Approaches to Solve Big Data Security Issues and Comparative Study of Cryptographic Algorithms for Data Encryption

Vinit G. Savant
Department of computer engineering,
Pimpri Chinchwad College of Engineering
Pune, India.

***Abstract*— In this paper main big-data security issues are discussed. The challenges of security in big data environment can be categorized into authentication level, data level, network level, and generic issues. We also discussed approaches like data encryption, network encryption, logging, node maintenance and algorithms for encryption techniques.**

***Keywords*— Big data, Encryption, RSA algorithm, ECC algorithm, DES algorithm, AES algorithm, RPC, SSL.**

I. Introduction

Big data refers to collection of massive data with processing and data retrieval. As big data collects very important and sensitive data from social sites and from government and hence security issues have to be concerned. This collected data have to encrypt by using appropriate algorithms to secure the data.

Big data have three important properties like volume, velocity and variety[2].

A. Volume

As name indicates data is in large amount. Daily terabytes to zettabytes of data is collected from various resources.

B. Velocity

Now a day's social sites are favorably used. Data comes at very high speed and with high frequency from social sites just like Gmail, Facebook, Twitter and WhatsApp.

C. Variety

Data comes in the structured or in the unstructured form just like image, video, sounds etc.

II. Security Issues

Big data deals with storing the data, processing the data, retrieval of data. Many technologies are used for these purposes just like memory management, transaction management, virtualization and networking. Hence security issues of these technologies are also applicable for big data.

The four important security issues of big data are authentication level, data level, network level and generic issues [1].

A. Authentication level issues

There are many clusters and nodes present. Every node has a different priorities or rights. Nodes with administrative rights can access any data. But sometimes if any malicious node got administrative priority then it will steal or manipulate the critical user data.

For faster execution with parallel processing, many nodes join clusters. In case of no authentication any malicious node can disturb the cluster.

Logging plays an important role in big data. If logging is not provided then no activity is recorded which modify or deleted data. If new node joins the cluster then that will not be recognized because of logging absence. Sometimes users may also used malicious data if log is not provided.

B. Data level issues

In big data, data is very important part and also plays vital role. Data is nothing but some important and personal information about us by the government or social networking sites. Data level issues deals with data integrity and availability such as data protection and distributed data.

To improve efficiency, big data environments kike Hadoop store the data as it is without encryption. If hacker access the machines, then there is impossible to stop him.

In distributed data store, information is stored in many nodes with replicas for quick access. But if any replica or information from other node is deleted or manipulated by hacker then it will be difficult to recover that data.

C. Network level issues

There are many nodes present in clusters and computation or processing of data is done in these nodes. This processing of data can be done anywhere among the nodes in cluster. So it is difficult to find on which node data is processing. Because of this difficulty on which node security should be provided is going to be complicated.

Two or more nodes can be communicate with each other or share their data/resources through network. Many times RPC (Remote Procedure Call) is used for communicating via network. But RPC is not secure until and unless it is encrypted.

D. *General level issues*

In big data environment many technologies are used for processing the data also some traditional security tools for security purposes.

Traditional tools are developed over years ago. So these tools may not be performed well with new distributed form of big data.

As big data uses many technologies for data storing, data processing and data retrieval, there may be some complexities occur because of these various technologies.

III. Approaches to Solve Security Issues

As discussed above, big data have many security issues. But these issues can be solved using some approaches like data encryption, network encryption and logging.

A. *Data encryption*

This approach is for data level issues. Data encryption is nothing but convert data into secret message using encryption algorithms. There are many encryption algorithms like AES, RSA, DES, ECC algorithm. These algorithms use private keys to encrypt data. Encryption of data can be done at sender's side and data decryption is done at receiver's side. For decryption of data same algorithms are used which mentioned above. For decryption of encrypted data, same private keys can be used which are used during encryption.

If data is in encrypted form then hacker cannot be able to steal the data. If any how hacker steals the data then he is not able to retrieve the data.

So now we are going to discuss data encryption algorithms:

For encryption/decryption process, in modern days is considered of two types of algorithms viz., Symmetric key cryptography and Asymmetric key cryptography [3].

- Symmetric key cryptography:
  Symmetric-key algorithms are those algorithms that use the same key for both encryption and decryption. Examples of symmetric key algorithms are Data Encryption Standard (DES) and Advanced Encryption Standard (AES).

- Asymmetric key cryptography:
  Asymmetric-key algorithms are those algorithms that use different keys for encryption and decryption. Examples of asymmetric-key algorithm are Rivest-Shamir-Adleman (RSA) and Elliptic curve cryptography (ECC).

1. *RSA (Rivest-Shamir-Adleman) algorithm*

Suppose any individual A wants to receive message M secretly will use pair of integers {e,n} as his public key also this A use {d,n} as his private keys.

Another individual who wants to send message M secretly to A will use A's public key to encrypt a message and it will create cipher text C. Now only A can decrypt message M using his private keys. Where, cipher text C= $(M^e)*|n|$.

2. *ECC (Elliptic Curve Cryptography) algorithm*

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. Elliptic curves are also used in several integer factorization algorithms that have applications in cryptography. The primary benefit promised by ECC is a smaller key size, reducing storage and transmission requirements, i.e. that an elliptic curve group could provide the same level of security afforded by an RSA-based system with a large modulus and correspondingly larger key – e.g., a 256-bit ECC public key should provide comparable security to a 3072-bit RSA public key . For current cryptographic purposes, an *elliptic curve* is a plane curve over a finite field (rather than the real numbers) which consists of the points satisfying the equation, $y^2=x^3+ax+b$.

