International Journal of Computer Applications Technology and Research
Volume 5– Issue 3, 159 - 164, 2016, ISSN:- 2319–8656

Solve Big Data Security Issues

Abhinandan Banik
IBM India Pvt. Ltd.

Samir Kumar Bandyopadhyay
Department of Computer Science and Engineering,
University of Calcutta
Kolkata, India

Abstract: The advent of Big Data has presented nee challenges in terms of Data Security. There is an increasing need of research in technologies that can handle the vast volume of Data and make it secure efficiently. Current Technologies for securing data are slow when applied to huge amounts of data. This paper discusses security aspect of Big Data.

Keywords: Big Data; Challenges; Security and Privacy

1. INTRODUCTION

Big data is a term for massive data sets having large, more varied and complex structure with the difficulties of storing, analyzing and visualizing for further processes or results. The concept of big data has been endemic within computer science since the earliest days of computing. "Big Data" originally meant the volume of data that could not be processed by traditional database methods and tools. Each time anew storage medium was invented, the amount of data an accessible exploded because it could be easily accessed.

Big Data is characterized by three aspects: (a) the data are numerous, (b) the data cannot be categorized into regular relational databases, and (c) data are generated, captured, and processed very quickly. Big Data is promising for business application and is rapidly increasing as a segment of the IT industry. The variety, velocity and volume of big data amplifies security management challenges that are addressed in traditional security management. Big data security challenges arise because of incremental differences, not fundamental ones. The differences between big data environments and traditional data environments include:

- The data collected, aggregated, and analyzed for big data analysis
- The infrastructure used to store and house big data
- The technologies applied to analyze structured and unstructured big data

The variety, velocity and volume of big data amplifies security management challenges that are addressed in traditional security management. Big data repositories will likely include information deposited by various sources across the enterprise. This variety of data makes secure access management a challenge. Each data source will likely have its own access restrictions and security policies, making it difficult to balance appropriate security for all data sources with the need to aggregate and extract meaning from the data.

Big data is an evolving term that describes any voluminous amount of structured, semi-structured and unstructured data that has the potential to be mined for information. The 3Vs that define Big Data are Variety, Velocity and Volume.

1) Volume: There has been an exponential growth in the volume of data that is being dealt with. Data is not just in the form of text data, but also in the form of videos, music and large image files. Data is now stored in terms of Terabytes and even Petabytes in different enterprises. With the growth of the database, we need to re-evaluate the architecture and applications built to handle the data.
2) Velocity: Data is streaming in at unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time. Reacting quickly enough to deal with data velocity is a challenge for most organizations.
3) Variety: Today, data comes in all types of formats. Structured, numeric data in traditional databases.

Information created from line-of-business applications. Unstructured text documents, email, video, audio, stock ticker data and financial transactions. We need to find ways of governing, merging and managing these diverse forms of data.

As computer applications were developed to handle financial and personal data, the need for security is necessary. The data on computers is an extremely important aspect for processing and transmitting secure data in various applications. Security is the process of preventing and detecting unauthorized use of computer or network.

Prevention measures help us to stop unauthorized users

from accessing any part of computer system. Detection helps to determine whether or not someone attempted to break into the system. The goal of cryptography is to make it possible for two people can exchange a message in such a way that other people cannot understand the message. There is no end to the number of ways this can be done, but here we will be concerned with methods of altering the text in such a way that the recipient can undo the alteration and discover the original text.

2. REVIEW WORKS

Big data is a large set of unstructured data even more than tera and peta bytes of data. Big Data[1] can be of only digital one. Data Analysis become more complicated because of their increased amount of unstructured or semi-structured data set. Predictions, analysis, requirements etc., are the main things that should be done using the unstructured big data. Big data is a combination of three v's those are namely Volume, Velocity and Variety. Big data will basically processed by the powerful computer. But due to some scalable properties of the computer, the processing gets limited.

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

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).

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.

3. PROPOSED METHOD

In a symmetric cryptosystem model we use the same key for encryption and decryption. In this proposed model our main key has two parts. The first part of the main key is substitution key or sub-key and the last part is shuffling key or suf-key. Here last 10bits of main key represents suf-key and apart from that last 10bits rest of the part represents sub-key. We have to remember that there is a restriction of choosing sub-key; it can only bean 8-bit or a multiple of 8 i.e.16, 24, 32 and so on. So, main key of our proposed Big data have many security issues and these issues can be solved using some approaches like data encryption.

