# A new security notion for asymmetric encryption Draft #8

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, bivariate function hard problem

## 1 Introduction

In this article we provide a new asymmetric encryption design based on the difficulty of solving ***a system of equations where the variables forming the system outnumber the equations***. Further discussion on this problem will be provided in the following sections.

## 2 A new security notion for asymmetric encryption

The following 2 sub-sections provide definitions and discussion on the the so-called ***underlying security primitive*** which the our asymmetric scheme relies on.

### 2.1 Solving a system of $m$ equations with $n$ variables where $n > m$

**Definition 1.** *To determine the value of variables (which are private) utilized initially to form a system of equations where the number of variables are more than the equations.*

### 2.2 Linear diophantine equations with infinitely many solutions

Before we discuss this subsection we will first observe a remark by Herrmann and May [1]. It discusses the ability to retrieve variables from a given linear Diophantine equation. But before that we will put forward a famous theorem of Minkowski relates the length of the shortest vector in a lattice to the determinant[1]:

**Theorem 1.** *In an $\omega$-dimensional lattice, there exists a non-zero vector $v$ with*

$$||v|| \leq \sqrt{\omega}\det(L)^{\frac{1}{\omega}}$$

We now put forward the remark.

*Remark 1.* There is a method for finding small roots of linear modular equations $a_1x_1 + a_2x_2 + ... + a_nx_n \equiv 0 \pmod{N}$ with known modulus $N$. It is further assumed that $\gcd(a_i, N) = 1$. Let $X_i$ be upper bound on $|y_i|$. The approach to solve the linear modular equation requires to solve a shortest vector problem in a certain lattice. We assume that there is only one linear independent vector that fulfills the Minkowski bound (Theorem 1) for the shortest vector. Herrmann and May showed that under this heuristic assumption that the shortest vector yields the unique solution $(y_1, ..., y_n)$ whenever

$$\prod_{i=1}^{n} X_i \leq N.$$

If in turn we have

$$\prod_{i=1}^{n} X_i > N^{1+\epsilon}.$$

then the linear equation usually has $N^{\epsilon}$ many solutions, which is exponential in the bit-size of $N$. So there is no hope to find efficient algorithms that in general improve on this bound, since one cannot even output all roots in polynomial time.

We now put forward a corollary.

**Corollary 1.** *A linear diophantine equation $f(x_1, x_2, ..., x_n) = a_1x_1 + a_2x_2 + ... + a_nx_n = N$, with*

$$\prod_{i=1}^{n} x_i > N^{1+\epsilon}.$$

*is able to ensure secrecy of the sequence $\mathbf{x} = \{x_i\}$.*

*Remark 2.* In fact if one were to try to solve the linear diophantine equation $N = a_1x_1 + a_2x_2 + ... + a_nx_n$, where $\prod_{i=1}^{n} X_i > N^{1+\epsilon}$, any method will first output a short vector $\mathbf{x} = \{x_i\}$ as the initial solution. Then there will be infinitely many values from this initial condition that is able to recontruct $N$.

## 3 A new asymmetric primitive

In this section we provide the reader with a working cryptographic primitive that is based upon the the mentioned "hard" problem as above.

• **Key Generation by Along**

INPUT: The size $n$ of the parameters and a pair of public prime integer values $(g_1, g_2) \approx 2^n$.

OUTPUT: A public key tuple $(n, s, e_1, e_2, e_3, g_1, g_2)$ and private keys $(d, v, a_1)$.

1. Generate random $n$-bit secret prime integers $(a_1, a_2, a_3, a_4, a_5)$.
2. Compute secret parameter $u = a_2 + a_3 + a_4, v = a_2 + a_4 + a_5, w = g_1(a_2 + a_3 + a_4) + g_2(a_2 + a_4 + a_5) \pmod{a_1}, x = g_1(a_2 + a_4 + a_5) + g_2(a_2 + a_3 + a_4) \pmod{a_1}$.
3. Compute public key-1: $e_1 = \frac{u}{v} \pmod{a_1}$.
4. Compute public key-2: $e_2 = \frac{w}{v} \pmod{a_1}$.
5. Compute public key-3: $e_3 = \frac{x}{v} \pmod{a_1}$.
6. Compute private $d = s^{-1} \pmod{a_1 - 1}$.
7. Return the public key pair $(n, s, e_1, e_2, e_3, g_1, g_2)$ and private keys $(d, v, a_1)$.

• **Encryption by Busu**

INPUT: The message $M$ tuple $(b_0, b_1, b_2)$ where $b_0 \approx 2^{n-1}$ and $b_1, b_2 \approx 2^{sn}$, and Along's public key set $(n, s, e_1, e_2, e_3, g_1, g_2)$.

OUTPUT: A ciphertext pair $(C_1, C_2)$.

1. Compute secret parameter $B_1 = g_1 b_1 + g_2 b_2, B_2 = g_2 b_1 + g_1 b_2$.
2. Compute ephemeral parameter $b_3 = b_0^s - b_2$.
3. Compute the first ciphertext $C_1 = B_1 e_1 - b_1 e_2 + b_3 e_3 + B_2$. Equivalently $C_1 = b_1(g_1 e_1 - e_2 + g_2) + b_2(g_2 e_1 + g_1) + b_3 e_3$.
4. Compute the second ciphertext $C_2 = b_1 + b_2 + b_3$.
5. Send the ciphertext pair $C = (C_1, C_2)$.

