JNATIA Volume 2, Nomor 3, Mei 2024 p-ISSN: 2986-3929
Jurnal Nasional Teknologi Informasi dan Aplikasinya e-ISSN: 3032-1948

Perlindungan *Seed Phrase* dengan Enkripsi *Dual-Layer* Menggunakan Algoritma AES dan *Caesar Cipher*

Raihan Akbar Maulana[a1], I Wayan Santiyasa[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]raihanakbarmaulana29@gmail.com
[2]santiyasa@unud.ac.id

Abstract

In the digital era, securing sensitive information such as seed phrases are crucial to prevent unauthorized access and potential loss of digital assets. This paper proposes a novel approach for protecting seed phrases using dual-layer encryption with AES algorithm and Caesar cipher. The AES algorithm is utilized to encrypt the seed phrase itself, providing a strong first layer of defense. Additionally, Caesar cipher is employed to encrypt the AES secret key, adding an extra layer of security to the encryption process. By combining these two encryption techniques, the security of the seed phrase is significantly enhanced, as both the phrase and its encryption key are protected. Furthermore, the encrypted seed phrase and key can be securely stored using email as a digital storage medium, enhancing accessibility while maintaining robust security measures.

Keywords: *Cryptocurrency, Seed Phrase, Encryption, Advanced Encryption Standard, Caesar Cipher*

1. Pendahuluan

Dalam ekosistem *cryptocurrency* yang terus berkembang, keamanan informasi memegang peran yang semakin krusial. Di antara berbagai jenis informasi sensitif, *seed phrase* atau frasa benih adalah elemen yang sangat penting. *Seed phrase* digunakan sebagai kunci utama untuk mengakses dan mengontrol aset digital pengguna dalam dompet kripto mereka. Saat seorang pengguna mulai menggunakan dompet kripto untuk pertama kalinya, dompet tersebut akan menghasilkan sebuah *seed phrase* panjang yang terdiri dari 12,15, 18, 21, atau 24 kata dari daftar 2048 kata yang ditentukan oleh standar BIP39 [1]. Biasanya pengguna diminta untuk menulisnya di sebuah kertas dan menyimpannya di tempat yang aman, seed phrase harus dijaga dengan sangat rahasia karena siapapun yang mengetahuinya dapat mencuri semua aset pengguna [2]. Jika komputer, laptop, atau perangkat seluler pengguna rusak, korup, dicuri, atau hancur, pengguna dapat memulihkan semua koin kripto mereka dengan menginstal ulang dompet kripto yang sama pada sistem baru dan menggunakan *seed phrase* yang sama [2]. Serta, untuk menghindari risiko seperti kehilangan atau dicurinya *seed phrase*, karena hal ini dapat berakibat pada kehilangan akses ke dana kripto yang dimiliki pengguna, penting untuk menerapkan lapisan keamanan yang kuat seperti perlindungan *seed phrase* dan menyimpannya secara digital. Salah satu pendekatan yang umum digunakan adalah pengenkripsian. Secara sederhana, enkripsi mengubah teks biasa menjadi *ciphertext* dan mendekripsi *ciphertext* menjadi teks biasa [3]. Pada penelitian ini, kami menggunakan *Advanced Encryption Standard* (AES), sebuah algoritma kriptografi simetris yang telah menjadi standar industri untuk enkripsi data. Meskipun AES memberikan tingkat keamanan yang tinggi, penggunaannya sendiri tidak selalu cukup untuk melindungi *seed phrase* secara menyeluruh. Oleh karena itu, dalam penelitian ini, kami mengusulkan sebuah pendekatan baru yang menggabungkan lebih dari satu algoritma enkripsi untuk meningkatkan keamanan *seed phrase*. Dalam pendekatan ini, kami memanfaatkan kombinasi dari AES dan *Caesar Cipher*. AES digunakan untuk mengenkripsi *seed phrase* itu sendiri, sementara *caesar cipher* digunakan untuk mengamankan kunci rahasia AES, dan penggunaan email sebagai media penyimpanan digital. Penggunaan email sebagai media

