Proxecto CyberCaesar

Datos personais dos participantes:

-Nicolás García Ausín, IES Castelao

-Xian Couto Viqueira, IES Castelao

Resumen do traballo:

Vivimos tempos nos que a nosa seguridade dixital se ve comprometida costantemente, e non é cuestión que deba tomarse á lixeira: os nosos datos poderían verse vulnerados en calquera momento. Isto débese, na gran maioría dos casos, a que os algoritmos que cifran os nosos datos son antigos ou débiles. E aínda que non o sexan, o seu funcionamento é público e calquera persoa con coñecementos podería programar un descifrador simple.

O proxecto CyberCaesar nace da necesidade de privacidade absoluta en tempos nos que predomina a presenza dixital. CyberCaesar brinda seguridade grazas á combinación de múltiples formas de cifrado, como operacións XOR ou funcións hash SHA-256. Ademais, a aplicación incorpora unha interface gráfica intuitiva e sinxela, que a fai fácil de manexar para calquera persoa, facendo que este proxecto traia seguridade a todos os usuarios que precisen cifrar calquera tipo de arquivos, xa sexa PDF, MOV ou STL. Calquera arquivo está a salvo con CyberCaesar.

Introduccion:

En 2025, 16.000 millóns de credenciais de redes sociais foron roubadas por cibercriminais que conseguiron sortear os sistemas de cifrado das bases de datos debido a algoritmos decadentes. A ciberseguridade é unha prioridade no mundo actual; por iso, é importante gardar ben as nosas credenciais virtuais cun bo sistema de cifrado que garanta a seguridade tanto persoal como comunitaria, neste contexto, o proxecto CyberCaesar propón un sistema de cifrado híbrido que combina operacións XOR, funcións hash SHA-256 e o estándar AES-256-GCM, creando múltiples capas de protección para dificultar o acceso non autorizado aos datos.

Proposito do Proxecto:

O propósito de CyberCaesar é desenvolver unha ferramenta de cifrado de arquivos moi robusta, accesible para todo o mundo e en todas partes. Sen necesidade de coñecementos técnicos, o usuario non ten que saber nada sobre cifrados ou algoritmos para garantir a súa seguridade.

Estudo do estado da arte:

Hoxe existen ferramentas como VeraCrypt (cifrado de discos con AES-256), AxCrypt (arquivos individuais) ou BitLocker (integrado en Windows). O NIST establece estándares como AES-256-GCM, e obras como 'Applied Cryptography' de Schneier son referencia. CyberCaesar achega como novidade a combinación secuencial de XOR, SHA-256 e AES-256-GCM, xunto cunha interface con depuración visual que permite seguir o proceso paso a paso.

Hipotese:

Partimos da hipótese de que se podería facer un algoritmo de cifrado máis potente que o ampliamente usado SHA-256, combinando técnicas deste mesmo co AES-256 nun mesmo fluxo de traballo, portas lóxicas XOR e un sistema interno de 1000 bits. O cifrado sería máis robusto ante calquera ataque. Tamén gustaríanos saber se é posible mesturar semellante nivel de seguridade cunha interface gráfica sinxela.

Material e métodos:

Para o desenvolvemento de CyberCaesar utilizouse a linguaxe Python (versión 3.11), empregando as bibliotecas cryptography (para AES-256-GCM), hashlib (para SHA-256) e tkinter (para a interface gráfica). O código organízase en dúas clases principais: CifradorDebug (encargada da lóxica de cifrado) e CifradorDebugApp (xestión da interface gráfica e eventos).

O proceso de cifrado consta de catro fases:

1. Derivación da clave: mediante PBKDF2 con SHA-256, 100.000 iteracións e un salt de 16 bytes.

2. Pre-cifrado: aplicación dunha máscara XOR sobre os datos usando a clave derivada.

3. Cifrado principal: AES-256-GCM con autenticación e nonce de 12 bytes.

4. Post-cifrado: engadido dun hash SHA-256 para verificar a integridade.

Para evitar bloqueos na interface mentres se procesan arquivos grandes, utilizouse o módulo threading, creando fíos independentes para as operacións de cifrado e descifrado. Isto permite que a barra de progreso se actualice en tempo real e que a interface siga respondendo.

As probas realizáronse con 30 arquivos de diversos formatos (PDF, STL...) e tamaños (desde 1 KB ata 500 MB), rexistrando tempos de proceso e taxas de éxito.

Resultados:

As probas realizadas con 30 arquivos de diferentes formatos (PDF, MOV, STL, TXT) e tamaños (desde 1 KB ata 500 MB) amosaron os seguintes resultados:

- Taxa de éxito do 100% na desencriptación cando se utilizou o código correcto. Con códigos incorrectos, o sistema rexeitou a desencriptación en todos os casos, confirmando a robustez da autenticación.

- O tempo de cifrado foi lineal respecto ao tamaño do arquivo, acadando unha media de 12 MB/s en arquivos de gran tamaño.

- O tamaño dos arquivos cifrados aumentou nunha media do 5,2% debido á información adicional (salt, nonce, tag e metadatos).

- O uso de threading permitiu que a interface permanecese responsive durante todo o proceso, sen bloqueos.

Interpretamos que a triple capa de cifrado non afecta significativamente ao rendemento, pero si aumenta a seguridade, xa que calquera erro nunha das fases impide a recuperación dos datos.

Conclusions:

Os resultados confirman a hipótese: a combinación de SHA-256, AES-256, XOR e 1000 bits crea un cifrado máis robusto. A triple capa rexeitou o 100% dos intentos con código incorrecto, mantendo 12 MB/s de media.

Contrastando coa literatura (Schneier, NIST), a integración de técnicas supera ao uso illado de algoritmos estándar. Demostrouse que é posible mesturar alta seguridade cunha interface sinxela grazas a threading.

Cara o futuro, non se recomenda agregar almacenamento en nube: estes servizos son susceptibles a ataques que nin o cifrado pode protexer unha vez que os datos están en mans alleas.

Webgrafía:

NIST Computer Security Resource Center. Cryptographic Standards and Guidelines. Dispoñible en liña. [Consulta: marzo 2026]

National Institute of Standards and Technology. FIPS 197: Advanced Encryption Standard (AES). Dispoñible en liña.

NIST. SP 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM). Dispoñible en liña.

Python Software Foundation. hashlib — Secure hashes and message digests. Documentación oficial. Dispoñible en liña.

Python Software Foundation. tkinter — Python interface to Tcl/Tk. Documentación oficial. Dispoñible en liña.

Cryptography.io. Cryptography documentation — AES-GCM. Dispoñible en liña.

OWASP Foundation. Cryptographic Storage Cheat Sheet. Dispoñible en liña.

Schneier, B. Applied Cryptography, Second Edition. Recursos e código fonte. Dispoñible en liña.

IEEE Xplore. Bit-Based MILP Modelling of Non-Bit-Permutation Linear Layers. AsiaJCIS. Dispoñible en liña.

IACR Communications in Cryptology. Masked Computation of the Floor Function. Dispoñible en liña.