ANALISIS PERBANDINGAN WAKTU ENKRIPSI DAN DEKRIPSI PADA ALGORITMA ECC DAN RSA

Yosef Adrian[1], Chikana Friscilla[2], Nicholas Suardiman[3], Antoni Wijaya[4], Sudimanto[5]

[1, 2, 3, 4, 5] Sekolah Tinggi Manajemen Informatika dan Komputer
LIKMI Jl. Ir. H. Juanda no. 96 Bandung

kristian_yosef@yahoo.com[1]

ABSTRAK

Informasi merupakan kebutuhan utama bagi masyarakat. Hadirnya internet membuat pertukaran informasi menjadi mudah bagi semua orang. Namun hal tersebut menyebabkan peluang munculnya kejahatan *cyber* dan dapat terjadi pada siapa saja. Untuk menangani hal tersebut dikembangkanlah sistem keamanan data dimana salah satunya adalah kriptografi. Kriptografi adalah teknik merubah pesan yang dapat dimengerti menjadi pesan yang tidak dapat dimengerti, lalu diubah kembali menjadi pesan yang dapat dimengerti. Algoritma yang digunakan pada analisis ini menggunakan algoritma *Elliptic Curve Cryptography* (ECC) dan *Rivest Shamir Adleman* (RSA). Data yang digunakan pada analisis diambil dari jurnal *International Journal of Applied Engineering Research* dengan artikel *RSA and ECC: A Comparative Analysis* dan jurnal *International Journal of Network Security* dengan artikel *Performance Analysis of RSA and Elliptic Curve Cryptography* untuk dianalisis dan dibandingkan. Berdasarkan hasil analisis yang telah dilakukan, bahwa algoritma RSA memiliki waktu lebih cepat dalam melakukan enkripsi, sedangkan untuk algoritma ECC lebih cepat dalam melakukan dekripsi. Perbedaan besar kunci yang dimiliki oleh masing-masing algoritma mempengaruhi waktu pada saat enkripsi ataupun melakukan dekripsi.

Kata Kunci : *Elliptic Curve Cryptography* (ECC), Kriptografi, *Rivest Shamir Adleman* (RSA)

ABSTRACT

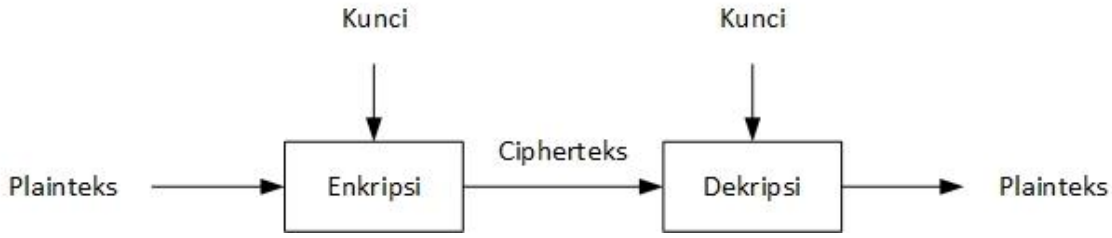
Information become a top priority for community. The advent of internet made us exhange information easily. However, this creates opportunities for cyber crimes to arise and may attack anyone. To prevent this problem, a data security system was developed in which one of security system is cryptography. Cryptography is a technique to convert a writen message that can be understood into a message that can't be understood, then it converts back to it's original state. The algorithms used in this analysis are Elliptic Curve Cryptography (ECC) and Rivest Shamir Adleman (RSA). The data used in the article was taken from International Journal of Applied Engineering Research with the name of RSA and ECC: A Comparative Analysis and from International Journal of Network Security with the name of Performance Analysis of RSA and Elliptic Curve Cryptography to be analyzed and compared. Based on the result, RSA algorithm have a quicker time in encryption while ECC algorithm have a quicker time in decryption. The difference in key size in each of the algorithm affects both encryption and decryption time.