penyimpanan digital dipilih karena keunggulannya dalam keamanan dan kemudahan aksesibilitas. Dengan memanfaatkan protokol SMTP, data yang dienkripsi dapat dikirim dengan aman dan andal ke alamat email pengguna. Outlook dan Gmail dipilih karena fasilitas pemulihan bencana yang kuat dan infrastruktur yang andal, memastikan bahwa seed phrase pengguna tetap tersimpan dengan aman dan dapat diakses kembali saat diperlukan. Dengan menggabungkan kedua teknik enkripsi ini, kami bertujuan untuk menciptakan perlindungan yang lebih kuat terhadap *seed phrase*, mengurangi risiko kehilangan atau pencurian aset kripto pengguna. Kami percaya bahwa pendekatan ini akan menjadi langkah penting dalam memperkuat keamanan informasi di ranah *cryptocurrency*, membantu pengguna untuk merasa lebih aman dalam menyimpan dan mengelola aset digital mereka. Dalam tulisan ini, kami akan merinci metode yang diusulkan, menguji keberhasilan proses enkripsi dan dekripsi, serta menguji penyimpanan menggunakan *email*.

2. Metode Penelitian

2.1. Gambaran Umum Sistem

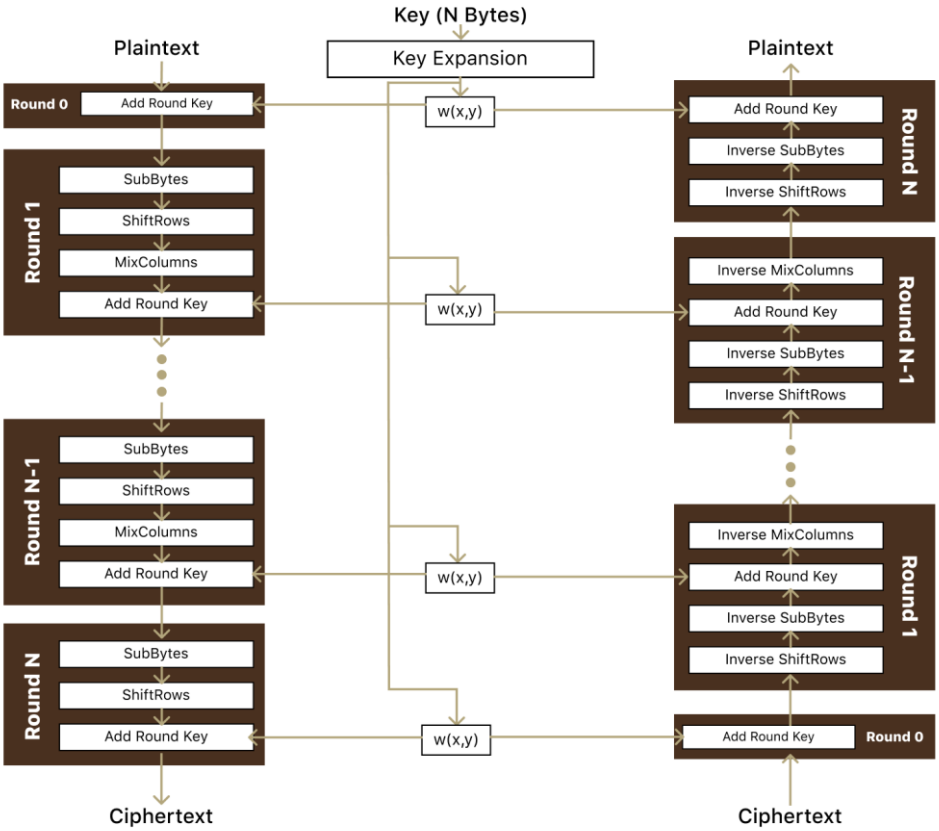
Sistem ini mengimplementasikan dua tahapan enkripsi untuk meningkatkan keamanan *seed phrase*, yaitu dengan menggunakan *Advanced Encryption Standard* (AES) untuk mengenkripsi *seed phrase* dan *caesar cipher* untuk mengamankan *secret key* yang dihasilkan dari proses AES.

