SURVEI TENTANG ALGORITMA KRIPTOGRAFI ASIMETRIS

Triyas Hevianto Saputro[1], Nur Hidayati[2], E.I.H. Ujianto[3]

[1,2, 3] Program Studi Teknologi Informasi Program Magister, Universitas Teknologi Yogyakarta
[1]triyas.hevianto.saputro@student.uty.ac.id, [2] nur.hidayati@student.uty.ac.id, [3] erik.iman@uty.ac.id

Abstrak

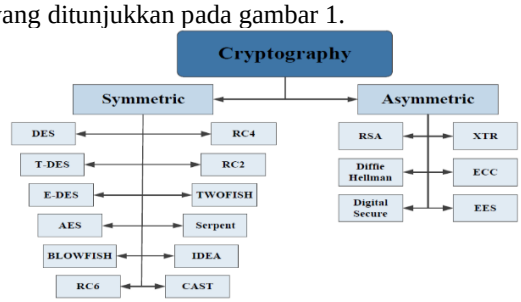
Keamanan merupakan salah satu faktor penting dalam penyimpanan dan pengiriman data atau pesan. Salah satu cara untuk mengamankan dokumen adalah dengan menggunakan algoritma kriptografi. Kriptografi berdasarkan jenis kuncinya dibedakan menjadi dua yaitu kriptografi simetris dan asimetris. Algoritma enkripsi asimetris termasuk *Rivest–Shamir–Adleman (RSA), Diffie-Hellman, Digital Secure Algorithm (DSA), XTR, Elliptic Curve Cryptography (ECC), dan Elgamal Encryption System (ESS).* Dalam makalah ini akan melakukan survei paper terkait algoritma-algoritma enkripsi asimetris. Implementasi kriptografi asimetris dapat dikembangkan menggunakan algoritma-algoritma tersebut.

Kata kunci : ***cryptography, encryption, decryption, public key***

1. Pendahuluan

Keamanan merupakan salah satu faktor penting dalam penyimpanan dan pengiriman data atau pesan. Keamanan dapat diartikan sekelompok langkah, prosedur, dan strategi yang digunakan untuk menghentikan dan mengamati akses ilegal, pemecahan masalah, pengungkapan, gangguan dan penyesuaian sumber jaringan komputer (Abood & Guirguis, 2018). Kriptografi salah satu komponen penting untuk komunikasi dan transmisi informasi melalui layanan keamanan (Mushtaq, Jamel, Disina, Pindar, & Ahmad, 2017). Salah satu cara untuk mengamankan dokumen adalah dengan menggunakan algoritma kriptografi (Sumarno, Gunawan, Tambunan, & Irawan, 2018). Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya (Sumarno et al., 2018). Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan enkripsi dan dekripsi. Pesan yang akan dienkripsi disebut sebagai *plaintext* (teks biasa), karena informasi ini dengan mudah dapat dibaca dan dipahami oleh siapa saja (Toyib & Wijaya, 2019). Enkripsi adalah suatu proses yang dilakukan untuk mengubah pesan asli menjadi *ciphertext*. Sedangkan suatu proses yang dilakukan untuk mengubah pesan tersembunyi menjadi pesan biasa (yang mudah dibaca) disebut dekripsi. Pesan yang telah diubah atau disandikan supaya tidak mudah dibaca disebut dengan *ciphertext* (Sumarno et al., 2018) dan (Aribowo, 2008).

Kriptografi berdasarkan jenis kuncinya dibedakan menjadi dua yaitu kriptografi simetris dan asimetris. Kunci simetris merupakan kunci yang dipakai dalam proses enkripsi dan dekripsi. Sedangkan kunci asimetris memiliki kunci yang berbeda dalam proses enkripsi dan dekripsi, proses enkripsi menggunakan kunci publik dan proses dekripsi menggunkan kunci privat. Algoritma kriptografi asimetris lebih kuat keamanannya dibanding dengan algoritma simetris (Abood & Guirguis, 2018) . Berikut ini kategori kriptografi yang ditunjukkan pada gambar 1.