• **Decryption by Along**

INPUT: The ciphertext pair $C = (C_1, C_2)$ and private key tuple $(d, v, a_4)$.

OUTPUT: The message tuple $M = (b_0, b_1, b_2)$.

1. Compute $b_0 = \left(\frac{C_1 v}{x}\right)^d \pmod{a_1}$.
2. Compute $b_1 = C_2 - b_0^s$.
3. Solve the simultaneous equations $C_1 - b_1(g_1 e_1 - e_2 + g_2) = b_2(g_2 e_1 + g_1) + b_3 e_3$ and $C_2 - b_1 = b_2 + b_3$ to obtain $b_2$.
4. Return the message tuple $M = (b_0, b_1, b_2)$.

**Proposition 1.** *The decryption process is correct.*

*Proof.*

$$\begin{aligned} z \equiv C_1 v(\tfrac{1}{x}) &\equiv [B_1 u - b_1 w + b_3 x + B_2 v](\frac{1}{x}) \\ &\equiv [(b_1 g_1 + b_2 g_2)(a_2 + a_3 + a_4) + (b_1 g_2 + b_2 g_1)(a_2 + a_4 + a_5) - b_1 w + b_3 x](\frac{1}{x}) \\ &\equiv [b_1 w + b_2 x - b_1 w + b_3 x)](\frac{1}{x}) \\ &\equiv [b_2 + b_3] \\ &\equiv b_0^s \pmod{a_1} \end{aligned} \tag{1}$$

Then,

$$z^d \equiv b_0 \pmod{a_1} \tag{2}$$

We obtain the exact $b_0$ since $b_0 < a_1$, which ensures that no modular reduction has occurred. Next, to obtain $(b_1, b_2)$ is trivial.

In the next section we will point out locations where the fundamental source of security situated.

## 4 The fundamental source of security

We will dissect the mathematical structures introduced in the above so-called "cryptosystem". We will begin at looking at Along's parameters first.

### 4.1 Security of the ciphertext

- Observe the ciphertext given by $C_1 = b_1(g_1 e_1 - e_2 + g_2) + b_2(g_2 e_1 + g_1) + b_3 e_3$.
- We have $b_1, b_2, b_3 \approx 2^{sn}$ while $g_1 e_1 - e_2 + g_2, g_2 e_1 + g_1 \approx 2^{2n}$ and $e_3 \approx 2^n$.
- We have $C_1 \approx 2^{(s+2)n}$ while $b_1 b_2 b_3 \approx 2^{3sn}$. Thus, $b_1 b_2 b_3 > C_1$.
- As for $C_2 = b_1 + b_2 + b_3$ its clear that $b_1 b_2 b_3 > C_2$.
- To solve the simultaneous equations of $C_1, C_2$ it is a system of 2 equations with 3 variables.

### 4.2 Security of the public key

Observe the following public key equations:

$$e_1 v = u + a_1 t_1 \tag{3}$$

$$e_2 v = w + a_1 t_2 \tag{4}$$

$$e_3 v = x + a_1 t_3 \tag{5}$$

This is a system of 3 equations with the following unknown tuple $(v, u, w, x, a_1, t_1, t_2, t_3)$. Now lets assume the following strategy:

- Set $a_1 = a'_1, t_1 = t'_1, t_2 = t'_2$.
- Multiply equation (7) with $t'_2$ and (8) with $t'_1$.
- Obtain $v(e_1t'_2 - e_2t'_1) = ut'_2 - wt'_1$. Let $\delta = ut'_2 - wt'_1$. Proceeding to solve $v(e_1t'_2 - e_2t'_1) = \delta$ (diophantine equation in 2 variables), let $v_0 = K$ and $\delta = e_1t'_2 - e_2t'_1$.
- Then proceed to to solve for $ut'_2 - wt'_1 = K\delta$, which gives $u = u_0 = Ke_1$ and $w = w_0 = Ke_2$.
- Finally, $x_0 \equiv Ke_3 \pmod{a'_1}$.
- We will now analyze the result when an adversary proceeds to decrypt using the parameters that he obtained as mentioned above as outlined in Proposition 2.

$$\begin{aligned} z \equiv C_1v_0 &\equiv [B_1u_0 - b_1w_0v_0 + b_3x_0v_0 + B_2v_0] \\ &\equiv [B_1u_0 - b_1e_2v_0 + b_3e_3v_0 + B_2v_0] \\ &\equiv [(b_1g_1 + b_2g_2)(a'_2 + a'_3 + a'_4) + (b_1g_2 + b_2g_1)(K) - b_1e_2(K) + b_3e_3(K)] \\ &\equiv [(b_1g_1 + b_2g_2)(a'_2 + a'_3 + a'_4) + K(b_1g_2 + b_2g_1) - Kb_1e_2 + Kb_3e_3] \\ &\equiv [b_1(g_1u_0 - Ke_2 + Kg_2) + b_2(g_2u_0 + Kg_1) + Kb_3e_3)] \\ &\equiv [b_1(g_1(a'_2 + a'_3 + a'_4) - Ke_2 + Kg_2) + b_2(g_2(a'_2 + a'_3 + a'_4) + Kg_1) + Kb_3e_3] \pmod{a'_1} \end{aligned} \tag{6}$$

