Efficient Security Model for RDF Files Used in IoT Applications

Mohamed El kholy[1]
Computer Engineering Department
Pharos University in Alexandria
Alexandria, Egypt

Abdel baes Mohamed[2]
Computer Engineering Department
AASTMT
Alexandria, Egypt

***Abstract*—The openness environment of IoT ecosystem arises several security and privacy issues. However, the huge amount of data produced by several IoT devices restricts using traditional security methods. Another security challenge for IoT system is the interoperability between heterogeneous IoT devices. Semantic Web has risen as a promising technology that provides semantic annotations allowing interoperability between IoT devices. Semantic web uses RDF triples to allow semantic data exchange between heterogeneous applications. Hence, RDF files used in IoT systems require specific security mechanism that regards large data size as well as rapidly data updates. The proposed work introduces a security novel that provides RDF files with a fine grained partial encryption. The proposed method allows applying security for the sensitive parts of RDF files without affecting the public parts. Encryption metadata is stored in a container related to each individual sensitive triple. Thus accessing public data in RDF file is not affected with the encryption overheads. A motivation scenario for privacy in a smart city is used to evaluate the proposed method. Experimental results showed that the proposed methodology enhances the access time of RDF triples from 10.4 msec to 6.2 msec. Moreover the proposed method facilitates integration of separated parts of a RDF graph together. The empirical evaluation proved the enhancement in efficiency and flexibility by applying the proposed method to RDF files used in IoT systems. Moreover the insensitive triples in RDF files are not affected with the security overheads.**

Keywords—Semantic Web; Internet of Things (IoT); resource description framework (RDF); smart cities; security mechanism; web ontology language (OWL); partial encryption; SPARQL protocol and RDF query language (SPARQL); data encryption standard (DES) component

I. Introduction

IoT ecosystem connects several heterogeneous devices scattered all over the world [1]. IoT system relies on a set of sensors and actuators. Sensors are responsible of collecting data from the surrounding environment [2]. While actuators act on performing different actions for controlling devices [3]. Sensors and actuators are connected to the Internet by several heterogeneous protocols [1]. Each IoT device has its own hardware manufacture and can transmit and receive data with a specific defined format [4]. Interoperability between these heterogeneous devices at the hardware level or at the physical network layer is strongly complex [2, 3]. Moreover, different IoT devices use different protocols for data transfer [2]. The traditional protocols for ether net and Wi-Fi are used as well as other protocols that maintain power saving such as zegbee and blue tooth [3]. Thus, interoperability between different IoT devices is considered one of the significant challenges for the success of IoT ecosystem.

On the other hand Semantic Web modified the Internet contents from documents for humans to read towards information for machines to manipulate [5]. Semantic web uses RDF triples as well as defined Web Ontology Language (OWL) to provide defined meaning to data [6]. RDF triples consists of subject, predicate, and object [5]. The subject identifies anything wanted to be described, while the predicate identifies the property or the attribute of description. The object is the value of the identified property. Representing data in RDF style allows different machines to get useful information about the status of the described subjects [7]. Thus heterogeneous machines can exchange data and also understand the meaning of such data [8]. Hence, Semantic RDF triples can provide significant benefit to IoT system [6]. RDF annotations allows IoT ecosystem to structure and enrich data coming from different IoT devices. Converting signals collected from IoT devices to RDF data allows applying semantic calculations on these signals [9]. Hence, using RDF for representing data provides IoT system with the required interoperability between different heterogeneous devices.

IoT system allows monitoring the surrounding environment and performing intelligent actions on behalf of human [10, 11]. Thus, IoT systems are widely used in smart cities. Such automated environment arise significant challenges of privacy and security. Several devices in smart homes submit different types of data. Among these data there exist private data that should not be available to public users [12]. Supplying IoT data with security and privacy is a key challenge for the success of smart cities. On the other hand, transferring IoT data to RDF triples increases the size of data [13]. Hence, traditional approaches of encryption and decryption are not suitable for RDF data used in IoT system. Traditional encryption and decryption techniques need high computational power resulting in high latency time [14]. Hence, such techniques are not suitable for IoT systems that are characterized with rapid data updates and the need to take quickly decisions [15].