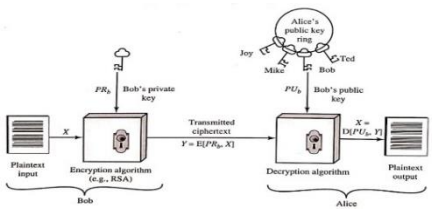


Gambar 1. Klasifikasi Kriptografi (Abood & Guirguis, 2018)

Algoritma kriptografi simetris dibagi menjadi dua kategori yaitu algoritma aliran *(Stream Ciphers)* dan algoritma blok *(Block Ciphers).* Dimana pada algoritma *Stream Ciphers* proses penyandiannya akan beriorientasi pada satu *bit/byte* data. Sedangkan pada algoritma *Block Ciphers*, proses penyandiannya berorientasi pada sekumpulan *bit/byte data* (per blok) (Irawan, 2017). Keamanan dari algoritma ini terletak pada kuncinya, jika kunci diberitahukan atau dibocorkan maka siapa saja dapat mengenkrip dan mendekrip data, jadi kunci harus benar-benar rahasia dan aman (Toyib & Wijaya, 2019).

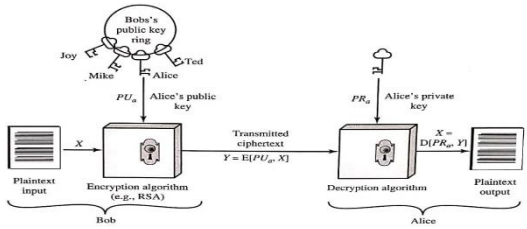
Algoritma Asimetris (*Asymmetric* atau *Public Key*) adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsi. Algoritma enkripsi asimetris termasuk *Rivest–Shamir–Adleman (RSA), Diffie-Hellman, Digital*

Secure Algorithm (DSA), XTR, Elliptic Curve Cryptography (ECC), dan Elgamal Encryption System (ESS) (Abood & Guirguis, 2018).



Gambar 2. Enkripsi dengan Kunci Private (Simetris) (Stallings & Brown, 2018)

RSA, *ECC*, dan El-Gamal merupakan salah satu teknik paling terkenal yang digunakan untuk melakukan enkripsi dan dekripsi (Mallouli, Hellal, Sharief Saeed, & Abdulraheem Alzahrani, 2019). Keamanan dari algoritma ini terletak pada kuncinya, jika kunci diberitahukan atau dibocorkan maka siapa saja dapat mengenkrip dan mendekrip data, jadi kunci harus benar-benar rahasia dan aman (Zuli & Irawan, 2017). Keuntungan utama kunci enkripsi asimetris adalah memiliki enkripsi yang kuat yang akan membuat dekripsi teks asli menjadi sulit dan tidak dapat diprediksi oleh peretas (Mallouli et al., 2019). Secara umum (Mallouli et al., 2019), kriptografi kunci publik membawa masalah kompleks ke dalam kriptografi.



Gambar 3. Enkripsi dengan Kunci Publik (Asimetris) (Stallings & Brown, 2018)

Berdasarkan uraian sebelumnya dijelaskan bahwa algoritma kriptografi berdasarkan Teknik kunci enkripsi diklasifikasikan menjadi kriptografi kunci simetris dan asimetris. Dalam makalah ini akan melakukan survei paper terkait algoritma-algoritma enkripsi asimetris, sehingga survei ini dapat mengetahui perkembangan dalam implementasi algoritma asimetris sampai saat ini. Dengan demikian diharapkan dapat ilmu pengetahuan algoritma ini dapat dikembangkan dan diimplementasikan dengan baik untuk masa depan.

2. Survei Literatur

RSA adalah algoritma enkripsi kunci publik yang dikembangkan oleh Ron Rivest, Adi Shamir dan Len Adleman pada tahun 1977 dan pertama kali dipublikasikan pada tahun 1978 (Stallings & Brown, 2018). *RSA* adalah singkatan dari huruf depan 3 orang yang menemukannya di *MIT (Massachussets Institute of Technology).* Pada tahun 1983, *Massachusets Institute of Technology* menerima hak paten atas sebuah makalah yang berjudul "*Cryptography Communication System and Method*" yang mengaplikasikan pengguna algoritma kriptografi *RSA* (Sumarno et al., 2018). *RSA* menggunakan kunci enkripsi asimetris yang memiliki dua kunci berbeda (Mallouli et al., 2019). Stallings (1995) menjelaskan bahwa sistem ini merupakan satu-satunya sistem yang diterima dan diterapkan secara luas sebagal sistem kriptografi kunci publik (Setiaji, 2015). Kunci publik digunakan untuk proses enkripsi dan diketahui semua orang. Kunci privat digunakan untuk proses dekripsi. Balasubramanian menganalisis keamanan tiga varian *RSA* terhadap serangan *cryptanalytic* (Balasubramanian, 2014). Makalah (Balasubramanian, 2014) menyajikan varian pertama membandingkan antara *Efficient RSA* dan *RSA* asli, varian kedua membahas Dependent *RSA* dan varian ketiga adalah *Carmichael RSA*. Langkah dalam algoritma *RSA* adalah membuat pasangan kunci yaitu Kunci publik dan kunci privat. Keamanan algoritma *RSA* terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat (Sumarno et al., 2018). Algoritma *RSA* memiliki 3 proses penting: *key generation*, *encryption*, dan *decryptio*n (Aufa, Endroyono, & Affandi, 2018).

