A Cryptography and Machine Learning Based Authentication for Secure Data-Sharing in Federated Cloud Services Environment A Cryptography and Machine Learning Based Authentication fo...

Article *in* Journal of Applied Security Research · February 2021

DOI: 10.1080/19361610.2020.1870404

CITATIONS
32

READS
981

3 authors:

Ashutosh Kumar
Washington State University
23 PUBLICATIONS **79** CITATIONS
SEE PROFILE

Deepika Saxena
The University of Aizu
75 PUBLICATIONS **935** CITATIONS
SEE PROFILE

Ashutosh Kumar Singh
National Institute of Technology, Kurukshetra
371 PUBLICATIONS **4,568** CITATIONS
SEE PROFILE

Routledge
Taylor & Francis Group

Journal of Applied Security Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/wasr20

A Cryptography and Machine Learning Based Authentication for Secure Data-Sharing in Federated Cloud Services Environment

Ashutosh Kumar Singh & Deepika Saxena

To cite this article: Ashutosh Kumar Singh & Deepika Saxena (2021): A Cryptography and Machine Learning Based Authentication for Secure Data-Sharing in Federated Cloud Services Environment, Journal of Applied Security Research

To link to this article: https://doi.org/10.1080/19361610.2020.1870404

Published online: 08 Feb 2021.

JOURNAL OF APPLIED SECURITY RESEARCH
https://doi.org/10.1080/19361610.2020.1870404

A Cryptography and Machine Learning Based Authentication for Secure Data-Sharing in Federated Cloud Services Environment

Ashutosh Kumar Singh and Deepika Saxena

Department of Computer Applications, National Institute of Technology, Kurukshetra, India

ABSTRACT

Secure mutual authentication is an indispensable requirement to share organizational invaluable data among collaborating entities in federated cloud environment. This work presents a novel mutual authentication method which incorporates machine learning based ensemble Voting Classifier for online threat detection and Elliptic Curve Cryptography with Schnorr's signature scheme based key agreement to ensure secure communication among the participating entities by prior detection and mitigation of security breaches. The performance evaluation by using a benchmark from Canadian Institute for Cybersecurity Datasets and ProVerif security analysis tool verifies its efficiency in terms of security features and communication cost over existing approaches.

KEYWORDS

Elliptic curve cryptography; mutual authentication; ProVerif; Schnorr signature; Voting Classifier

Introduction

Across the world, the corporate organizations provide e-services like e-business, online banking, e-commerce, e-finance, e-marketing, etc. to make their clients' life easier and comfortable by leveraging hardware and software services from the multiple cloud service providers (Krombholz et al., 2015; Saxena & Saxena, 2015; Saxena & Singh, 2020a). To ensure safe and smooth online business or data sharing, it has become mandatory to secure all the online communication among sending and receiving entities. Therefore, a robust mutual authentication scheme is needed to allow protected key exchange and secure session set up during online data sharing (Li, Xuelian, et al., 2019; Li, Yang, et al., 2019). Enterprises keep their information on multiple cloud servers for the purpose of security and scalability (Saxena et al., 2018). Hence, multiple clouds collaborate to provide Information Technology (IT) services to these organizations, which have their branch offices distributed at different locations throughout the world (Li et al., 2012; Saxena et al., 2016; Saxena & Saxena, 2015).

CONTACT Deepika Saxena 13deepikasaxena@gmail.com Department of Computer Applications, National Institute of Technology, Kurukshetra 136119, India.

According to Thales report 2020, 47% of organizations have experienced security threats or compliance audit failure in the last year. Moreover, multi-server based environment has further raised the challenges and triggered the security vulnerabilities. In 2017, Equifax, which is a largest credit bureau in the U.S. faced application vulnerability that revealed confidential data of about 147.9 million consumers, as reported in Armerding (2018). Corresponding to the security breaches of online data exchange, many researchers have proposed a number of security solutions including secure authentication protocols, right to access data schemes, protected credential storage, secure secret key sharing or key agreement via mutual authentication and secure data storage and data sharing, etc. (Li, Xuelian, et al., 2019; Li, Yang, et al., 2019; Saxena et al., 2018; Saxena & Singh, 2020b). Some of the available security solutions includes dynamic identity based mutual authentication protocol using smart card in multi-server system to provide perfect security against user anonymity attack is present in (Li et al., 2012). In smart grid, online communication is secured by constructing public and private keys using ECC (Abbasinezhad-Mood & Nikooghadam, 2018; Mahmood et al., 2018). Mutual authentication is essential for every user to prove his identity to an opposite party for data sharing, which is mandatory for protection against frauds like, man-in-the-middle attack, key loggers, user anonymity, etc. Usually, mutual authentication involves three phases: registration, log in or authentication and password change phase. The registration phase allows clients to create their official identity or account at the trusted cloud server. An authentication phase provides secret key sharing for a specific session establishment (Abbasinezhad-Mood & Nikooghadam, 2018; Balaji et al., 2019; Mahmood et al., 2016; 2018; Nimmy & Sethumadhavan, 2014). The password change phase enable clients to renew the security locks of their identities at the cloud server.