Keywords: Elliptic Curve Cryptography (ECC), Cryptography, Rives Shamir Adleman (RSA)

1 PENDAHULUAN

Zaman modern ini membuat teknologi informasi menjadi kebutuhan utama bagi masyarakat. Hadirnya internet membuat pertukaran informasi menjadi mudah diakses oleh semua orang. Namun hal tersebut menjadi peluang muncul banyaknya kejahatan *cyber* dan dapat terjadi pada siapa saja. Dalam menangani hal tersebut, dikembangkanlah berbagai sistem keamanan data yang salah satunya adalah teknik penyandian data atau dapat disebut dengan kriptografi[1].

Kriptografi terbagi menjadi dua jenis yaitu kriptografi simetris dimana menggunakan satu kunci untuk mengenkripsi dan mendekripsi data, sementara kriptografi asimetris menggunakan dua kunci dimana kunci untuk mengenkripsi dinamakan *public key* sementara kunci untuk mendekripsi dinamakan *private key* [2][3]. Pada kriptografi terdapat proses enkripsi dan dekripsi dimana enkripsi adalah proses merubah pesan yang dapat dimengerti (*plainteks*) menjadi pesan yang tidak dapat dimengerti secara kasat mata (*cipherteks*). Sedangkan dekripsi adalah proses penerjemahan *cipherteks* agar kembali menjadi *plainteks* kembali[4]. Kedua proses tersebut dapat dilihat pada gambar berikut,



Gambar 1.
Cara Kerja Kriptografi Asimetris Dalam Melakukan Enkripsi dan Melakukan Dekripsi Data[4]

Algoritma kriptografi asimetris yang dipakai untuk dibandingkan pada analisis ini adalah algoritma *Rivest Shamir Adleman* (RSA) dan *Elliptic Curve Cryptography* (ECC). Algoritma RSA merupakan algoritma asimetris tertua dan algoritma ini mudah dipahami serta memiliki tingkat keamanan cukup tinggi sehingga cukup terkenal pada masanya sementara algoritma ECC diciptakan pada abad ke-19 dengan struktur yang cukup sulit tetapi memiliki kinerja yang dapat diharapkan untuk kedepannya [5]. Tingkat keamanan pada algoritma ECC dan RSA memiliki kekuatan keamanan yang sama namun berbeda pada ukuran besar kuncinya [6]. Analisis perbandingan terhadap algoritma ECC dan RSA dilakukan dengan melihat perbedaan waktu enkripsi dan dekripsi serta hal yang menyebabkan perbedaan tersebut.

Analisis perbandingan dengan menggunakan algoritma RSA dan ECC pernah dilakukan oleh Wahid Miftahul Ashari yang mana analisis perbandingan yang dilakukan hanya digunakan untuk *key agreement* dan *digital signature* [7]. Penelitian lain yang dilakukan oleh Niko, menyimpulkan bahwa penggunaan spesifikasi perangkat keras yang diperlukan untuk menerapkan algoritma RSA lebih tinggi daripada algoritma ECC karena pembentukan dari kunci RSA. Penelitian ini dilakukan pada protokol Secure Socket Layer (SSL) [8].

2 ALGORITMA RSA, ECC SERTA BESAR KUNCINYA

Data yang digunakan dalam analisis ini menggunakan dua buah data. Data yang pertama berasal dari artikel yang berjudul *ECC and RSA: A Comparative Analysis* dan data yang kedua berasal dari artikel yang berjudul *Performance Analysis of RSA and Elliptic Curve Cryptography* [6] [9]. Dari artikel tersebut, data yang diambil mengenai struktur algoritma

ECC dan RSA serta besar kunci pada setiap algoritma dengan standar tingkat keamanan *National Institute of Standards and Technology* (NIST) [10] [9]. Data yang sudah diambil serta sudah disatukan secara sistematis, akan digunakan untuk proses pembandingan serta penganalisisan pada jurnal ini demi keperluan pemahaman mengenai kriptografi asimetris. Berikut ini adalah data yang diperlukan serta sudah diambil dari kedua jurnal tersebut mengenai pengertian RSA, ECC, dan besar kunci yang direkomendasikan oleh NIST.

