_2 e_1 + (a_1^2 e_2)) \equiv 0 \pmod{p}$

Remark 2. We can also have a few other relations as in Remark 1 that results in 0 modulo p, but each relation need secret parameters.

2 Encryption

1. Message is $b_0 \approx 2^{n-1}$.
2. Compute $b_3 = b_0^e - b_1 - b_2$.
3. Compute $C_1 = b_1 + b_2(e_2e_3 + e_4 + 1) + b_3(e_1e_3 + 1)$ and $C_2 = b_2e_1 + b_3e_2$.
4. Send (C_1, C_2) to recipient.

3 Decryption

1. Compute $(C_1 + C_2(r - d_1))^{d_0} \equiv b_0 \pmod{p}$.

Proposition 1. *The decryption procedure is correct.*

Proof. $(C_1 + C_2(r - d_1))^{d_0} \equiv (b_1 + b_2(a_2C + 1) + b_3 + b_2(-a_2C - re_1 + re_1)^{d_0} \equiv (b_1 + b_2 + b_3)^{d_0} \equiv (b_0^e)^{d_0} \equiv b_0 \pmod{p}$.

To obtain (b_1, b_2) is trivial.□

4 Desirable properties

This section must be treated with caution. It is only meaningful if there does not exist "trivial" attacks on the scheme.
In this section we list down some "desirable" properties induced within the key generation procedures that disallows an adversary to construct "usable" information; either to reconstruct the plaintext or the private keys.

1. From (C_1, C_2), determine (b_0, b_1, b_2). From section 2, this is impossible because of $b_1b_2b_3 \gg C_1$ and $b_2b_3 \gg C_2$ (refer to article by Hermann and May).
2. From section 2 it can be viewed as the problem to solve 3 unknowns in 2 equations.
3. If $a_1 \equiv a_2 \equiv 1 \pmod{p}$ then $d_1 \equiv e_3 \pmod{p}$ but $a_1, a_2 \not\equiv 1 \pmod{p}$.
4. From e_1 determine the tuple (g_1, t_1).
5. From e_2 determine the tuple (g_2, t_2).
6. From e_3 determine the tuple $(a_1, g_1, t_3, a_2, g_2, t_4)$.
7. From e_4 determine (a_2, C).
8. From (e_1, e_2, e_3, e_4) determine a function F such that $F(e_1, e_2, e_3, e_4) \equiv 0 \pmod{p}$, in order to reduce the problem to the integer factorization problem.
9. Observe that $e_1 \pm (e_2e_3 + e_4 + 1) \not\equiv 1 \pmod{p}$ and $e_2 \pm (e_1e_3 + 1) \not\equiv 1 \pmod{p}$. Thus, $C_1 \pm C_2 \not\equiv b_1 + b_2 + b_3 \equiv b_0^e \pmod{p}$.
10. Determine p' and $x \in \mathbb{Z}_{p'}$ such that $(C_1 + C_2(x))^{d_0} \equiv b_0 \pmod{p'}$.