Attribute-based access control and authentication mechanism using smart cards for cloud-based IoT applications

B.B. Gupta* and Megha Quamara

National Institute of Technology Kurukshetra,
Mirzapur Part, Haryana 136119, India
Email: gupta.brij@gmail.com
Email: meghaquamara@gmail.com
*Corresponding author

Abstract: With exploding growth in information technology (IT), numerous services and applications having enhanced capabilities are coming into picture with an aim to serve the users. Internet of things (IoT) along with its enabling cutting-edge technologies is establishing a scenario where these services can be utilised effectively. However, with large number of users and applications, it becomes challenging to safeguard the identifying information being transmitted to provide access to these services. This paper presents a refined version of an integrated attribute-based access control and authentication mechanism using smart cards for cloud-based IoT applications. System-wide attributes not only restrict the users to access the remote cloud services, but also ensure user anonymity. We also implement the proposed mechanism on ACPT and AVISPA tool for its validation and to verify its correctness. Moreover, we present an analysis of its security and performance efficiency on the basis of different parameters.

Keywords: attribute; access control; authentication; authorisation; smart cards; cloud; internet of things; IoT; access control policy testing; ACPT; AVISPA; on-the-fly model checker; OFMC.

Reference to this paper should be made as follows: Gupta, B.B. and Quamara, M. (2020) 'Attribute-based access control and authentication mechanism using smart cards for cloud-based IoT applications', *Int. J. Embedded Systems*, Vol. 13, No. 1, pp.40–49.

Biographical notes: B.B. Gupta received his PhD degree from Indian Institute of Technology Roorkee, India in the area of information security. He has published more than 200 research papers in international journals and conferences of high repute. He has visited several countries to present his research work. His biography has published in the Marquis Who's Who in the World, 2012. At present, he is working as an Assistant Professor in the Department of Computer Engineering, National Institute of Technology Kurukshetra, India. His research interest includes information security, cyber security, cloud computing, web security, intrusion detection, computer networks and phishing.

Megha Quamara received her Master of Technology (MTech) specialised in Cyber Security from National Institute of Technology (NIT), Kurukshetra, India, in 2018, and Bachelor of Technology (BTech) in Computer Science and Engineering from University Institute of Engineering and Technology (UIET), Kurukshetra University, India, in 2015. Her research interests include Security in Internet of Things (IoT) and cloud computing, authentication in smart card technology, and data privacy.

This paper is a revised and expanded version of a paper entitled 'An identity based access control and mutual authentication framework for distributed cloud computing services in IoT environment using smart cards' presented at International Conference on Computational Intelligence and Data Science (ICCIDS 2018), Gurugram, India, 7–8 April 2018.

1 Introduction

Internet of things (IoT) being an extension of the conventional internet and information technology (IT), and an emerging domain of research, has become a far-reaching concept in the present day world. Many powerful IoT applications of remarkable value along with energy-aware (Xiao and Hao, 2016), location-aware (Catarinucci et al., 2015), time-aware (Cheng et al., 2016) services are being extensively used to serve the users in a variety of domains. With reduced or almost no human intervention, IoT is able to establish direct connection between the physical world and the internet accompanied with enhanced performance, efficiency and economic benefits to the society (Jiang et al., 2015; Zhong and Ge, 2018). Cloud computing enables remote processing, storage and management of data by a shared pool of resources. Evolutionary cloud computing

aspects including high-end computations, scalable storage and wide scale network access are enabling factors in designing and deploying numerous IoT-based business models, services and applications (Parasher et al., 2018). Thus, IoT and cloud computing can be considered as tightly coupled futuristic paradigms for the establishment of real-time cloud-based IoT ecosystems. While IoT provides wider acceptability, cloud computing enhances the performance efficiency and resource availability for the applications (Stergiou et al., 2018; Gupta, 2018; Gupta et al. 2016, Plageras et al., 2018).

Smart cards are easy-to-carry chip-based cards that enable processing or storage of the identifying information or other valuable data of the users (Liang et al., 2017; Nedjah et al., 2017). These cards have found their use in diverse applications areas including telecommunication, transportation, banking, healthcare, user identification, and so on. They enable auditing, offline data management, access control and support of mobility in these applications. Although industrial adoption of smart cards came into picture at least a decade ago, organisations across the world are still trying to explore their integration with other technologies. Commercial use of smart cards in the area of security has become significant due to the use of cryptographic tools and mechanisms in their development.

A typical cloud-based IoT environment consists of following entities – users, devices, communication channels and the cloud application servers (Gupta and Quamara, 2018). The users utilise devices along with the communication links to obtain access to the services and data provided by the cloud service providers (CSP). In this process, the whole system is susceptible to a number of malicious physical and logical attacks. Users are vulnerable to identity theft, location tracing, identity tracing, and so on. Because of the resource-constrained nature, complex and computational intensive security mechanisms cannot be implemented over IoT devices used by the users and thus, these are also vulnerable to a number of security attacks including unauthorised monitoring and logging, installation of rogue software, device compromise, and so forth. The communication channels are susceptible to denial of service attack (DoS), replay attack, man-in-the-middle (MITM) attack, eavesdropping, and so on (Devi and Kannammal, 2016; Panica et al., 2018). Moreover, the cloud application servers can also be hijacked by the malicious entities.