2a. Rivest, Shamir and Adleman (RSA)

Algoritma RSA merupakan salah satu algoritma kriptografi *public key* yang paling umum digunakan dan tertua. Keamanannya terletak pada masalah faktorisasi bilangan bulat. Algoritma ini cocok digunakan untuk melakukan asimetrik kriptografi dengan memberikan keamanan, kerahasiaan, integritas, dan autentikasi perjalanan dan penyimpanan pesan yang dapat dipercaya namun mengorbankan kecepatan. Untuk keamanan data yang lebih baik dan lebih kuat, diperlukan ukuran kunci yang lebih besar, yang berarti lebih banyak overhead pada komputasi sistem [9][5]. RSA memiliki 3 langkah dalam prosesnya yaitu pembangkitan kunci yang dapat dilihat pada Tabel 1, enkripsi yang dapat dilihat pada Tabel 2, dan dekripsi yang dapat dilihat pada Tabel 3. Berikut adalah algoritma dari tiap langkahnya :

Tabel 1
Struktur *Key Generation* Algoritma Kriptografi Asimetris RSA [9][6]

Pembangkitan Kunci	
Langkah 1	Pilih P,Q *(P dan Q adalah bilangan prima, P≠Q*
Langkah 2	Menghitung n = p x q
Langkah 3	Menghitung Φ(n) = (p-1) x (q-1)
Langkah 4	Pilih Bilangan Bulat e (int e) gcd(Φ(n), e) = 1 dan (1<e< Φ(n))
Langkah 5	Menghitung d [$d=e^{-1} \pmod{\Phi(n)}$]
Langkah 6	Public key PU = {e, n}
Langkah 7	Private key PR = {d, n}

Tabel 2
Struktur Enkripsi Algoritma Kriptografi Asimetris RSA [9][6]

Enkripsi	
Langkah 1	*Plaintext*: M<n
Langkah 2	*Ciphertext*: $C=M^e \bmod n$

Tabel 3
Struktur Dekripsi Algoritma Kriptografi Asimetris RSA [9][6]

Dekripsi	
Langkah 1	*Ciphertext*: C
Langkah 2	*Plaintext*: $M=C^d \bmod n$

2b. *Elliptic-Curve Cryptography* (ECC)

Elliptic-Curve Cryptography (ECC) adalah pendekatan kriptografi *public key* berdasarkan struktur aljabar kurva eliptik di atas bidang berhingga. Kurva elipsnya digunakan dalam beberapa algoritma faktorisasi bilangan bulat yang memiliki aplikasi dalam kriptografi seperti contohnya faktorisasi kurva elips Lenstra (penggunaan kurva elips ini bukan yang biasanya disebut sebagai "*Elliptic-Curve Cryptography*") [11]. ECC merupakan salah satu alternatif kriptografi asimetris yang dapat dipercaya karena menghadirkan keamanan yang tak kalah dari RSA. ECC memiliki 3 langkah dalam

prosesnya yaitu pembangkitan kunci pada Tabel 4 sampai dengan Tabel 8, enkripsi pada Tabel 9, dan dekripsi pada Tabel 10. Akan tetapi langkah tersebut berbeda dengan yang dimiliki oleh RSA karena dalam ECC pada langkah Pembangkitan Kuncinya dibagi kembali menjadi beberapa bagian. Untuk dapat mengetahui lebih lengkap tiap langkah pada ECC, berikut adalah algoritma dari tiap langkahnya :

Tabel 4
Struktur *Global Public Elements* Algoritma Kriptografi Asimetris ECC [6][9]

Global Public Elements	
Langkah 1	$E_q(a,b)$ kurva eliptik dengan parameter a, b, dan q, di mana q adalah bilangan prima atau bilangan bulat berbentuk 2^m.
Langkah 2	Tentukan Titik G(x,y) pada kurva eliptik yang ordenya bernilai besar n

Tabel 5
Struktur *User* Alice *Key Generation* Algoritma Kriptografi Asimetris ECC [6][9]

User Alice Key Generation	
Langkah 1	Pilih *private key*, n_A dimana $n_A < n$
Langkah 2	Menghitung *public key* $P_A(x,y)$
Langkah 3	$P_A(x,y) = n_A \times G(x,y)$