The proposed work contributes in providing RDF files with a technique that allows partial fine grained encryption for RDF triples. The proposed technique links the encryption metadata to the encrypted triple directly without any overheads to the main RDF file. Thus, unlike the traditional encryption containers, the insensitive data is not affected by the encryption overheads. The proposed technique has two significant enhancements; the first is shorter access time to insensitive data to RDF files even if it contains another encrypted data. The second is enabling to integrate the encrypted triples to another RDF file directly without processing the complex metadata in the RDF header. At the sender side the sensitive triples are encrypted and the encryption metadata is linked to each encrypted triple individually. At the recipient side the encrypted triples are decrypted using the schema send with each individual triple. Hence, a public user could access the public part of the encrypted RDF file without any encryption overheads. While the sensitive parts is restricted to authorized users only who can decrypt the sensitive triples. Hence, RDF triples are provided with the required security and privacy constrains while maintaining performance and efficiency aspects. The proposed work is limited to IoT sub systems that communicate by sending RDF files.

The remaining of the paper is organized as follows; a literature review is presented in section two. Section three introduces the problem definition. The proposed solution is discussed in section four. Finally the proposed model is evaluated in section five.

II. Literature Review

A significant number of published works discuss providing robust security for IoT systems without regarding the drawbacks of increasing data size and data access time. Other researches focus on securing only defined scenarios of using IoT systems. Securing RDF stores is also an attractive area for researchers. Several mechanisms for data encryption and access control are defined to provide security for semantic data. However, a little amount of work discusses securing semantic data associated with IoT systems. This literature review discusses the work done to secure RDF data in the spirit of openness and heterogeneous environment of IoT ecosystem.

Fern´andez et al. [16] defines a fine grained security for RDF triples. Their mechanism of encryption depends on the triples rather than a dedicated mediator. Their work combined symmetric and asymmetric encryption to reach high efficient security for RDF triples. They applied functional encryption to RDF data. The functional encryption allows the encrypted RDF triples to self-enforce its access restrictions. The Authors defined an encryption function derived from the RDF graph and randomly generated seeds. This function is used to construct a triple encryption vector for each RDF triple. Their encryption technique provides high security, however it is inefficient for large size of data. Moreover encrypting triples in such complicated technique makes security recovery challenging in case of different errors. Thus such security technique is not suitable for IoT environment.

Prajit Kumar Das et al. [17] designed a security framework that regard different security policies for data transfer between IoT devices. Authors represent security polices in semantically annotated statements. The framework defines different polices for access control depending on user attribute and the context of IoT devices usage. Such polices are represented semantically using OWL to allow different computer machines to deal with it. Access to data associated with an IoT resource depends on the context of requesting this data. The context includes predefined relationship between user, and the RDF triple (subject, predicate and object). Thus access control is granted by particular permission for a specific user to use specific RDF at a specific situation. The framework shows high complexity and lacks flexibility needed for IoT open environment. The context of using IoT devices is a subject of continuous change. Thus defining access control according to the context will decrease the efficiency of using IoT data. Moreover it is complicated to include all scenarios of using IoT devices.

Pedro Gonzalez-Gil et al. introduces data-security ontology for IoT [18]. Authors represent a common vocabulary describing the practical security aspects related to data access that is relevant to producers and consumers. They defined two main classes one for secure data and the other for access control. The secure data is divided into hidden data and encrypted data, while the access control defines the authority for each party to access the secure data. Their work integrates security metadata such as access control and data protection to the traditional semantic data annotation. Then they used triples to describe the security aspects of different parts of data. Their work focused on defining the security requirements rather than applying theses security to semantic data. Moreover, such mechanism is complicated to be implemented in a rapidly data changing environment of IoT ecosystem.

Another significant contribution for semantic security was done by Guangquan Xu et al. [19]. They defined a set of different Ontologies to describe security. Their work used Ontology to describe context and other Ontology to describe network attack as well as Ontology for system vulnerability. The network attack Ontology allows detecting complex attacks using a set of inference rules. While the vulnerabilities Ontology is responsible to detect elements exposed to danger to warm about this danger and its possible attacks. However, using different Ontologies increases the system complexity and increases the size of semantic data.

