

A new security notion for asymmetric encryption

Draft #15

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 $p, p_1, p_2, t_1, t_2, k_1, g_1, r$ n-bits.
2. Generate e where $\gcd(e, p-1) = 1$. Compute $d_0 \equiv e^{-1} \pmod{p-1}$.
3. Compute $A = p_1 - t_1, B = p_2 - t_2$ and $u = \gcd(B, p-1)$.
4. Compute $a \equiv g_1^A \pmod{p}$.
5. Compute $d_B \equiv \frac{u}{B} \pmod{\frac{p-1}{u}}$.
6. Compute $g_2 \equiv a^{d_B} \pmod{p}$. That is $g_1^A \equiv g_2^{\frac{B}{u}} \pmod{p}$.
7. Generate random v n-bits. Compute $k_2 = vu + t_2$.
8. Compute $w \equiv g_1^{p_1-k_1} g_2^{\frac{k_2-t_2}{u}} - g_1^{k_1-t_1} g_2^{\frac{p_2-k_2}{u}} \pmod{p}$.
9. Compute $e_1 \equiv g_1^{k_1-t_1} \pmod{p}$ and $e_2 \equiv g_2^{\frac{k_2-t_2}{u}} \pmod{p}$.
10. Denote $y_1 = 1 - re_1, y_2 = re_2 - 1$. Compute $a_1 \equiv \frac{y_1}{w} \pmod{p}$ and $a_2 \equiv \frac{y_2}{w} \pmod{p}$.
11. Compute $e_3 \equiv a_1 g_1^{p_1-k_1} + a_2 g_2^{\frac{p_2-k_2}{u}} \pmod{p}$.
12. Compute $d_1 \equiv a_1 g_2^{\frac{p_2-k_2}{u}} + a_2 g_1^{p_1-k_1} \pmod{p}$.
13. Publish (e, e_1, e_2, e_3, p) as public keys.
14. Publish (d_1, r) as private keys and keep $(a_1, a_2, p_1, p_2, t_1, t_2, k_1, k_2, g_1, u, w, y_1, y_2)$ secret.

Remark 1. We can re-write e_2, e_3 and d_1 as follows:

1. $e_2 \equiv g_1^{Ad_B \frac{k_2-t_2}{u}} \pmod{p}$.
2. $e_3 \equiv \frac{1}{w} \left(y_1 g_1^{p_1-k_1} - y_2 g_2^{\frac{p_2-k_2}{u}} \right) \pmod{p}$

$$3. d_1 \equiv \frac{1}{w} \left(y_1 g_2^{\frac{p_2-k_2}{u}} - y_2 g_1^{p_1-k_1} \right) \pmod{p}$$

Remark 2. We can have the congruence relation:

$$(e_2 - e_1)(e_3 e_1 + e_3 e_2 - 1) \equiv 0 \pmod{p}$$

2 Encryption

1. Message is $b_0 \in \mathbb{Z}_p$.
2. Compute $b_3 = b_0^e - b_1 - b_2 \pmod{p}$.
3. Compute $C_1 = b_1 + b_2(e_2 e_3) + b_3(e_1 e_3) \pmod{p}$ and $C_2 = b_2 e_1 + b_3 e_2 \pmod{p}$.
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. $(C_1 + C_2(r - d_1))^{d_0} \equiv (b_1 + (a_1 b_3 - a_2 b_2)(g_1^{p_1-t_1} - g_2^{\frac{p_2-t_2}{u}}) + (a_1 b_2 - a_2 b_3)(g_1^{p_1-k_1} g_2^{\frac{k_2-t_2}{u}} - g_1^{k_1-t_1} g_2^{\frac{p_2-k_2}{u}}) + r C_2)^{d_0} \equiv (b_1 + b_2 y_1 + b_3 y_2 + r C_2)^{d_0} \equiv (b_0^e)^{d_0} \equiv b_0 \pmod{p}$.
□

Remark 3. An alternative encryption procedure would be not to conduct modular p . In this case the parameters (b_0, b_1, b_2) can be extracted.

4 Alternative Encryption

1. Message is $b_0 \in \mathbb{Z}_p$ and $b_1, b_2 \approx 2^{en}$.
2. Compute $b_3 = b_0^e - b_1 - b_2$.
3. Compute $C_1 = b_1 + b_2(e_2 e_3) + b_3(e_1 e_3)$ and $C_2 = b_2 e_1 + b_3 e_2$.
4. Send (C_1, C_2) to recipient.

5 Alternative Decryption

1. Compute $(C_1 + C_2(r - d_1))^{d_0} \equiv b_0 \pmod{p}$.
2. Solve the system of equations:

$$b_0^e = b_1 + b_2 + b_3 \tag{1}$$

$$C_1 = b_1 + b_2(e_2 e_3) + b_3(e_1 e_3) \tag{2}$$

$$C_2 = b_2 e_1 + b_3 e_2 \tag{3}$$

to obtain (b_1, b_2) .

Proposition 2. *The decryption procedure is correct.*

$$\begin{aligned} \text{Proof. } (C_1 + C_2(r - d_1))^{d_0} &\equiv (b_1 + (a_1b_3 - a_2b_2)(g_1^{p_1-t_1} - g_2^{\frac{p_2-t_2}{u}}) + (a_1b_2 - \\ a_2b_3)(g_1^{p_1-k_1}g_2^{\frac{k_2-t_2}{u}} - g_1^{k_1-t_1}g_2^{\frac{p_2-k_2}{u}}) + rC_2)^{d_0} &\equiv (b_1 + b_2y_1 + b_3y_2 + rC_2)^{d_0} \equiv \\ (b_0^e)^{d_0} &\equiv b_0 \pmod{p} \end{aligned}$$

To obtain (b_1, b_2) is trivial. \square

6 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 the modular p process. In section 5, $b_1b_2b_3 \gg C_1$ 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 5 unknowns in 2 equations, while in section 5 as solving 3 unknowns in 2 equations.
3. If $y_1 \equiv y_2 \equiv 1 \pmod{p}$ then $d_1 \equiv -e_3 \pmod{p}$ but $y_1, y_2 \not\equiv 1 \pmod{p}$.
4. From e_1 determine the tuple (g_1, k_1, t_1) .
5. From e_2 determine the tuple $(g_1, p_1, t_1, p_2, t_2)$.
6. From e_3 determine the tuple $(g_1, p_1, k_1, p_2, k_2, y_1, y_2)$.
7. Observe that $e_1 + e_2e_3 \not\equiv 1 \pmod{p}$ and $e_2 + e_1e_3 \not\equiv 1 \pmod{p}$. Thus, $C_1 + C_2 \not\equiv b_1 + b_2 + b_3 \equiv b_0^e \pmod{p}$.
8. Determine $x \in \mathbb{Z}_p$ such that $(C_1 + C_2(x))^{d_0} \equiv b_0 \pmod{p}$.