Tabel 6
Struktur *User* Bob *Key Generation* Algoritma Kriptografi Asimetris ECC [6][9]

User Bob Key Generation	
Langkah 1	Pilih *private key*, n_B dimana $n_B < n$
Langkah 2	Menghitung *public key* $P_B(x,y)$
Langkah 3	$P_B(x,y) = n_B \times G(x,y)$

Tabel 7
Perhitungan *Secret Key* oleh *User* Alice menggunakan Algoritma Kriptografi Asimetris ECC [6][9]

Perhitungan *Secret Key* oleh *User* Alice	
Langkah 1	$K(x,y) = n_A \times P_B(x,y)$

Tabel 8
Perhitungan *Secret Key* oleh *User* Bob menggunakan Algoritma Kriptografi Asimetris ECC [6][9]

Perhitungan *Secret Key* oleh *User* Bob	
Langkah 1	$K(x,y) = n_B \times P_A(x,y)$

Tabel 9
Enkripsi oleh Alice menggunakan *Public key* Bob pada Algoritma Kriptografi Asimetris ECC [6][9]

Enkrpsi oleh Alice menggunakan *Public key* Bob	
Langkah 1	Alice memilih pesan Pm(x,y) dan bilangan bulat positif acak ‘k’ dimana $1<k<q$
Langkah 2	*Ciphertext*: $C_m((x,y),(x,y)) = \{ k \times G(x,y), P_m(x,y) + k \times P_B(x,y) \}$

Tabel 10
Dekripsi oleh Bob menggunakan *Private key* sendiri pada Algoritma Kriptografi Asimetris ECC [6][9]

Dekripsi oleh Bob menggunakan *Private key*nya sendiri		
Langkah 1	*Ciphertext*: $C_m((x,y),(x,y))$	*Keterangan:* Di sini, P_m adalah titik (x,y) yang dikodekan dengan bantuan pesan teks biasa 'm'. P_m adalah titik yang digunakan untuk enkripsi dan dekripsi.
Langkah 2	*Plaintext*: $P_m(x,y)$, $= (P_m(x,y) + k \times P_B(x,y)) - (n_B \times k \times G(x,y))$ $= (P_m(x,y) + kP_B(x,y)) - (n_B \times kG(x,y))$ $= P_m(x,y)$	

2c. Besar Kunci

Besar kunci pada algoritma ECC lebih kecil jika dibandingkan dengan algoritma RSA. Pada tingkat keamanan 80 bit algoritma ECC hanya membutuhkan besar kunci 160 bit sementara algoritma RSA membutuhkan besar kunci 1024 bit. Tabel 11 merupakan data perbandingan besar kunci antara algoritma ECC dan RSA dengan tingkat bit keamanan yang direkomendasikan oleh *National Institute of Standards and Technology* (NIST) [10]:

Tabel 11
Tingkat Keamanan (Bit) yang direkomendasikan NIST [6][9][10]

Tingkat Keamanan (Bit)	Besar Kunci RSA (Bit)	Besar Kunci ECC (Bit)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

3 HASIL ANALISIS DAN PEMBAHASAN

3a. Waktu Enkripsi dan Dekripsi Antara Algoritma ECC dan RSA

Perbandingan waktu enkripsi dan dekripsi antara algoritma ECC dan RSA pada analisis ini menggunakan data dari jurnal *International Journal of Applied Engineering Research* dengan artikel yang berjudul *RSA and ECC: A Comparative Analysis* [6] dan jurnal *International Journal of Network Security* dengan artikel yang berjudul *Performance Analysis of RSA and Elliptic Curve Cryptography* [9]. Tabel 12 merupakan spesifikasi sistem dan cara pengetesan Algoritma yang digunakan untuk percobaan yang telah dilakukan dalam artikel tersebut.

Tabel 13 merupakan data perbandingan waktu enkripsi dan dekripsi antara algoritma ECC dan RSA dari jurnal *International Journal of Applied Engineering Research* dengan artikel yang berjudul *RSA and ECC: A Comparative Analysis* [6].