DIFFIE-HELLMAN key agreement adalah algoritma kunci public yang dipublikasikan oleh Diffie Hellman (Stallings & Brown, 2018). Algoritma ini digunakan (Dey & Hossain, 2019) untuk skema pembentukan kunci sesi yang ringan dan aman untuk jaringan rumah pintar dan menggabungkan Pertukaran kunci Diffie-Hellman sebagai metode alternatif. *Diffie-Hellman Key Agreement (DHKA)* merupakan salah satu protokol key agreement yang didasarkan pada perkalian bilangan bulat. *DHKA* dikenalkan oleh Whitfild Diffie dan Martin Hellman pada tahun 1976 (Permana & St, 2013) (Mehibel & Hamadouche, 2017). Salah satu jenis protokol kriptografi yaitu *key agreement protocol* atau protokol pertukaran kunci. Protokol *key agreement* merupakan salah satu protokol kriptografi yang menyediakan layanan kriptografi untuk menghasilkan kunci (Permana & St, 2013). Pada dasarnya, Algoritma ini adalah solusi yang efisien untuk masalah menciptakan rahasia umum antara dua peserta melalui saluran yang tidak aman (Mehibel & Hamadouche, 2017). Dalam Johnston dan Gemmell (2002) algoritma pertukaran kunci Diffie-Hellman tidak aman terhadap serangan-serangan menengah (Mehibel & Hamadouche, 2017).

Digital Signature Algorithm (DSA) adalah algoritma *digital signature* teknik baru (Stallings & Brown, 2018). Dokumen yang hendak dikirim terlebih dahulu dikenal dengan fungsi *hash* sehingga menjadi bentuk yang ringkas yang disebut dengan *message digest*. Kemudian, *message digest* dienkripsi menggunakan algoritma kriptografi kunci publik,

kunci privat milik penandatangan atau pengirim dokumen akan digunakan untuk melakukan enkripsi *message digest*. Hasil dari enkripsi inilah yang disebut sebagai tanda tangan digital (*digital signature)* (Anshori, et. al. 2019). *DSA* memiliki 3 proses penting: *key generation, signing*, dan *verifying* (Balasubramanian, 2014).

Elliptic Curve Crytography (ECC) adalah produk dan standart yang menggunakan kriptografi kunci publik untuk enkripsi dan *digital signature* menggunakan *RSA* (Stallings & Brown, 2018). Neal Koblitz (Loi & Ko, 2015) dan Victor S. Miller (Loi & Ko, 2015) memperkenalkan secara independen *Elliptic Curve Crytography* pada tahun 1985 dan 1987 (Mallouli et al., 2019). Implementasi *ECC* telah terbukti lebih efisien daripada algoritma kriptografi kunci publik lainnya, seperti Rivest-Shamir-Adleman (Loi & Ko, 2015). *ECC* telah mendapatkan peningkatan jumlah perhatian dalam beberapa tahun terakhir karena algoritma kriptografi kunci publik yang lebih efisien diperlukan untuk semakin banyaknya transaksi aman melalui jaringan (Loi & Ko, 2015). Ansah, et. all. (2018) menjelaskan *ECC* didasarkan pada struktur aljabar *elliptic Curve* di atas bidang yang terbatas (Mallouli et al., 2019). Algoritma ini didasarkan pada aritmatika pada *elliptic Curve* dan keamanan kekerasan dari *Elliptic Curve Discrete Logarithm Problem (ECDLP)* (Mehibel & Hamadouche, 2017). Dalam Menezes dan Vanstone (1993) *Elliptic Curve Cryptography* dapat digunakan untuk enkripsi, tanda tangan digital, dan pertukaran kunci (Mehibel & Hamadouche, 2017).