III. Problems Definition and Motivation Example

A. *Problems Associated with Securing RDF Data used in IoT Systems*

The proposed work contributes in filling the knowledge gap for the methodologies of securing RDF stores while maintaining its openness and semantic features. Traditional security approaches are not suitable to provide RDF triples with the required security and data privacy while maintaining openness features. RDF stores provide IoT system with semantic meaning that allows linking data from different IoT devices [20, 21]. In such an open and heterogeneous environment that lacks human monitoring security and privacy requirements increase significantly [22]. Methods for specifying the role of each agent to access or to use specific pieces of data are considered a significant challenge.

1) First problem: RDF stores lack a trustworthy infrastructure for specifying access control. Such problem is a reflect of the openness environment of semantic RDF triples. Another challenge will appear even a robust access control mechanism is applied to RDF stores. RDF data is transferred in non-secure channels so it is not safe from different sniffing attacks [23]. Thus using access control to secure RDF stores cannot provide the required security and privacy for IoT system.

2) Second problem: Traditional RDF encryption solutions are not suitable to the openness and rapidly changing environment of IoT system. IoT includes billions of sensors that transmit huge volume of data which is frequently changed [24]. Encryption and decryption of such a huge amount of data consume high computational power and significantly affect performance. Moreover, such approach affects real time applications that depend on the speed of reasoning IoT data. RDF partial encryption is used to solve such a problem but still include significant drawbacks. A significant number of RDF files is serialized in XML files to allow interaction between heterogeneous machines [25]. RDF partial encryption selects the sensitive data and encrypts it, and stores the encryption metadata in XML file header. Thus increase the size of RDF files when represented in XML files and needs more time to process data in the file headers. Moreover IoT environment is characterized with rapidly data updates which will needs continuous updates to the file header consuming more computational power. Another drawback in traditional partial encryption approach is that IoT data includes a small size of sensitive data and a large size of public data. In traditional partial encryption approach even accessing public data is affected with processing the encryption metadata which is stored in the XML file header.

B. Motivation Example

Smart cities include huge number of sensors that are responsible of monitoring the surrounding environment. These sensors vary from temperature sensors, humidity sensors, cameras and others. Sensors are distributed everywhere on the streets, over the buildings and also inside homes and homes' gardens. Cameras on home gardens record different images from different angles and may be supported with facial recognition facilities. Data emitted by cameras include private data for the home owner. However, for security reasons police station or city authority should be able to access such data in specific intervals of time. The data associated with the home owner also includes sensitive parts such as the bank account or bank balance. To be more general the motivation example is based on sensor that emits data that should be accessed by specific users and restricted from another. Fig. 1 represents a general overview of sensors used in IoT systems. Thus as mentioned before the size of private data is small compared with the total size of monitoring data. Applying security features for the whole RDF file is not efficient. Instead, security should be applied only to the sensitive part of data.

IV. Proposed RDF Partial Encryption Model for IoT System

The proposed model perform a fine grained partial encryption for RDF files in which sensitive triples are encrypted while non-sensitive triples are represented in plain text. Unlike traditional approaches, encryption metadata is linked to individual RDF triples not to the whole graph. Thus each encrypted triple encapsulates its cipher text as well as its encryption metadata. RDF documents contain a small part of sensitive data and a large part of public data. Hence, it is more efficient to apply security features to the sensitive triples and store the encryption metadata as extension of the encrypted triple. Any access to public data is not affected by the metadata of the encrypted triples. Thus the openness and rapidly changing environment of IoT is not affected. Moreover storing encryption metadata as separated triples extended from the encrypted triple increase coherence features of RDF graphs. RDF graphs can be divided to smaller sub graphs. Then sub graphs can be shared in another RDF graph.

A. First Step: Selecting Private Data

The In the first step all the sensitive data are selected over the RDF graph. To illustrate the idea the motivation scenario discussed in Section 3.2 is used. As shown in Fig. 1 the RDF graph presents an image captured by a fixed camera installed in a home garden to monitor the motion of persons. The captured images include private data that should be limited to the home owners. However at specific time the police station may require to use these images. Analyzing the RDF graph shows that sensitive data is limited to the image URL which include the link to the captured image. Other data does not include high level of privacy such as camera type or file format. Thus the first encrypted triple is the object in the following triple.

(Subject: captured image, Predicate: saved in, Object: image URL)

Another sensitive data related to the person how owns the home is his bank account. The second encrypted triple is the subject and object in the triple (Subject: bank account, Predicate: has balance, Object: balance).