Tabel 12
Spesifikasi Sistem dan Cara Pengujian Algoritma ECC dan RSA [6][9]

Judul Artikel	Spesifikasi Sistem	Alat Bantu Uji	Cara Uji
RSA and ECC : A Comparative Analysis	Perangkat : Laptop Prosesor : *Intel Pentium dual-core* (1.60 GHz, 533 MHz, 1 MB L2 cache) RAM : DDR2 2GB *Platform:*MsWindows	Menggunakan *software* MATLAB R2008a	Menguji tiga input data sampel dengan ukuran 8 bit, 64 bit, dan 256 bit pada kedua algoritma
Performance Analysis of RSA and Elliptic Curve Cryptography		Menggunakan bahasa pemrograman C dengan GMP *library*	Menguji dua input data sampel dengan ukuran 27 bit dan 270 bit pada kedua algoritma

Tabel 12
Perbandingan Waktu Enkripsi, Dekripsi dan Total Waktu Antara Algoritma ECC dan RSA (dalam detik) [6]

Input	Tingkat Keamanan (Bit)	Enkripsi (detik)		Dekripsi (detik)		Total Waktu (detik)	
		ECC	RSA	ECC	RSA	ECC	RSA
64 Bit	80	2.1685	0.1366	5.9099	5.5372	8.0784	5.6738
	112	9.9855	0.1635	6.9333	20.4108	16.9188	20.5743
	128	15.0882	0.1672	7.3584	46.4782	22.4466	46.6454
	144	20.2308	0.1385	8.4785	77.7642	28.7093	77.9027
256 Bit	80	7.9240	0.5596	22.8851	19.3177	30.8091	19.8772
	112	39.7008	0.5815	26.3331	102.0337	66.0339	102.6153
	128	58.4386	0.5611	27.4060	209.6086	85.8446	210.1697
	144	77.5034	0.5718	32.1522	311.0649	109.6556	311.6368

Tabel 14 merupakan data perbandingan waktu enkripsi dan dekripsi antara algoritma ECC dan RSA dari jurnal *International Journal of Network Security* dengan artikel yang berjudul *Performance Analysis of RSA and Elliptic Curve Cryptography* [9].

Tabel 13
Perbandingan Waktu Enkripsi, Dekripsi dan Total Waktu Antara Algoritma ECC dan RSA (dalam detik) [9]

Ukuran Input Data (Bit)	Tingkat Keamanan	Enkripsi (detik)		Dekripsi (detik)		Total Waktu (detik)	
		ECC	RSA	ECC	RSA	ECC	RSA
27	80	0.1563	0	0.0521	0.4688	0.2084	0.4688
	112	0.2083	0	0.0521	0.2813	0.2604	0.2813
	128	0.2604	0.1563	0.1042	0.8906	0.3646	1.0469
270	80	0.3073	0.1563	0.1510	0.5313	0.4583	0.6876
	112	0.2969	0.4688	0.2031	3.6250	0.5	4.0938
	128	0.3906	0.6250	0.1979	8.9844	0.5885	9.6094

Berdasarkan hasil perbandingan pada Tabel 13 dan 14, dapat diketahui bahwa pada saat proses enkripsi algoritma RSA memiliki waktu enkripsi yang lebih unggul jika dibandingkan

dengan algoritma ECC. Sementara pada saat proses dekripsi, algoritma ECC memiliki waktu proses yang lebih unggul jika dibandingkan dengan algoritma RSA. Pada total waktu enkripsi dan dekripsi, algoritma ECC memiliki waktu proses yang lebih unggul. Disamping perbandingan waktu tersebut, dapat dilihat pada tingkat keamanan yang sama ECC memiliki ukuran besar kunci yang lebih kecil jika dibandingkan dengan RSA berdasarkan data pada Tabel 11. Dengan begitu, dapat diketahui jika waktu enkripsi dan dekripsi pada algoritma ECC dan RSA dipengaruhi oleh besar kunci.

