Jurnal Teknik Komputer AMIK BSI
Volume 8, No.2, Juli 2022
P-ISSN 2442-2436, E-ISSN: 2550-0120
Akreditasi Ristekdikti, No: 36/E/KPT/2019 (Sinta 4)
DOI: 10.31294/jtk.v4i2

Kriptanalisis Kunci Publik ElGamal Menggunakan Jaringan Syaraf Ridge Polynomial

Rina Pramitasari[1], Imam Rofiki[2]

[1]Universitas Amikom Yogyakarta
[2]Universitas Negeri Malang
[1]e-mail: rina.pramitasari@amikom.ac.id
[2]e-mail: imam.rofiki.fmipa@um.ac.id

Diterima	Direvisi	Disetujui
31-07-2022	31-07-2022	01-08-2022

Abstrak – Keamanan kriptografi dapat di uji dengan menggunakan kriptanalisis. Tingkat keamanan kriptografi kunci publik ElGamal tergantung pada tingkat kesulitan menemukan logaritma diskrit. Penelitian ini melakukan kriptanalisis kunci public ElGamal dengan menggunakan Jaringan Syaraf Ridge Polynomial. Hasil penelitian menunjukkan Jaringan Syaraf Ridge Polynomial mampu menentukan kunci private. Dengan menggunakan metode yang diusulkan dapat mengatasi iterasi yang terjebak ke dalam minimal local. Oleh karena itu menghasilkan pengujian iterasi yang konvergen.

Kata Kunci: Jaringan Syaraf Ridge Polynomial, Kunci Public ElGamal, Kriptanalisis

Abstract *– Cryptographic security can be tested using cryptanalysis. The level of security of the ElGamal public key cryptography system depends on the difficulty of finding the discrete logarithm. This study performs ElGamal's public key cryptanalysis using Ridge Polynomial Neural Network. The results showed that the Ridge Polynomial Neural Network was able to determine the private key. Using the proposed method can overcome iterations that are stuck into the local minimum. Therefore it produces a convergent iteration test.*

Keywords: Ridge Polynomial Neural Network., ElGamal Public Key, Cryptoanalysis

PENDAHULUAN

Tingkat keamanan sistem kriptografi kunci publik ElGamal tergantung pada tingkat kesulitan menemukan logaritma diskrit. Menggunakan algoritma ElGamal dapat dienkripsi dalam berbagai ciphertext, sehingga memberikan tambahan lapisan keamanan. Tetapi algoritma ElGamal memiliki kelemahan yaitu meningkatnya ukuran ciphertext dua kali lipat dari plaintext. Sama seperti enkripsi asimetris lainnya, algoritma ElGamal lebih lambat dan membutuhkan lebih banyak waktu komputasi (Jitcharadze & Iavich, 2020).

Kriptanalisis dapat digunakan untuk menguji keamanan kriptografi. Jaringan syaraf tiruan dipilih untuk menyelesaikan masalah tersebut dimana terdistribusi parallel pada prosesor dan mempunyai penyimpanan pengalaman dan tetap tersedia untuk digunakan kembali (Markowska-Kaczmar & Switek, 2009)

Jaringan pi-sigma adalah jaringan feedforward yang berbentuk perkalian dari penjumlahan input. Jaringan pi-sigma menghasilkan neuron input, neuron *single hidden* berbentuk unit penjumlahan dan neuron output berbentuk unit perkalian. Bobot yang menghubungkan neuron input ke neuron single hidden diperbarui selama proses pelatihan. Sedangkan, bobot yang menghubungkan neuron single hidden ke neuron output adalah tetap (fixed). Derajat jaringan pi-sigma adalah jumlah unit dari lapisan hidden (Epitropakis & Vrahatis, 2005).

Menurut Karnavas & Papadopoulos (2004), jaringan pi-sigma mendapatkan hasil aproksimasi yang terbatas. Untuk mendapatkan aproksimasi menyeluruh yaitu dengan cara menjumlahkan output dari beberapa jaringan pi-sigma. Hasil

penggabungan jaringan pi-sigma disebut jaringan ridge polynomial (Karnavas & Papadopoulos, 2004).