3. *DES (Data encryption standard) algorithm*

DES algorithm uses cipher key known as Feistel block cipher. DES expects two inputs - the plaintext to be encrypted and the secret key. The manner in which the plaintext is accepted, and the key arrangement used for encryption and decryption, both determine the type of cipher it is. DES is therefore a symmetric, 64 bit block cipher as it uses the same key for both encryption and decryption and only operates on 64 bit blocks of data at a time.

TABLE I. COMPARATIVE STUDY

Factors	DES	AES	RSA	ECC
Contributor	IBM 75	Rijman, Joan	Rivest, Shamir	Neil, Victor
Key length	56 bits	128, 198 and 256 bits	Based on no. of bits	135 bits
Block size	64 bits	128 bits	varies	Varies
Security rate	Not enough	Excellent	Good	Less
Execution time	Slow	More fast	Slowest	Fastest

4. *AES (Advanced Encryption Standard) algorithm*

AES is new cryptographic algorithm that can be used to protect electronic data. It uses 10, 12, or fourteen rounds. Depending on the number of rounds, the key size may be 128, 192, or 256 bits. AES operates on a 4×4 column-major order matrix of bytes, known as the state.

When encrypting data with a symmetric block cipher, which use block of n bits. With AES, n=128(AES-128, AES-192 and AES-256 all use 128-bit blocks).This means a limit of more than 250 millions of terabytes.When encrypting data with a symmetric block cipher, which uses block of n bits. With AES, n=128(AES-128, AES-192 and AES-256 all use 128-bit blocks). This means a limit of more than 250 millions of terabytes.

So, by observing above table it is clearly understand that AES cryptographic algorithm is best algorithm among the all cryptographic algorithms.

- Advantages of AES algorithm:

i. Extremely secure

One of the most widely used symmetric key encryption systems is the U.S. Government-designated Advanced Encryption Standard. When you use it with its most secure 256-bit key length, it would take about a billion years for a 10 petaflop computer to guess the key through a brute-force attack. Since, as of November 2012, the fastest computer in the world runs at 17 petaflops, 256-bit AES is essentially unbreakable.

ii. Relatively fast

One of the drawbacks to public key encryption systems is that they need relatively complicated mathematics to work, making them very computationally intensive. Encrypting and decrypting symmetric key data is relatively easy to do, giving you very good reading and writing performance. In fact, many solid state drives, which are typically extremely fast, use symmetric key encryption internally to store data and they are still faster than unencrypted traditional hard drives.

- Disadvantages of AES algorithm:

i. Sharing the key

The biggest problem with AES encryption is that you need to have a way to get the key to the party with whom you are sharing data. Encryption keys aren't simple strings of text like passwords. They are essentially blocks of gibberish. As such, you'll need to have a safe way to get the key to the other party. Of course, if you have a safe way to share the key, you probably don't need to be using encryption in the first place.

ii. More damage if compressed

When someone gets their hands on a symmetric key, they can decrypt everything encrypted with that key. When you're using AES encryption for two-way communications, this means that both sides of the conversation get compromised. With asymmetrical public-key encryption, someone that gets your private key can decrypt messages sent to you, but can't decrypt what you send to the other party, since that is encrypted with a different key pair.

B. *Network encryption*

This approach is for network level issues. As data or message is sent over the network for communication, network must be encrypted.

If network is encrypted with appropriate industry standards then hacker will not able to crack the network. So data is secure over the network. For communicating between two or more nodes, RPC (Remote procedure call) is used [4]. But RPC is not more secure until and unless it is encrypted with some techniques like SSL (Secure Socket Layer).

SSL is a standard security technology for establishing an encrypted link between a server and a client. Normally, data sent between browsers and web servers is sent in plain text-leaving you vulnerable to eavesdropping. If an attacker is able to intercept all data being sent between a browser and a web server they can see and use that information. To be able to create an SSL connection a web server requires an SSL Certificate. When you choose to activate SSL on your web server you will be prompted to complete a number of questions about the identity of your website and your company. Your web server then creates two cryptographic keys a Private Key and a Public Key. The Public Key does not need to be secret and is placed into a Certificate Signing Request (CSR) - a data file also containing your details. You should then submit the CSR. During the SSL Certificate application process, the Certification Authority will validate your details and issue an SSL Certificate containing your details and allowing you to use SSL. Your web server will match your issued SSL Certificate to your Private Key. Your web server will then be able to establish an encrypted link between the website and your customer's web browser.

C. *Logging*

This approach is for authentication level issues. Logging is very important to record the logs for maintaining the changes in data. So if we maintain the logs then any changes, manipulation, deletion of data is recorded. If every node have separate log then whatever activity it performs is maintained and malicious node can be detected easily.

IV. Conclusion

The paper has taken quick review of the approaches to solve big data security issues along with the basic properties of Big data. The later part of the paper has presented the data encryption cryptographic algorithms with the comparative study of these algorithms.

References

[1] Venkata Narasimha Inukollu , Sailaja Arsi and Srinivasa Rao Ravuri. “Security Issues Associted With big Data In Cloud Computing”.

[2] Elisa Bertino. “Big Data- Opportunities and Challenges”.

[3] Vanya Diwan, Shubhra Malhotra,Rachna Jain. “Cloud Security Solutions: Comparison Among Various Algorithms”, volume 4, issue 4, April 2014.

[4] Dave Marshall, May 1995. “Remote Procedure Call”. www.cs.cf.ac.uk/Dave/C/node33.ht