Thus, the output of the first step determents which fragments of data should be encrypted. These fragments are named Sensitive Triples (ST) while the remaining triples are named Plain Triples (PT). The proposed work does not restrict a method to select sensitive data. ST selection may be done statically by enumerating the encryption fragment in the document before run time. Selection also can be done dynamically during run time by specifying selection patterns which check specific properties.

B. Second Step: Encrypting the Selected Fragment

After defining ST and PT an Encryption Function (EF) is added to each RDF ST in the RDF file. Thus for a graph G of RDF triples, ST ε G is an encrypted triple EF (ST) = (ST, has encryption container, E_c) where ST is the sensitive triple that will undergo encryption, as a whole triple or parts of it. E_c is the encryption container associated with the sensitive triple.

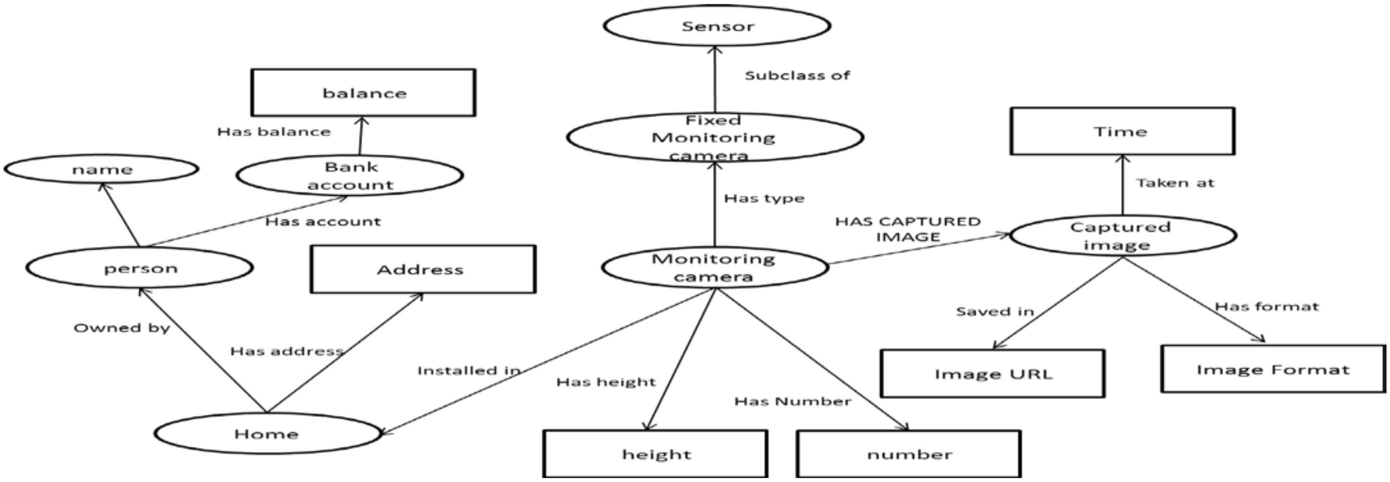

Fig. 1. RDF Representation of IoT Sensors in a Smart City.

The encryption container contains all the required encryption metadata. The encryption container contains a number specifying which part of the triple to be encrypted, as well as the encryption key and the encryption method. It also contains reference to the triple subject, predicate, and object as plain or cipher text according to the encryption numberE_n. The encryption number has a value from 1 to 7 as there are only seven possible probabilities to choose the EF from each triple (Subject, Predicate, Object) (S, P, O). E_n= 1 for O only, 2 for P only 3 for O,P, 4 for S only, 5 for S, O, 6 for S,P , 7 for S, P, O. The encryption key K_t is associated with the encrypted triple and is encrypted with the public key of the authorized user.

To clarify the parameters of the encryption function, it is applied to the motivation example in section 3.2 as follows:

(Subject: captured image, Predicate: saved in, Object: image URL)

The sensitive data is the URL of the image which is the object so the encryption function is defined as follows:

EF (ST) = (Sensitive Triple, has container, E_c) then the encryption container has the following related properties:

(E_c, has number, E_n) $E_n$is the encryption number which has the value of 1 (only the object will be encrypted).

(E_c, has key, E_k) a new E_k is generated for each sensitive triple and is encrypted by the public key of authorized user.

(E_c, has method, DES) the object is encrypted with DES.