3b. Pengaruh Besar Kunci Pada Algoritma ECC dan RSA

Perbedaan besar kunci antara algoritma ECC dan RSA tentunya akan mengakibatkan waktu proses enkripsi dan dekripsi yang berbeda. Hal tersebut dapat dilihat pada perbandingan waktu enkripsi ECC dan RSA dimana besar kunci pada algoritma ECC yang lebih kecil dibandingkan dengan besar kunci algoritma RSA pada tingkat keamanan yang sama memiliki total waktu yang lebih unggul.

Tabel 145
Perbandingan Besar Kunci dan Total Waktu Antara Algoritma ECC dan RSA dengan Tingkat Keamanan 80 Bit (dalam detik) [6][9]

Hasil Percobaan	Besar Kunci (Bit)		Enkripsi		Dekripsi		Total Waktu	
	ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA
RSA and ECC : A Comparative Analysis (Input 64 Bit)	160	1024	2.1685	0.1366	5.9099	5.5372	8.0784	5.6738
Performance Analysis of RSA and Elliptic Curve Cryptography (Input 270 Bit)	160	1024	0.3073	0.1563	0.1510	0.5313	0.4583	0.6876

Tabel 156
Perbandingan Besar Kunci dan Total Waktu Antara Algoritma ECC dan RSA dengan Tingkat Keamanan 112 Bit (dalam detik) [6][9]

Hasil Percobaan	Besar Kunci (Bit)		Enkripsi		Dekripsi		Total Waktu	
	ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA
RSA and ECC : A Comparative Analysis (Input 64 Bit)	224	2048	9.9855	0.1635	6.9333	20.4108	16.9188	20.5743
Performance Analysis of RSA and Elliptic Curve Cryptography (Input 270 Bit)	224	2048	0.2969	0.4688	0.2031	3.6250	0.5	4.0938

Tabel 167
Perbandingan Besar Kunci dan Total Waktu Antara Algoritma ECC dan RSA dengan Tingkat Keamanan 112 Bit (dalam detik) [6][9]

Hasil Percobaan	Besar Kunci (Bit)		Enkripsi		Dekripsi		Total Waktu	
	ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA
RSA and ECC : A Comparative Analysis (Input 64 Bit)	256	3072	15.0882	0.1672	7.3584	46.4782	22.4466	46.6454
Performance Analysis of RSA and Elliptic Curve Cryptography (Input 270 Bit)	256	3072	0.3906	0.6250	0.1979	8.9844	0.5885	9.6094

Berdasarkan hasil perbandingan pada Tabel 15 sampai Tabel 17, terlihat bahwa semakin tinggi tingkat keamanan dan besar kunci, maka waktu enkripsi dan dekripsi juga akan semakin besar. Pada tingkat keamanan 80-bit, RSA memiliki total waktu yang lebih unggul dan hampir mendekati total waktu algoritma ECC. Tetapi pada tingkat keamanan di atas 80-bit yang menyebabkan semakin besarnya ukuran besar kunci, algoritma ECC memiliki waktu proses yang jauh lebih unggul. Besar kunci pada algoritma ECC lebih kecil jika dibandingkan dengan algoritma RSA dikarenakan perbedaan konsep pada kedua algoritma dimana algoritma RSA menggunakan metode pemfaktoran bilangan bulat sementara itu algoritma ECC menggunakan perhitungan kurva eliptik di atas bidang terbatas [12]. Oleh karena itu, besar kunci mempengaruhi waktu proses enkripsi dan dekripsi pada kedua algoritma.