XTR (Lenstra & Verheul, 2012) adalah singkatan dari *'ECSTR*', yang merupakan singkatan dari *Efficient and Compact Subgroup Trace Representation.* Ini adalah metode baru yang menggunakan jejak untuk mewakili dan menghitung kekuatan elemen subkelompok bidang terbatas. Penerapan *XTR* dalam protokol kriptografi mengarah pada penghematan substansial baik dalam komunikasi dan overhead komputasi tanpa mengorbankan keamanan. *XTR* adalah sistem logaritma diskrit tradisional: untuk keamanannya bergantung pada sulitnya memecahkan masalah logaritma diskrit terkait dalam kelompok multiplikatif bidang terbatas (Lenstra & Verheul, 2012). *XTR* tidak didasarkan pada masalah primitif baru, yang mendasari *cryptosystem* kunci publik pertama ini adalah protokol perjanjian kunci Diffie-Hellman (Lenstra & Verheul, 2012).

The Escrowed Encryption Standard (EES) (Blaze, 1994) mendefinisikan keluarga prosesor kriptografi Pemerintah AS, yang dikenal sebagai *chip "Clipper",* yang dimaksudkan untuk melindungi komunikasi dan data sektor pemerintah dan sektor swasta yang tidak rahasia. Fitur dasar pengaturan kunci antara pasangan prosesor *EES* melibatkan pertukaran *"Law Enforcement Access Field" (LEAF)* yang berisi salinan kunci kunci sesi terenkripsi. *LEAF* dimaksudkan untuk memfasilitasi akses pemerintah ke teks bagian data yang dienkripsi di bawah sistem. Beberapa aspek desain *EES*, yang menggunakan algoritma sandi rahasia dan perangkat keras yang tahan terhadap kerusakan, berupaya membuatnya tidak layak untuk menggunakan sistem tanpa mentransmisikan *LEAF*.

3. Perkembangan Kriptografi Asimetris

Algoritma RSA diimplementasikan untuk penyandian data (Setiaji, 2015) untuk meningkatkan kehandalan keamanan data dalam jaringan komputer. Dalam penelitian ini didapatkan algoritma *RSA* dari segi pembangkitan lebih unggul untuk penyandian data. Implementasi algoritma ini untuk keamanan data pada jaringan komputer juga lebih cepat dan *valid*. Keuntungan utama (Mallouli et al., 2019) dari algoritma *RSA* adalah penggunaan kunci pribadi, yang tidak ditransmisikan dengan teks yang dienkripsi, yang membuatnya tidak mungkin diketahui oleh peretas. Selain itu menurut Mitali dan Sharma bahwa *RSA* memberikan tanda tangan digital dengan kunci publik di *RSA* memberikan fitur yang kuat. Hardik Gohel (2015) menjelaskan tanda tangan digital dapat ditentukan sebagai tanda tangan *online* yang akan memberikan dua poin utama yaitu pesan itu dikirim ke orang yang diperlukan tanpa perubahan, dan identitas pengirim dijamin (Mallouli et al., 2019). Menurut Mitali and Sharma bahwa kelemahan utama dari algoritma *RSA* adalah pemrosesan yang lambat. Metode kombinasi *RSA* 1024 dan *DSA* 512 untuk mendapatkan waktu komputasi yang relatif cepat (Aufa et al., 2018). Metode kombinasi ini tidak hanya dapat mengenkripsi pesan, tetapi juga menyediakan tanda tangan digital untuk proses otentikasi dengan aman dan cepat (Aufa et al., 2018).

Digital signature telah menyediakan layanan keamanan untuk mengamankan transaksi elektronik melalui internet. Algoritma *RSA* paling banyak digunakan untuk menyediakan teknik keamanan. Dalam penelitian ini (Jaju & Chowhan, 2015) telah memodifikasi algoritma *RSA* untuk meningkatkan tingkat keamanannya. Perbandingan antara algoritma *RSA* dan algoritma *RSA* yang dimodifikasi bersama dengan waktu dan keamanan dengan menjalankan beberapa pengaturan enkripsi dan dekripsi untuk memproses data dari ukuran yang berbeda (Jaju & Chowhan, 2015). Efisiensi dari algoritma ini dianggap memiliki kecepatan generasi kunci dan tingkat keamanan. Simulasi ini membuktikan bahwa dalam pembangkitan kunci algoritma *RSA* yang dimodifikasi lebih cepat dan meningkatkan keamanan oleh dua tingkat (Jaju & Chowhan, 2015). Tetapi algoritma *RSA* lebih cepat daripada *RSA* yang dimodifikasi dalam hal kecepatan enkripsi dan dekripsi (Jaju & Chowhan, 2015).