In this case the object will be in cipher text, while the subject and predicate will be in its plain text.

For the second sensitive data in the motivation example:

(Subject: bank account, Predicate: has balance, Object: balance)

The sensitive data are the bank account and the balance, which are the subject and object so the encryption function will be:

EF (ST) = (Sensitive Triple, has container, E_c)

(E_c, has number, E_n) $E_n$is the encryption number which has the value of 5(Subject, Object)

(E_c, has key, E_k) a new E_k is generated for each sensitive triple.

(E_c, has method, DES) the subject and object is encrypted with DES

Thus each encrypted triple has an associated metadata which is inserted to the RDF graph as an encryption container. The encryption container contains a set of triples that presents the metadata of the original encrypted triple.

The encryption container includes three parts of metadata associated with the encrypted triple as well as the three triple subject, predicate, and object. First part is the encryption number which defines which parts of the triple are encrypted. Second part is the encryption key that used to perform symmetric encryption of the triple. Third part is the cipher text of the encrypted part of the triple. There is no need to encrypt all the data in the triple the sensitive parts are encrypted while public parts are presented in plain text.

C. Third Step: Decryption

Each reference to a sensitive triple will be directed to the encryption container associated with this triple Fig. 2. The decryption is done according to the parameters specified in the encryption container. Authorized recipient will perform asymmetric encryption for the triple key using their asymmetric private key. Then the triple session key is used to decrypt the cipher text of the triple. If a receiver does not have an appropriate triple key, the decryption fails. Public users who access public triples are not affected with the encryption process as the encryption containers are associated with sensitive triples only. Fig. 3 represents the decryption process.

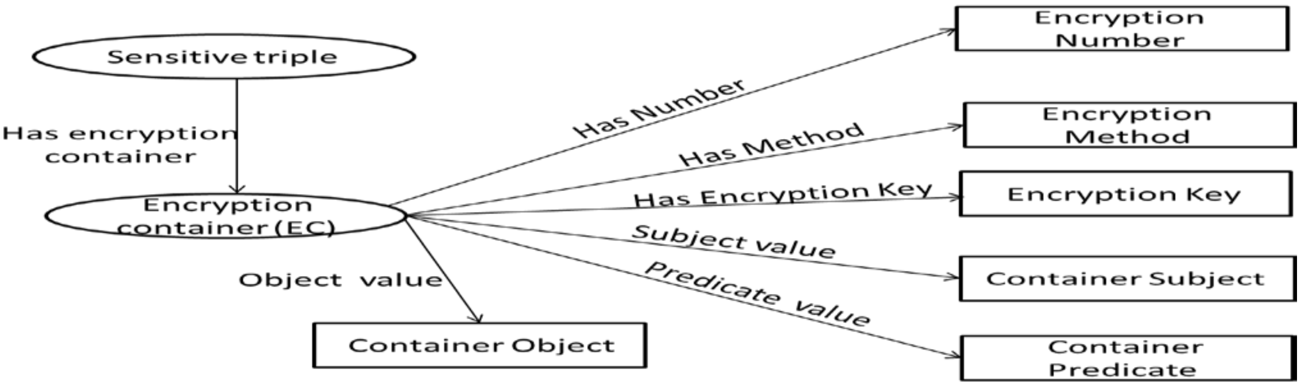

Fig. 2. Encryption Container.