Data Encryption Standards or DES is a block encryption algorithm. When 64-bit blocks of plain text go in, 64-bitblocks of cipher text come out. It is asymmetric algorithm, meaning the same key is used for encryption and decryption. It uses a 64-bitkey, 56 bits make up the true key, and 8 bits are used for parity. When the DES algorithm

is applied to data, it divides the message into blocks and operates on them one at a time. A block is made up of 64bits and is divided in half and each character is encrypted one at a time. The characters are put through 16 rounds of transposition and substitution functions. The order and type of transposition and substitution functions depend on the value of the key that is inputted into the algorithm. The result is a 64-bitblock of cipher text. There are several modes of operations when using block ciphers. Each mode specifies how a block cipher will operate. One mode may work better in one type of environment for specific functionality, whereas another mode may work in a different environment with totally different types of requirements.

The Advanced Encryption Standard or AES [6] algorithm is also a symmetric block cipher that can encrypt and decrypt information. This algorithm is capable of using cryptographic keys of 128,192and 256 bits to encrypt and decrypt data in blocks of 128 bits. The input and output for the AES algorithm each consist of sequences of 128bits(digits with values of 0 or 1).These sequences will sometimes be referred to as blocks and the number of bits they contain will be referred to as their length. The bits within such sequences will be numbered starting at zero and ending at one less than the sequence length (block length or key length).The number i attached to a bit is known as its index and will be in one of the ranges between 0 to127or 0 to255 or 0 to 255 depending on the block length and key length.

In this paper, we concentrate on developing an innovative cryptosystem for information security which is more advantages than existing symmetric key standards like DES and AES in context of security.

model will always be sub-key +10bits,i.e.18, 26, 34,42 and so on. Therefore this proposed model is based on substitution-expansion-shuffling technique.

In this method first determine the sub-key from the main key. On the basis of this sub-key, first step, method of substitution will occur. Let take an 8-bit suf-key ,i.e. 00111010.Firstcalculate the key value to perform the substitution. In case of 8-bit key this values can be

determine by calculating the decimal value of first three bits then next and so on. In our key 1st three bits i.e. 0,1and 2 positions are 001whose corresponding decimal value is 1.Consider first value. Then calculate next three bits i.e.1,2 and 3 three bit positions i.e. 011 is 3. Consider second value. Perform same technique until come to last bit positioni.e.7^{th} position .Now, what happen if we try to take next three bits from the 6th position where only 2-bits left or in case of 7th position where only 1-bitleft? In such case we take rest of the bits from the beginning e.g. for 6^{th} position it should be 100 and for 7^{th} position it is 000.In case of 8-bits sub-key we'll get eight key values lie in between 0 to7.Key values of our sub-key are1, 3, 7, 6,5,2,4 and 0.As soon as we'll get that key values we create a block of eight rows and 1+Ncolumn (N is number of characters in the file). Here first column represent the key values of the sub-key and remaining columns represent the characters of the file. Now, we convert each character to its corresponding 8-bit binary stream(as we take 8-bit sub-key)and put that into the block serially from top to bottom. After the whole block is filled up it will rearrange in ascending order from top to bottom on the basis of the key values and thus the bits of the whole plain text will have substituted and we will get the substituted binary bit stream. Though we have to determine that key values therefore the number of sub-key combinations we can generate by a particular sub-key length is$2^{(\text{Length of sub-key}/2)}$,i.e.for8-bit we can generate up to 2^4 combinations of sub-key,for16-bit we can generate 2^8combinations of sub key and so on. Now, key values are varies with the sub-keys, like in case of 16-bit key it lies between 0 to15, for24-bit key it lies between 0-23 and so on. Now we convert the characters to corresponding binary stream of that bit which our sub-key has, i.e. incase of16-bit key we convert the character to 16-bit binary stream, for24-bit key we convert to 24-bit binary stream and so on.

