IMPLEMENTASI AES-256 DALAM APLIKASI MANAJEMEN PASSWORD MENGGUNAKAN BAHASA PEMROGRAMAN JAVA BERBASIS DESKTOP

Akbar Hendra Jaya*[1], Afnan Rosyidi[2]
[12]Program Studi Manajemen Informatika, STMIK Amikom Surakarta
[12]Sukoharjo Indonesia
Email: [1]akbar.30549@mhs.amikomsolo.ac.id , [2]afnan@dosen.amikomsolo.ac.id

Abstract

Data breaches and cybersecurity incidents are increasingly alarming in this digital era, making secure password management crucial. This research develops a password management application that uses AES-256 and a master password to protect sensitive information. The application also offers a strong password generator, which helps users create passwords that are difficult to guess and more secure. With these features, the application is designed to minimize the risk of weak or repeated password usage. This application helps users manage their passwords more securely, reduces the risk of data breaches, and provides a sense of security in managing personal and sensitive information.

Keywords: *Data Security, Digital Security, Encryption, Advanced Encryption Standard (AES)*

Abstraksi

Kebocoran data dan pelanggaran keamanan siber semakin mengkhawatirkan di era digital ini, sehingga manajemen kata sandi yang aman menjadi sangat penting. Penelitian ini mengembangkan aplikasi manajemen kata sandi yang menggunakan enkripsi AES-256 dan master password untuk melindungi informasi sensitif. Aplikasi ini juga menawarkan generator kata sandi kuat, yang membantu pengguna membuat kata sandi yang sulit ditebak dan lebih aman. Dengan fitur-fitur ini, aplikasi dirancang untuk meminimalkan risiko penggunaan kata sandi yang lemah atau berulang. Dengan demikian, aplikasi ini membantu pengguna mengelola kata sandi mereka dengan lebih aman, mengurangi risiko kebocoran data, dan memberikan rasa aman dalam pengelolaan informasi pribadi dan sensitif.

Kata Kunci: *Advanced Encryption Standard (AES), Enkripsi, Keamanan Data, Keamanan Digital*

1. PENDAHULUAN

Seiring berjalannya waktu, kemajuan teknologi komputer dan telekomunikasi telah menjadi kebutuhan yang sangat berguna untuk melakukan banyak tugas dengan cepat, tepat, dan akurat. Namun, diyakini bahwa aspek keamanan data penting untuk informasi sensitif, karena juga memiliki efek negatif dari orang yang tidak berwenang menyadap data penting.

Data merupakan salah satu aset penting dan berharga baik bagi individu maupun organisasi. Dalam Sistem Informasi Manajemen (SIM), data yang disimpan dapat berupa data keuangan, data pelanggan, data karyawan, dan data penting lainnya. Oleh karena itu, keamanan data dalam SIM sangat penting untuk melindungi informasi dari ancaman seperti pencurian data, kerusakan fisik terhadap sistem informasi, dan ancaman lainnya. Artikel ini bertujuan untuk membahas tentang keamanan data dalam SIM, risiko yang dapat terjadi pada keamanan data, dan strategi perlindungan yang dapat dilakukan untuk mengatasi risiko tersebut[1].

Enkripsi adalah salah satu solusi atau metode keamanan data terbaik untuk menjaga kerahasiaan dan keandalan data, serta dapat meningkatkan keamanan data atau informasi data. Kriptografi adalah bidang ilmiah yang mempelajari teknologi enkripsi data. Kriptografi dibagi menjadi dua jenis, klasik dan modern, dengan kunci simetris dan asimetris[1]. Kunci simetris adalah *Advanced Encryption Standard* (AES) atau kadang disebut sebagai *Rijndael*. AES adalah algoritma enkripsi aman untuk melindungi data atau informasi sensitif menggunakan berbagai teknik enkripsi dan dekripsi panjang kunci seperti 128-bit, 192 bit, dan 256 bit[2].

Oleh sebab itu, implementasi manajemen *password* berbasis desktop yang aman menjadi krusial dalam menjaga keamanan data sensitif. Dalam konteks ini, penggunaan algoritma enkripsi AES-256 dengan bahasa pemrograman Java menjadi solusi yang efektif untuk melindungi *password* dari ancaman data bocor dan dapat digunakan untuk berbagai sistem operasi. Karena AES-256 menawarkan tingkat keamanan yang tinggi dengan panjang kunci 256 bit, yang membuatnya sangat sulit untuk dipecahkan oleh pihak yang tidak berwenang[3].

