Securing High-Velocity Data: Authentication and Key Management Model for Smart City Communication

M. Mazhar Rathore[1], Yaser Jararweh[2], Muhammad Raheel[3], Anand Paul*, [4]

[1]Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha Qatar
[2]Duquesne University, PA, USA
[3]School of Computer Science and Technology, Anhui University, Hefei, China
[4]The School of Computer Science and Engineering, Kyungpook National University, Daegu, Korea
rathoremazhar@gmail.com, jararwehy@duq.edu, raheelmuhammad66@yahoo.com, *paul.editor@gmail.com

***Abstract*—The smart city is established through continuous analytics of the city data that is harvested from various smart systems and IoT sensors deployed in the city such as, smart homes, buildings, and parking, smart pollution monitoring, transportation sensors, etc., which generates continuous streams of high-speed data. Lack of security and privacy in the smart city system while communicating with the central analysis building gives the control of the city to the cyber criminals. However, offering privacy and security for such a continuous high-speed communication requires an efficient security model without introducing any delay in the real-time communication of the smart city. Thus, to cater with these challenges, in this paper, we proposed an efficient authentication and key management model among the smart city entities such as, remote smart systems (RSS) (i.e., smart homes, smart parkings, smart health systems, environment control system, etc.), remote users (U), and central city analysis building. First, every remote system and citizen have to be registered with the system through secure registration phase. Later, the session keys are generated and shared among the smart city entities that are used in secure routine communication. Moreover, the protocol is enabled with an efficient secure communication mechanism with the ability of parallel processing that makes it possible to work in a real-time environment. Finally, the security of the proposed model is verified informally through mathematical analysis and formally by implementing it using AVISPA tool (Automated Validation of Internet Security Protocols and Applications). Also, the proposed protocol is evaluated in terms of its efficiency and capability to work in a real-time environment. The results show that the protocol is secure against known cyber-attacks, is efficient, and faster than the existing schemes.**

Keywords — High-Speed Data, Smart City, Cryptosystem, IoT

I. INTRODUCTION

The city data is monitored through IoT devices that are deployed at various places in the city such as, at home, at the road, at bridges, at cars, at mountains, at buildings, etc., and is analyzed at central city server to make a smarter decision related to the city. Thus, the data is transmitted from remote devices to the central server, as depicted in Figure 1. In a smart city, the data is harvested by connecting the central city building with the remote smart systems working within a city such as, smart homes and building, parking, environmental and air control systems, city traffic system [1, 2], health care systems [3], and city surveillance cameras. The central intelligent city building (ICB) makes use of the collected data to make real-time decisions, generates alerts, and respond to citizens' request. Initially, all of the mentioned remote smart systems (RSS) produce their data by sensing the environment. Every RSS has a central node that is connected with the ICB via Internet. The ICB is responsible for city related decision-makings and responding citizens' queries [4, 5] using machine learning, soft computing, deep learning, statistical methods, and other decision making models. The users (U) and RSSs are required to be registered with the system through a registration procedure before they interact with the ICB.

Overall, the smart city [6] is bringing a revolution in the life of humans with respect to their knowledge, health, transport, safety, and living style. Similarly, storing and using city data for urban planning upturn the ability of authorities to modernize their city through pre-examining the future need and citizen's demands [7]. Regardless of all the modern facilities delivered to citizens through smart city, the smart city uses the citizens' data in order to take intelligent decision-making. This raises the legal and social concerns on the right of privacy by citizens.

The smart city and urban planning systems employ data related to city assets as well as related to its citizens such as, their mobility, health, behavior, and living style. If all the city data is communicated and stored without security measurements on the internet, it may leave a terrible impact on the life of citizens and the country as a whole. For instance, the vehicular movement analyzed by the city monitors the mobility of citizens. Thus, with the insecure user's mobility information, an attacker can perform a criminal activity that destroys his assets or harm him physically. Likewise, we can also think about the affect if the city assets are compromised. There are number of active and passive attacks that can be launched on city data and assets. Therefore, securing data and assets is extremely important in a smart city environment. Most of the attacks on data can partially be catered trough powerful confidentiality, integrity, and authentication mechanisms used by different security models. However, in smart cities, thousands of sensors and actuators work together and continuously generate real-time data with a very high-speed.

978-1-7281-1796-6/19/$31.00 ©2019 IEEE

The security models perform complex computations on large numbers that can only be implemented on a delay tolerant systems. Thus, providing security to such a high-volume and high-speed data analytics system in order to achieve reliable real-time decision-making for smart city is a major challenge. We need an efficient security model for authentication and key management to secure the smart city communication while introducing no delay.

Therefore, in this paper, we proposed an efficient authentication and key management model among the smart city entities such as, remote smart systems (RSS) (i.e., smart homes, smart parkings, smart health systems, environment control system, etc.), remote users, and central city analysis building. The proposed authentication model provides all the needed security services including integrity, confidentiality, authentication, and availability while introducing a very minor delay. First, every remote system and citizen have to be registered with the system through secure registration phase. Later, the session keys are generated and shared among the smart city entities that are used in secure routine communication. Moreover, the protocol is enabled with an efficient secure communication mechanism with the ability of parallel processing that makes it possible to work in a real-time environment. Finally, the security of the proposed model is verified informally through mathematical analysis and formally by implementing it using AVISPA tool (Automated Validation of Internet Security Protocols and Applications). Also, the proposed protocol is evaluated in terms of its efficiency and capability to work in a real-time environment.

II. The Proposed System

In this section, firstly, we have presented the current security needs of the smart city. Afterward, we introduced the proposed security model by highlighting its various phases.