1.1 Motivation

In order to deal with these attacks and to protect the identity of the users, authentication and access control mechanisms play a vital role (Lohachab and Karambir, 2018). Whereas authentication helps in verifying the identity of the user, access control mechanisms provide access to the resources based on the privileges given to the users. For cloud-based IoT environment in which different applications are residing in a multi-server setting, authentication and access control mechanisms incorporating light-weight operations are required to facilitate access to multiple services and applications. Moreover, it is desirable to ensure the access using same set of credentials so that the user is not required to manage different set of credentials for different applications. In this regard, we propose an attribute-based access control and user authentication mechanism based on smart cards for applications residing in cloud-based IoT environment.

1.2 Research contributions

The fundamental contributions of the present work are summarised as follows:

- We utilise attribute-based access control (ABAC) model in order to restrict user's access to the cloud services, while at the same time to ensure user anonymity.
- We ensure user and server authentication using light-weight cryptographic operations for resource-constrained IoT devices in multi-server environment.
- To validate and verify the correctness of our mechanism, we implement the proposed mechanism on access control policy testing (ACPT) tool. Moreover, to perform the cryptographic check, we implement the proposed solution on automated validation of internet security protocols and applications (AVISPA) tool.
- We analyse the security of the proposed mechanism based on different security aspects and compare the same with other related schemes.
- We also analyse the performance of the proposed mechanism on the basis of computational cost incurred while performing the cryptographic operations. In addition, we compare the same with other related schemes.

1.3 Paper organisation

The remaining paper is organised into following sections. In Section 2, we discuss some related work in the field. Section 3 presents the details of our proposed solution including the preliminary concepts involved while designing the mechanism, system entities and its detailed working. In Section 4, we analyse the mechanism based on the security and performance aspects, and validate and verify its correctness on ACPT and AVISPA tool. Lastly, Section 5 concludes the paper and provides insights for future research directions.

2 Related work

With the progressive advancements in the field of cloud computing, different authentication and access control mechanisms have been proposed in the literature in order to ensure secure and efficient resource access to the remote users. In this section, we discuss some of the related work done in the field, and briefly summarise their concerns and the corresponding solutions provided by them in Table 1.

Table 1 Related work

Authors (year)	*Key concerns*	*Solutions*
Ma and Sartipi (2015)	• Identity management in cloud-based diagnostic imaging (DI) systems • Ensuring equivalent access control across various system entities	• User-centric single sign-on (SSO) • Integrating access control policies
Liu et al. (2016)	• Device sharing by multiple users to access cloud services	• Attribute-based access control • Two-factor authentication using one time passwords (OTP)
Yang et al. (2017)	• Account handling in distributed cloud computing environment • Access control in multimedia cloud	• SSO mechanism to access multiple applications • Group-based role-based access control (RBAC)
Namasudra and Roy (2017)	• Security and access control in cloud environment	• Smart card-based authentication using chaotic maps
Kumari et al. (2017)	• Security of data out-sourced to the cloud • Credential management in multi-server environment	• User authentication • One-time registration of the users
Amin et al. (2018)	• Security issues in multi-server cloud environment • Resource-constrained nature of IoT devices	• Light-weight user authentication
Roy et al. (2018)	• Handling heterogeneous data attributes in mobile cloud computing environment • Access to cloud server's data by resource-constrained devices	• Fine-grained access control • Light-weight authentication using hash and XOR operations

Ma and Sartipi (2015) presented a decentralised identity management and access control model for diagnostic imaging (DI) systems over the cloud. The authors incorporated fine-grained access control by integrating OpenID connect authorisation server with eXtensible access control markup language (XACML) policies. Liu et al. (2016) devised a two-factor attribute-based access control mechanism for web-based cloud computing services. The mechanism is based on bi-linear pairings and ensures the anonymity of the users. Yang et al. (2017) proposed a user authentication and data authorisation framework based on smart cards for multimedia cloud in order to ensure the protection of the privacy of the users and the data. The authors used group-based role-based access control (RBAC) model for content control and home group privacy. Namasudra and Roy (2017) developed a smart card-based authentication scheme for cloud computing environment using chaotic maps to ensure a number of security properties including mutual authentication, two-factor security, user anonymity, user un-traceability, and so on.

Kumari et al. (2017) proposed an authentication scheme based on user biometrics for multi-server cloud environment. The authors used bio-hashing to enhance the overall accuracy of the biometric pattern matching. Amin et al. (2018) proposed a light-weight authentication protocol for IoT devices for distributed cloud computing environment. The authors considered different security aspects including user anonymity, password secrecy, protection from insider attacks, and so on. Roy et al. (2018) proposed an integrated fine-grained access control and provably secure authentication framework for healthcare applications in mobile cloud computing environment containing multiple cloud servers. The authors incorporated light-weight cryptographic operations in order to ensure that the scheme is suitable for mobile devices having limited battery and resource-constrained smart cards.