Resiko terjadinya data bocor akan mengakibatkan kerugian yang sangat fatal dalam perusahaan, negara, ataupun instansi lainnya, selain dari resiko finansial dan reputasi terdapat juga tekanan dari pihak hukum dan beberapa tahun kemarin terdapat perusahaan kredit Equifax mengalami data bocor dan pemerintah Indonesia [4].

Manajemen *Password* memiliki penelitian terdahulu tetapi masih berbasis *web* dan dalam penelitian ini akan menggunakan basis *desktop* untuk memberikan fitur yang lebih luas, belum lagi kerawanan untuk diretas dengan mudah jika terbuka untuk umum [5].

Dengan demikian karena rawannya menyimpan password yang mudah bocor serta pembuatan password yang mudah diretas oleh sebab itu penulis mengembangkan aplikasi manajemen password berbasis desktop.

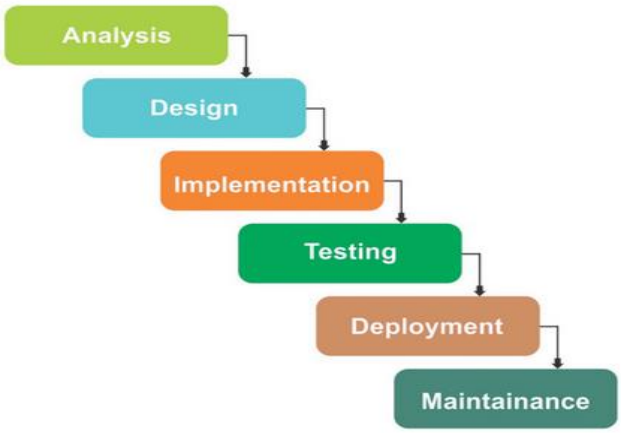
2. TINJAUAN PUSTAKA

Dari penelitian [5], peneliti tersebut menggunakan teknologi web untuk membuat sistem manajemen password, oleh sebab itu saya sebagai penulis akan membuat sistem manajemen password berbasis *desktop* menggunakan bahasa pemrograman Java dan algoritma kriptografi AES-256, karena AES-256 terbilang lebih baru dan lebih aman.

Dari penelitian [6], peneliti tersebut menggunakan algoritma AES-256 dan SHA-256 untuk mengamankan aplikasi Al-Quran digital, makah saya sebagai penulis menggunakan algoritma AES-256 memiliki keyakinan bahwa algoritma tersebut lebih aman dibandingkan dengan AES-128.

3. METODE PENELITIAN

Metode penelitian yang digunakan ialah metode Waterfall. Waterfall adalah suatu model pengembangan perangkat lunak yang menggunakan pendekatan yang berurutan serta sistematis seperti air terjun yang mengalir dari atas kebawah[7] tersaji pada gambar 1.



Gambar 1. Metode *Waterfall*

Peneliti akan menggunakan metode tersebut karena cocok untuk melakukan pengujian algoritma AES-256, dimulai dari studi literatur tentang AES-256 agar peneliti menggunakan algoritma tersebut dengan benar, lalu melakukan desain dan implementasi, dan percobaan dilapangan.

4. HASIL DAN PEMBAHASAN

Secara garis besar algoritma AES sebagai berikut :

1. Ekspansi Kunci : Algoritma AES memerlukan 128bit blok kunci untuk setiap fase.
2. Fase pertama : setiap byte dari state dikombinasikan dengan kunci fase dengan bitwise xor.
3. 9, 11 atau 13 fase:
    1. Dilakukannya substitusi non linier pada setiap byte dan diganti menggunakan data di lookup table.

2. Menggeser Baris dimana 3 baris state digeser secara bergantian dengan step tertentu.
3. Mencampur Kolom.
4. Menjumlahkan kunci fase.

4. Fase terakhir:
    1. Dilakukannya substitusi non linier pada setiap byte dan diganti menggunakan data di lookup table.
    2. Menggeser Baris dimana 3 baris state digeser secara bergantian dengan step tertentu.
    3. Menjumlahkan kunci fase.