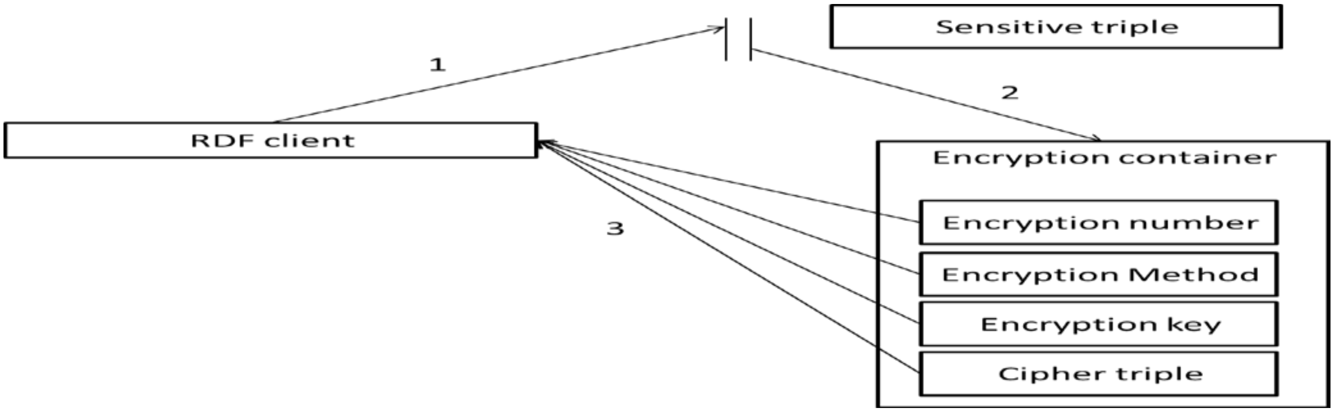

Fig. 3. Decryption Process of Encrypted RDF Triple.

V. Implementation and Empirical Evaluation

The aim of the evaluation is to proof the efficiency of the proposed method while maintaining strong security aspects for sensitive data. The motivation example of smart city is implemented using XML serialization. A data set including 1000 RDF triples is used to perform different calculations. The evaluation metrics are then compared with traditional partial RDF approach to proof the efficiency enhancement of the proposed work.

A. Evaluation Metrics

As the proposed RDF partial encryption method is mainly concerned with IoT echo system, it is supposed to maintain openness and rapidly data updates environment. Thus the encrypted RDF triples should allow rapid access time and rapid data updates. Other significant attributes in IoT systems is the ability to integrate several parts from different RDF graphs together. Hence, such attribute is considered while choosing evaluation metrics of the proposed work. Thus two metrics were chosen, the first is the access time, the second is the ability to integrate the RDF file with another files.

B. Implementation

To clarify the proposed method the motivation example was implemented using XML serialization. All the encryption metadata for each triple is linked to the triple itself rather than to the RDF header. Regarding the motivation example the object in the following triple contain private data.

Captured image: (hasimagefile) URL: image URL

According to the proposed work any access to the object of this triple will be directed to the Encryption container with the following properties:

Number = 1; as the encrypted part is the object

Triple key = K_t which is the session key encrypted by public key of authorized user.
Encryption method: DES
Cipher subject= the same as plain subject
Cipher predicate= the same as plain predicate
Cipher object= encrypted URL of the image
Thus the XML serialization of the sensitive triple is as shown below in listing 1.

```
</rdf:Describtion>
</rdf:Description
rdf:about="http://www.smartcity/..../CapturedImage/EncryptionContainer#">
<EC: Number> 1 </EC:Number>
<EC: tripleKey> EncryptionKey </EC:Number>
<EC: EncryptionMethod> DES </EC: EncryptionMethod>
<EC:PlanSubject>http://www.smartcity/..../CapturedImage#/IMG2013
</EC:planSubject>
<EC: PlanPredicate> Ci </EC:PlanPridicate>
<EC: CipherObject> ####### </EC:CipherObject>
```

Listing. 1. XML Serialization of the Encryption Container.

C. Experimental Results

To evaluate the proposed method, 1000 triples of the proposed RDF graph of the motivation example were implemented using Apache Jena API in Java. Then the RDF file was encrypted twice. First time, the RDF file was encrypted using the proposed method by applying the encryption metadata to individual sensitive triples. The other time the traditional encryption approach was applied to the RDF file by encrypting each triple and inserting the encryption metadata in the RDF file header. Then SPARQL query was used to access different sensitive and non-sensitive triple for the two RDF files. SPARQL query was applied to each triple in the file once, twice, and three times. The time to get data was calculated for the two files. The results were clarified in Table I. Results prove the enhancement of access time for the proposed partial encryption method. Using the proposed approach any access to non-sensitive data in RDF file will not process the encryption metadata. As the metadata is encapsulated in encryption container associated with each encrypted triples only. However, traditional RDF encryption approach requires processing the header for each access to the file whether the data is sensitive or not.