Ren dan Harn (2006) mengusulkan *ring signature* berdasarkan skema tanda tangan ElGamal. Mereka mengklaim bahwa skema yang diusulkan memiliki keuntungan sebagai berikut *ring signature* yang diusulkan secara inheren merupakan *ring signature* yang dapat dikonversi dan memungkinkan penandatangan pesan yang sebenarnya untuk membuktikan kepada pemverifikasi bahwa hanya dia yang mampu menghasilkan *ring signature*. Kemudian Wang, Han, Deng, & Zhang (2009), menemukan *ring signature* yang tidak dapat memenuhi keuntungan ini, yaitu, tidak dapat diubah. Untuk mewujudkan konvertibilitas, mereka meningkatkan skema aslinya. Skema *ring signature* yang ditingkatkan dapat memiliki keuntungan dan aman.

Inam dan Ali (2018) mengusulkan sistem kriptografi mirip ElGamal berdasarkan matriks *over groupring*. Menggunakan pendekatan aljabar linier, Jia, Wang, Zhang, Wang, dan Liu (2018) memberikan cryptanalysis untuk cryptosystem dan mengklaim bahwa serangan mereka dapat memulihkan semua kunci yang setara. Namun, mereka telah meningkatkan pendekatan kriptanalisisnya dan menurunkan semua pasangan kunci yang setara yang dapat digunakan untuk menghancurkan sistem kriptografi mirip ElGamal yang diusulkan oleh Inam dan Ali. Menggunakan dekomposisi matriks *over groupring* ke matriks berukuran lebih dari *over ring*, Pandey, Gupta, dan Singh (2021) membuat algoritma cryptanalysing lebih praktis dan efisien. Mereka membuktikan bahwa kriptosistem ElGamal yang diusulkan oleh Inam dan Ali tidak mencapai keamanan IND-CPA dan IND-CCA.

Joye dan Quisquater (1998) menyurvei implementasi tipe RSA berdasarkan urutan Lucas dan pada kurva eliptik. Fokus utamanya adalah beberapa serangan yang diketahui pada RSA diperluas ke sistem LUC, KMOV, dan Demytko. Penelitian Shamir (1995) menyajikan ikhtisar algoritma kriptografi asinkron RSA, konsep matematika yang mendasarinya dan serangan yang diusulkan padanya, menggunakan algoritme kepemilikan baru untuk menggabungkan pendekatan matematika dan brute-force. Berdasarkan hasil penelitian-penelitian sebelumnya, terdapat celah untuk menyelidiki lebih lanjut kriptanalisis kunci public ElGamal dengan menggunakan Ridge Polynomial Neural Network. Oleh karena itu, penelitian ini dilakukan untuk mengisi gap penelitian tersebut.

METODOLOGI PENELITIAN

Langkah penelitian adalah sebagai berikut. Menggunakan Ridge Polynomial Neural Network untuk kriptanalisis kunci public ElGamal (Hussein, Mstafa, Mohammed, & Younis, 2022).

1. Kriptografi ElGamal

Kriptografi ElGamal berdasarkan protocol pertukaran kunci Diffie-Hellman, yang mana keamanannya tergantung pada pemecahan masalah logaritma diskrit. Informasi kunci public (p, a, y) dan kunci privat (x) sehingga

$y = a^x \ mod \ p$ ----------(1)

1.1. Pembuatan Pasangan Kunci

Penerima kemudian menerima pesan terenkripsi, menetapkan kunci public dan kunci privat yang diperlukan untuk enkripsi dan dekripsi. Prosesnya sebagai berikut.

a. Generate p (sebuah bilangan prima terbesar)
b. Inisialisasi a (sebuah akar primitive dari p) sehingga $1 < a < p - 1$
c. Inisialisasi x (sebuah bilangan bulat random) sehingga $1 < x < p - 2$
d. Hitung y seperti pada persamaan (1)

Setelah pasangan kunci dibuat, penerima hanya meneruskan kunci public kepada pengirim sehingga kunci privat tetap rahasia.

1.2. Algoritma Proses Enkripsi

Setelah menerima kunci public dari penerima, pengirim mengenkripsi pesan yang akan diberikan. Prosesnya sebagai berikut.

a. Ubah setiap karakter di dalam pesan yang akan diberikan (m) ke representasi unicode yang sesuai $1 \leq m_i \leq p - 1$
b. Inisialisasi k (sebuah bilangan bulat random yang mewakili kunci privat) sehingga $1 < k < p - 2$
c. Hitung seperti pada persamaan (2)
d. Menggunakan persamaan (3) untuk mengenkripsi setiap m_i dimana $i = 0, 1, 2, \cdots, L - 1$ dan L mewakili panjang m

$d = a^x \ mod \ p$ ----------(2)

$z_i = (y^k \times m_i) \ mod \ p$ ----------(3)

Setelah proses enkripsi selesai, pengirim meneruskan ciphertext (d, z) ke penerima dan menjaga kerahasian kunci privat.