In this work, an enhanced mutual authentication protocol is developed based on the integration of cryptography and machine learning approaches at the trusted cloud server to provide more robust and efficient solution, coping the limitations for online data sharing. We propose the proper registration phase, machine learning and cryptography scheme based threat detection and key agreement respectively during session establishment and password change phase for secure online data exchange. The key contributions of the proposed work can be summarized as follows:

- A novel **C**ryptography and **M**achine learning based **A**uthentication **P**rotocol (**CMAP**) is proposed to allow secure exchange of data among users in federated cloud server environment.

- An online threat detector based on ensemble Voting Classifier is developed at trusted cloud server to mitigate denial of service attacks and other security breaches.
- Security analysis of proposed protocol against various attacks including, insider attack, ID and password leakage attack, client impersonation attack, third party impersonation attack, session key computation attack, replay attack, forward secrecy attack, middle man attack and user's anonymity attack.
- Formal verification by using ProVerif security analysis tool and comparison of proposed protocol with state-of-the-arts to validate its efficiency.

Organization

Section "Online data sharing threat scenarios" discusses online data sharing threat scenarios. The recent key contributions pertaining to this work are discussed in two sub-divisions including cryptography based mutual authentication and machine learning based data sharing approaches in Section "Recent key contributions." Preliminaries are discussed in Section " Preliminaries." The proposed CMAP is presented in Section "CMAP" followed by security analysis of proposed protocol in Section "Security analysis." Algorithm and formal verification of proposed protocol driven by using an automated security analysis tool, ProVerif, is given in Section "Algorithm and formal verification of proposed mutual authentication protocol." The results and comparison with existing work is discussed in performance evaluation in Section "Performance analysis" followed by the conclusion in Section "Conclusion." Additionally, the proposed work's semantic verification implementation code is given in Appendix A.

Online data sharing threat scenarios

Figure 1 shows a credential leakage threat scenario, where a legitimate client initiates login process on trusted server, to share data with a remote receiver over secure public channel. Anyhow, some malicious user sense the public medium and launches attack by sending a fake login interface, to allow victim data owner to inert his credentials, thereby, stealing confidential information.

Figure 2 shows a sensitive data leakage threat scenario. A data owner has uploaded his encrypted documents over trusted server site in order to send some confidential information to an authenticated receiver. However, attacker has also got the decryption key due to unsecure key exchange before data sharing between sender and receiver. Hence, along with

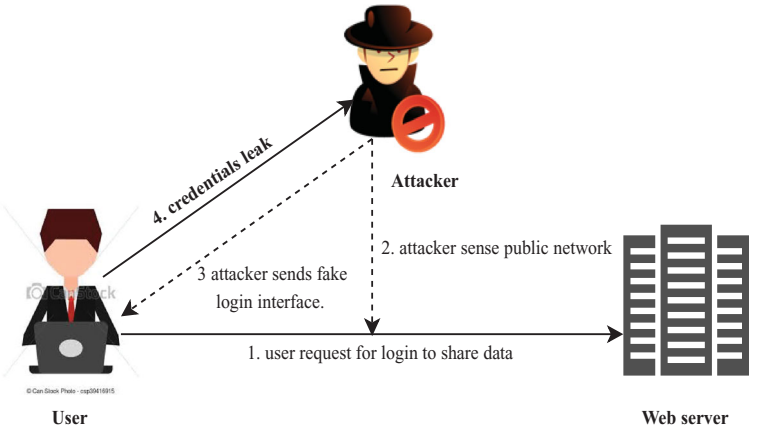


Figure 1. Credentials leakage during login phase.