a. Pertama, *plaintext* dari *seed phrase* akan dienkripsi menggunakan AES. Pada tahap ini, *plaintext* akan diproses menjadi *ciphertext* menggunakan algoritma AES dengan kunci yang disesuaikan pengguna. Proses enkripsi ini menghasilkan *ciphertext* yang tidak dapat dibaca secara langsung oleh pihak yang tidak berwenang.
b. Selanjutnya, kunci rahasia yang dihasilkan dari proses enkripsi AES akan dienkripsi menggunakan *Caesar Cipher.* Proses ini meningkatkan keamanan *secret key* dengan melakukan enkripsi menggunakan *caesar cipher*, sehingga menghasilkan nilai yang sulit untuk diretas.
c. Pada tahapan dekripsi, dilakukan dengan mengembalikan langkah-langkah enkripsi yang dilakukan secara terbalik: Pertama, *secret key* yang telah dienkripsi dengan *caesar cipher* akan didekripsi untuk mendapatkan *secret key* asli. Selanjutnya, *secret key* tersebut akan digunakan untuk mendekripsi *ciphertext* AES menjadi *plaintext seed phrase* yang asli.
d. Dengan menggunakan kedua tahapan enkripsi ini, sistem bertujuan untuk memberikan perlindungan yang lebih kuat terhadap *seed phrase*, mengurangi risiko kehilangan, dicurinya aset kripto pengguna, dan memberikan keamanan yang lebih tinggi terhadap informasi sensitif dalam *cryptocurrency*.

2.2. Algoritma *Advanced Encryption Standard* (AES)

Advanced Encryption Standard (AES) adalah salah satu algoritma enkripsi yang paling umum digunakan secara global. AES menggunakan blok *cipher* yang beroperasi pada blok-blok data tetap berukuran 128 bit, dengan kunci yang dapat memiliki panjang 128, 192, atau 256 bit [4]. Pada bulan September 1997, AES (*Advanced Encryption Standard*) dipublikasikan[4]. AES telah diadopsi oleh banyak organisasi pemerintah dan industri sebagai standar enkripsi, digunakan untuk melindungi data sensitif, seperti data pribadi, transaksi keuangan, dan pesan rahasia. AES juga berjalan cukup cepat dibandingkan dengan ECC dan RSA [5]. AES menggunakan beberapa putaran transformasi linier, penggantian *byte*, pergeseran baris, dan campuran kolom untuk mengubah teks biasa menjadi teks teracak (*ciphertext*) dan sebaliknya. Hal ini membuatnya sangat kuat terhadap serangan kriptanalisis yang canggih dan cocok untuk digunakan dalam berbagai aplikasi keamanan, mulai dari komunikasi *online* hingga penyimpanan data.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 1. AES *Transformation*

a. Cara Kerja AES

- Inisialisasi Kunci: Tahap awal melibatkan inisialisasi kunci enkripsi, yang terdiri dari pengaturan kunci yang digunakan untuk mengacak data pada *Round* 0.
- Tahap *SubBytes*: Pada tahap ini, setiap byte dalam blok data diubah menjadi nilai baru menggunakan sebuah tabel substitusi yang disebut S-Box. Ini adalah langkah non-linear yang memperkenalkan kebingungan ke dalam proses enkripsi.
- Tahap *ShiftRows*: Di sini, setiap baris dalam blok data bergeser ke kiri sejumlah langkah tertentu. Ini menciptakan efek dispersi horizontal dalam blok data.
- Tahap *MixColumns*: Pada tahap ini, setiap kolom dalam blok data diubah menggunakan operasi matriks yang kompleks. Ini mencampur nilai-nilai dalam setiap kolom untuk memberikan dispersi vertikal dalam blok data.
- Tahap *AddRoundKey*: Ini adalah tahap kunci *round*, di mana setiap *byte* dalam blok data di-XOR-kan dengan bagian kunci rahasia yang disesuaikan oleh *Key Expansion*.

Tahapan-tahapan ini diulang secara berurutan sejumlah *round* tertentu yang tergantung pada panjang kunci (kunci 128 bits akan menghasilkan 10 *round*, kunci 192 bits akan menghasilkan 12 *rounds*, dan kunci 256 bits akan menghasilkan 14 *rounds*), kecuali pada round terakhir di mana tahap *MixColumns* tidak dijalankan dalam enkripsi dan sebaliknya pada dekripsi, serta di mana kunci rahasia diubah untuk setiap *Round*. Pada tahap terakhir, blok data yang dienkripsi atau didekripsi akan menjadi hasil akhir dari proses tersebut.