1.3. Algoritma Proses Dekripsi

Setelah menerima ciphertext (d, z) dari pengirim, penerima mendekripsi pesan enkripsi. Prosesnya sebagai berikut.

- Hitung d seperti pada (4)
- Gunakan rumus (5) untuk mendekripsi setiap z_i

$r = d^{(p-1-x)}$----------(4)

$m_i = (r \times z_i) \ mod \ p$----------(5)

2. Ridge Polynomial Neural Network

2.1. Jaringan Pi-Sigma

Persamaan output jaringan pi-sigma dikenalkan oleh Gupta, Jin, dan Homma (2004) sebagai berikut.

$$y = f\left(\prod_{i=1}^{N}\left[\sum_{j=1}^{n} w_{ij}x_j + w_i\right]\right) \quad \text{----------(6)}$$

Dengan:

w_{ij} = Bobot terbarui.
w_i = Threshold untuk neuron hidden ke-j.
$f(x)$ = Fungsi aktivasi nonlinear.
$(n+1)N$ = Jumlah total bobot pada jaringn pi-sigma order N dengan n input.

2.2. Jaringan Ridge Polynomial

Jaringan ridge polynomial adalah perluasan dari jaringan pi-sigma yang menggunakan bentuk *ridge polynomial*.

Definisi 1 *Diberikan himpunan compact* $K \subset \mathbb{R}^d$, *semua fungsi didefinisikan pada K dengan bentuk,* $f(\langle ., \mathbf{w}\rangle): K \mapsto \mathbb{R}$. *Dimana* $\boldsymbol{w} \subset \mathbb{R}^d$ *dan* $f(.): \mathbb{R} \mapsto \mathbb{R}$ *adalah kontinu, disebut* ***fungsi ridge***

Ridge polynomial adalah fungsi ridge yang dapat dijelaskan sebagai berikut:

$$\sum_{i=0}^{N}\sum_{j=0}^{M} a_{ij}\,\langle \boldsymbol{x}, \boldsymbol{w_{ij}}\rangle^i \quad \text{----------(7)}$$

untuk beberapa $a_{ij} \in \mathbb{R}$ dan $\langle \boldsymbol{x}, \boldsymbol{w_{ij}}\rangle \in \mathbb{R}^d$

Teorema menyatakan setiap polynomial multivariate dapat dinyatakan dalam istilah ridge polynomial, dan dapat direalisasikan dengan jaringan ridge polynomial yang disesuai kan.

Teorema 1 *Setiap polynomial multivariate dapat dinyatakan sebagai ridge polynomial.*

$$p(x) = \sum_{j=0}^{k}\sum_{m=1}^{n_j} c_{jm}\,x^{i_{jm}} \Leftrightarrow p(x) = \sum_{j=1}^{N}\prod_{i=1}^{j}\left(\langle x, w_{ji}\rangle + w_{ji}\right) \quad \text{--(8)}$$

Jaringan ridge polynomial memiliki kemampuan pemetaan yang bagus didalam artinya bahwa setiap fungsi kontinu pada himpunan *compact* di $\mathbb{R}^d$ dapat diaproksimasi secara merata dengan jaringan.

Teorema 2 *Setiap fungsi kontinu pada himpunan compact di* $\mathbb{R}^d$ *dapat diaproksimasi secara merata dengan jaringan ridge polynomial.*

Persamaan output jaringan ridge polynomial dapat dijelaskan sebagai berikut

$$y = \emptyset\left(\sum_{j=1}^{N}\prod_{i=1}^{j}\left(\langle x, w_{ij}\rangle + \theta_{ji}\right)\right) \quad \text{----------(9)}$$

Atau

$$y = f\left(\sum_{j=1}^{N}\prod_{i=1}^{j}\left(\sum_{k=1}^{n} w_{ijk}x_k + \theta_{ji}\right)\right) \quad \text{----------(10)}$$