Algoritma *RSA* (Karakra & Alsadeh, 2016) adalah cryptosystem kunci publik yang paling banyak digunakan di seluruh dunia. Algoritma ini digunakan untuk keamanan dalam segala hal, mulai dari belanja online hingga ponsel. Namun, *RSA* dasar tidak semantik aman, misalnya untuk mengenkripsi pesan yang sama lebih dari sekali selalu memberikan ciphertext yang sama. Dengan ini *RSA* dasar rentan terhadap serangkaian serangan tidak langsung, seperti plaintext yang diketahui, *plaintext* yang dipilih, waktu, modulus umum, dan serangan *frequency of blocks (FOB)*. Selain itu, *RSA* dikenal jauh lebih lambat daripada enkripsi kunci simetris standar dan tidak digunakan untuk mengenkripsi data besar. Dalam makalah ini (Karakra & Alsadeh, 2016) dirancang dan diimplementasikan varian *RSA* yang cepat dan aman berdasarkan koding Rabin dan Huffman yang disebut *Augmented RSA (A-RSA)* untuk mengatasi keterbatasan *RSA* dasar yang disebutkan sebelumnya. Peneliti menemukan bahwa sistem *RSA* meningkatkan ukuran *ciphertext* sebesar 1% dibandingkan dengan ukuran file asli, sedangkan ukuran rata-rata file *A-RSA* sama dengan 0,46 dari ukuran aslinya

Ansah et. all. (2016) mengatakan *ECC* didasarkan pada struktur aljabar dari kurva eliptik di atas bidang yang terbatas (Mallouli et al., 2019). Keuntungan utama *ECC* dibandingkan algoritma kunci publik lainnya terletak pada pertukaran kunci yang membutuhkan panjang kunci yang lebih pendek (Mallouli et al., 2019). *Elliptic Curve Cryptography* telah menjadi pilihan kriptografi untuk jaringan dan perangkat komunikasi karena ukuran dan manfaat efisiensinya (Mallouli et al., 2019).

Loi dan Ko menyajikan arsitektur dan implementasi *elliptic curve cryptography processor (ECP)* kinerja tinggi yang dapat diskalakan (Loi & Ko, 2015). Dalam makalah ini memberikan arsitektur *ECP* yang sangat efisien dan *scalable.* Desain yang diusulkan mengambil keuntungan dari *DSP48E slices* performa tinggi yang tersedia pada *Xilinx FPGAs* untuk meningkatkan kinerjanya. Dengan memparalelkan operasi *point additions (PADD*) dan *point doublings (PDBL)* operasi *addition*, *subtraction*, dan *reduction* berjalan paralel dengan *integer multiplication*. Operasi inversi bilangan prima menggunakan algoritma inversi biner, yang diimplementasikan dengan modifikasi minimal pada blok aritmatika yang digunakan untuk operasi lain. Menurut penulis *ECP scalable* yang diusulkan adalah *ECP* tercepat dan terkecil yang mendukung kelima bidang utama yang direkomendasikan *NIST* tanpa perlu mengkonfigurasi ulang perangkat keras.

Kekuatan kriptografi kunci publik menggunakan *Elliptic Curves cryptography* bergantung pada kesulitan menghitung logaritma diskrit di bidang yang terbatas (Deligiannidis, 2015). Algoritma kriptografi kunci publik lainnya, mengandalkan kesulitan faktorisasi bilangan bulat. Penulis menjelaskan bagaimana dapat mengimplementasikan beberapa algoritma *ECC* kriptografi di *java* (Deligiannidis, 2015). Dalam makalah ini, penulis mengimplementasikan beberapa algoritma *ECC* seperti pertukaran kunci *Diffie-Hellman (ECDH)*, dua algoritma tanda tangan digital seperti: ElGamal dan *ECDSA*, dan tiga algoritma enkripsi seperti ElGamal, Massey-Omura, dan ECIES (Deligiannidis, 2015).

Pendekatan baru untuk *Elliptic Curve Method (ECM)* (Kobrle & Lorencz, 2015) yang mempercepat waktu faktorisasi dalam koordinat *affine*, dengan mengoptimalkan langkah-langkah perhitungan untuk kebutuhan *Double and Add algorithm*. Persamaan yang digunakan dalam penelitian ini adalah *Elliptic Curve Cryptography (ECC)* atau *Elliptic Curve Digital Signature Algorithm (ECDSA)*, di mana prinsip yang sama digunakan dan dengan demikian dapat membuat operasi lebih cepat. Metode yang diusulkan membuktikan di mana penggandaan dan penambahan poin dieksekusi lebih cepat. Pendekatan penggabungan operasi ini dapat juga digunakan dalam sistem koordinat lainnya sebagai proyektif atau Jacobi, untuk mencapai percepatan. Implementasi pendekatan ini dapat bermanfaat pada perangkat lunak komputer klasik yang ditulis dalam bahasa seperti C atau *Java*. Karena *ECDSA* atau *ECC* umum tidak selalu diprogram dengan cara *FPGA* atau *VGA*.