2.3. Algoritma *Caesar Cipher*

Dalam kriptografi, *Caesar Cipher* adalah salah satu teknik enkripsi yang paling dikenal dan paling sederhana. Sandi ini termasuk dalam kategori sandi substitusi, di mana setiap huruf dalam teks terang digantikan oleh huruf lain dalam alfabet dengan pergeseran posisi tertentu. Sebagai contoh, jika kita menggunakan pergeseran tiga, maka huruf A akan digantikan oleh huruf D, huruf

B akan digantikan oleh huruf E, dan seterusnya. Dalam contoh enkripsi *Caesar Cipher* dengan menggunakan pergeseran dua, setiap huruf dalam kata "*snatia*" akan digantikan oleh huruf yang berada dua posisi setelahnya dalam alfabet.

a. S digantikan oleh U.
b. N digantikan oleh P.
c. A digantikan oleh C.
d. T digantikan oleh V.
e. I digantikan oleh K.
f. A digantikan oleh C.

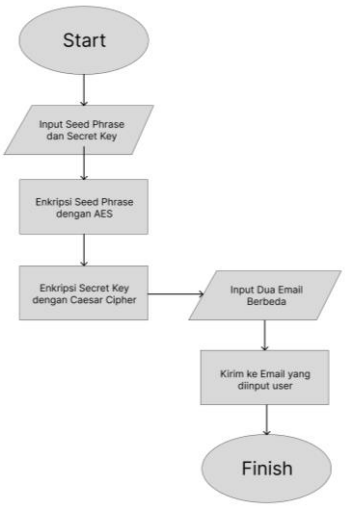
Sehingga, hasil enkripsi dari kata "*snatia*" dengan menggunakan *Caesar Cipher* dan pergeseran dua adalah "upcvkc". Dalam proses dekripsi, kita akan menggunakan pergeseran yang sama namun ke arah yang berlawanan untuk mengembalikan teks ke bentuk aslinya.

2.4. Penyimpanan Menggunakan *Email*

Dalam pengembangan sistem kami untuk penyimpanan *seed phrase*, kami memanfaatkan protokol SMTP (*Simple Mail Transfer Protocol*) untuk mengirimkan data yang dienkripsi ke alamat *email* pengguna. Penggunaan SMTP memastikan pengiriman yang aman dan andal, serta memfasilitasi integrasi yang mudah dengan layanan *email* yang umum digunakan seperti Outlook dan Gmail. Kedua layanan ini dipilih karena keunggulan mereka dalam keamanan dan pemulihan data. Outlook dan Gmail menawarkan fasilitas pemulihan bencana yang kuat, yang memungkinkan pemulihan data yang cepat dan efisien dalam situasi darurat seperti kegagalan *server* atau kerusakan perangkat keras. Selain itu, keduanya memiliki infrastruktur yang sangat andal dan dapat dipercaya, dengan fasilitas replikasi dan keberlanjutan yang memastikan ketersediaan layanan yang optimal bahkan dalam kondisi yang tidak terduga. Dengan demikian, Outlook dan Gmail tidak hanya menyediakan pengiriman *email* yang aman dan handal, tetapi juga memberikan jaminan bahwa *seed phrase* pengguna akan tetap tersimpan dengan aman dan dapat diakses kembali saat diperlukan.

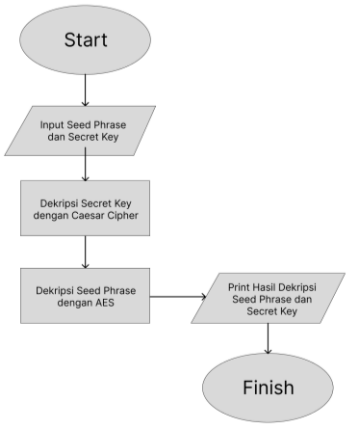
2.5. Perancangan Sistem

Perancangan sistem dijelaskan dengan menggunakan diagram alur. Secara garis besar, jalannya sistem dibagi menjadi dua proses proses enkripsi dan proses dekripsi.



Gambar 2. *Flowchart* Enkripsi

a. Proses Enkripsi:
- Program dimulai.
- Pengguna diminta untuk memasukkan *seed phrase* dan *secret key*.
- *Seed phrase* dan *secret key* dimasukkan ke dalam algoritma enkripsi AES untuk menghasilkan *ciphertext seed phrase*.
- *Secret key* dienkripsi menggunakan algoritma *caesar cipher*.
- Pengguna diminta untuk memasukkan dua alamat *email* yang berbeda untuk penyimpanan hasil enkripsi.
- Hasil enkripsi *seed phrase* dan *secret key* disimpan secara digital di dua alamat *email* yang telah ditentukan.
- Proses selesai.