Dimana:

j = Jumlah jaringan pi-sigma dari 1 sampai *N*,
i = Jumlah jaringan pi-sigma order dari *i* sampai *j*,
k = Jumlah input dari 1 sampai *n*,
w_{ijk} = Bobot yang diperbarui dari input x_k ke unit PSN order ke-*i* dari PSN ke-*j*.
$f(x)$ = Fungsi aktivasi nonlinear.
$N(N+1)(n+1)/2$ = Jumlah total bobot yang digunakan

2.3. Pelatihan Ridge Polinomnial Neural Network

a. Menambah jaringan pi-sigma
   Jika belum terpenuhi maka lakukan b – j
b. Inisialisasi bobot dan bias
c. STOP jika belum mencapai eror atau maksimum iterasi maka lakukan d – i.
d. Untuk setiap pola data maka lakukan e – h.
e. Menerima masukan pola data
f. Menghitung output jaringan pi-sigma dengan persamaan (6)
g. Menghitung delta bobot jaringan pi-sigma

$$\delta_l = \eta(d - y)y' \prod_{\substack{z=1 \\ z \neq l}}^{j} h_z \quad \text{----------(11)}$$

$\Delta w_{kl} = \delta_l x_k$
$\Delta w_{0l} = \delta_l$
Dimana:
η = Learning rate.
d_i = Target output jaringan pi-sigma
y_i = Hasil 0utput jaringan pi-sigma
y_i' = Turunan pertama dari fungsi aktifasi nonlinear.
h_z = Neuron hidden dimana *z* adalah jumlah neuron hidden tetapi bukan bagian dari indek ke-*l*.
x_k = masukan pola data.

h. Memperbarui bobot dan bias
   w_{kl} (baru) = w_{kl} (lama) + Δw_{kl} ----------(12)
   Dimana:
   w_{kl} = Bobot yang diperbarui
i. Memeriksa kondisi STOP
j. Menghitung output jaringan ridge polynomial. Output jaringan ridge polynomial adalah jumlah output tiap PSN. Dengan persamaan (10)

HASIL DAN PEMBAHASAN

1. Pengumpulan Data

Penelitian menggunakan pemrograman Java. Data pelatihan menggunakan punci public latih berjumlah 20 baris dan cyphertext latih berjumlah 20 baris. Data pengujian menggunakan kunci public uji berjumlah 6 baris dan cyphertext uji berjumlah 6 baris. Target data pelatihan dan data pengujian menggunakan kunci privat. Pada Ridge Polynomial Neural Network, maksimum iterasi = 5000, error target = 0,0001. Data pelatihan dan pengujian berturut-turut disajikan pada Gambar 1 dan Gambar 2.

no	p	a	y	d	z1	z2	kunci
1	40819	18494	31507	38104	45126	113102	39827
2	45757	15256	35212	1426	4397	461	42641
3	47513	18359	44802	3128	3788	23120	35111
4	49409	33757	7264	11842	4325	838	41611
5	41843	14276	27764	99126	1367	5114	35759
6	57667	16429	19923	3993	3756	9859	53693
7	62563	28830	23579	3554	4481	810	58537
8	55291	40877	10955	6348	6686	42108	33247
9	33493	9426	13466	6778	4864	52118	33053
10	65327	61501	37083	11742	9176	3274	38119
11	63103	10125	47057	11573	2067	44104	53819
12	50383	8969	9079	6737	3853	3172	37619
13	12123	12123	25896	10982	5656	4912	34259
14	52127	28149	41293	5598	2012	5767	36587
15	43487	15862	30846	2576	7145	5964	35401
16	39631	6834	17589	95124	2411	7334	38039
17	45763	2487	23376	10235	6516	6474	37021
18	60449	41247	59165	7931	2007	590	40283
19	51343	2650	11421	6184	5878	9620	48799
20	62827	15804	25750	38	5978	759	55051

Sumber: Data Penelitian 2022

Gambar 1. Data Pelatihan

no	p	a	y	d	z1	z2	kunci
1	40699	22334	5132	11069	566	328	35407
2	40087	1887	16749	1637	6112	12737	37607
3	56957	48009	2364	2194	3112	6132	47711
4	37171	11039	15812	9085	3317	10138	33071
5	61129	59652	42303	10160	10576	45118	47797
6	64747	58786	33674	348	99107	96125	60169

Sumber: Data Penelitian 2022

Gambar 2. Data Pengujian

2. Hasil Pengujian

Hasil pengujian pertama Ridge Polinomial Neural Network menggunakan Jaringan Pi-Sigma order 3 dengan learning rate 0.15 , 0.1 dan 0.05 adalah sebagai berikut.