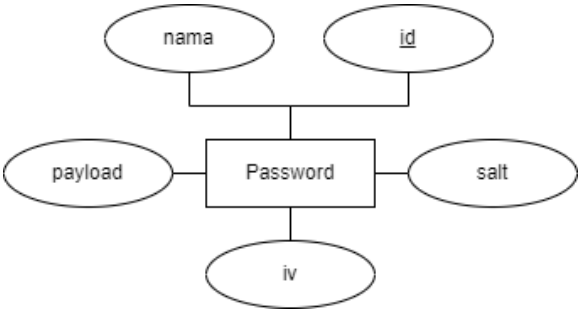
Untuk sistem manajemen *password* ini penulis menggunakan AES-256 sebagai algoritma enkripsi dalam manajemen *password* didasarkan pada keamanan tinggi yang ditawarkannya. Dengan panjang kunci 256 bit, AES-256 sangat sulit untuk diretas atau didekripsi tanpa kunci yang tepat, menjadikannya pilihan ideal untuk melindungi data sensitif seperti password. Selain itu, AES telah diadopsi secara luas sebagai standar internasional dalam kriptografi, yang berarti telah diuji dan terbukti aman dalam berbagai aplikasi di seluruh dunia. Keamanan yang tinggi dari AES-256 juga tidak mengorbankan efisiensi dan kinerja, memungkinkan operasi enkripsi dan dekripsi yang cepat dan responsif. Implementasi AES-256 juga didukung oleh banyak perangkat lunak dan perpustakaan kriptografi, memudahkan integrasi dalam berbagai lingkungan dan platform. Secara keseluruhan, AES-256 memenuhi berbagai persyaratan keamanan, baik dari segi teknis maupun regulasi, menjadikannya pilihan yang andal dan terpercaya dalam melindungi informasi sensitif seperti password dalam sistem informasi manajemen[2], [8]. Berikut tabel struktur dari AES tersaji pada tabel 1.

Tabel 1. Struktur AES

Tipe AES + Kunci	Blok Data	Blok Matriks	Jumlah Fase
AES-128	128	4*4	10
AES-192	128	4*6	12
AES-256	128	4*8	14

1.1. Uraian Alur Sistem

1.1.1. ERD

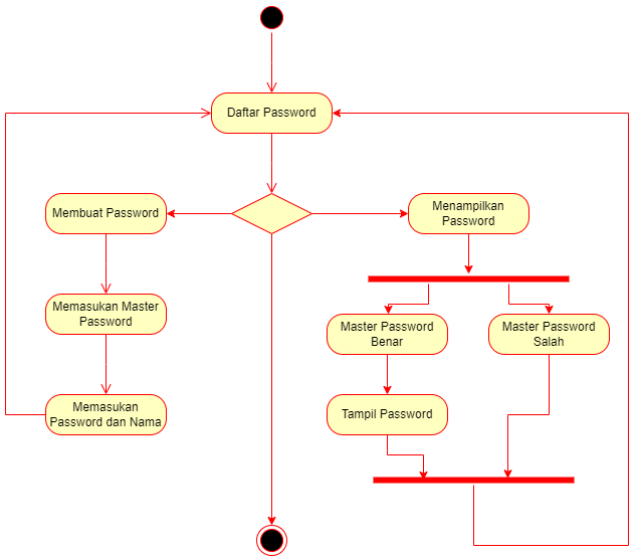


Gambar 2. ERD Aplikasi

Pada gambar 2 diatas merupakan Diagram ERD untuk aplikasi sistem manajemen password untuk memudahkan penulis untuk merancang database yang digunakan dalam sistem tersebut. Dari gambar diatas terdapat entitas password yang memiliki nama untuk memberikan deskripsi dari password tersebut dan juga terdapat *payload, salt,* dan *iv* untuk menyimpan hasil enkripsi AES-256.