A. Smart City Security Needs

Figure 1 depicts the smart city architecture [4, 5] that we are focusing on for the security proposal. Our security proposal did not consider the internal security aspects of the RSS (i.e., how they secure data from sensors to the main RSS node). The proposed authentication and key management model only emphasizes on the communication security between remote entities (i.e., RSS and user U) and ICB. As, the data transmitted from RSS to ICB (we called it COM1) uses the public internet link, so, the possibility of data capturing and misuse by the intermediate intruder is more. The same risk is with the communication from U to the ICB (We called it COM2). Hence, both COM1 and COM2 are vulnerable to active and passive attacks that have to be avoided by the provision of an efficient security model that delivers the services of confidentiality, integrity, and authentication. The proposed model should also somehow be able to reduce the chances of availability compromising attack. The proposed Authentication and Key Management provide all of these services.

B. Proposed Authentication and Key Management Model

Providing the security to a communication between RSS to ICB and users is becoming more essential. Thus, it is our main aim to provide security to a high-speed environment without interrupting the overall efficiency of the smart city system. As to provide all of the security services that have been discussed above in the Figure 1, the security protocol are divided into different phases, i.e. 1) Registration, 2) Key exchange, 3) Session key revocation, 4) Data transmission, which is required

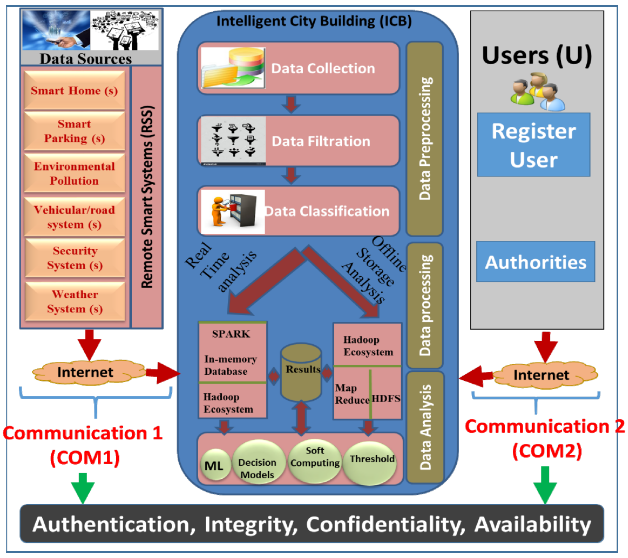

Fig. 1. Security Architectures of the Smart City

from RSS to ICB, and 5) Data transmission phase from the user to ICB. To overcome the shortcomings in existing security systems, we mainly focused on the privacy, efficiency, and security, while designing each phase of the protocol. Each of the protocol phase handled data in a real-time scenario.

Firstly, to grasp over the security and to design a reliable and efficient security protocol, every remote entity i.e. users (U) and remote smart system (RSS) needs to be enrolled with the smart city system. Secondly, at the time of registration process, each entity goes through stamping or verification process, where each new entity (its ID and public key (PU)) is stamped by the registration server. Later, while communicating with ICB for session key exchange, this public key (PU) and private key (PR) pair is used for encryption and signature. Nevertheless, before starting the formal transmission of data, the remote smart system (RSS) exchanges shared session key by using key exchange process. Although, the secure official transmission of data can be done by using public-private key pairs (asymmetric cryptography). However, encrypting the comprehensive data with asymmetric ciphers, the performance becomes very slow. Thus, to exchange and generate the shared keys for the symmetric cryptosystem, key exchange process is proposed. Thirdly, keys must be re-generated and revoked again by the session key revocation process after any key is lost, damaged or suspected to be compromised. Also, from time to time, the revocation process is executed to make the keys fresh. Fourthly, data transmission phase is executed, where generated keys are further used for protected communication among smart city entities. Lastly, the communication between User and ICB is also achieved by symmetric ciphers. All of these processes briefly discussed above are explicitly described in upcoming subsections.

Registration phase: Private and stamped keys are made to be important when the communication from remote entities (i.e., RSS and user) to the ICB is needed. However, the RSS and U required to be registered with the City Registration Server (CRS) before they communicate with the ICB. CRS is the authority, who registers the remote entities by stamping

their *IDs* and public keys (*STAMP)*. Figure 2 illustrates the overall abstract level registration process. For registration the *RSS* and the *U*, the procedure remains same. A registration request (*Reg_Req*) is sent by *RSS*/*U* to *CRS* by giving some required information. After getting the request, *CRS* verify the request of *RSS*/User either manually or automatically by the system. Afterwards, the authenticated *RSS*/*U* is successfully registered on *CRS*, and he is responded with the stamped identity. Later, the stamped information (*STAMP)* is used by the remote entities to be authenticated by the *ICB*. The Figure 3 presented the detailed formal explanation of the registration. We supposed that *CRS* already got a digital certificate (*Cert*) signed by a presumed signing authority. A simple and open registration request is sent to *CRS* from a remote entity (*RSS* or User). The response is sent from *CRS* by sending its digital certificate for the purpose of secure communication. Afterward, *RSS*/*U* confirms the digital signature, as any security protocol does by validating the signature on the public key (*PU*).

The RSS generates its asymmetric-keys pair i.e., (*PU*, *PR*). Subsequently, he generates a registration request by including his location, type, and other information and encrypting by *CRS* public key. He adds his time stamp T_{RSS} and nonce N_{RSS} to defeat against the replay attack. Lastly, to provide the integrity to the system, the hash of the encrypted nonce, timestamp, and the message is taken. To obtained the signature on the message for authentication, the hash digest is encrypted with the private key (*PR*) of the sender as $E\_AS\ (PR_{RSS}, h(\ EM \| N_{RSS} \| TS_{RSS}\ ))$. Afterward, at the end of the first phase, the public key, nonce, signature, the encrypted message, and timestamp of the *RSS* are sent as a registration request.

When *CRS* receives a registration request from any remote entity (*RSS*/*U*), *CRS* verifies the signature on the request by the public key of RSS/*U*. The timestamp and nonce cannot be changed by anyone, as they are signed. The verification of digital signature is done by comparing $h(\ EM \| N_{RSS} \| TS_{RSS}\ ))$ and $E\_AS(PU_{RSS}, S)$. If both parameters are not equal then it means there is a change in either *EM*, $T_{RSS,}$ or N_{RSS}. Further, the *CRS* uses own private key (*PR*) to decrypt the message using asymmetric cipher once the signature is verified. *CRS* generates an *ID* for the remote entity (RSS/*U*) and stores its credentials in the database for future use. Next, The *ID* and the public key of *RSS*/*U* is stamped by *CRS* using his private key as: $STAMP_{CRS} := E\_AS\ (PR_{CRS}, (ID_{RSS} \| PU_{RSS}))$. This *Stamp* is used by *RSS* to communicate with *ICB* by proving him with this *Stamp*. Subsequently, the *Stamp*, nonce, and time stamp are encrypted and sent as a response to *RSS*, i.e., Response: $(EM \| S \| N_{CRS} \| TS_{CRS})$.

The response from RSS is validated by *RSS* using hash and the public key of *CRS*. After validation, the message is decrypted and the *Stamp* is stored for future use. To acknowledge the successful reception of the *ID* and *Stamp*, the *Stamp* and *ID* is encrypted and signed by *RSS* and is sent to the *CRS*. Finally, *CRS* verifies the received *Stamp* and the *ID* by comparing the received S*tamp* and *ID* with the S*tamp* and *ID* generated in the prior steps for the corresponding remote entity. After that verification, the registration process is considered to be completed.

Session Key Exchange phase: For both remote entities (*RSS* and *ICB*), the process of key exchange with the *ICB* is similar. The use of public key cryptography (*PKC*) for a long message encryption and transmission of data is not convenient, because, the public key cryptosystem is very slow. A symmetric system is used for larger data encryption, which encrypts and decrypts

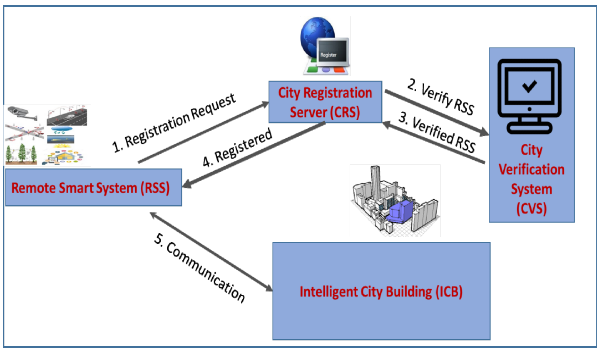

Fig. 2. Registering RSS with CRS at phase 1.

the data by using a shared key. The sharing of secured key among the communicating parties is offered by session key exchange phase, illustrated by Figure 4. First, the *ID* and *Stamp* is encrypted by the sender (*RSS*/*U*) with the *ICB's* public key, then nonce and time stamp are generated. Afterward, the hash of the encrypted message is taken and signed by the sender with his own private key (*PR*). Then, the RSS sends the encrypted message along with its nonce, time stamp and signature to *ICB*. The *ICB* validates the signature and the *Stamp* as $E\_AS(PR_{ICB}, EM))$, *Is* $((ID_{RSS} \| PU_{RSS}) = E\_AS(PU_{CRS}, STAMP_{CRS}))$. After the verification process, secret seed value and a shared key are generated by *ICB*. Both of the parameters are further encrypted and signed by *ICB* and sent back to RSS. The received information is verified and authenticated by *RSS* again as *Is (h* $(EM \| N_{ICB} \| TS_{ICB}) = E\_AS(PU_{ICB}, S)) = YES$, $PM := E\_AS(PR_{RSS}, EM))$.

RSS also generates its secret seed value and *XoR* it with the received seed value from *ICB*. Further, the seed and the shared key is encrypted by the public key of *ICB* and signed with the private key of *RSS*, and then sent to the *ICB*. Again, the all information is verified and authenticated by *ICB*. Furthermore, a final seed is formed by XoR operation on the seed value of *ICB* and the seed value of *RSS*. Then, a temporary key (*TK*) is computed by XoRing the seed and the shared key. By this process, the key turn into more secure key and key cannot be released as the message is encrypted by shared key. Constrainedly, he can only encrypts the seed value and then *ACK* with temporary key. Then, the seed value and the ACK are signed and transmitted to *RSS*. The received remote system authenticates the message from *ICB* and decrypts it with *TK*. After decryption, the successfully retrieval of the seed value ensures the success exchange of session key. The communication will be performed onward by using Key *K* and seed value SD_{RSS} and SD_{ICB}. To exchange the key between User and *ICB*, the same procedure will be performed but the only difference is that the notation of *RSS* will be changed into U.

Session Key Revocation phase: The procedure of revocation of session key is identical for both *ICB* to *U* and *RSS* to *ICB*. The use of seeds and session keys is limited for particular session or time duration. After the session expiration, the process of revocation and exchange of session key is carried out again. Possibly, any of the keys can be compromised for various reasons. Therefore, in such condition, revocation procedure can be carried out by any of the party for the purpose of session key refreshment. The procedure of session key revocation is almost

same as the session key exchange. However, at revocation phase, for remote entities, there is no need of sending the *ID*, *Stamp* and public key again. Rest of the process will remain the same. The figure 5 demonstrates the overall key revocation process.

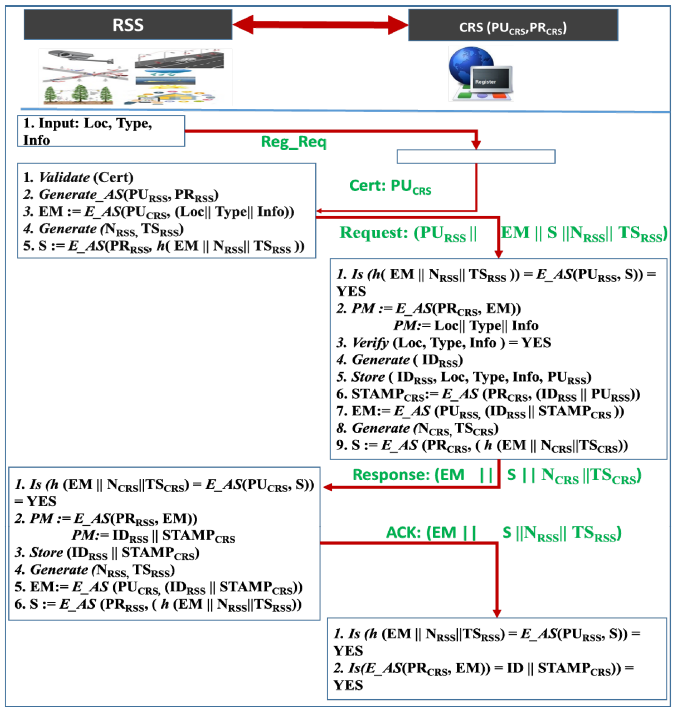

Fig. 3. Phase 1: Smart Remote Systems (*RSS*) and Users (*U*) Registration

Data Transmission from RSS to ICB: After the process of Key exchange, both types of entities (*RSS/U* and *ICB)* hold a share key *K*, the seed of *RSS* SD_{RSS}, and the seed of *ICB* SD_{ICB}. Using these secrets, the message can be transmitted securely between them. To maintain the continuous secure transmission of data among smart city entities as fast as possible, we need to overcome the extra delay and perk up the overall performance of smart city system. For this purpose, the phase of secure data transmission is designed to optimize the performance while introducing parallelism. The Figure 6 portraits the overall procedure of secure data transmission. For encryption of data the stream cipher (Pseudo Random Number Generator (*PRNG*)) is used with the combination of block cipher.

At the start, when any information that we desired to transmit from *RSS* to *ICB*, a random number is generated by *RSS* using the seed value *SD* i.e. SD_{RSS} XoR SD_{ICB}. Afterward, the random number is further encrypted by block cipher (*AES*) for the purpose of more security. The overall performance cannot be affected due to the fact that the block cipher is just applied on the small random number (not on large size message). The encryption is achieved by just XoRing the message with the encrypted random number, working as a key (*S_K*). To make the key prediction impossible for attackers, the key is refreshing for every transmission. Later on, by using the *PKC*, the signature is applied on the data, nonce, and time stamp to achieve authentication and integrity. The incoming data is divided into an encrypted message and the signature at *ICB*. The parallel processing is performed on both parts. The message is decrypted as by the same mechanism as encryption and the verification of signature is done similarly as discussed earlier.

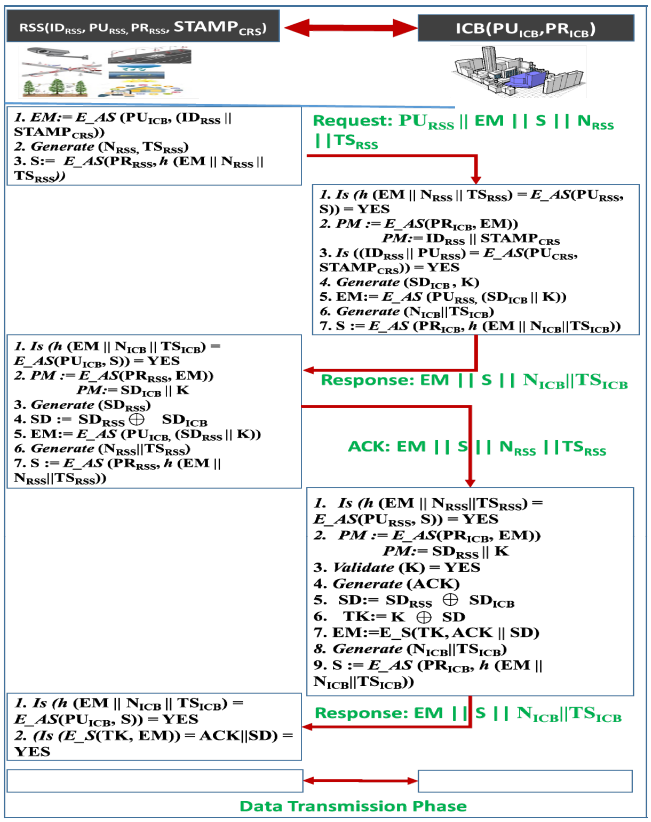

Fig. 4. Phase 2: Exchanging Session Key between Remote City Entity (RSS) and Central Analysis Building (*ICB*).

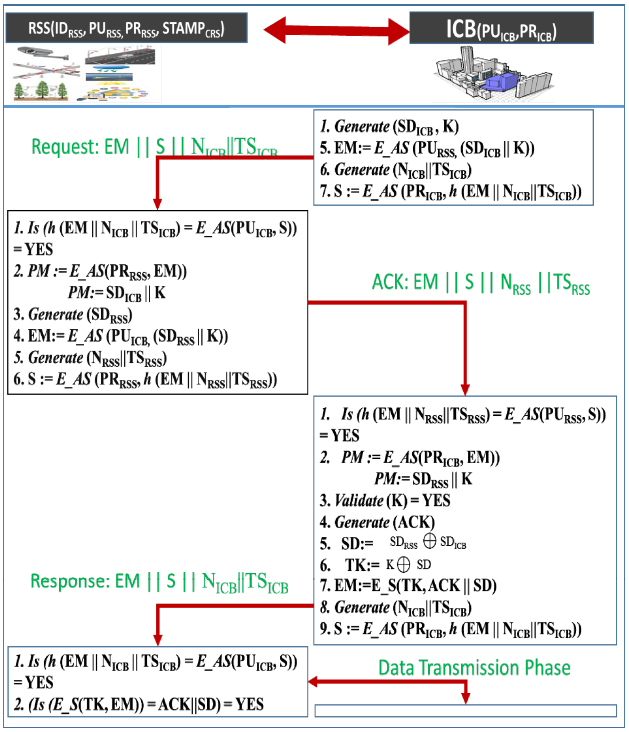

Fig. 5. Phase 3: Revocation of Session Key between Remote City Entity (RSS) and Central Analysis Building (*ICB*).

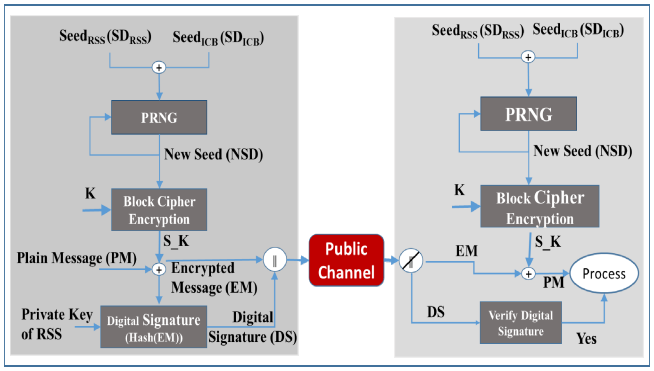

Fig. 6. Phase 4: Data Transmission between Remote City Entity (RSS) and Central Analysis Building (ICB).

Data Transmission phase between User and ICB: The requirement of efficiency for the data communication between the user U and ICB is totally different than the efficiency in data communication from *RSS* to *ICB*. Usually the query is transmitted by the user or the user gets back the response or any alert from the *ICB*. The communication between User and the *ICB* is done infrequently. Thus, from *U* to *ICB,* the system has delay tolerance and needs more security. Therefore, to make the data communication more secure between *ICB* to *U*, a symmetric block cipher is used for the encryption, *PKC* for the signature and a hash function for integrity. The data is encrypted by *K* and then the signature is taken on the hash-digest of the message. At the destination, the reverse process is performed.

III. System Evaluation

We have verified the proposed communication level security by informal mathematical analysis as well as by formally implementing the protocol in *AVISPA*.

A. Informal Security Analysis

The proposed security method for initial authentication and key exchange, is obliging to grasp over the various types of cyber-attacks, such as, man in the middle attack, denial of service (DoS) attack, replay attack, compromise session key or change of authorized keys attack, change of nonce, time stamp attack, compromise key attack and brute force attack.

Resistance to Denial of Service (DoS) attack: When the hackers attack on the server and suspend its services for legal user, it is known to be a denial of service attack. By using the strong digital signature mechanism, the proposed system handles this issue. In this mechanism, the public and private key encryption is used as signature, which is taken as *E_AS* (PR_{source}, *h* (EM || other-info)) and the verification is confirmed as *Is (h* (*EM* || other-info)) = *E_AS* (PU_{source}, Signature)). Thus, the *ICB* would only allow the data to be processed that is from the authentic user or *RSS*. The request of the user who is not authenticated by digital signatures is discarded and not allowed to enter into the system.

Resistance to man-in-the-middle attack: When the adversary tried to get access to the information which was communicated from the source to the destination and attempts to break through in-between the transmission, this illegal action is known as man-in-the-middle-attack. In this attack, the communication between two parties can either be replayed and or possibly be altered by the attacker. By providing a strong encryption mechanism, this attack is defied and confidentiality service is provided to the transmission. Furthermore, the hashing procedure is provided by our solution, which ensures that any attacker can not alter the information during transmission. Moreover, by using public/private key cryptosystem, influential digital signature method is used. This mechanism ensures that the data is receiving from the authentic user and there is no change while transmission of data.

Resistance to compromise session.: ICB and remote entities mutually computes the session key for secure data transmission. However, the compromising session key results in revealing secret data to the adversary. Thus, to avoid from compromising sessions, we proposed to use multiple shared secrets instead of one session key i.e., two secret seed-values generated by each party and a shared session key. In case any one of the secret values is revealed, the other remains the unknown. In addition, the revocation phase introduces fresh secret values in case the hacker guesses any of the secret. The secrets are also refreshed after the session time is expired.

Resistance to replay attack: The attack in which the valid data transmission deceitfully or spitefully delayed or repeated in known as replay attack or it is also known as playback attack. Suppose, an attacker might replay any request (that is encrypted and signed) from User *U* at the time (*T*+1) towards *ICB* to get the information required by a user at time T. By introducing a timestamp and a nonce at each communicating packet, this attack is resisted. To avoid the alteration of these parameters, they are hashed and signed as S:= *E_AS* (PR_{RSS}, *h* (*EM* || ID_{RSS} ||N_{RSS}||TS_{RSS})). Therefore, the system will detect any replay attack by verifying the timestamp and nonce.

Resistance Change of Nonce and Time Stamp attack: To address the replay attack, we use nonce and time stamp each communication. The time stamp and the nonce are hashed by each party and then signed using the private key. Therefore, the proposed solution avoided the possibility of the alteration in either timestamp or nonce.

Resistance to Brute force attack: The trial and error method is brute-force attack in which adversary attempts to decrypt/decode the data by using all possible keys through exhaustive effort. As, the strong encryption method is used by the proposed protocol with large key size. Hence, the cracking the key by using all possible combination is impossible. The brute force attack is resisted due to a large size of keys i.e., 128, 192, 256-bit keys that are used by *AES*, while *RSA* uses 1024 bits and at least 160-bit key size is used by *ECC*.

Resistance key to change of public key attack: Change of public key attack refers to the change in a public key while in the process of registration. The proposed resists this attack by encrypting and signing the initial registration request by the sender. Due to this mechanism, no change can be made by the attacker.

B. Formal Security Analysis using AVISPA Tool

The proposed authentication and key management model is implemented using broadly-accepted verification tool called "Automated Validation of Internet Security Protocols and Applications (*AVISPA*)" [8]. The proposed protocol is modeled in *AVISPA's IDE* called *SPAN* 1.6 [9] using High-Level Protocol Specification Language (*HLPSL*) [10] to validate the security of the model. *AVISPA* mainly focused on the detection

of a replay attack and any other man-in-the-middle attack on data secrecy and authentication. *AVISPA* used Dolev-Yao model [11] to include an intruder (i) in the communication while giving him a legitimate role with the full access over the channel and its own key set. Thus, while implementing the protocol for evaluation, we give full access of the channel to intruders. Also, the proposed protocol uses the timestamp in each phase to avoid the replay attack by the intruder. However, the current version of *AVISPA* does not support timestamp. It takes a timestamp as a nonce. Thus, while implementing the protocol, the *AVISPA* shows timestamp as a nonce. Figure 7, Figure 8, 9, and Figure 10 shows the implemented simulation model generated by *AVISPA* for all of the phases of the protocol including registration phase, key exchange phase, key revocation phase, and data transmission phase (from RSS-to-*ICB*) respectively. The simulation is run with intruder to show that how an intruder has full access of the channel. We can see in the simulation model with the intruder, the intruder receives every single message transmitted from any source to any destination.

Each phase of the protocol is run with intruder knowledge for security verification. Two broadly used *AVISPA* back-end verification models i.e., *OFMC* [12] and *CL-ATSE* [13], are used. The result of these models consists of following things:

- SUMMARY: Main part of the result that shows whether the protocol is safe/unsafe.
- DETAILS: it describes the conditions for which the protocol is acknowledged safe/unsafe.
- PROTOCOL: Protocol name and its location.
- GOAL: Specifies the goal of the analysis.
- BACKEND: shows the back-ends security verification model used for testing, such as *OFMC* and *ATSE*.

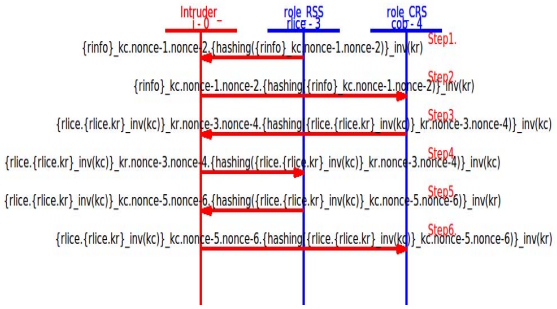

Fig. 7. Registration phase: protocol simulation with full access to intruder.

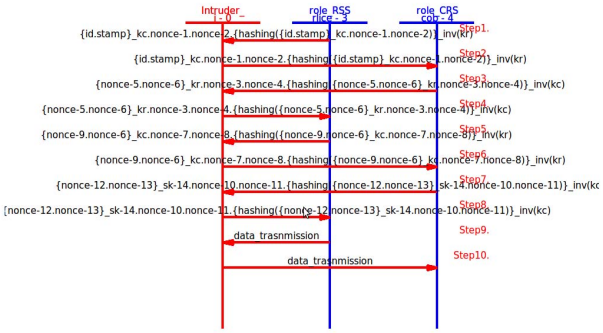

Fig. 8. Key Exchange phase: protocol simulation with full access to intruder.

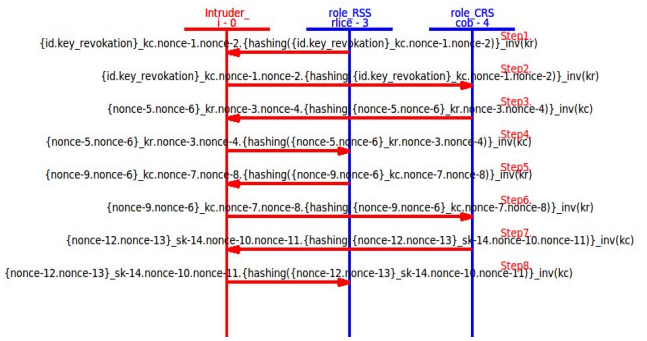

Fig. 9. Key revocation phase: protocol simulation with full access to intruder.

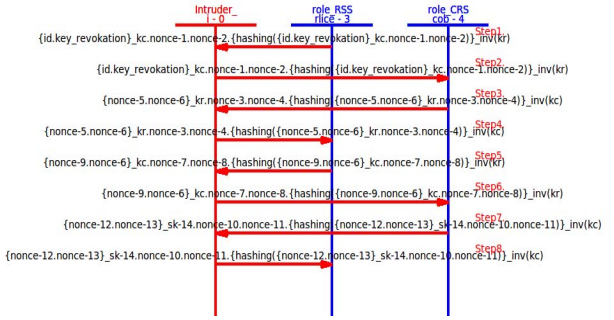

Fig. 10. Data Transmission phase: protocol simulation with full access to intruder.

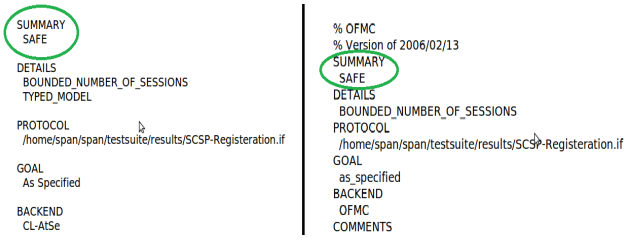

Fig. 11. Registration phase: security verification result by ATSE and OFMC models.

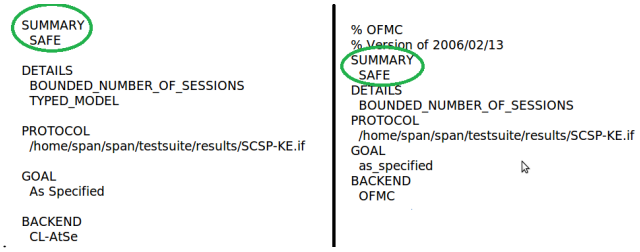

Fig. 12. Key Exchange phase: security verification result by ATSE and OFMC models.

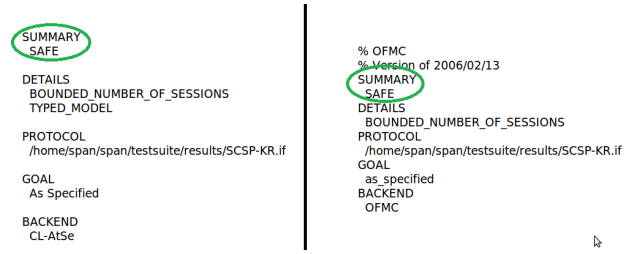

Fig. 13. Key revocation phase: security verification result by ATSE and OFMC models.

TABLE I. OVERALL COMPLEXITY OF THE PROPOSED SECURITY MODEL

Protocol Phases	T_{ASLE}	T_h	T_{ASSE}	T_{SLE}	T_{SSE}	Overall Complexity	Total: Messages Exchanged
Phase1: Reg. Phase	7 T_{ASLE}	6 T_h	6 T_{ASSE}	0 T_{SLE}	0 T_{SSE}	$7T_{ASLE}+6T_h+6T_{ASSE}+0T_{SLE}+0T_{SSE}$	5
Phase2: Session Key Exchange	6 T_{ASLE}	8 T_h	12 T_{ASSE}	0 T_{SLE}	2 T_{SSE}	$6T_{ASLE}+8T_h+12T_{ASSE}+0T_{SLE}+2T_{SSE}$	5
Phase3: Key Revocation	0 T_{ASLE}	6 T_h	10 T_{ASSE}	0 T_{SLE}	2 T_{SSE}	$0T_{ASLE}+6T_h+10T_{ASSE}+0T_{SLE}+2T_{SSE}$	4
Phase4A:Transmission RSS--->*ICB*	0 T_{ASLE}	2 T_h	2 T_{ASSE}	0 T_{SLE}	1 T_{SSE}	$0T_{ASLE}+2T_h+2T_{ASSE}+0T_{SLE}+1T_{SSE}$	1
Phase1:Transmission U ---> *ICB*	0 T_{ASLE}	4 T_h	4 T_{ASSE}	4 T_{SLE}	0 T_{SSE}	$0T_{ASLE}+4T_h+4T_{ASSE}+4T_{SLE}+0T_{SSE}$	2
Total	13 T_{ASLE}	26 T_h	34 T_{ASSE}	4 T_{SLE}	5 T_{SSE}	$13T_{ASLE}+26T_h+34T_{ASSE}+4T_{SLE}+5T_{SSE}$	17

The OFMC and ATSE security analyses show that all of the phases of the proposed smart city security protocol are safe. Any intruder with expert knowledge cannot perform any man-in-the middle-attack to exploit secrecy or authentication of the data or legitimate users. Figure 11, Figure 12, and Figure 13 shows the security verification results generated by OFMC and ATSE model for each of the phases of the protocol including registration phase, key exchange phase, and key revocation phase.

C. System Efficiency and Comparison

The main concern of the proposed model is to make sure the same level of security and more speed than existing security systems. The efficiency is a core issue that is catered by the proposed model to ensure that there is no delay in smart city services due to the provision of security procedures.

In proposed authentication and key exchange model, the hash function is considered as key process that consumes major portion of the overall processing time. So, the proposed protocol is evaluated with respect to computational cost by considering 1) the time consumed on the encryption of asymmetric large message (T_{ASLE}), 2) the time taken by the hash function (T_h), 3) consumption of time in asymmetric encryption for a small message for example the signature on small digest (T_{AASE}), 4) time taken by symmetric block ciphers for large message (T_{SLE}), for example payload, and 5) the time utilized by symmetric encryption on the small message such as, keys (T_{SSE}). In each step, the overhead of the system is considered regarding some messages exchange. The overall computational cast is shown in the table II. The data transmission from *RSS* to *ICB* is most important aspect, as the real-time high-speed processing is required at this step. The cost of data transmission at this step is very low, i.e., $2T_h+2T_{ASSE}$. The cost of data transmission is low due to the fact of using only two hash functions and two asymmetric encryptions on the small digest for the purpose of signature. Although, not all the steps are carried out every time, the overall cost of computation including all the phases is concluded as: $13T_{ASLE}+26T_h+34T_{ASSE}+4T_{SLE}+5T_{SSE}$. Where, $13T_{ASLE}$ symbolized that thirteen asymmetric large message encryptions are used, $26T_h$ points towards twenty-six hash functions, $34T_{ASSE}$ denotes that thirty-four asymmetric encryption for messages of small size like the signature on small digest, $4T_{SLE}$ signify four symmetric block ciphers for messages of large size like payload, and $5T_{SSE}$ indicate the five symmetric encryption on short messages such as keys.

As BLAKE-2 hash function is used for the authentication control and integrity, which are two main security services that cannot be avoided. The proposed model uses two, four, and six hash functions for most of the transmissions. Therefore, the system is evaluated by considering the time spent on one *BLAKE*-2 hash function, two *BLAKE*-2 hash, four *BLAKE*-2 hash function, and six *BLAKE*-2 hash function. These hash function are examined with respect to the increasing size of data. Blake-2 process 890 mebibytes/sec on 3.1 GHz (1 MiB=2^20 bytes). The time taken by one *BLAKE*-2 hash function, two *BLAKE*-2 hash functions, four *BLAKE*-2 hash functions, six *BLAKE*-2 hash functions are demonstrated in Figure 14. It is discovered that performing two hash functions on 10-12MB of data, *BLAKE*-2 just takes less than ten milliseconds, which is quite fast for a smart city system.

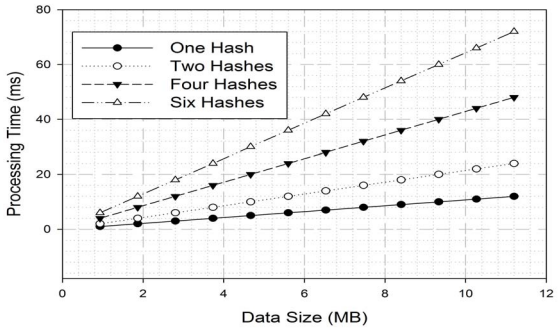

Fig. 14. Processing Time Comparison of the number of hashes (BLAKE 2) used by the proposed model

For the evaluation, the proposed security model for key management and authentication is also compared against the existing authentication schemes. Since, the smart city is the emerging technology, which has the security as the main concern. The existing system are secure enough, but they are not efficient enough to work in smart city environment, which does not bear any delay introduced by the existing security mechanisms. The complexity of our proposed protocol in the comparison of existing protocols is shown in the Table III. Six hashes functions are used for registration and eight hash functions for the key exchange, the proposed solution performed proved to be more efficient.

IV. CONCLUSION

In this paper, we proposed an efficient authentication and key management model among the smart city entities such as, remote smart systems (RSS) (i.e., smart homes, smart parkings, smart health systems, environment control system, etc.), remote users, and central city analysis building. There

TABLE II. COMPARISON OF NUMBER OF HASHES USED IN FIRST TWO PHASES OF THE PROTOCOL

Phase	Reddy et al. [14]	Lu et al.[15]	Lin[16]	Lee[17]	Gope and Hwang [18]	Proposed Algorithm
Phase1: Reg. Phase	6 T_h	9T_h	5T_h	3T_h	18T_h	6T_h
Phase2: Session Key Exchange	10 T_h	30T_h	10T_h	12T_h	26T_h	8 T_h

proposed model consists of various phases including registration phase, key exchange and key revocation phase, and data transmission phase. Registration phase securely registers remote systems and citizens with the overall city system. Next, key exchange and key revocation phase allows the generation of session key and other secret values. Later, it permits the secure sharing of these secret values over the communicating parties for the secure data communication. The proposed model is enabled with an efficient secure communication mechanism with the ability of parallel processing that makes it possible to work in a real-time environment. The validation and evaluation results show that the protocol is secure against known cyber-attacks, is efficient, and faster than the existing schemes.

REFERENCES

[1] M. M. Rathore, A. Ahmad, A. Paul and U. K. Thikshaja, "Exploiting real-time big data to empower smart transportation using big graphs," 2016 IEEE Region 10 Symposium (TENSYMP), Bali, 2016, pp. 135-139. doi: 10.1109/TENCONSpring.2016.7519392.

[2] M. M. Rathore, A. Ahmad, A. Paul and G. Jeon, "Efficient Graph-Oriented Smart Transportation Using Internet of Things Generated Big Data," 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Bangkok, 2015, pp. 512-519. doi: 10.1109/SITIS.2015.121.

[3] M. Mazhar Rathore, Awais Ahmad, Anand Paul, Jiafu Wan, Daqiang Zhang. "Real-time Medical Emergency Response System: Exploiting IoT and Big Data for Public Health." Journal of Medical Systems December 2016, 40:283.

[4] M. Mazhar Rathore, Awais Ahmad, Anand Paul, Seungmin Rho. "Urban planning and building smart cities based on the Internet of Things using Big Data analytics." Computer Networks, Vol 101, Pages 63-80, 2016

[5] Rathore, M. Mazhar, Anand Paul, Awais Ahmad and Gwanggil Jeon. "IoT-Based Big Data: From Smart City towards Next Generation Super City Planning." IJSWIS 13.1 (2017): 28-47. Web. 30 Nov. 2016. doi:10.4018/IJSWIS.2017010103

[6] Rathore, M. Mazhar, Awais Ahmad, and Anand Paul. "Big Data and Internet of Things: An Asset for Urban Planning." Proceedings of the 2015 International Conference on Big Data Applications and Services. ACM, eju Island, Republic of Korea, October 2015, pp. 58-65. Doi: 10.1145/2837060.2837067

[7] Jin, Jiong, Jayavardhana Gubbi, Slaven Marusic, and Marimuthu Palaniswami. "An information framework for creating a smart city through Internet of things." Internet of Things Journal, IEEE 1, no. 2 (2014): 112-121.

[8] The AVISPA Tool v1.1. "Available at http://www.avispa-project.org/"2006.

[9] Glouche, Yann, et al. "A security protocol animator tool for AVISPA." ARTIST2 workshop on security specification and verification of embedded systems, Pisa. 2006

[10] AVISPA. "Deliverable 2.1: The High-Level Protocol Specification Language." Available at http://www.avispa-project.org/publications.html, 2003.

[11] D. Dolev and A. C. Yao. ''On the security of public key protocols.'' IEEE Trans. Inf. Theory, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[12] D. Basin, S. M¨odersheim, and L. Vigan`o. "OFMC: A Symbolic Model-Checker for Security Protocols." International Journal of Information Security, 2004.

[13] Turuani, Mathieu. "The CL-Atse protocol analyser." International Conference on Rewriting Techniques and Applications. Springer, Berlin, Heidelberg, 2006.

[14] Reddy, Alavalapati Goutham, et al. "An Enhanced Biometric Based Authentication with Key-Agreement Protocol for Multi-Server Architecture Based on Elliptic Curve Cryptography." PloS one 11.5 (2016): e0154308.

[15] Lu Y, Li L, Yang X, Yang Y. "Robust Biometrics Based Authentication and Key Agreement Scheme for Multi-Server Environments Using Smart Cards." PLoS ONE, 10(5): e0126323. 2015. doi: 10.1371/journal.pone.0126323 PMID: 25978373

[16] Lin H., Wen F., & Du C. "An Improved Anonymous Multi-Server Authenticated Key Agreement Scheme Using Smart Cards and Biometrics." Wireless Personal Communications, 1‑12. 2015. doi: 10.1007/s11277-015-2708-4

[17] Lee C. C., Lou D. C., Li C. T., & Hsu C. W. "An extended chaotic-maps-based protocol with key agreement for multiserver environments." Nonlinear Dynamics, 76(1), 853‑866. 2014. doi: 10.1007/s11071-013-1174-3

[18] P. Gope and T. Hwang, "A Realistic Lightweight Anonymous Authentication Protocol for Securing Real-Time Application Data Access in Wireless Sensor Networks," in IEEE Transactions on Industrial Electronics, vol. 63, no. 11, pp. 7124-7132, Nov. 2016. doi: 10.1109/TIE.2016.2585081.