Tabel 1 Hasil Uji Dengan learning rate 0.15

No	Hasil	Error
1	33108	0.08483414637135502
2	33151	0.16443711373659373
3	34576	0.4847000393213097
4	33090	7.124339667476889E-4
5	35265	0.46246071692508206
6	47379	0.4719781229690748

Sumber: Data Penelitian 2022

Tabel 2 Hasil Uji Dengan learning rate 0.1

No	Hasil	Error
1	33127	0.08412376875710241
2	33195	0.16279143732411422
3	35227	0.46066643410597635
4	33099	0.0010500313383376655
5	35260	0.46262387126297716
6	36803	0.8622498034772367

Sumber: Data Penelitian 2022

Tabel 3 Hasil Uji Dengan learning rate 0.05

No	Hasil	Error
1	33156	0.08305947132342063
2	33307	0.15868278841571853
3	35392	0.45460471585381557
4	33124	0.00196812727316846
5	35108	0.4682288817312826
6	44461	0.5796434507028132

Sumber: Data Penelitian 2022

Berdasarkan data pada Tabel 1, Tabel 2, dan Tabel 3, data pengujian yang hampir mendekati akurasi paling bagus adalah no 4, disusul no 1, no 2. Dengan learning rate tertentu yang tidak mempengaruhi hasil. Data pengujian tersebut adalah kunci paling kecil dari kunci yang lainnya.

Hasil pengujian kedua Ridge Polinomial Neural Network menggunakan Jaringan Pi-Sigma order 5 dengan learning rate 0.15 , 0.1 dan 0.05 adalah sebagai berikut.

Tabel 4 Hasil Uji Dengan learning rate 0.15

No	Hasil	Error
1	33092	0.08542300385144502
2	33151	0.16441554242291387
3	34427	0.49018896634010795
4	33083	4.5795820925449525E-4
5	33484	0.5281780954439781
6	42252	0.6611773811485612

Sumber: Data Penelitian 2022

Tabel 5 Hasil Uji Dengan learning rate 0.1

No	Hasil	Error
1	33119	0.08442822941627344
2	33184	0.16321773887015004
3	34801	0.47641132497665795
4	33097	9.832781095766366E-4
5	34912	0.4754796523485369
6	44435	0.5806225252296315

Sumber: Data Penelitian 2022

Tabel 6 Hasil Uji Dengan learning rate 0.05

No	Hasil	Error
1	33156	0.08305156655504309
2	33306	0.15869683986339836
3	35387	0.45479118483964753
4	33125	0.002020715308643854
5	35308	0.4608474409330202
6	44569	0.5756516084077427

Sumber: Data Penelitian 2022

Berdasarkan data pada Tabel 4, Tabel 5, dan Tabel 6, data pengujian yang hampir mendekati akurasi paling bagus adalah no 4, disusul no 1, no 2. Dengan learning rate tertentu yang tidak mempengaruhi hasil. Data pengujian tersebut adalah kunci paling kecil dari kunci yang lainnya.

Hasil pengujian ketiga Ridge Polinomial Neural Network menggunakan Jaringan Pi-Sigma order 7 dengan learning rate 0.15 , 0.1 dan 0.05 adalah sebagai berikut.

Tabel 7 Hasil Uji Dengan learning rate 0.15

No	Hasil	Error
1	33107	0.08486721396843817
2	33145	0.1646500010076751
3	34524	0.48663728619849
4	33089	6.722982850424472E-4
5	34462	0.4920869475979674
6	49059	0.409973915864449

Sumber: Data Penelitian 2022

Tabel 8 Hasil Uji Dengan learning rate 0.1

No	Hasil	Error
1	33128	0.0841003689399227
2	33194	0.16283528900694605
3	35297	0.45811288118973387
4	33099	0.0010357095879196753
5	36465	0.41815844100473964
6	40885	0.7116230573140722

