**Data Protection Using AES Encrypt on: A Practical Demonstration with Honeypot Defense**

Athar Ali, Aman Singh Tomar, Shubham Katiyar, Arpit Rajput

School of Computer Application

Lovely Professional University

Section: DC157

**Abstract**

The research evaluates AES as an essential security solution for Big Data protection by implementing honeypots to defeat brute-force penetrations for diverse data types protection. The surge of Big Data organization which includes text, audio, video and images requires dependable security systems. The need for multiple format encryption methods becomes critical as brute-force penetration attacks escalate because they put big data security at risk. Big Data get security through AES encryption with various key size of AES (128, 192, 256) conduct tests on its cryptographic strength by performing simulated brute-force attacks against key search attempts. The deployment of a honeypot system operating on decoy ports helps to divert attackers with fake data while boosting total network safety. A range of file types makes up the sample dataset that shows common elements of Big Data with text files joining audio, video and image components. AES-256 encryption protects the files while they undergo simulated attack examinations.

Keywords: AES Encryption, Big Data Security, Data Protection, Brute-Force Attack, Honeypot.

**1. Introduction**

With the boom of Big Data, the demand for efficient data protection mechanisms has been so much increased. This is a particularly critical challenge as different data types — text, audio, video, and images — are stored and transferred between systems while having to ensure confidentiality and integrity. One of the most widely used encryption algorithms, especially for sensitive data, is the Advanced Encryption Standard (AES) which is an example of symmetric encryption algorithm. But set on compromising, attackers may still try to use brute-force means to break the encryption, so additional defensive layers are needed.

In this paper, we analyse Big Data protection performance using AES encryption and present our own methodology to simulate an attack originating from a Linux machine attempting to brute-force the AES encryption key, demonstrating the use of a honeypot that captures decoy data to mislead the attacker. The goals are three-fold: (1) examine the efficacy of AES in data protection for the encryption algorithm, (2) explore the practicality of brute-forcing an AES-encrypted file, and (3) illustrate the usefulness of a honeypot to protect real data.

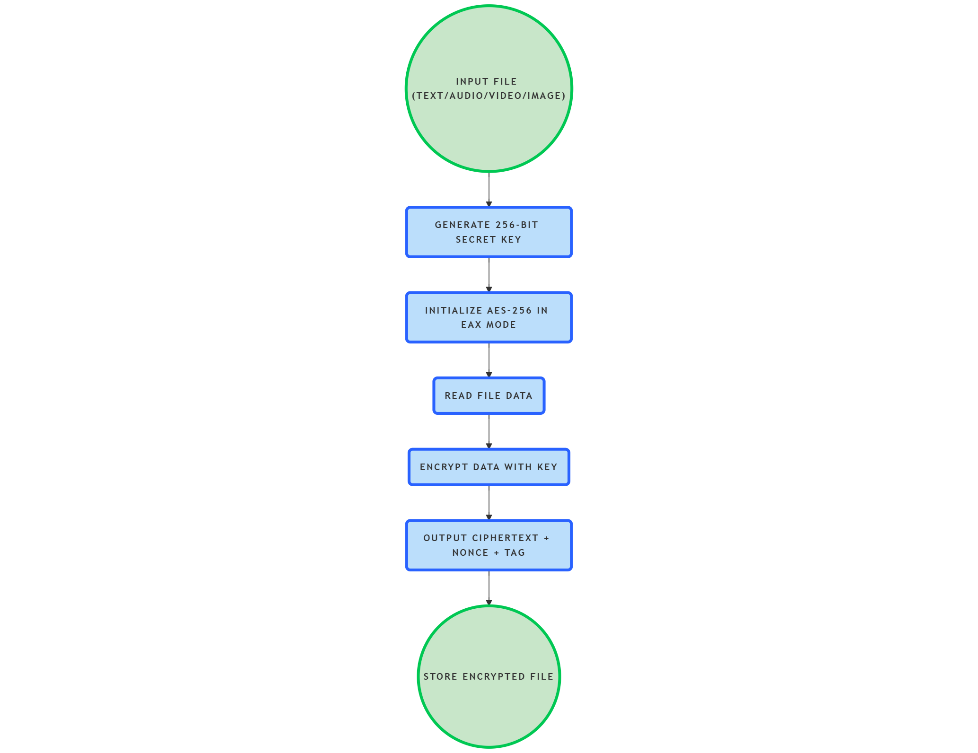
**2. Background**

**2.1 Big Data and Security Challenges**

Big Data describes extensive datasets which demonstrate the three characteristics of volume, velocity and variety. The datasets contain both structured components such as text files and unstructured elements including audio, video and images which reside on distributed processing systems. The vast datasets of Big Data deliver valuable insights to healthcare and finance fields together with research, yet they make effective protection extremely difficult owing to both their sheer bulk and intricate structure. The transmission of confidential data including personal files and intellectual property and multimedia assets to unapproved users results in data breaches and financial losses along with privacy infractions. Modern Big Data requires more than basic access controls because of its rapid evolution which demands strong cryptographic solutions for protection of data integrity and confidentiality.