3 Proposed mechanism

In this section, we discuss the details of our proposed work. We begin with discussing the assumptions related to the security threats and functionality along with some specific properties that we consider while formulating our solution. Afterwards, we discuss the system entities involved, detailed system working and phases.

3.1 Threat and functional notions

Some of the valid threat and functional notions used in the formulation of our proposed mechanism are discussed as follows:

- Single sign-on (SSO) support – Smart card can be used only with CSPs that facilitate SSO.
- Breach in client's privacy – An adversary may pretend to be an authorised CSP and can breach the privacy of the client by collecting the required information.
- Second authorisation – An adversary may pretend to be an authorised client in order to access the cloud services, applications and data from the CSP without having adequate privileges.
- Eavesdropping – An adversary may try to snoop the ongoing transmission in order to gather the confidential information of the communicating entities.
- Message manipulation – An adversary may modify, delete, re-route and re-send the messages captured during eavesdropping.

3.2 Specific properties

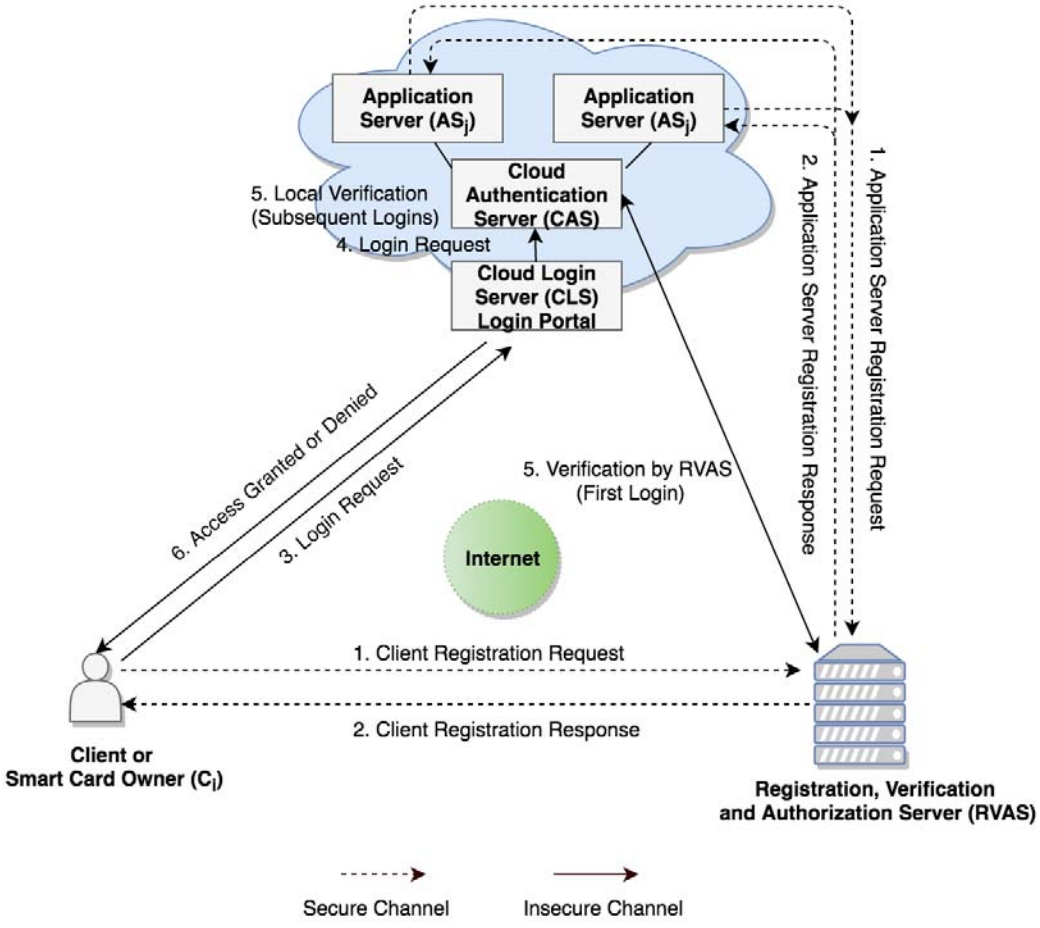
- Resource constrained environment – While designing the proposed mechanism, we consider that the devices implementing this mechanism are resource-constrained in terms of processing power and storage.
- Multi-server application environment – We consider that the applications may reside at multiple servers constituting a multi-server application environment and thus, each server is required to be registered.
- Communication channels – We consider both secure and insecure channels of communication. Secure channels are considered for initial registration, while for actual communication among the entities, insecure communication channels are utilised.