Perkalian skalar *elliptic curve* (Thomas & Sheela, 2015) adalah operasi yang paling penting dalam *elliptic curve cryptosystems*. Dalam pengembangan multiplikasi *elliptic curve* menggunakan Karatsuba multiplier dan juga menghasilkan pasangan kunci rahasia untuk enkripsi dan dekripsi dalam *elliptic curve cryptography*. Implementasi *elliptic curve point multiplication* dicapai dengan menggunakan *Galois Field arithmetic* yang disimulasikan pada *ModelSim.*

Makalah ini (Singh, Khan, & Singh, 2016) mengevaluasi kinerja *cryptosystem elliptic curve* secara kritis. Bentuk cryptosystem didasarkan pada kurva eliptik atas beberapa *Galois fields*. *Elliptic Curve Cryptography* atau *Cryptosystem (ECC)* adalah *cryptosystem* kunci publik yang menggunakan pasangan kunci publik dan pribadi sebagai sarana mengenkripsi dan mendekripsi informasi di internet. Keamanan kritis *ECC* bergantung pada kemampuan untuk menghitung perkalian titik dengan bilangan bulat dengan mudah dan ketidaklayakan atau kegagalan untuk menemukan pengali dan multiplikasi dan titik produk yang diberikan dalam jendela waktu. Keuntungan dan manfaat utama menggunakan *ECC* dibandingkan dengan *cryptosystems* lainnya adalah bahwa kita dapat menggunakan kunci yang lebih kecil tanpa mengurangi keamanan sistem.

Infrastruktur kriptografi kunci publik (Shaikh, Nenova, Iliev, & Valkova-Jarvis, 2017) dari perdagangan elektronik (*E-commerce*), penggunaan algoritma *RSA* sangat umum. Dengan persyaratan keamanan, ukuran kunci *RSA* yang diperlukan

meningkat secara eksponensial sehingga ukuran kunci yang besar ini, tidak nyaman untuk menggunakan *RSA*. *Elliptic Curve Cryptography* (*ECC*) adalah alternatif terbaik untuk RSA tradisional. *ECC* menyediakan tingkat keamanan yang sama dengan *RSA* tetapi dengan ukuran kunci yang dikurangi. Faktor paling penting dalam penggunaan *ECC* adalah pemilihan kurva eliptik yang benar. Penulis menguraikan (Shaikh et al., 2017) set kurva eliptik yang direkomendasikan oleh berbagai standar untuk kriptografi dipilih, dan kurva yang dipilih dianalisis, fokus pada fitur kinerja dan keamanan. Penulis menganalisis dengan mempertimbangkan setiap kurva untuk implementasi *Elliptic Curve Diffie-Hellman (ECDH)* dan *Elliptic Curve Digital Signature Algorithm (ECDSA)*. Dari hasil analisis, jelas bahwa ketika ukuran bidang kurva meningkat dan waktu komputasi yang diperlukan meningkat. Bergantung pada persyaratan keamanan dan komputasi aplikasi tertentu, kurva eliptik yang sesuai dapat dipilih (Shaikh et al., 2017).

Implementasi DIFFIE-HELLMAN algorithm

Skema Elliptic Curve Cryptosystem (ECC) adalah mekanisme kunci publik yang menyediakan enkripsi, tanda tangan digital dan kemampuan pertukaran kunci. Keuntungan dari kurva eliptik adalah bahwa mereka memastikan tingkat keamanan yang setara dengan sistem kunci publik yang ada tetapi dengan panjang kunci yang lebih pendek. Dalam tulisan ini (Mehibel & Hamadouche, 2017), penulis tertarik pada pertukaran kunci publik dari Diffie-Hellman. Penulis mengusulkan pendekatan baru pertukaran kunci kurva elips Diffie-Hellman. pendekatan baru pertukaran kunci yang aman dan otentik di saluran publik menggunakan Elliptic Curve. Kriptografi kunci publik menyediakan solusi untuk distribusi kunci dan pertukaran informasi yang aman. Keamanan *ECC* didasarkan pada kesulitan *Elliptic Curve Discrete Logarithm Problem (ECDLP)* menggunakan pertukaran kunci ECC, yaitu, penulis dapat memastikan kerahasiaan, otentikasi, dan non-penolakan. Dalam karya masa depan, kita dapat menerapkan pertukaran kunci publik dan kunci publik tertentu yang diusulkan dalam makalah ini untuk enkripsi dan dekripsi pesan dengan memastikan keaslian pesan, untuk memiliki kriptosistem asimetris yang diamankan oleh kekerasan *Elliptic Curve Discrete Logarithm Problem (ECDLP)*.