*Remark 3.* One can see that if via the candidate parameters $(K, a'_1, a'_2, a'_3, a'_4)$ equation (6) has both the following occurring then the adversary is able to decrypt passively. That is if the adversary obtains simultaneously:

$$g_1(a'_2 + a'_3 + a'_4) - Ke_2 + Kg_2 \equiv 0 \pmod{a'_1} \tag{7}$$

and

$$g_2(a'_2 + a'_3 + a'_4) + Kg_1 \equiv Ke_3 \pmod{a'_1} \tag{8}$$

That is, the adversary is able to reconstruct $b_2 + b_3 \equiv b_0^s \pmod{a'_1}$.

- The adversary can also view the above public key equations (3),(4) and (5) as:

$$\begin{aligned} (e_1 - 1)a_2 - a_3 + (e_1 - 1)a_4 + e_1a_5 &\equiv 0 \pmod{a_1} \\ (e_2 - g_1 + g_2)a_2 - g_1a_3 + (e_2 - g_1 + g_2)a_4 + (e_2 + g_2)a_5 &\equiv 0 \pmod{a_1} \\ (e_3 - g_1 + g_2)a_2 + g_2a_3 + (e_3 - g_1 + g_2)a_4 + (e_3 - g_1)a_5 &\equiv 0 \pmod{a_1} \end{aligned}$$

Let

$$G = \begin{pmatrix} e_1 - 1 & -1 & e_1 - 1 \\ e_2 - g_1 + g_2 & -g_1 & e_2 - g_1 + g_2 \\ e_3 - g_1 + g_2 & g_2 & e_3 - g_1 + g_2 \end{pmatrix}.$$

For benefit of adversary assume $det(G) \neq 0$. Next, the adversary chooses $a_5'$ and $a_1'$ where $gcd(a_1', det(G)) = 1$. This results in the following:

$$G \begin{pmatrix} a_2 \\ a_3 \\ a_4 \end{pmatrix} = -a_5' \begin{pmatrix} e_1 \\ e_2 + g_2 \\ e_3 - g_1 \end{pmatrix} \pmod{a_1'}.$$

The adversary would then obtain the tuple $(a_2', a_3', a_4')$ which would be used together with $a_1'$ and $a_5'$ to construct the tuple $(u_0, v_0, w_0, x_0)$. This would lead back to equation (7) and (8) in Remark 3.

*Remark 4.* From equations (7) and (8) we can have the following:

$$(g_1 - g_2)u_0 + K(g_2 - e_2 - g_1 + e_3) \equiv 0 \pmod{a_1'} \tag{9}$$

which can also be viewed as

$$(g_1 - g_2)u_0 + (g_2 - e_2 - g_1 + e_3)K - a_1'j = 0 \tag{10}$$

for some $j \in \mathbb{Z}$.

In other words, the adversary has to search for integers $(u_o, K, a_1', j)$ such that equation (10) holds together simultaneously with following equations (remember: $K = v_0$):

$$e_1 v_0 \equiv u_0 \pmod{a_1'} \tag{11}$$
$$e_2 v_0 \equiv w_0 \pmod{a_1'} \tag{12}$$
$$e_3 v_0 \equiv x_0 \pmod{a_1'} \tag{13}$$

**This is analogous to the subset sum problem?**

*Remark 5.* If the parameter $a_1$ is given as a public parameter, will the above situation (experienced by the adversary) remain?

## 5 Collision type attacks

We dedicate this section to discuss the possibility of designing a collision type attack on our new scheme.

## 6 Achieving IND-CCA2

It is obvious that the new scheme achieves IND-CPA. But how about IND-CCA2?

## 7 Conclusion

This paper presents a new cryptosystem that has advantages in the following areas against known public key cryptosystems:

1. It has a complexity order of $O(n^2)$ during encryption and $O(n^3)$ during decryption.
2. Mathematically, an adversary does not have any advantage to attack the published public key (i.e. because of the number of variables is much more than the equations) or the ciphertext.
3. Does the new scheme produce "cylic-type" features that would allow a collision type attack to be designed?
4. If a collision type attack cannot be designed, how do we propose to evaluate the scheme in order to suggest a minimum key length?

## 8 Acknowledgment

I would like to acknowledge Prof Abderrahmane Nitaj of University of Caen, France and Dr Yanbin Pan of the Chinese Academy of Science, China for all discussion and feedbacks while designing the scheme.

## References

1. Herrmann, M., May, A.: Solving Linear Equations Modulo Divisors: On Factoring Given Any Bits. In ASIACRYPT 2008, volume 5350 of Lecture Notes in Computer Science, pp. 406-424. Springer-Verlag (2008)