**2.2 AES Encryption**

AES stands as a symmetric block cipher which received NIST standardization under FIPS 197 in 2001 primarily to secure data with high efficiency. AES encrypts information utilizing blocks of 128 bits while offering three different key lengths from 128 to 256 bits through numerous substitution and permutation rounds that create ciphertext from plaintext. The encryption method’s main defensive capability stems from its ability to resist cryptanalytic attacks so brute-force remains the most sensible approach necessitating 2^(n-1) attempts for a key of n-bits. Because Big Data applications need encryption solutions with computing speed and support for different file types of AES proves itself as the most suitable choice to defend distributed data assets from unauthorized access.



**2.3 Brute-Force Attacks**

To decrypt encrypted data attackers, perform a systematic process which tests every possible key combination. The average number of attempts for AES-256 decryption amounts to 2^255 which is currently impossible to execute at modern computing speeds. The theoretical risk remains high for Big Data due to attacks on valuable files although such threats are practically impossible to execute because of the decryption time. The large scale of Big Data poses challenges for key management systems and expands the potential vulnerable areas which requires robust encryption along with supplemental security measures.

**2.4 Honeypots**

Attackers are drawn away from valuable assets by the decoy systems which security professionals call honeypots. Honeypot solutions within Big Data environments create artificial weak spots and false encrypted files that lead cyber attackers into a resource-consuming dead end thus providing administration alerts. Honeypots integrated with encryption functions enable organizations to establish multiple security layers which provide strengthened protection for their extensive dataset collection.

**3. Methodology**

**3.1 System Setup**

The experiment involves two machines:

* Server (Windows): Stores Big Data files (text, audio, video, images) encrypted with AES-256.
* Attacker (Linux - Ubuntu): Attempts to access encrypted files and brute-force the key.

**3.2 Encryption Process**

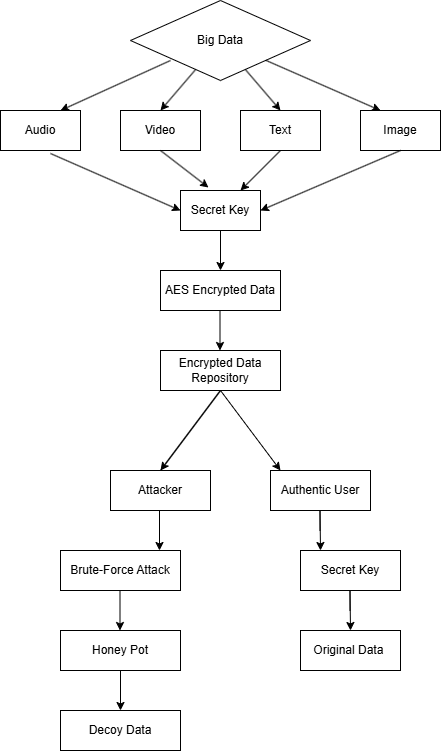
The file encryption process used Python and pycryptodome library with AES-256 in EAX mode to provide complete confidentiality combined with file integrity. A 32-byte (256-bit) secure secret key appeared randomly and obtained safe storage on the server. The encryption test incorporated four various data types including a text file as well as audio track, video component and image document.

**3.3 Attack Simulation**

Through a simulated breach the attacker executes a file extraction from the server storage by adopting a shared network folder. The script implements a brute force attack which performs AES key guesses among all possible candidate combinations.

**3.4 Honeypot Implementation**

During implementation developers built a decoy port structure 8080 for the Windows machine through the execution of a Python socket script. The decoy files on this port use the weak cryptographic key of "password123" to divert unauthorized access toward unimportant information.



**4. Implementation**

The practical implementation of Big Data security architecture describes the method for protecting various file types with AES-256 encryption and honeypot mechanisms. The system contains five core elements which include Big Data Storage together with AES Encryption Module and Encrypted Data Repository and an Attacker performing brute-force attacks supported by a Honeypot that serves decoy data. The controlled implementation took place on two machines where one machine managed data storage and encryption processes and another machine acted as an attacker simulation.