4. KESIMPULAN

Berdasarkan hasil analisis dan perbandingan pada kedua algoritma di atas, dapat disimpulkan bahwa kedua algoritma bekerja dengan baik dalam hal enkripsi dan dekripsi. Perbedaan algoritma tentu menghasilkan waktu enkripsi dan dekripsi yang berbeda (perhitungan kurva eliptik untuk ECC dan pemfaktoran bilangan bulat untuk RSA). Pada proses enkripsi, waktu yang dibutuhkan algoritma ECC lebih banyak dibandingkan RSA namun pada saat proses dekripsi, waktu yang dibutuhkan algoritma ECC lebih sedikit dibandingkan RSA. Hal tersebut diakibatkan oleh perbedaan ukuran besar kunci antara algoritma ECC dan algoritma RSA. Pada tingkat keamanan yang sama, algoritma ECC memiliki besar kunci yang lebih sedikit jika dibandingkan dengan algoritma RSA. Sebagai contoh, pada tingkat keamanan 128 bit algoritma ECC hanya memerlukan 256 bit besar kunci, sementara itu algoritma RSA memerlukan 3072 bit besar kunci. Hal ini menyebabkan algoritma RSA memiliki waktu enkripsi dan dekripsi yang lebih besar jika dibandingkan dengan algoritma ECC. Oleh karena itu, besar kunci yang lebih kecil pada algoritma ECC mengakibatkan algoritma ECC mempunyai total waktu yang lebih unggul jika dibandingkan dengan algoritma RSA. Hal tersebut dibuktikan dengan perbandingan waktu enkripsi dan dekripsi antara algoritma ECC dengan algoritma RSA.

DAFTAR PUSTAKA

[1] A. Suroso, "STUDI PERBANDINGAN KRIPTOGRAPI MENGGUNAKAN METODE DES, TRIPLE DES DAN RSA Amat Suroso," *SIGMA - J. Teknol. Pelita Bangsa*, vol. 8, no. 2, pp. 17–25, 2018, [Online]. Available: https://jurnal.pelitabangsa.ac.id/index.php/sigma/article/view/150.

[2] R. Sinha, H. K. Srivastava, and S. Gupta, "Performance Based Comparison Study of RSA and Elliptic Curve Cryptography," *Int. J. Sci. Eng. Res.*, vol. 4, no. 5, pp. 720–725, 2013.

[3] A. Tanenbaum, *Modern Operating System*, Fourth. Pearson Education, Inc., 2015.

[4] R. Munir, "Pengantar Kriptografi," *Bahan Kuliah*, 2006.

[5] Z. Vahdati, S. M. D. Yasin, A. Ghasempour, and M. Salehi, "Comparison of ECC and RSA algorithms in IoT devices," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 16, pp. 4293–4308, 2019.

[6] D. Mahto and D. K. Yadav, "RSA and ECC: a comparative analysis," *Int. J. Appl. Eng. Res.*, vol. 12, no. 19, pp. 9053–9061, 2017.

[7] W. M. Ashari, "Perbandingan Performa Kriptografi Asimetris Pada Proses Key Exchange," *Sci. Tech J. Ilmu Pengetah. dan Teknol.*, vol. 6, no. 1, pp. 26–32, 2020, doi: 10.30738/jst.v6i1.6609.

[8] N. Adianson, Y. Yupianti, and A. Kurniawan, "Analisa Perbandingan Performansi Rsa (Rivest Shamir Adleman) Dan Ecc (Elliptic Curve) Pada Protokol Secure Socket Layer (Ssl)," *Media Infotama*, vol. 11, no. 1, pp. 71–80, 2015.

[9] D. Mahto and D. Kumar Yadav, "Performance Analysis of RSA and Elliptic Curve Cryptography," *Int. J. Netw. Secur.*, vol. 20, no. 4, pp. 625–635, 2018, doi: 10.6633/IJNS.201807.

[10] E. Barker, W. Barke, W. Burr, W. W. Polk, and M. Smid, "Recommendation for Key Management – Part 1 : General," *NIST Spec. Publ. 800-57 Part 1, Revis.*, no. May, pp. 1–142, 2007, [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf.

[11] V. B. Kute, P. R. Paradhi, and G. R. Bamnote, "A software comparison of RSA and ECC," *Int. J. Comput. Sci. Appl.*, vol. 2, no. 1, pp. 61–65, 2009, [Online]. Available: http://www.researchpublications.org/IJCSA/issue4/2009-IJCSA-02-01-15.pdf.

[12] D. Rubiagatra, "KRIPTOGRAFI KURVA ELIPTIK ELGAMAL UNTUK PROSES ENKRIPSI- DEKRIPSI CITRA DIGITAL BERWARNA," Institut Teknologi Sepuluh Nopember, 2017.