The next step after method of substitution is method of expansion. As the name suggested we can guess that there we will perform some sort of expansion. This expansion will perform on the substituted binary bit stream and it will also need the key values of the sub-key. Now, key value of our sub-key is1,3,7,6,5,2,4 and 0. At the beginning we take the first key value which is1. Now ,from the substituted binary bit stream we will take 1^{st} bit because our key value is1 and we also perform only one time expansion as our value is1.We expand up to 8 bit in case of 8-bit sub-key. As we have 1^{st} bit 1 so we add another 7 bits i.e.0101010.0101010 because last bit is1, thus we add alternate 0 and 1.We then move to next key value which is 3,therefore we perform up to three times expansion. But we perform expansion from the starting key value (1^{st} key value to 3^{rd} key value) i.e.1,3 and7. We follow this technique until the substituted binary stream found empty. As soon as we found it is empty,we will stop expansion and go to next step.We will expand up to8-bit to make expanded bit stream divisible by 8.It helps to convert binary bit stream to corresponding cipher text at the end. Apart from8-bit substitution key we follow quite different technique for expansion. The basic idea is same with only one change. For rest of the sub-keys, if key values are less than the half of the length of current sub-key we will expand it up to the length of previous sub-key and if key values are greater than the half of the length of current sub-key we will expand it up to the length of current sub-key. E.g. in case of 16 bit sub-key ifkeyvalueis6thenwewillexpandup to 8 bit, but if key value is 10 then we will expand up to 16 bit. Now, it can be a situation that our key value is 5 but only two bits are left in the substituted binary bit stream for expansion. In that case, though it can't be a major issue in time of expansion but it will leads to major problem at the time of decryption as we will get some redundant values. So, we need to keep that redundant information to get the actual message. We keep that information as redundant bit information at the end of expanded binary bit stream. In case of 8-bit sub-key we can keep that information by adding extra eight bits at the end.

Now, by 8-bit (2^8)we can represent a number between 0 to 255,so we can use extra 8-bit to keep redundant bit information up to 256bitssub-key as the key value lies between 0 to255. But we can't keep the information of the redundant bits if the sub-key size is greater than 256.So, solve that problem we will use8-bit if the sub-key is less than or equal to2^8,if sub-key is greater than 2^8then we will use 16-bit to keep that information until the length of sub-keyisgreater2^{16}andsoon.Thatiswewilluse8-bitorthe multipleof8-bitto keep the redundant bit information and this depends on the length of sub-key. So, the redundant bit information will determine by 2^{8n},(here n is1,2,3and soon) where, sub-key>$2^{8(n-1)}$ and sub-key<=2^{8n}.

As the binary expansion of substituted binary bit stream completes and redundant bit information adds we will get the actual expanded binary bit stream. The next technique, method of shuffling will start after that and it take expanded binary bit stream as input. This method has two scenarios. One what type of shuffling case it going to perform? And how many rounds this shuffling will occur? We need10-bitshuffling key to determine that as in our proposed model were strict shuffling cases in four types and shuffling rounds by 255.In this method, the four cases will occur and these are as follows: case1.Unchanged, case2.Swap,case3.Inverse and case4.Swap-Inverse.

*Case1.Unchanged:*In this case we will take expanded binary bit stream as final cipher binary bit stream and convert it to its corresponding cipher text.

Case2.*Swap:*In this case we will take expanded binary bit stream as initial cipher binary bit stream and then perform bit swapping on the basis of key value. Here, key value of our sub-key was 1,3,7,6,5,2,4 and 0.

Let the expanded bit length is24 and we perform 3 shuffling rounds. At the beginning we take the first key

valuewhichis1.Now,in the 1st round we divide the expanded binary bit stream into twoparts,thelefthalfwhichcontains1-bit(as our key valueis1)and the right half which contain rest 23 bits and then swap them. So, it forms a new intermediate cipher binary bit stream and we'll take it as input for next round.

Then, in the 2nd round we take the next key value which is

binary bit stream.

Case3.Inverse: This case is similar as previous. Here we inverse the expanded binary bit stream. If we take the previous conditions then in the 1st round we divide the expanded binary bit stream into two parts, the left half For the 2nd round we divide it into the left half which contains 3-bit (as our key value is 3) and the right half which contain 21b its and inverse those 21 bits. And at the end we divide it into the left half which contains 7-bit (as our key value is 7) and the right half which contain17 bits and inverse those 17bits.After 3rd round we'll get our final cipher binary bit stream.

Case4.Swap-Inverse: This case is nothing but combination of previous methods i.e. Case2 and Case3. Here in each round we perform first the swap and then inverse operation to form the intermediate cipher binary bit stream.

The algorithm for encryption and decryption process in presented next.

Algorithm1 Encryption

Input: input file to be encrypted

K=key

Output: encrypted file

Begin

Step1.Divide the key into three parts such as substitution key, shuffling case and shuffling rounds.

Step2.Take the all character of plain text of the file into a string and generates corresponding binary bit steam on the basis of the size of substitution key

Step3.Substitute input bit stream using substitution key and calculate the key value of that substitution key.

3, again divide it into two parts, the lefthalfwhichcontains3-bit(as our key value is 3) and the right half which contains rest 21 bits and then swap them. And in the last round our key value will be 7 and we'll divide the left half which contains7-bit(as our key value is 7 now) and the right half which contains 17 bits and swap them. So, after 3rd round we'll get our final cipher

which contains1-bit(as our key value is 1)and the right half which containrest23bitsas previous. Then we just inverse the rest of 23 bits and form intermediate cipher binary bit stream.

Step4.Expand substituted binary bit stream on the basis of key value and add redundant bit information in the end of that bit stream.

Step5. Shuffle expanded binary bit stream using shuffling case, shuffling round sand key value information and generates final cipher bit stream.

Go to Step5 until expanded bit stream found null.

Step6. Convert the final cipher bit stream to corresponding cipher text and write it into the encrypted file.

End

Algorithm2 Decryption

Input: input file to be decrypted

K=key

Output: decrypted file

Begin

Step1.Dividethekey into three parts such as substitution key, shuffling case and shuffling rounds, also determine the key value of substitution key.

Step2.Take the all character of cipher text of the file into a string and generates corresponding binary bit steam on the basis of the size of substitution key.

Step3.Shuffle initial cipher bit stream using shuffling case, shuffling rounds and key value information and

generates expanded plain text bit stream.

Go to Step3 until cipher bit stream found null.

Step4. Keep the information of redundant bits into an array and compress expanded plain text bit stream into initial plain text bit stream

Step5. Reconstitute initial plain text bit stream using substitution key and generates corresponding original plain text binary bit stream.

Step6. Convert original plain text binary bit stream to corresponding cipher text and write it into the decrypted file.

End

The length of expanded bit stream is depending of the sub-key, therefore this expanded binary bit stream is varies with the sub-key and it also possible that two different sub-key producing same size of expanded binary bit stream. This approach confuses the unauthorized users about the size of the key. Another important technique we inherit in our model is shuffling cases and shuffling rounds. Here, as we stated, we use four different cases and up to255 rounds which will determine only on the basis of key. Now the major advantage of this mechanism is we can produce various different cipher texts from same expanded binary bit stream by varying different the cases or the rounds.

4. CONCLUSIONS

The paper has taken review of the big data security issues along with the basic properties of Big Data. It shows successful achievement of the encryption and decryption of the given text files. The later part of the paper has presented the encryption and decryption algorithms to demonstrate the security issues of Big Data. These encryption techniques make data more secure. It was observed that size of text file varies to count the effect of volume due to big data.

5. REFERENCES

[1] S.Sen, C.Shaw, R.Chowdhuri, N. Ganguly, and P. Chaudhuri, “Cellular Automata Based Cryptosystem(CAC)”, Fourth International Conference on Information and Communication Security (ICICS02), Dec. 2002, PP. 303-314.

[2] Feng. Bao, ”Cryptanalysis of a Partially Known Cellular Automata Cryptosystem”, VOL.53, No.11, Nov. 2004.

[3] Biham, Eli and Adi Shamir, Differential Cryptanalysis of the Data Encryption Standard, Springer Verlag, 1993.

[4] Coppersmith, D. "The Data Encryption Standard(DES) and Its Strength Against Attacks." IBM Journal of Research and Development, May 1994,pp. 243 - 250.

[5] K. Naik, D. S.L. Wei, Software Implementation Strategies for Power-Conscious Systems,” Mobile Networks and Applications -6, 291-305, 2001.

[6] Farina, A., "Simultaneous Measurement of Impulse Response and Distortion with a Swept-Sine Technique", presented at the AES 108th Convention, Paris, France, 2002 February 19 -22.

[7] N. Tippenhauer, C. Pöpper, K. Rasmussen, S. Capkun, “On the requirements for successful GPS spoofing attacks,” in Proceedings of the 18th ACM conference on Computer and communications security, pp. 75-86, Chicago, IL, 2011.

[8] P. Gilbert, J. Jung, K. Lee, H. Qin, D. Sharkey, A. Sheth, L. P. Cox, “YouProve: Authenticity and Fidelity in Mobile Sensing,” ACM SenSys 2011, Seattle, WA, November, 2011.

[9] B. Levine, C. Shields, N. Margolin, “A Survey of Solutions to the Sybil Attack,” Tech report 2006-052, University of Massachusetts Amherst, Amherst, MA, October 2006.

[10] B. Agreiter, M. Hafner, and R. Breu, “A Fair Non-repudiation Service in a Web Service Peer-to-Peer

Environment," Computer Standards & Interfaces, vol 30, no 6, pp. 372-378, August 2008.

[11] A. Anagnostopoulos, M. T. Goodrich, and R. Tamassia, "Persistent Authenticated Dictionaries and Their Applications," in Proceedings of the 4th International Conference on Information Security, pp. 379-393, October, 2001.