1.1.2. Activity Diagram

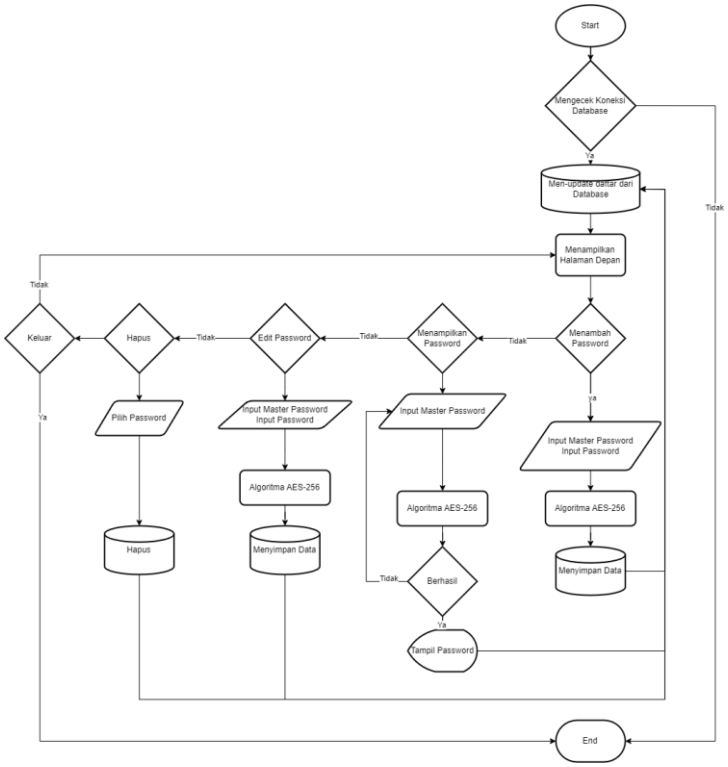
Activity Diagram adalah suatu diagram yang mengambarkan alur kerja suatu aplikasi atau sistem yang sedang berjalan[9] tersaji pada gambar 3.



Gambar 3. Alur Sistem *(Activity Diagram)*

Alur sistem dari sistem yang penulis buat adalah pengguna akan diarahkan daftar *password* yang sudah disimpan dan dapat untuk menambah password ataupun menampilkan *password* baru.

1.1.3. Flowchart



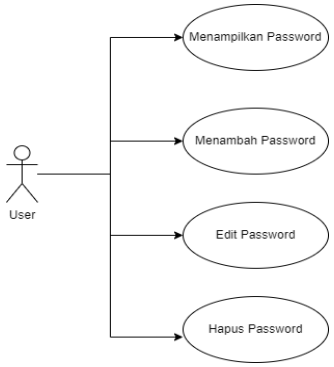
Gambar 4. Flowchart sistem manajemen password

Pada gambar 4 diatas merupakan *Flowchart* untuk membuat sebuah sistem manajemen *password*, *flowchart* ini bertujuan untuk memudahkan penulis untuk membuat sebuah program. Pertama *flowchart* dimulai dengan *Start* kemudian dilanjutkan dengan pengecekan koneksi database dan men-*update* tampilan utama dan menunggu interaksi pengguna untuk menggunakan fitur seperti membuat password, men-edit *password*, menghapus *password* ataupun keluar.

Sebelum password disimpan akan diminta Master *password* dan nama *password* tersebut untuk disimpan, dan untuk di edit akan men-edit keseluruhan entri password tersebut, *password* akan diberikan *salt* dan *initial vector* secara random agar meningkatkan keamanan data, dan hapus akan menghapus entri *password* setelah perubahan database sistem akan melakukan *refresh* menu utama agar daftar *password* tetap *up to date*.

1.1.4. Use Case Diagram

Use Case Diagram adalah Visualisasi yang menunjukan interaksi antara sistem dengan komponen yang lainnya, seperti pengguna ataupun perangkat lain. Visualisasi ini dapat membantu kita memahami sistem bekerja[9].



Gambar 5. Use Case Diagram

Pada gambar 5 diatas merupakan *Use Case Diagram* untuk memudahkan penulis dalam pembuatan fitur-fitur apa saja yang dapat digunakan oleh pengguna. Pada fitur User dapat mengakses *password*, menambah *password*, men-edit *password* dan menghapus *password*.

1.2. Pengujian Algoritma AES-256 dengan Algoritma Yang Lainnya

Peneliti akan menggunakan AES-256 dan AES-128 dalam perbandingan ini karena kedua algoritma tersebut merupakan satu tipe yang sama, dalam pengujian ini kedua algoritma tersebut akan di implementasikan didalam aplikasi peneliti buat. Berikut merupakan hasil dari perbandingan kecepatan enkripsi, dan dekripsi, pengujian ini termasuk dengan pembuatan *initial vector* dan *salt* tersaji pada tabel 2.