3.3 Proposed system entities

In our proposed model, different system entities are involved having different functionalities that are described as follows:

- Registration, verification and authorisation server (RVAS) – It is responsible for the registration of the clients and the cloud application servers. It is also responsible for verifying the identity of the client when the client tries to log into the remote application server for the first time, based on the already stored credentials of the client and the smart card. In addition, it defines the access control policies for the client using ABAC model.
- Client or smart card owner (C_i) – A client is a person who can make use of the services provided by CSP. To access the same, client needs to be registered first using appropriate information at RVAS in order to get a smart card.
- Cloud login server (CLS) – It facilitates client login by providing a login portal to the client in order to access the services provided by CSP.
- Cloud authentication server (CAS) – It verifies the legitimacy of the client logging into CLS after the first login in order to allow or deny access to the services.
- Application server (AS_j) – It serves the clients with required services, applications and data as per the privileges defined for the client.

Figure 1 System model for the proposed mechanism (see online version for colours)



3.4 Proposed system model and working

Figure 1 depicts the system model of our proposed mechanism (Gupta and Quamara, 2018). Secure communication channel can be seen between the client C_i and RVAS and between application server AS_j and RVAS for initial registration. Rest of the channels are insecure or open, and are used for later communication. The detailed working as per Figure 1 is given as follows. In the initial steps, the clients and the application servers are registered over RVAS. The client is issued a smart card as registration response, while the application server gets some secret information which is used by it for future communication. Once the client is successfully registered, he/she can send a login request to the remote CLS through a login portal which in turn is forwarded to the CAS. If this is the first login request from the client, the request is forwarded to RVAS. It verifies the identity of the client and the AS_j which will serve the client's request. Otherwise, CAS and the client mutually authenticate each other. After the identities have been verified, the client is able to access the cloud services.

3.5 Mathematical construction of the proposed mechanism

Now, we discuss the working of our proposed mechanism which is partitioned into six phases, namely – registration phase (client registration and application server registration), client login phase, mutual authentication and authorisation phase (identity verification by RVAS or mutual authentication with CAS), service access phase, client logout phase and password update phase. Table 2 contains the useful notations and their description used in the proposed mechanism.

Table 2 Summary of notations used

Symbol	*Description*
ID_i	Client's original identity
CID_i	Client's anonymous identity
PW_i	Client's password
RPW_i	Client's randomised password
BIO_i	Client's biometrics
h	One-way collision-free hash function
AID_j	Application server's identity
t_i	Timestamp
K	Secret key of registration, verification and authorisation server (RVAS)
OTP	One-time password

1 *Registration phase* – This phase is further divided into two sub-phases as given below:

1.1 *Client registration* – This phase involves registration of the client $_C$i who wishes to access the cloud services, who in return is issued a smart card from registration, verification and authorisation server RVAS.

a C_i computes anonymous identity $CID_i = h(ID_i \| BIO_i)$ and randomised password $RPW_i = h(PW_i \| BIO_i)$.

b C_i sends the registration request $\{CID_i, RPW_i\}$ along with the mobile number to RVAS through a secure channel.

c RVAS checks whether the client is already registered or not using CID_i.

d RVAS utilises its secret key K in order to compute the secret information S_i as –

$$S_i = h\left(CID_i \| RPW_i \| K\right)$$

e RVAS issues smart card to C_i having S_i stored over it.

1.2 *Application server registration* – In this step, the application server AS_j which is going to offer the services to the clients is registered at RVAS.

a AS_j sends its identity AID_j to RVAS for registration through a secure channel.

b RVAS computes a secret information $A_j = h(AID_j \| K)$ and sends it back to AS_j.

2 *Client login phase* – The client C_i who is already registered and wants to access the cloud services sends login request containing anonymous identity, randomised password and current timestamp, along with smart card information through the service portal provided by the cloud login server CLS, which in turn is forwarded to the CAS as:

$$C_i \rightarrow CLS \rightarrow CAS : \left(CID_j', RPW_i', t_i\right)$$

3 *Mutual authentication and authorisation phase* – This phase involves the verification of the identity of the client C_i and the application server AS_j which can occur by following two means:

3.1 *Identity verification by RVAS* – For the first login of the client, CAS stores the client's identity and then forwards the login request to RVAS along with $\{AID_j', A_j\}$. RVAS performs the necessary computations to determine the legitimacy of the client and application server, and sends the corresponding response.

a RVAS matches the current timestamp with the received timestamp t_i to ensure the freshness of the message.

b It matches CID_i' with CID_i to verify the correctness of the value with the already stored one.

c It computes $S_i' = h(CID_i' \| RPW_i \| K)$ and compares it with the value of S_i already stored over the smart card.

d) It computes $A_j' = h(AID_j' \| K)$ and matches A_j' with A_j.

e After comparing all the parameters, it sends the authentication response to CAS.

f If CAS finds that the identity of the client and the application server is valid, it sets the status bit corresponding to the client as 1.

3.2 *Mutual authentication with CAS* – For subsequent login requests by the client C_i, both the client and the CAS verify each other's identity through mutual authentication.

a CAS compares the value of CID_i' with already stored one.

b CAS sends an OTP to the client's registered mobile number.

c Client enters the OTP and CAS verifies the client's identity

d Afterwards, CAS sets the status bit corresponding to the client as 1.