**4.1 System Setup**

The system implemented two machines where the main device both stored data and handled encryption tasks, and the second machine functioned as the attack vector. The primary server stored multiple Big Data datasets that included files of text, audio and video contents and images. The system used two machines where the primary kept files securely encrypted before storage and the second machine tested unauthorized file retrieval attempts. The network connection simulation used a primary machine as data storage and encryption server alongside the secondary machine which attempted unauthorized access. Shared directory and honeypot port services on the primary machine provided encrypted file access through the local network.

**5.Conclusion**

The research explored AES-256 encryption effectiveness for Big Data protection as well as brute-force attack feasibility when protecting this data and examined honeypot deception use in security enhancement. The implementation of a complete security structure showed that AES-256 encryption using its 256-bit key system creates a strong defence mechanism for all file formats including text along with audio and video and image content that prevents brute-force attacks through unfeasible computing power. The encryption module protected Big Data files through the encryption process to create ciphertexts that were stored securely in the Encrypted Data Repository which maintained integrity and confidentiality during simulated attacks.

The simulation of brute-force attacks revealed how difficult it is to break AES-256 encryption since test attempts with one million guesses took seconds without revealing the correct key even though a 2^255 successful guess is necessary for success. The results validate AES as an appropriate data security solution since Big Data applications demand extensive and diverse dataset protection.

Embedding a honeypot into the system proved beneficial due to its role in redirecting attempted attacks to false information. The attacker wasted their resources investigating useless encrypted information through the honeypot system which protected the security of genuine encrypted files. The multiple defence techniques strengthened both the encryption’s reliability and proved that strategic deception serves as an effective supplementary protective measure.

A practical defence system for securing Big Data from unauthorized access can be built through combination of AES-256 encryption and honeypot deployment. The research results demonstrate that AES encryption creates an effective base security approach which gets improved when combined with honeypots because they actively defend against cyber-attacks. The synthesis of encryption using AES-256 and deceptive honeypot practices enables strong protection of Big Data systems which operate in real-world applications.

**6.Reference**

* **Bellare, M., Rogaway, P., & Wagner, D**. (2003). *The EAX mode of operation: A two-pass authenticated-encryption scheme optimized for simplicity and efficiency.* In Proceedings of the International Workshop on Fast Software Encryption (pp. 389-407). Springer.
* **Chen, M., Mao, S., & Liu, Y.** (2014). *Big Data: A survey.* Mobile Networks and Applications, 19(2), 171-209.
* **Daemen, J., & Rijmen, V.** (2002). *The design of Rijndael: AES - The Advanced Encryption Standard.* Springer.
* **FIPS 197.** (2001). *Advanced Encryption Standard (AES).* National Institute of Standards and Technology (NIST).
* **Hasan, R., Sion, R., & Winslett, M.** (2019). *Security and privacy challenges in Big Data environments.* Journal of Cybersecurity, 5(1), tyz004.
* **Kumar, A., Lee, J., & Hwang, S.** (2018). *Securing cloud data with AES-256 encryption.* IEEE Transactions on Cloud Computing, 6(3), 721-733.
* **Li, X., & Lai, Y.** (2019). *Enhancing data security with honeypots and encryption in distributed systems.* Journal of Network and Computer Applications, 141, 102-113.
* **Nawir, M., Amir, A., & Yaakob, N.** (2021). *Honeypot deployment in Big Data security: A practical approach.* Computers & Security, 102
* **Patel, S., & Sharma, P.** (2020). *Efficient AES encryption for Big Data security.* International Journal of Computer Applications, 175(10), 45-52.
* **Schneier, B.** (2007). *Applied cryptography: Protocols, algorithms, and source code in C* (2nd ed.). Wiley.
* **Spitzner, L.** (2003). *Honeypots: Tracking hackers.* Addison-Wesley.
* **Stallings, W.** (2017). *Cryptography and network security: Principles and practice* (7th ed.). Pearson.
* **Wang, C., Wang, Q., Ren, K., & Lou, W.** (2010). *Privacy-preserving public auditing for data storage security in cloud computing.* In Proceedings of the IEEE INFOCOM (pp. 1-9). IEEE.
* **Zhang, Y., & Chen, J.** (2016). *A survey on security and privacy in Big Data.* In 2016 IEEE International Conference on Big Data (pp. 2689-2695). IEEE.
* **Zhou, L., Varadharajan, V., & Hitchens, M.** (2018). *Achieving secure Big Data storage with encryption and access control.* Future Generation Computer Systems, 87, 214-224.