Tabel 2. Perbandingan Kecepatan AES-128 dan AES-256

Pengujian	AES-128	AES-256
Enkripsi	89ms	142ms
Dekripsi	39ms	41ms

Dari tabel diatas peneliti dapat menyimpulkan bahwa algoritma AES-256 lebih lambat tetapi walaupun memiliki kelemahan dibidang kecepatan bukan berarti disaat terjadi peretasan atau kebocoran data dapat diretas dengan mudah akan tetapi akan memperlama peretasan.

Peneliti juga melakukan *brute force* dengan menggunakan aplikasi HashCat yang digunakan dalam penelitian [10], terhadap enkripsi tersebut akan tetapi karena keamanan AES-256 terlalu kuat dan membuat pengujian secara *brute force* tidak dapat dilakukan jika *secret key* yang digunakan ataupun *salt* ataupun *initial vector* tidak

ditemukan oleh karena itu harus melakukan peretasan dalam database tersaji pada tabel 3.

Tabel 3. Perbandingan Kecepatan Brute Force AES-128, AES-256, MD5, SHA256

MD5	SHA256	AES-128	AES-256
34s	30s	> 24 jam (Terlalu Lama)	> 48 jam (Terlalu Lama)

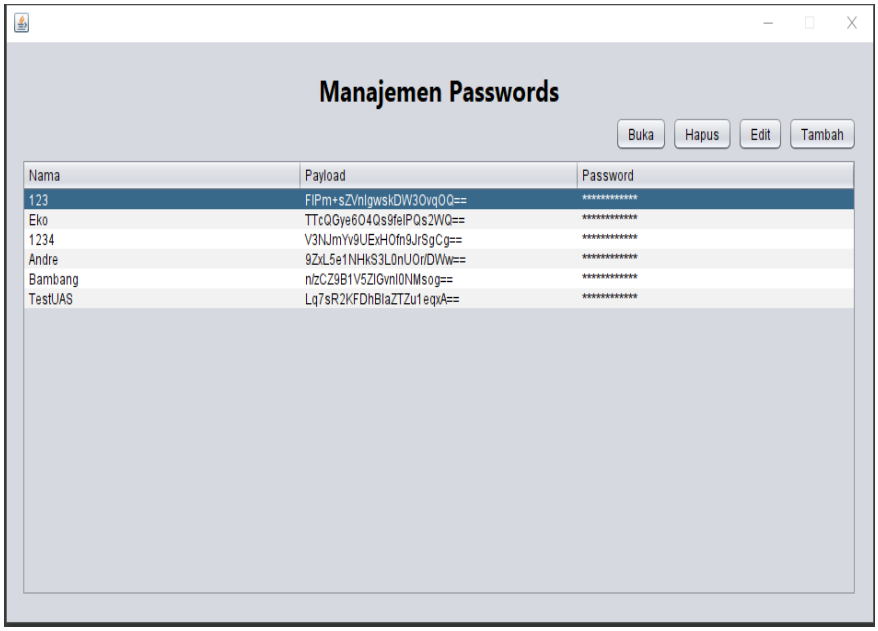
Hardware yang digunakan dalam pengujian tersebut menggunakan spesifikasi hardware yang cukup dengan sumber daya dan memorinya. Berikut ini spesifikasi hardware tersaji pada tabel 4.

Tabel 4. Spesifikasi Hardware

Hardware	Spesifikasi
PC Laptop HP 14-cm0xxx	Processor : AMD Ryzen 5 2500U @2.0 Ghz VGA : AMD RADEON(TM) Vega 8 Graphics RAM : 8 GB DDR 4 Storage : - SSD SanDisk SD9SN8W 128GB - HDD TOSHIBA MQ04ABF100 1TB OS Windows 10 x64 bit

1.3. Screenshot Hasil Aplikasi

Pada tampilan ini pengguna dapat melihat, menghapus, men-edit *password* yang sudah ada dan juga dapat menambah *password* baru dengan mengisikan nama *password* dan master *password* tersaji pada gambar 5.