Gambar 3. *Flowchart* Dekripsi

b. Proses Dekripsi:
- Program dimulai.
- Pengguna diminta untuk memasukkan *secret key* yang terenkripsi dan *seed phrase* terenkripsi.
- *Secret key* dan *seed phrase* terenkripsi dimasukkan ke dalam algoritma dekripsi masing-masing.
- Hasil dekripsi *secret key* dan *seed phrase* ditampilkan.
- Proses selesai.

2.6. Pengujian Sistem

Pengujian sistem bertujuan untuk mengevaluasi tingkat keberhasilan dalam mengamankan *seed phrase* dan *secret key*, serta untuk memastikan bahwa sistem yang telah dibuat dapat berjalan tanpa mengalami *error*. Pengujian dilakukan dengan menggunakan skenario-skenario simulasi yang telah disiapkan sebelumnya.

3. Hasil dan Pembahasan

Penelitian ini menciptakan sebuah sistem perlindungan *seed phrase* dengan menggunakan kombinasi algoritma *Advanced Encryption Standard* (AES) dan *Caesar Cipher*. Sistem ini bertujuan untuk meningkatkan keamanan *seed phrase*, yang merupakan kunci akses utama dalam ekosistem *cryptocurrency*. Dengan menerapkan dua tahapan enkripsi, yaitu pertama, enkripsi *seed phrase* menggunakan AES, dan kedua, enkripsi kunci rahasia AES menggunakan *Caesar Cipher*, serta penggunaan email sebagai media penyimpanan digital. Sistem ini memberikan lapisan perlindungan tambahan terhadap serangan dan pencurian *seed phrase*. Dengan demikian, pengguna dapat memiliki keyakinan yang lebih tinggi dalam keamanan aset kriptonya.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

3.1. Pengujian Sistem

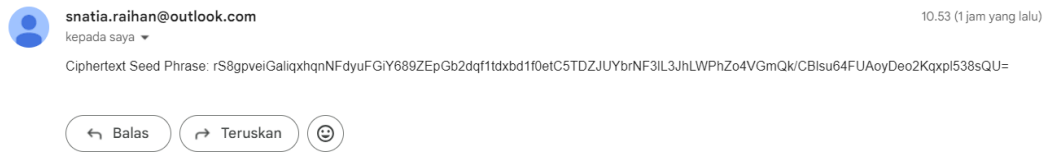
a. Pengujian Program Enkripsi

```
PERLINDUNGAN SEED PHRASE
Menu:
1. Enkripsi Seed Phrase
2. Dekripsi Seed Phrase
Pilih menu (1/2): 1
Masukkan seed phrase: belt rose change fire resource general churn walk turkey maximum bicycle crush
Masukkan secret key (huruf lebih aman): raihanseedphrase
Masukkan Email Untuk Penyimpanan Seed Phrase: raihanakbarmaulana29@gmail.com
Masukkan Email Untuk Penyimpanan Secret Key: legalistahd@gmail.com
Pesan enkripsi berhasil dikirim.
```

Gambar 4. *Output* Program Enkripsi

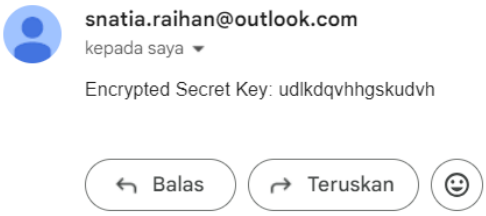
Pengujian pertama kami lakukan dengan mengenkripsi *seed phrase*, *output* dari hasil enkripsi seed *phrase* terlihat pada gambar 4. Dimasukkan *plaintext seed phrase* "*belt rose change fire resource general churn walk turkey maximum bicycle crush*" dengan *secret key* "raihanseedphrase" maka akan menghasilkan *ciphertext* yang dikirim ke dua *email* yang berbeda yaitu *email* 1 ke "raihanakbarmaulana29@gmail.com" dan *email* 2 ke "legalistahd@gmail.com". Tampil pesan enkripsi berhasil dikirim yang menandakan hasil enkripsi *seed phrase* dan *secret key* berhasil disimpan ke alamat *email* yang ditentukan.

b. Pengujian Pengiriman *Email*



Gambar 5. *Email Seed Phrase* Terenkripsi

Melanjutkan dari pengujian program enkripsi, kami menguji keberhasilan pengiriman *email* pertama. Dapat dilihat pada gambar 5, *email ciphertext seed phrase* berhasil terkirim dengan baik ke alamat *email* yang sudah ditentukan.