XTR algorithm memiliki keuntungan bahwa parameternya yang sangat cepat dan pemilihan tombol (jauh lebih cepat dari *RSA*, urutan besarnya lebih cepat dari *ECC*), ukuran kunci kecil (jauh lebih kecil dari *RSA*, sebanding dengan *ECC* untuk pengaturan keamanan saat ini), dan kecepatan (secara keseluruhan sebanding dengan *ECC* untuk pengaturan keamanan saat ini) (Lenstra & Verheul, 2012). Penelitian ini memberikan bukti bahwa keamanan yang disediakan oleh subkelompok *XTR* lebih baik daripada yang disediakan oleh kelompok *isomorfik* pada *supersingular elliptic curves*. Bukti tambahan diberikan oleh fakta bahwa keputusan masalah Diffie-Hellman secara efisien dapat dihitung pada kelompok yang terakhir, sementara masalah ini diyakini sulit di subkelompok *XTR* (Lenstra & Verheul, 2012).

4. Kesimpulan dan Saran

Kriptografi asimetris atau enkripsi kunci publik merupakan topik yang menarik dan masih luas untuk diteliti. Paper ini melakukan survei terkait dengan kriptografi enkripsi kunci publik, untuk penelitian selanjutnya dapat mengkaji lebih dalam salah satu algoritma-algoritma enkripsi kunci publik dengan mengimplementasikannya dalam bentuk perangkat lunak.

Daftar Pustaka:

Abood, O. G., & Guirguis, S. K. (2018). A Survey on Cryptography Algorithms. *International Journal of Scientific and Research Publications (IJSRP)*, *8*(7). https://doi.org/10.29322/ijsrp.8.7.2018.p7978

Anshori, Y., Erwin Dodu, A. Y., & Wedananta, D. M. P. (2019). Implementasi Algoritma Kriptografi Rivest Shamir Adleman (RSA) pada Tanda Tangan Digital. *Techno.Com*, *18*(2), 110–121. https://doi.org/10.33633/tc.v18i2.2166

Aribowo, E. (2008). Aplikasi Pengamanan Dokumen Office Dengan Algoritma Kriptografi Kunci Asimetris Elgamal. *Jurnal Informatika*, *2*(2), 209–219. https://doi.org/10.26555/jifo.v2i2.a5252

Aufa, F. J., Endroyono, & Affandi, A. (2018). Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm. *Proceedings - 2018 4th International Conference on Science and Technology, ICST 2018*, *1*, 1–5. https://doi.org/10.1109/ICSTC.2018.8528584

Balasubramanian, K. (2014). Variants of RSA and their Cryptanalysis. *2014 International Conference on Communication and Network Technologies*, *2*, 145–149. https://doi.org/10.1109/CNT.2014.7062742

Blaze, M. (1994). Protocol failure in the Escrowed Encryption Standard. *Proceedings of the ACM Conference on Computer and Communications Security*, (April 1993), 59–67. https://doi.org/10.1145/191177.191193

Deligiannidis, L. (2015). Elliptic curve cryptography in Java. *2015 IEEE International Conference on Intelligence and Security Informatics: Securing the World through an Alignment of Technology, Intelligence, Humans and Organizations, ISI 2015*, 193.

https://doi.org/10.1109/ISI.2015.7165975

Dey, S., & Hossain, A. (2019). Session-Key Establishment and Authentication in a Smart Home Network Using Public Key Cryptography. *IEEE Sensors Letters*, *3*(4), 1–4. https://doi.org/10.1109/lsens.2019.2905020

Irawan, M. D. (2017). Implementasi Kriptografi Vigenere Cipher Dengan Php. *Jurnal Teknologi Informasi*, *1*(1), 11. https://doi.org/10.36294/jurti.v1i1.21

Jaju, S. A., & Chowhan, S. S. (2015). A Modified RSA algorithm to enhance security for digital signature. *2015 International Conference and Workshop on Computing and Communication, IEMCON 2015*, 1–5. https://doi.org/10.1109/IEMCON.2015.7344493