Gambar 5. Tampilan Menu Utama

5. KESIMPULAN

Aplikasi manajemen password ini menawarkan sejumlah kelebihan yang signifikan, seperti keamanan yang lebih tinggi berkat penggunaan algoritma AES-256, ringan dan tidak memerlukan spesifikasi perangkat yang tinggi, serta kompatibilitas lintas platform yang memungkinkan penggunaan di berbagai sistem operasi dan perangkat melalui penyimpanan data di database MySQL. Namun, terdapat juga beberapa kekurangan, antara lain kecepatan enkripsi AES-256 yang sedikit lebih lambat dibandingkan AES-128, kebutuhan akan database untuk operasional aplikasi, serta aplikasi ini belum dirilis secara resmi karena masih terdapat beberapa kekurangan yang perlu diperbaiki. Secara keseluruhan, meskipun memiliki potensi yang kuat dalam hal keamanan dan fleksibilitas, pengembangan lebih lanjut diperlukan untuk mengatasi kendala yang ada sebelum aplikasi ini dapat digunakan secara luas.

6. SARAN

Aplikasi manajemen password dapat dikembangan ke bagian android ataupun iOS, menambahkan fitur pengingat dan serta pembuatan password yang secara acak tanpa diperlukannya input oleh pemakainya, serta pengujian selain *brute force* dikarenakan terlalu lama untuk melakukan pengujian untuk algoritma AES.

DAFTAR PUSTAKA

[1] R. Rivaldi and S. Subandi, “Implementasi Keamanan Data Arsitektur Menggunakan Algoritma Kriptografi Dengan Metode Rivest Code (4 Rc4)Pada Pt.Naviri Indah Cemerlang,” *SKANIKA Sist. Komput. Dan Tek. Inform.*, vol. 4, no. 2, Art. no. 2, Jul. 2021, doi: 10.36080/skanika.v4i2.2249.

[2] T. B. I. Guy-Cedric and S. R, “A Comparative Study on AES 128 BIT AND AES 256 BIT,” *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 6, no. 4, pp. 30–33, Aug. 2018.

[3] E. S. Marsiani, I. Setiadi, and A. Cahyo, “Implementasi Sistem Keamanan AES 256-Bit GCM Guna Mengamankan Data Pribadi,” *JRKT J. Rekayasa Komputasi Terap.*, vol. 1, no. 02, Art. no. 02, Jun. 2021, doi: 10.30998/jrkt.v1i02.4096.

[4] “Pentingnya Kepatuhan Keamanan Informasi Dalam Mengurangi Risiko Data Breach | Maeswara : Jurnal Riset Ilmu Manajemen dan Kewirausahaan.” Accessed: Nov. 13, 2024. [Online]. Available: https://journal.arimbi.or.id/index.php/Maeswara/article/view/587

[5] G. K. Rydwanto, “Pembuatan Aplikasi Personal Digital Assistant dan Web untuk Manajemen Password.” Accessed: Nov. 13, 2024. [Online]. Available: http://digilib.ubaya.ac.id/pustaka.php/137087

[6] A. S. Dewi and H. Setiawan, “Implementation of SHA-256 and AES-256 for Securing Digital Al Quran Verification System,” in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, Oct. 2019, pp. 1–8. doi: 10.1109/ICIC47613.2019.8985960.

[7] "Pengembangan Aplikasi MySaku Menggunakan Metode Waterfall | Indonesian Technology and Education Journal." Accessed: Nov. 13, 2024. [Online]. Available: https://journal.diginus.id/ITEJ/article/view/178

[8] R. Andriani, S. E. Wijayanti, and F. W. Wibowo, "Comparision Of AES 128, 192 And 256 Bit Algorithm For Encryption And Description File," in *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*, Nov. 2018, pp. 120–124. doi: 10.1109/ICITISEE.2018.8720983.

[9] S. Pranoto, S. Sutiono, Sarifudin, and D. Nasution, "Penerapan UML Dalam Perancangan Sistem Informasi Pelaporan Dan Evaluasi Pembangunan Pada Bagian Administrasi Pembangunan Sekretariat Daerah Kota Tebing Tinggi," *Surpl. J. Ekon. Dan Bisnis*, vol. 2, no. 2, Art. no. 2, Jun. 2024.

[10] "Prob-Hashcat: Accelerating Probabilistic Password Guessing with Hashcat by Hundreds of Times | Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses." Accessed: Nov. 13, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3678890.3678919