TABLE I. ENHANCEMENT OF ACCESS TIME USING RDF THE PROPOSED APPROACH

Number of time of applying SPARQL queries to each triple in the file	Response time for proposed method	Response time for traditional RDF encryption
One time	2.5 msec	5.3 msec
Two times	3.8 msec	8.2 msec
Three times	6.2 msec	10.4 msec

D. Discussion

To illustrate the efficiency of the proposed model it was compared with existing work. The results in Table I compare between the proposed work and traditional state-of-the-art methods for RDF encryption methods. It is observed that the proposed technique decreases the response time significantly for more than 50%. Thus the proposed technique provides IoT application with high respond time for SPARQL queries while maintain high level of security and data privacy. Moreover, the capsulation of encryption metadata in triples associated with the encrypted triple allows linking this encrypted triple to another RDF graph Moreover the proposed technique allows flexible integration of RDF triples from one RDF file to another without consuming time in processing metadata in RDF header.

VI. CONCLUSION AND FUTURE WORK

The proposed work provides RDF files with a fine grained partial encryption method suitable to be used in IoT ecosystem. The proposed method allows applying security aspects for each RDF triple individual reducing the time to process encryption metadata while accessing non-sensitive triples. The benefit of such approach increases in IoT systems hence RDF files used in IoT system include small size of sensitive data and large size of public data. Thus the proposed method maintains the security of sensitive triples while enhancing the access time of public data.

Moreover the proposed method provides IoT systems with the ability to integrate triples from different RDF file together to deal with different environments. The proposed approach supports such requirement as the security metadata is related to each individual triple rather than to the file as a whole. A future work will analyze the ability to compress XML files that includes encrypted RDF triples. Our future work will also discuss the ability to select the sensitive data dynamically at run time

REFERENCES

[1] Sankar Mukherjee, G.P. Biswas, "Networking for IoT and applications using existing communication technology," Egyptian Informatics Journal, Volume 19, Issue 2, 2018, Pages 107-127, ISSN 1110-8665, https://doi.org/10.1016/j.eij.2017.11.002.

[2] Michael Haslgrübler, Peter Fritz, Benedikt Gollan and Alois Ferscha," Getting through: modality selection in a multi-sensor-actuator industrial IoT environment" IoT '17: Proceedings of the Seventh International Conference on the Internet of ThingsOctober 2017 Article No.: 21 Pages 1–8 https://doi.org/10.1145/3131542.3131561.

[3] Ramadevi Chappala, Ch.Anuradha and P. Sri Ram Chandra Murthy, "Adaptive Congestion Window Algorithm for the Internet of Things Enabled Networks" International Journal of Advanced Computer Science and Applications(IJACSA), 12(2), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0120214.

[4] Minhaj AhmadKhan, KhaledSalah, "IoT security: Review, blockchain solutions, and open challenges" Future Generation Computer Systems, May 2018, Pages 395-41 https://doi.org/10.1016/j.future.2017.11.022.

[5] Sabrina Kirrane, Serena Villata, Mathieu d'Aquin, Mathieu d'Aquin, Sabrina Kirrane, Serena Villata, "Privacy security and policies: A review of problems and solutions with semantic web technologies", Semantic Web, vol. 9, pp. 153, 2018.

[6] Haytham Al-Feel, Hanaa Ghareib Hendi and Heba Elbeh, "Enrichment Ontology with Updated user Data for Accurate Semantic Annotation" International Journal of Advanced Computer Science and Applications(IJACSA), 10(12), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0101223.

[7] N. Seydoux, K. Drira, N. Hernandez, T. Monteil, "Capturing the Contributions of the Semantic Web to the loT: A Unifying Vision," Semantic Web Technologies for the Internet of Things Workshop colocated with 16th ISWC-2017, (2017).

[8] Kirrane, S., Mileo, A., Decker, S ,"Access control and the resource description framework: a survey," SemanWeb8(2), 311–352 (2017). doi:10.3233/SW 160236. http://dx.doi.org/10.3233/SW-160236.

[9] Uceda-Sosa R, Srivastava B, Schloss RJ, "Building a highly consumable semantic model for smarter cities," In: Proceedings of the AI for an Intelligent Planet on - AIIP '11. New York, New York, USA: ACM Press; 2011:1-8. doi:10.1145/2018316.2018319.

[10] Mamdouh Alenezi, Khaled Almustafa, Khalim Amjad Meerja, "Cloud based SDN and NFV architectures for IoT infrastructure," Egyptian Informatics Journal, Volume 20, Issue 1, 2019, Pages 1-10, ISSN 1110-8665, https://doi.org/10.1016/j.eij.2018.03.004.