Karakra, A., & Alsadeh, A. (2016). A-RSA : Augmented RSA. *SAI Computing Conference*, 1016–1023. https://doi.org/10.1109/SAI.2016.7556103

Kobrle, D., & Lorencz, R. (2015). Optimization of elliptic curve operations for ECM using double and add algorithm. *2015 4th International Conference on E-Technologies and Networks for Development, ICeND 2015*, 34–37. https://doi.org/10.1109/ICeND.2015.7328534

Lenstra, A. K., & Verheul, E. R. (2012). An overview of the XTR public key system. *Public-Key Cryptography and Computational Number Theory*. https://doi.org/10.1515/9783110881035.151

Loi, K. C. C., & Ko, S. B. (2015). Scalable Elliptic Curve Cryptosystem FPGA Processor for NIST Prime Curves. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *23*(11), 2753–2756. https://doi.org/10.1109/TVLSI.2014.2375640

Mallouli, F., Hellal, A., Sharief Saeed, N., & Abdulraheem Alzahrani, F. (2019). A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms. *Proceedings - 6th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2019 and 5th IEEE International Conference on Edge Computing and Scalable Cloud, EdgeCom 2019*, *6*, 173–176. https://doi.org/10.1109/CSCloud/EdgeCom.2019.00022

Mehibel, N., & Hamadouche, M. (2017). A New Approach of Elliptic Curve Diffie-Hellman Key Exchange. *International Conference on Electrical Engineering – Boumerdes (ICEE-B)*, *5*(2), 3–8.

Mushtaq, M. F., Jamel, S., Disina, A. H., Pindar, Z. A., & Ahmad, N. S. (2017). *A Survey on the Cryptographic Encryption Algorithms*. *8*(11), 333–344.

Permana, A. D., & St, S. (2013). *Pengamanan Sistem Login Aplikasi Menggunakan Protokol ID Based Diffie-Hellman Key Agreement*. (70), 9–13.

Setiaji, B. (2015). Analisis Dan Implementasi Algoritma Kriptografi Kunci Publik Rsa Dan Luc Untuk Penyandian Data. *Data Manajemen Dan Teknologi Informasi (DASI)*, *16*(3), 27.

Shaikh, J. R., Nenova, M., Iliev, G., & Valkova-Jarvis, Z. (2017). Analysis of standard elliptic curves for the implementation of elliptic curve cryptography in resource-constrained E-commerce applications. *IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems, COMCAS 2017*, 1–4. https://doi.org/10.1109/COMCAS.2017.8244805

Singh, S. R., Khan, A. K., & Singh, T. S. (2016). On the Performance of Elliptic Curve Public Cryptosystem. *International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, *3*(2), 24–29. https://doi.org/10.1109/ICACDOT.2016.7877545

Stallings, W., & Brown, L. (2018). *Computer Security: Principles and Practice, Global Edition*, *4/E* (4th ed.). Retrieved from www.pearsonglobaleditions.com

Sumarno, Gunawan, I., Tambunan, H. S., & Irawan, E. (2018). Analisis Kinerja Kombinasi Algoritma Message-Digest Algortihm 5 (Md5), Rivest Shamir Adleman (Rsa) Dan Rivest Cipher 4 (Rc4) Pada Keamanan E-Dokumen. *JUSIKOM PRIMA (Junal Sistem Informasi Ilmu Komputer Prima)*, *2*(1), 41–48.

Thomas, C., & Sheela, K. G. (2015). Analysis of Elliptic Curve Scalar Multiplication in Secure Communications. *2015 Global Conference on Communication Technologies (GCCT)*, (Gcct), 623–627. https://doi.org/10.1109/GCCT.2015.7342737

Toyib, R., & Wijaya, A. (2019). ANALISIS PERBANDINGAN ALGORITMA SIMETRIS RIVEST CODE 5 DENGAN ALGORITMA SIMETRIS RIVEST CODE 6) (Studi Kasus : SMK Negeri Seluma). *Jurnal Informatika Upgris*, *4*(2), 203–209. https://doi.org/10.26877/jiu.v4i2.2840

Triase. (2015). Kriptografi elgamal menggunakan metode mersenne. *Integritas*, *1*(4), 1–2.

Zuli, F., & Irawan, A. (2017). Implementasi Kriptografi Dengan Algoritma Blowfish Dan Riverst Shamir Adleman (Rsa) Untuk Proteksi File. *Jurnal Ilmiah FIFO*, *9*(1), 5. https://doi.org/10.22441/fifo.v9i1.1437