4 *Service access phase* – Once the identity is verified, the client C_i can access the services on the basis of the privileges provided until he/she logs out.

5 *Client logout phase* – After the client logs out, CAS sets the status bit corresponding to the client to 0.

6 *Password update phase* – The client C_i can update the password by using the smart card and submitting the old credentials to RVAS for verification. Once the old credentials are verified, registration, verification and authorisation server RVAS ask for the new password $NRPW_i$ and the client provides the same. RVAS then replaces the older value of S_i over the smart card with the one recently calculated as:

$$C_i \rightarrow RVAS : \{CID_i \| NRPW_i\}$$

$$RVAS : S_i = h(CID_i \| NRPW_i \| K)$$

4 Security and functional analysis

In this section, we analyse our proposed mechanism based on the security and performance parameters, and compare the same with some other related schemes. We also implement our mechanism on ACPT tool to verify its correctness.

4.1 Informal security analysis

The security features supported by our proposed mechanism are discussed as follows:

- Client anonymity and un-traceability – The identity of the client is kept anonymous by applying concatenation and hash operation over the original identity ID_i and biometric information BIO_i. This ensures that even if the login message $\{CID_i', RPW_i', t_i\}$ is intercepted by the attacker, the original identity of the client will not be revealed. In addition, the time stamp t_i in the login request ensures that the attacker cannot trace the client in different session.
- Prevention from replay attack – In order to get the access permission, the client sends the credentials in the login request message along with the timestamp t_i at which the message is generated. At the server side, this timestamp is matched which ensures that the message if replayed is discarded.
- Prevention from stolen smart card attack – Our proposed mechanism also resists stolen smart card attack. Since the smart card contains the value S_i which is obtained by applying hash operation over the anonymous identity, randomised password and secret key of RVAS, original credentials of the client cannot be obtained by the attacker.
- Password-guessing attack – Since the original password PW_i of the client is masked using biometrics BIO_i and hash operation $h()$, it cannot be obtained by the attacker as the hash function is one-way collision-resistant hash function.
- Insider attack – A privileged client even if obtains the login message $\{CID_i', RPW_i', t_i\}$ and the smart card of another client, he/she cannot obtain the original credentials of the other client because of the masked credentials.
- Masquerade and skimming attack – In order for client to access the services, an OTP is sent to his/her registered mobile number, in the absence of which, any other person pretending as the original client cannot authenticate himself.
- Multiple login attack – Status bit ensures that a client cannot login multiple times in parallel using the same set of credentials.

In Namasudra and Roy (2017), user anonymity is not maintained as the original user identity is sent to the cloud service provider during registration phase. Also, client side verification of credentials is vulnerable to stolen smart card attacks. In Kumari et al. (2017), the original identity of the user is used by the registration authority which again fails to ensure user anonymity in registration phase. In Amin et al., (2018), the proposed solution is susceptible to stolen smart card attacks as the attacker on obtaining the smart card can input correct identity and password by hit-and-trial and can obtain access to the system. The solution proposed in Roy et al. (2018) also suffers from lack of user anonymity as original identity is used for user registration by the registration centre.

4.2 Performance analysis

Our proposed mechanism achieves the following functionalities:

- Choice of credentials and password change facility – The client is given authority to choose identity and password freely during the registration phase. Similarly, the client can change his/her password based on the requirements.
- Single-time registration – On receiving the registration request from the client in the registration phase, RVAS checks whether the client is already registered with same set of credentials or not, which prevents creation of multiple accounts for a single client.
- Ease of account management – The client can access the services from different service providers using single set of credentials and smart card. Thus, there is no need to remember or maintain multiple credentials.
- Computational efficiency – Involvement of RVAS is not required in every session to authenticate the client and the application server. Thus, it reduces the overall communication cost.
- Certificate-less application server registration – In our proposed model, the application server is registered and is provided a secret information for further computation instead of trust certificates, thereby saving storage and transmission cost.

Table 3 compares the computational cost of our proposed model with other related schemes in terms of number of hash operations involved. Phase-wise comparison is performed among the schemes. It can be seen that the proposed scheme incurs four hash operations in the registration phase, two in client login phase (in each first login and subsequent login), three in mutual authentication and authorisation phase (all three incurred during first login only), one in password update phase, thus a total of ten hash operations which is less than those in other related schemes.

Figure 2 shows the performance comparison in terms of number of rounds of messages in our proposed scheme and other related schemes. In the proposed scheme, the client login phase involves transmission of message from client to CAS only once which consists of the client identity CID_i', randomised password RPW_i' and current timestamp ti. Similarly, mutual authentication and authorisation phase involves transmission of authentication response from RVAS to CAS only once. Scheme proposed in Roy et al., (2018) matches the number of rounds of messages, while other related schemes (Amin et al., 2018; Kumari et al., 2017; Namasudra and Roy, 2017; Yang et al., 2017) involve transmission of multiple rounds of messages in both the phases. Yang et al. (2017) involves use of complex modular and exponential operations unlike our scheme which only involves light-weight hash and concatenation operations. Also, it does not provide freedom to the users to choose their credentials and involves registration authority in every session unlike our scheme in which RVAS is only involved during first login only.