[11] 1Brambilla, M., Umuhoza, E. & Acerbis R, "Model-driven development of user interfaces for IoT systems via domain-specific components and patterns," Journal of Internet Services and Applications 8, 14 (2017). https://doi.org/10.1186/s13174-017-0064-1.

[12] A. Gharaibeh et al., "Smart Cities: A Survey on Data Management, Security, and Enabling Technologies," in IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2456-2501, Fourthquarter 2017, doi: 10.1109/COMST.2017.2736886.

[13] Bhavani Thuraisingham, "Security standards for the semantic web," Computer Standards & Interfaces Volume 27, Issue 3, March 2005, Pages 257-268 https://doi.org/10.1016/j.csi.2004.07.002.

[14] B. Lalitha and G. Murali, "Implementing deduplication technique for RDF files with enhanced security using multi cloud servers," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 3618-3621, doi: 10.1109/ICECDS.2017.8390137.

[15] A. Esfahani et al., "A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment," in IEEE Internet of Things Journal, vol. 6, no. 1, pp. 288-296, Feb. 2019, doi: 10.1109/JIOT.2017.2737630.

[16] .Kaleem Razzaq Malik, Yacine Sam, Majid Hussain, Abdelrahman Abuarqoub, "A methodology for real-time data sustainability in smart city: Towards inferencing and analytics for big-data, Sustainable Cities and Society," Volume 39, 2018, Pages 548-556, ISSN 2210-6707, https://doi.org/10.1016/j.scs.2017.11.031.

[17] P. K. Das, S. Narayanan, N. K. Sharma, A. Joshi, K. Joshi and T. Finin, "Context-Sensitive Policy Based Security in Internet of Things," *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, St. Louis, MO, 2016, pp. 1-6, doi: 10.1109/SMARTCOMP.2016. 7501684.

[18] N. Yorino, A. Muhammad, Y. Sasaki, Y. Zoka, “Robust Power System Security Assessment under Uncertainties Using Bi-Level Optimization,” IEEE Trans. on Power Syst., Vol. 33, No. 1, pp. 352-362, Jan. 2018.

[19] G. Xu, Y. Cao, Y. Ren, X. Li and Z. Feng, "Network Security Situation Awareness Based on Semantic Ontology and User-Defined Rules for Internet of Things," in IEEE Access, vol. 5, pp. 21046-21056, 2017, doi: 10.1109/ACCESS.2017.2734681.

[20] Vogt L., Baum R., Köhler C., Meid S., Quast B., Grobe P, "Using Semantic Programming for Developing a Web Content Management System for Semantic Phenotype Data," In: Auer S., Vidal ME. (eds) Data Integration in the Life Sciences. DILS 2018. Lecture Notes in Computer Science, vol 11371. Springer, Cham.

[21] S. Benbernou, X. Huang and M. Ouziri, "Semantic-based and Entity-Resolution Fusion to Enhance Quality of Big RDF Data," in IEEE Transactions on Big Data, doi: 10.1109/TBDATA.2017.2710346.

[22] Antonio Celesti, Maria Fazio, "A framework for real time end to end monitoring and big data oriented management of smart environments," Journal of Parallel and Distributed Computing," Volume 132,2019,Pages 262-273,ISSN 0743-7315, https://doi.org/10.1016/j.jpdc .2018.10.015.

[23] Farhan Ullah, Muhammad Asif Habib, Muhammad Farhan, Shehzad Khalid, Mehr Yahya Durrani, Sohail Jabbar, "Semantic interoperability for big-data in heterogeneous IoT infrastructure for healthcare, Sustainable Cities and Society, "Volume 34,2017,Pages 90-96, ISSN 2210-6707, https://doi.org/10.1016/j.scs.2017.06.010.

[24] H. Cai, B. Xu, L. Jiang and A. V. Vasilakos, "IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges," in IEEE Internet of Things Journal, vol. 4, no. 1, pp. 75-87, Feb. 2017, doi: 10.1109/JIOT.2016.2619369.

[25] Joe Tekli, Nathalie Charbel, Richard Chbeir, "Building semantic trees from XML documents," Journal of Web Semantics, Volumes 37–38, 2016, Pages 1-24, ISSN 1570-8268, https://doi.org/10.1016/ j.websem.2016.03.002.