Sumber: Data Penelitian 2022

Tabel 9 Hasil Uji Dengan learning rate 0.05

No	Hasil	Error
1	33159	0.08292921478027712
2	33317	0.1583030102553685
3	35622	0.4461166172501507
4	33124	0.0019867416685303034
5	35248	0.463074552256189
6	40078	0.7414051967571027

Sumber: Data Penelitian 2022

Berdasarkan data pada Tabel 7, Tabel 8, dan Tabel 9, data pengujian yang hampir mendekati akurasi paling bagus adalah no 4, disusul no 1, no 2. Dengan learning rate tertentu yang tidak mempengaruhi hasil. Data pengujian tersebut adalah kunci paling kecil dari kunci yang lainnya

KESIMPULAN

Berdasarkan hasil penelitian kriptanalisis ElGamal menggunakan Ridge Polinomial Neural Network yang telah dilakukan berhasil mencari kunci privat dengan akurat. Tingkat keberhasilan tersebut dipengaruhi oleh beberapa variable factor. Nilai error yang dihasilkan tidak bergantung pada besar kecilnya learning rate dan jumlah order jaringan Pi-Sigma. Data pengujian dari no 1 sampai 6, yang hampir mendekati akurasi adalah ada no 4, disusul no 1, no 2. Data pengujian tersebut adalah kunci paling kecil dari kunci yang lainnya. Dari hasil pembahasan didapatkan kesimpulan bahwa hasil output yang mendekati nilai kunci terkecil yang dapat mengatasi iterasi yang terjebak ke dalam minimal lokal. Hal ini karena menghasilkan pengujian iterasi yang konvergen.

REFERENSI

Epitropakis, M. G., & Vrahatis, M. N. (2005). Root finding and approximation approaches through neural networks. *ACM SIGSAM Bulletin*, *39*(4), 118-121.

Gupta, M., Jin, L., & Homma, N. (2004). *Static and dynamic neural networks: From fundamentals to advanced theory*. John Wiley & Sons.

Hussein, H. I., Mstafa, R. J., Mohammed, A. O., & Younis, Y. M. (2022). An enhanced ElGamal cryptosystem for image encryption and decryption. In *2022 International Conference on Computer Science and Software Engineering (CSASE)* (pp. 337-342). IEEE.

Inam, S., & Ali, R. (2018). A new ElGamal-like cryptosystem based on matrices over groupring. *Neural Computing and Applications*, *29*(11), 1279-1283.

Jia, J., Wang, H., Zhang, H., Wang, S., & Liu, J. (2018). Cryptanalysis of an ElGamal-Like Cryptosystem Based on Matrices Over Group Rings. In *Chinese Conference on Trusted Computing and Information Security* (pp. 255-269). Springer, Singapore.

Jintcharadze, E., & Iavich, M. (2020). Hybrid implementation of Twofish, AES, ElGamal and RSA cryptosystems. In *2020 IEEE East-West Design & Test Symposium (EWDTS)* (pp. 1-5). IEEE.

Ren, J., & Harn, L. (2006). Ring signature based on ElGamal signature. In *International Conference on Wireless Algorithms, Systems, and Applications* (pp. 445-456). Springer, Berlin, Heidelberg.

Joye, M., & Quisquater, J. J. (1998). Cryptanalysis of RSA-type cryptosystems: A visit. *Network Threats:* DIMACS Series in Discrete Mathematics and Theoretical Computer Science, *American Mathematical Society*, 38, 21-31.

Karnavas, Y. L., & Papadopoulos, D. P. (2004). Excitation control of a synchronous machine using polynomial neural networks. *Journal of Electrical Engineering*, *55*(7-8), 169-179.

Markowska-Kaczmar, U., & Switek, T. (2009). Combined unsupervised-supervised classification method. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (pp. 861-868). Springer, Berlin, Heidelberg.

Pandey, A., Gupta, I., & Singh, D. K. (2021). Improved cryptanalysis of a ElGamal cryptosystem based on matrices over group

rings. *Journal of Mathematical Cryptology*, *15*(1), 266-279.

Shamir, A. (1995). RSA for paranoids. *CryptoBytes*, *1*, 1-4.

Wang, H., Han, S., Deng, C., & Zhang, F. (2009). Cryptanalysis and improvement of a ring signature based on ElGamal signature. In *2009 WRI World Congress on Software Engineering* (Vol. 3, pp. 397-401). IEEE.