Figure 2 Number of rounds of messages (see online version for colours)

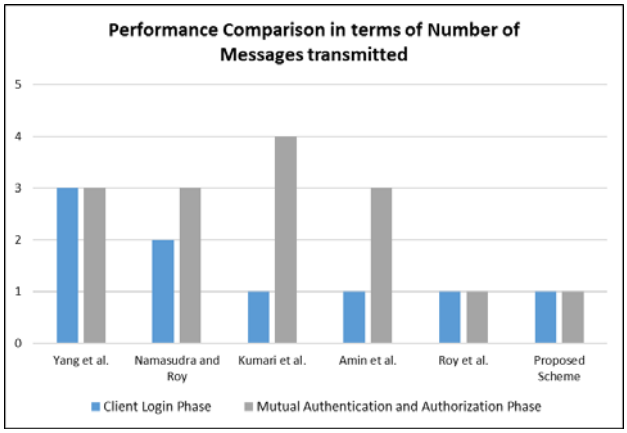


Table 3 Comparison of computational cost (on the basis of number of hash operations)

Schemes →		*Yang et al. (2017)*	*Namasudra and Roy (2017)*	*Kumari et al. (2017)*	*Amin et al. (2018)*	*Roy et al. (2018)*	*Proposed scheme*
Phases↓							
Registration phase		4	3	6	7	11	4
Client login phase	First login	5	6	18	5	9	2
	Subsequent login		3				
Mutual authentication and authorisation phase	First login		6		17	10	2
	subsequent login		0				0
Password update phase		-	4	4	8	11	1
Total		9	22	28	37	41	10

4.3 Verification over ACPT tool

Access control policy testing (ACPT) tool was developed by National Institute of Standards and Technology (NIST) as a prototype system to develop and verify the access control policies (CSRC-NIST, https://csrc.nist.gov/Projects/Access-Control-Policy-Tool). It provides Graphical User Interface (GUI) template for composing the policies, a symbolic model verification (SMV) model checker for checking the properties of the access control policy models, a combinatorial testing tool ACTS for test suite generation, and XACML mapping of the policies (Hwang et al., 2010).

In our proposed mechanism, we use ABAC model while implementing it on the ACPT tool. We use four categories of attributes:

- Subject attributes – Attributes associated with the client including biometrics, anonymous identity and randomised password.
- Action attributes – Attributes corresponding to the action being taken i.e. granting the client access to the remote application server.
- Resource attributes – Attributes corresponding to the resource being accessed, i.e., application server's identity.
- Environment attributes – Attributes associated with the operational environment, i.e., timestamp.

The possible results of the policies that we define for our proposed system model include:

- Client is given access to the remote application server after successful identity verification.
- Client is denied access to the remote application server after the failure of identity verification.

Table 4 Initial parameter settings in ACPT tool

Parameters	*Attributes*	*Attribute type*	*Attribute values*
Subject	Biometrics	String	Correct, incorrect
	Anonymous_Identity	String	Correct, incorrect
	Randomized_Password	String	Correct, incorrect
Resource	Identity	String	Correct, incorrect
Action	Access	String	Grant
Environment	Timestamp	String	Valid, invalid

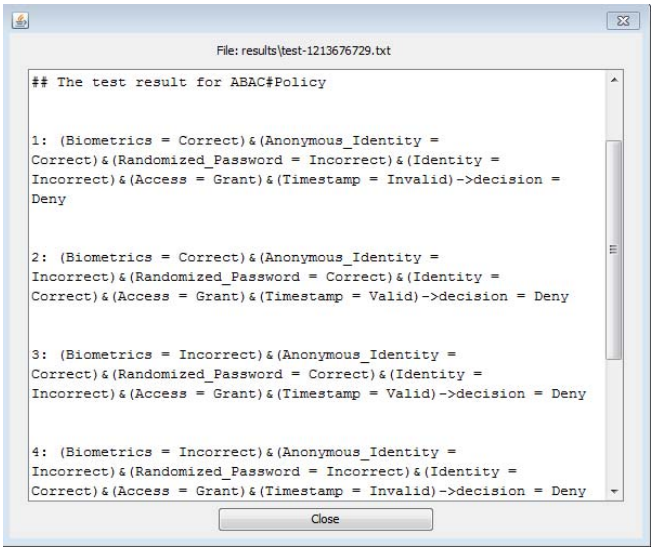
Table 4 shows the initial parameter settings for the proposed model in ACPT tool. As discussed earlier, three subject attributes, i.e., biometrics, Anonymous_Identity and Randomized_Password, are taken having string values correct or incorrect. Similarly, resource attribute which is identity of the application server is of string type. Attribute corresponding to action is access which is of type string and signifies access of remote application server to the client. Environment attribute is timestamp of type string having the values as valid or invalid.