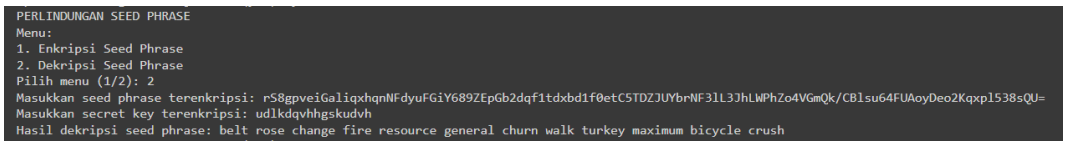


Gambar 6. Email *Secret Key* Terenkripsi

Melanjutkan dari pengujian program enkripsi, kami menguji keberhasilan pengiriman *email* kedua. Dapat dilihat pada gambar 6, *email encrypted secret key* berhasil terkirim dengan baik ke alamat *email* yang sudah ditentukan.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

c. Pengujian Program Dekripsi



Gambar 7. *Output* Program Dekripsi

Pengujian terakhir kami lakukan dengan mendekripsi *seed phrase*, *output* dari hasil dekripsi seed *phrase* terlihat pada gambar 7. Dimasukkan *ciphertext seed phrase* "rS8gpveiGaliqxhqnNFdyuFGiY689ZEpGb2dqf1tdxbd1f0etC5TDZJUYbrNF3lL3JhLWPhZo4VGmQk/CBlsu64FUAoyDeo2Kqxpl538sQU=" dengan *secret key* "udlkdqvhhgskudvh" yang kami dapatkan dari dua *email* yang menjadi tempat penyimpanan digital *seed phrase*. Maka akan menghasilkan *plaintext seed phrase* yang ditampilkan pada program yaitu "*belt rose change fire resource general churn walk turkey maximum bicycle crush*". Program berjalan dengan baik.

4. Kesimpulan

Dari penelitian yang telah dilakukan, kombinasi algoritma *Advanced Encryption Standard* (AES) dengan *Caesar Cipher* dapat diimplementasikan dalam enkripsi dan dekripsi *seed phrase*. Pada tahap enkripsi dijalankan menggunakan algoritma AES terlebih dahulu, lalu dilanjutkan dengan algoritma *Caesar cipher* untuk mengenkripsi *secret key* AES. Lalu hasil dari enkripsi di kirim ke dua *email* yang berbeda untuk penyimpanan secara digital. Selanjutnya, pada tahap dekripsi dijalankan dengan algoritma Caesar cipher dahulu untuk mendekripsi *secret key* lalu dilanjutkan dengan algoritma AES untuk mendekripsi *seed phrase*. Dengan penggabungan kedua algoritma ini membuktikan bahwa keamanan data *seed phrase* akan semakin kuat dan terjaga, karena menggunakan perlindungan enkripsi *dual-layer* algoritma untuk melindungi data *seed phrase*.

Daftar Pustaka

[1] Bukhari, S. T., Janjua, M. U., & Qadir, J. (2024). Secure Storage of Crypto Wallet Seed Phrase Using ECC and Splitting Technique. *IEEE Open Journal of the Computer Society*, 1–12. https://doi.org/10.1109/OJCS.2024.3398794

[2] Shaik, C. (2020). Unforgettable User Defined Seed Phrase for Cryptocurrency Wallets. *International Journal on Cryptography and Information Security*, *10*(4), 11–20. https://doi.org/10.5121/ijcis.2020.10402

[3] Gyawali, Y. (2020). *Encryption Algorithm Advanced Encryption Standard*. https://www.researchgate.net/publication/345684900

[4] Vincenttrijmen, J. (2020). *Information Security and Cryptography the Design of Rijndael the Advanced Encryption Standard (AES) Second Edition*. http://www.springer.com/series/4752

[5] Shaik, C. (2020). Securing Cryptocurrency Wallet Seed Phrase Digitally with Blind Key Encryption. *International Journal on Cryptography and Information Security*, *10*(4), 1–10. https://doi.org/10.5121/ijcis.2020.10401

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong