ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 2, Issue 12, December 2017

Survey on Data Stream Management and Stream Security Models

[1] K.Akshatha, [2] Mr.Mahavishnu, [3] Dr.B. Persis Urbana Ivy, [4] Dr.Sujatha.k
[1] II.M.E.PG Scholar, [2] Assistant Professor, [3] Professor &Head, [4]Associate Professor
[1][2][3][4] Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology
Kuniamuthur, Coimbatore

***Abstract:*-- The advancement in telecommunication and electronics results in enormous amount of data. The data or information collected from applications like traffic monitoring, weather monitoring, social networks (Facebook, twitter, etc...),web based retail applications (Amazon, Flipkart, etc..) are continuous data streams which cannot be handled by traditional data management systems. Data Stream Managers (DSM) are used to handle the data's with high volume and velocity ,also known as Big Data .DSM helps to analyze the data streams to obtain useful insight from it. Streaming datas are continuous in nature and it has to processed consecutively ensure the freshness of the data. The data transmitted via a network also consists of some sensitive information like patients health records, banking transactions,etc.. Since, the analytics are used for decision making the data's transmitted has to be preserved from internal and external malicious attacks .In this paper, various data stream management and stream security models are discussed along with the symmetric cipher algorithms to implement the security models.**

Keywords – Data stream, data confidentiality, Data Encryption, Symmetric Encryption

1. INTRODUCTION

The rapid growth in technology and communication produces large volume of data in all fields like banking, HealthCare, mobile applications, websites,etc...The efficient management of these datas helps in smart decision making. The Streaming datas are continuous in nature and hence it cannot be handled by traditional database systems.DSM is used to handle datas with high velocity and volume. Data stream Processing involves various steps like 1.filtering, 2. Data cleaning /integration, 3. query execution/Analysis 4. Data Interpretation/ Decision making.[21]. It is impossible to store and process huge volume of data's (batch processing) instead, Stream Processing Engines (SPE) for example: Apache S4, Apache Spark, Nepture, etc... are used to achieve real time data analytics in a short time. In SPE it is not necessary to store the large volume of data and SPE is capable of real time computation.. The Stream Processing Engines helps in obtaining 1. high Throughput,2. high scalability,3. low latency. Since, the data's collected from variety of sources are used for further processing and analysis it is necessary to ensure the confidentiality, authenticity and integrity of the data. It is important to preserve the freshness of the data to avoid improper use of the data ,that is the data has to be preserved from any external and internal attacks.

Cryptography is the most common and effective way of storing and transmitting confidential data. The cryptographic model uses two different types of encryption standards,
1. Symmetric key Encryption
2. Asymmetric key Encryption

In Symmetric key encryption the sender and receiver uses the same key to encrypt and decrypt the data, whereas in Asymmetric key encryption the sender and receiver uses different keys [public and private key] to encrypt and decrypt the data. The Symmetric key encryption is more secure than Asymmetric key encryption (i.e.)128 bit of symmetric key is equal to 3000 bit of asymmetric key[22] [23]. So, Symmetric key encryption algorithms (3DES, RC4, Twofish, Rijindael, etc...) are preferred to secure big data streams. If the encryption keys are properly managed , cryptographic encryption methods are the most efficient way to transmit the data.

In this paper we study about the different types of data stream security modes and the algorithms used to implement those security models .The organisation of the paper is as follows Section 2 contains the overview of related works. Section 3 consists of a brief description about the methodologies involved in implementing the existing security models. Section 4 concludes the paper by providing a quick glance on the benefits of securing big data streams.

2. RELATED WORKS

The real time data processing is highly challengeable [20] when compared to Batch Processing. In stream processing the input data has to be processed consecutively as they arrive at the system .The data's transmitted via the network is always vulnerable to malicious attacks. To protect the streaming data R.V.Nehme et al [21] proposed a StreamShield also known a stream centric approach towards security and privacy in data stream environments in 2009.

There have been several works on securing the confidentiality, integrity of the data[4][5][5][6][12][14] and also to provide end to end data security. This section is briefed on the literature of data stream processing, streaming data security, and also presents overview of existing techniques to preserve the data confidentiality and integrity.

2.1 Data Stream Management:

Data stream is a continuous flow of data at high velocity from one or more sources. Streaming data integrates wide range of information. For example: consider a Email account holder, the activity log of the account holder is updated every few seconds (i.e.) information collected from mobile applications, data transactions, geographic locations, search history, etc... All these data's collected has to be filtered, aggregated and processed with continuous queries consecutively to obtain the desired output. Results obtained from these stream processing operations helps to identify efficient way of doing business (i.e.) helps in smart decision making.

The real time Data streams processing through traditional database management systems(DBMS) is very challenging due to the high velocity and volume of data. Stream Processing Engines are used to process the data streams consecutively as soon as they enter the system. STandard stream data Manager (STREAM) also know as Data Stream Management System was proposed by Arsu et al.[2][22]. In 2003. According to STREAM, queries that are given by the same client share a common seq – window Administrators. Other stream processing engines include Aurora[33] and Borealis[34]. Aurora aggregate the query execution with message processing. The sources of data streams are not persistent ,so there is always a high risk of late arrival of packets and data loss. Borealis [23] is the improvised version of Aurora[33] used to handle the misplaced or data packets with high latency .SPE's focuses on improvising the performance of query processing, but not much in preserving the confidentiality and integrity of the data stream.

2.2 Data Stream Security

Data's thus collected and processed by SPE's are mainly used for decision making purposes. All the data's are dependent to one other so, it is important to use the data from a trusted source for decision making process. To ensure the authenticity, integrity and trustworthiness of the data, the data has to be preserved from reaching wrong people. Encryption standards are use to preserve the integrity of the data stream. There are several works that focuses on preserving the data streams from any malicious attacks [21][24][25][26][27][28] [29].

2.3 Access Control over Data Streams

A framework to enforce access control over the streaming data is proposed by R.V Nehme et al [22]. The queries written in descriptive languages like StreamSQL, CCL in Coral8 has to be written according to the access control of the streaming data, but the lack of standard query language does not provide the necessary output. So, the query rewriting policies are changed in a manner it returns the query processing results according to the access control of the data stream. Even though it enforces access control over the data streams , it does not prevent the illegitimate data entering into the data stream. To prevent the illegitimate data , it is necessary to verify the authenticity and trustworthiness of the data entering into the stream. Several works [31][10] have been proposed to verify the authenticity and integrity of the data .

2.4 Location Aware End to End Security

In 2008, Kui Ren Et al proposed a integrated security design to establish a location aware end-to-end data security(LEDS) framework [23] .In LEDS , the selected geographic area is divided into cells using virtual geographic grid. The geographic location of the cell is bounded with the symmetric key of each node, so the unauthorised nodes are at high risk of identification by the legitimate nodes. LEDS follow an /interleaves Hop-by-Hop authentication[14] and Statistical en-route filtering[15] to remove the false data from the network. LEDS is comprised of two major module 1.key management framework 2.End to End Data Security. LEDS overcome the threats like Denial of Service (DoS), Selective Forwarding attack[4] [5] [17].

LEDS assure End-to-End security (i.e.) no-to-node authentication and node-to-sink authentication and the availability of the data. The targeted geographic area is virtually divided into cells ,each wireless sensor nodes identify the location through a suitable positioning scheme[9] [12] [13] [22]. By identifying the geographic location of the nodes the action of it can be restricted to a specific geographic location, thereby reducing the effect of the compromised nodes. Every single data shared using LEDS framework needs approval from all the nodes in the network. The authentication status of the nodes can be verified individually using sink. Since, all the processes are bound to geographic locations the illegitimate data transmission of the compromised nodes can be easily eliminated by integrating the authentication status of the nodes. The datas transmitted are highly available in spite of the selective forwarding [4] and disruption attack [13]. Although it provides end-to-end data security, the data aggregation between the intermediate nodes is complex [1].

ISSN (Online) 2456 -1304

International Journal of Science, Engineering and Management (IJSEM)
Vol 2, Issue 12, December 2017

2.5 Selective Encryption:

In 2017, Deepak Puthal et al. [1] proposed A Selective Encryption Method to Ensure Confidentiality for Big Sensing Data Streams[SEEN]. SEEN is proposed to secure the data based on its sensitivity level. The sensitivity of the data is not just determined based on its applications, it is the type of data which helps in determining its sensitivity. Not all applications in a data is sensitive so, it is not necessary to apply same level of security to all data types. Instead ,the security level can be determined based on the data type so that we can save the time spent to secure the data's which are less sensitive. In general, data security can be categorized into three different types.

1. Highly secured
2. Partially Secured
3. Not secured

For Example: In Health Care monitoring, the data has to be highly secured [5][15] [20][36]to preserve the patient's personal health records. Whereas in military applications it is not necessary to apply high security for all collected data's , it is enough to implement partial security to less sensitive data's; high security can be applied for the data's like border security patrol, weapon details, etc...When unauthorised disclosure or alteration or destruction of a data is not a risk to a particular organization or individual, that data can be classified as not secured[4][6][8] (i.e.)it is not necessary to secure that data ,it can be made publically available.

Data's are encrypted using Symmetric key Encryption. It handles large volume of continuous data streams more effectively and provides end to end security mechanism. According to SEEN the rekeying process is done more effectively without any retransmissions. SEEN reduces the processing time due to flexible encryption, it also increases the transmission rate. Symmetric key Encryption algorithm Rijndael is used to encrypt the data stream in order to avoid the intrusion of illegitimate data into the data stream.

3. METHODOLOGIES TO ENCRYPT THE STREAMING DATA

Symmetric cryptographic cryptography is preferred over asymmetric key cryptography[19] as it is more efficient than the other. The most commonly used strong encryption algorithms DES,RC2,Rijndael are analysed in detail .In 2000, J.Bruke et al proposed architectural support to improvise the performance of the symmetric key algorithms.

Plaintext: the actual data which is given as input to the system.

Cipher text: The scrambled form of data created by processing the plaintext based on the secret key.

3.1 DES

Data Encryption Standard(DES) was first adopted by US government in the year 1977 .It is implemented in a wide variety of embedded systems like sim cards,credit cards,modems,etc..DES is applied to a block of data simultaneously , not for a single bit. DES uses 56 bit key to encrypt the data . Due to the small key length DES is no longer recommended to encrypt the sensitive data. National Institute of Standards and Technology (NIST) choose Triple - DES [33] to replace DES . 3-DES performs the performance of DES thrice, different encryption key is used for every single iteration to increase the key length to 168 bits. Even though it is secure it will still replaced by the algorithms which are ought to be faster and secure.

3.2 RC2

RC2 was developed by Ron Rivest in 1987 , the specifications of the algorithms are not revealed until 1996. RC2 is reasonably strong and it uses 64 bits to encrypt the data[34] , it is weak during the usage of certain keys. The input data is first divided as 8 bytes and then each block with 8 bytes is processed individually. Each and every block is considered as four words, each word is comprised as 16 bits . The array of four words is presented as R[0] R[1] R[2] R[3]. Both encryption and decryption consider these arrays as input and encrypt/decrypt the content of the arrays. The output is returned via the array in which the input is given.

3.3 Rijndael

NIST established Rijndael in 2001. Rijndael is very fast and compact cipher that supports key sizes of 128 bits or 192 bits and 256 bits long[35]. Rijndael is considered to be the best algorithm to secure the real time data and it can be further enhanced by improving the number of rounds. Each round is compraised of 4 different steps

Key_Size	No. of Round
128 bits	9
192 bits	11
256 bits	13

- SubBytes
- ShiftRows
- MixColumns
- Add Round key

CONCLUSION

As the technology grows , the applications associated with it generates a huge volume of data . In this paper,we overviewed the existing data stream management techniques and the security models used to preserve the confidentiality and integrity of the streaming data. A brief description of the strong symmetric key algorithms used to implement the security models are given

REFERENCES

[1] Deepak Puthal, Xindong Wu, Surya Nepal, Rajiv Ranjan, Jinjun Chen ,"SEEN: A Selective Encryption Method to Ensure Confidentiality for Big Sensing Data Stream."In IEEE Transactions on Big Data, pp. 2332-7790. IEEE 2017.

[2] A. Arasu, et al. "STREAM: the stanford stream data manager (demonstration description)." In ACM SIGMOD international conference on Management of data, pp. 665-665, ACM, 2003.

[3] H-S. Lim, Y-S. Moon and E. Bertino, "Provenance-based trustworthiness assessment in sensor networks." In Seventh nternational Workshop on Data Management for Sensor Networks, pp. 2-7. ACM, 2010.

[4] S. Sultana, G. Ghinita, E. Bertino and M. Shehab, "A lightweight secure provenance scheme for wireless sensor networks." In 18th International Conference on Parallel and Distributed Systems (ICPADS), pp. 101-108, 2012.

[5] R. A. Shaikh, S. Lee, M. AU Khan and Y. J. Song, "LSec: lightweight security protocol for distributed wireless sensor network." In IFIP International Conference on Personal Wireless Communications, pp. 367-377. Springer Berlin Heidelberg, 2006.

[6] G. Selimis, at al. "Evaluation of 90 nm 6 T-SRAM as Physical Unclonable Function for Secure Key Generation in Wireless Sensor Nodes", in IEEE ISCAS Brazil, pp. 567-570, 2011.

[7] G. Selimis, at al. "Evaluation of 90 nm 6 T-SRAM as Physical Unclonable Function for Secure Key Generation in Wireless Sensor Nodes", in IEEE ISCAS Brazil, pp. 567-570, 2011.

[8] M. Roesch,"Snort: Lightweight Intrusion Detection for Networks." LISA, vol. 99, no. 1, pp. 229-238. 1999.

[9] A. S. Wander, N. Gura, H. Eberle, V. Gupta and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks." In Third IEEE international conference on pervasive computing and communications, pp. 324-328. IEEE, 2005.

[10] T. Park and K. G. Shin, "LiSP: A lightweight security protocol for wireless sensor networks." ACM Transactions on Embedded Computing Systems (TECS), vol. 3, no. 3, pp. 634-660, 2004.

[11] T. A. Zia and A. Y. Zomaya, "A Lightweight Security Framework for Wireless Sensor Networks." JoWUA, vol. 2, no. 3, pp. 53-73, 2011.

[12] P. Ferreira and P. Alves, Distributed context-aware systems. Springer, 2014. DOI 10.1007/978-3-319-04882-6

[13] D. Ganesan, D. Estrin and J. Heidemann, "DIMENSIONS: Why do we need a new data handling architecture for sensor networks?." ACM SIGCOMM Computer Communication Review, vol. 33, no. 1. pp. 143-148, 2003

[14] D. Puthal, S. Nepal, R. Ranjan and J. Chen, "A Secure Big Data Stream Analytics Framework for Disaster Management on the Cloud." In 18th International Conference on High Performance Computing and Communications, pp. 1218-1225. 2016.

[15] D. Puthal, S. Nepal, R. Ranjan and J. Chen, "A dynamic prime number based efficient security mechanism for big sensing data streams." Journal of Computer and System Sciences vol. 83, no. 1, pp. 22-24, 2017.

[16] D. Puthal, S. Nepal, R. Ranjan and J. Chen, "DLSeF: A Dynamic Key Length based Efficient Real-Time Security Verification Model for Big Data Stream." ACM Transactions on Embedded Computing Systems, vol. 16, no. 2, pp. 51, 2017.

[17] R. Ranjan, "Streaming big data processing in datacenter clouds." IEEE Cloud Computing, vol. 1, no. 1, pp. 78-83, 2014.

[18] www.cloudflare.com (accessed on: 04.08.2014)

[19] J. Burke, J. McDonald and T. Austin, "Architectural support for fast symmetric-key cryptography." ACM SIGARCH Computer Architecture News, vol. 28, no. 5, pp. 178-189, 2000.

[20] M. Stonebraker, U. Çetintemel and S. Zdonik, "The 8 requirements of real-time stream processing." ACM SIGMOD Record, vol. 34, no. 4, pp. 42-47, 2005.

[21] R. V. Nehme, H-S. Lim, E. Bertino and E. A. Rundensteiner, "StreamShield: a stream-centric approach towards security and privacy in data stream environments." In ACM SIGMOD International Conference on Management of data, pp. 1027-1030.

[22] A. Arasu, et al. "Stream: The stanford data stream management system." Technical Report 2004-20, Stanford InfoLab, 2004.

[23] D. J. Abadi et al., "The Design of the Borealis Stream Processing Engine." CIDR, vol. 5, pp. 277-289, 2005.

[24] R. Adaikkalavan and T. Perez, "Secure shared continuous query processing." In ACM Symposium on Applied Computing, pp. 1000-1005. ACM, 2011.

[27] R. Adaikkalavan, X. Xie and I. Ray, "Multilevel secure stream processing: Architecture and implementation." Journal of Computer Security, vol. 20, no. 5, pp. 547-581, 2012.

[28] J. Cao, B. Carminati, E. Ferrari and K.-L. Tan, "Acstream: Enforcing access control over data streams." In IEEE 25th International Conference on Data Engineering, pp. 1495-1498, 2009.

[29] W. Lindner and J. Meier, "Securing the borealis data stream engine." In 10th International Database Engineering and Applications Symposium (IDEAS'06), pp. 137-147. IEEE, 2006.

[30] X. Xie, I. Ray, R. Adaikkalavan and R. Gamble, "Information flow control for stream processing in clouds." In 18th ACM symposium on Access control models and technologies, pp. 89-100. ACM, 2013.

[31] R. V. Nehme, E. A. Rundensteiner and E. Bertino, "A security punctuation framework for enforcing access control on streaming data." In IEEE 24th ICDE, pp. 406-415, 2008.

[32] K. Ren, W. Lou and Y. Zhang, "LEDS: Providing location-aware end-to-end data security in wireless sensor networks." IEEE Transactions on Mobile Computing, vol. 7, no. 5, pp. 585-598, 2008.

[33]Lars R. Knudsen, Vincent Rijmen, Ronald L. Rivest, Matthew J. B. Robshaw: On the Design and Security of RC2. Fast Software Encryption 1998: 206–221

[34] Christof Paar, Jan Pelzl, "The Data Encryption Standard (DES) and Alternatives" free online lectures on Chapter 3 of "Understanding Cryptography, A Textbook for Students and Practitioners". Springer, 2009.

[35] Daemen, Joan; Rijmen, Vincent (March 9, 2003). "AES Proposal: Rijndael"(PDF). National Institute of Standards and Technology. p. 1. Retrieved 21 February2013.

[36] D. Puthal, S. Nepal, R. Ranjan and J. Chen, "A Synchronized Shared Key Generation Method for Maintaining End-to-End Security of Big Data Streams." In 50th Hawaii International Conference on System Sciences, pp. 6011-6020, 2017.