Table 5 shows the results of the time taken for the verification and testing of the proposed model on ACPT tool. The static verification takes 1,545 ms, while dynamic verification takes 0.187 sec. Figure 3 shows the four out of six possible test cases during the execution of the proposed model in ACPT tool.

Table 5 Verification results in ACPT tool

Verification	*Running/execution time*
Static verification	1,545 ms
Dynamic verification	0.187 sec

Figure 3 Test cases during execution in ACPT tool (see online version for colours)



4.4 Verification over AVISPA tool

AVISPA stands for automated validation of internet security protocols and applications (AVISPA, http://www.avispa-project.org). It is a push-button tool used for the security analysis of web-based protocols and applications. High level protocol specification language (HLPSL) is used to define specifications for the protocol and security protocol animator (SPAN) builds message sequence chart (MSC) for the protocol execution. HLPSL specifications are mapped to intermediate format (IF) specifications that are provided as input to the back-ends of AVISPA for security analysis. The tool interactively builds security attacks over protocol to verify whether the protocol is resistant against the same or not. Figure 4 shows the MSC for the proposed solution which contains all the systems entities and the communication among them. This MSC corresponds to the intruder simulation of the solution in which an intruder is assumed to be eavesdropping all the communication. Figure 5 shows the result in on-the-fly model-checker (OFMC) back-end of AVISPA according to which the

proposed solution is secure against various security attacks for bounded number of sessions.

Figure 4 Message sequence chart of the proposed solution in AVISPA tool (see online version for colours)

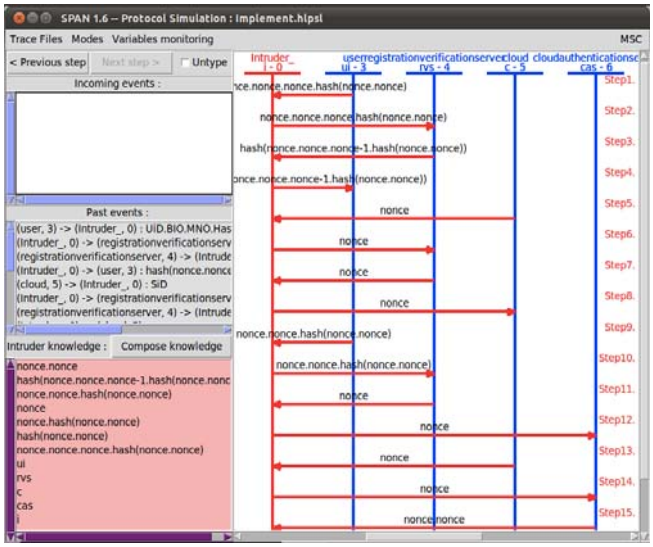


Figure 5 On-the-fly model checker results (see online version for colours)

SPAN 1.6 - Protocol Verification : Implement.hlpsl

File

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/Implement.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.04s
  visitedNodes: 97 nodes
  depth: 10 plies
```

5 Conclusions and future work

Concept of IoT aims at establishing inter-connectivity among a wide range of digital devices in order for them to communicate over the Internet and to share their resources. At the same time, cloud computing provides features of paramount importance to the Internet-connected users, such as device management, secure routing services, enhance storage and processing, quality of service (QoS), and so forth. In a cloud-based IoT environment, it becomes essential to ensure secure access to these services, applications and data shared over the cloud without compromising the anonymity of the user. In this regard, we proposed an attribute-based access control and authentication mechanism using smart cards for applications residing over cloud servers in IoT environment. We incorporated light-weight cryptographic operations considering the resource-constrained nature of IoT devices. In order to validate and verify the correctness of our model, we implemented the same on ACPT and AVISPA tool. The static verification took 1,545 ms, while dynamic verification took 0.187 sec, on ACPT tools. Results from AVISPA show that the scheme is secure against various security attacks. We also analysed our mechanism on the basis of security and performance aspects, and it can be inferred that our mechanism is more efficient than other related schemes.

For future work, the model can be enhanced to deal with attacks over an ongoing session by introducing the concept of re-authentication using key-caching-based mechanism (Wan et al., 2018) or by behavioural biometrics technique, and by data provenance concept. We also aim to consider transmission of data packets of different sizes along with fraud detection scenarios.

References

Amin, R., Kumar, N., Biswas, G.P., Iqbal, R. and Chang, V. (2018) 'A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment', *Future Generation Computer Systems*, Vol. 78, No. 3, pp.1005–1019.

AVISPA [online] http://www.avispa-project.org (accessed 14 May 2018).

Catarinucci, L., De Donno, D., Mainetti, L., Palano, L., Patrono, L., Stefanizzi, M.L. and Tarricone, L. (2015) 'An IoT-aware architecture for smart healthcare systems', *IEEE Internet of Things Journal*, Vol. 2, No. 6, pp.515–526.

Cheng, B., Zhu, D., Zhao, S. and Chen, J. (2016) 'Situation-aware IoT service coordination using the event-driven SOA paradigm', *IEEE Transactions on Network and Service Management*, Vol. 13, No. 2, pp.349–361.

CSRC-NIST, Computer Security Resource Center [online] https://csrc.nist.gov/Projects/Access-Control-Policy-Tool (accessed 25 June 2018).

Devi, P. and Kannammal, A. (2016) 'An integrated intelligent paradigm to detect DDoS attack in mobile ad hoc networks', *International Journal of Embedded Systems*, Vol. 8, No. 1, pp.69–77.

Gupta, B., Agrawal, D.P. and Yamaguchi, S. (Eds.) (2016) Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security, IGI Globa, USAl.

Gupta, B.B. (Ed.). (2018) *Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives*, CRC Press, Florida, USA.

Gupta, B.B. and Quamara, M. (2018) 'An identity based access control and mutual authentication framework for distributed cloud computing services in IoT environment using smart cards', *Procedia Computer Science*, Vol. 132, pp.189–197.

Hwang, J., Xie, T., Hu, V. and Altunay, M. (2010) 'ACPT: a tool for modeling and verifying access control policies', in *2010 IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY)*, pp.40–43.

Jiang, H., Shen, F., Chen, S., Li, K.C. and Jeong, Y.S. (2015) 'A secure and scalable storage system for aggregate data in IoT', *Future Generation Computer Systems*, Vol. 49, pp.133–141.

Kumari, S., Li, X., Wu, F., Das, A.K., Choo, K.K.R. and Shen, J. (2017) 'Design of a provably secure biometrics-based multi-cloud-server authentication scheme', *Future Generation Computer Systems*, Vol. 68, pp.320–330.

Liang, W., Xie, S., Li, X., Long, J., Xie, Y. and Li, K.C. (2017) 'A novel lightweight PUF-based RFID mutual authentication protocol', in *International Conference on Frontier Computing, Singapore*, pp.345–355.

Liu, J.K., Au, M.H., Huang, X., Lu, R. and Li, J. (2016) 'Fine-grained two-factor access control for web-based cloud computing services', *IEEE Transactions on Information Forensics and Security*, Vol. 11, No. 3, pp.484–497.

Lohachab, A. and Karambir (2018) 'Using quantum key distribution and ECC for secure inter-device authentication and communication in IoT infrastructure', in *Proceedings of 3rd International Conference on internet of Things and Connected Technologies (ICIoTCT)*, Jaipur, India.

Ma, W. and Sartipi, K. (2015), 'Cloud-based identity and access control for diagnostic imaging systems', in *Proceedings of the International Conference on Security and Management (SAM), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*, p.320.

Namasudra, S. and Roy, P. (2017) 'A new secure authentication scheme for cloud computing environment', *Concurrency and Computation: Practice and Experience*, Vol. 29, No. 20.

Nedjah, N., Wyant, R.S., Mourelle, L.M. and Gupta, B.B. (2017) 'Efficient fingerprint matching on smart cards for high security and privacy in smart systems', *Information Sciences*, Vol. 479, pp.622–639.

Panica, S., Irimie, B. and Petcu, D. (2018) 'Enabling and monitoring platform for cloud-based applications', *International Journal of High Performance Computing and Networking*, Vol. 12, No. 4, pp.328–338.

Parasher, Y., Kedia, D. and Singh, P. (2018) 'Examining current standards for cloud computing and IoT', *Examining Cloud Computing Technologies through the Internet of Things*, pp.116–124, IGI Global, Hershey, USA.

Plageras, A.P., Psannis, K.E., Stergiou, C., Wang, H. and Gupta, B.B. (2018) 'Efficient IoT-based sensor big data collection-processing and analysis in smart buildings', *Future Generation Computer Systems*, Vol. 82, pp.349–357.

Roy, S., Das, A.K., Chatterjee, S., Kumar, N., Chattopadhyay, S. and Rodrigues, J.J. (2018) 'Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications', *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 1, pp.457–468.

Stergiou, C., Psannis, K.E., Kim, B.G. and Gupta, B. (2018) 'Secure integration of IoT and cloud computing', *Future Generation Computer Systems*, Vol. 78, No. 3, pp.964–975.

Wan, Y., Luo, X., Qi, Y., He, J. and Wang, Q. (2018) 'Access-driven cache attack resistant and fast AES implementation', *International Journal of Embedded Systems*, Vol. 10, No. 1, pp.32–40.

Xiao, P. and Hao, Z. (2016) 'Improving energy-efficiency of large-scale workflows in heterogeneous systems', *International Journal of Computational Science and Engineering*, Vol. 13, No. 3, pp.258–267.

Yang, T.C., Lo, N.W., Liaw, H.T. and Wu, W.C. (2017) 'A secure smart card authentication and authorization framework using in multimedia cloud', *Multimedia Tools and Applications*, Vol. 76, No. 9, pp.11715–11737.

Zhong, R.Y. and Ge, W. (2018) 'Internet of things enabled manufacturing: a review', *International Journal of Agile Systems and Management*, Vol. 11, No. 2, pp.126–154.