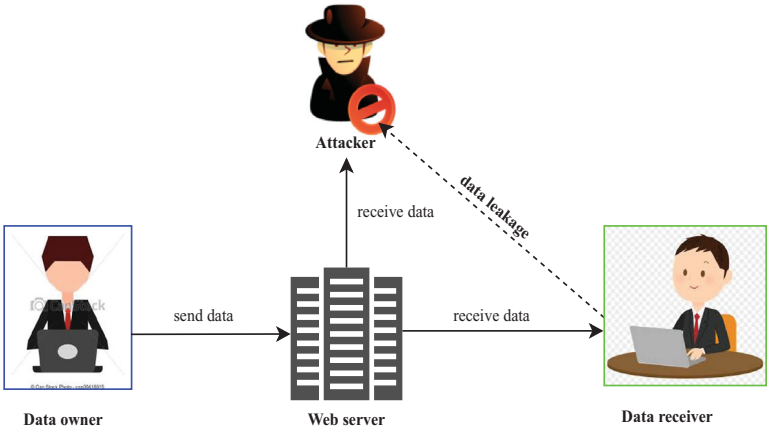


Figure 2. Information leakage during data sharing phase.

legitimate receiver, attacker also gets the chance to download and decrypt the confidential documents and thus, exploit the valuable information.

Recent key contributions

Cryptography based mutual authentication

In 1981, Lamport was the first to propose remote password authentication protocol for insecure communication, where the server must store password list that cannot resist interpolation attack. Hwang and Li (2000), Fan et al. (2005), Hwang et al. (2010) have presented different authentication schemes (Li et al., 2012). Moreover, Vincent et al. (2010) and Kumar et al. (2019) have proposed a secure e-commerce transaction scheme based on elliptic curve cryptosystem (ECC) in order to combat the slow response time of SSL protocol and traditional RSA protocol (Kalra & Sood, 2015).

proposed dynamic identity-based authentication scheme for multi-server architecture using smart card, that involves three parties- the user, service provider server, and control server (for registration). The user inserts smart card into card reader, which generates a random number for user log in and mutual authentication.

Li et al. (2012) indicates the vulnerability of Sood et al. (2011) scheme that it cannot resist leak-of-verifier attack, stolen smart card attack, incorrect session key agreement, as Sood's authentication scheme never use real identity of the user entity in any of the phases (registration, log in, authentication). In addition, they updated Sood's scheme by applying user's ID and Password at registration phase to compute secret number for the user that is required at log in or authentication phase. Here, the different masked identity of user is use for session set up, that is, exclusively created for each session. Moreover, in Wu and Zhou (2011), ECC based mutual authentication scheme was proposed. Later, Xia and Wang (2012) exposed the weakness of Wu et al. scheme that it cannot resist middle-man attack. Liu et al. (2013) proposed Cloud Mutual Authentication Scheme, where user simply enters, ID and password to login to cloud server. Cloud server has trusted platform module (TPM) which generates private secret key and does encryption. It claims to prevent replay attack, password guessing attack, impersonation attack, forward secrecy attack but it is unable to prevent DOS attack, stolen ID and password attack. Nimmy and Sethumadhavan (2014) proposed mutual authentication scheme in cloud computing environment using secret sharing and steganography in data sharing. Here one part of the share is kept at user's side and second part is kept at cloud server's side and secret can only be revealed when two shares combines together. This protocol cannot resist client anonymity attack as anybody can register at Authentication server (AS) and AS without properly verifying the authentication of client starts sharing secret with malicious user.

Lim et al. (2017) have surveyed cloud security and privacy and concluded that research still lacks detailed analysis of authentication for cloud security regarding protected data sharing. The acknowledgment by trusted third party (TTP) within cloud brings necessary protection and trust and preserve confidentiality, integrity, mutual authentication while sharing data online. Mahmood et al. proposed RSA-based authentication scheme in Mahmood et al. (2016) and later they presented authentication scheme for smart grid using ECC in Mahmood et al. (2018), that do not depends on public key infrastructure (PKI), instead it used Trusted Authority (TA) for private key generation. Recently, Abbasinezhad-Mood and Nikooghadam (2018) analyzed that in Mahmood et al. (2018), TA generated private key so, TA also knows private key of user which could become loophole. Therefore, in their authentication scheme, they applied ECC for public key

generation at TA and user entity itself generated private key using Schnorr's signature. Moreover, Elliptic curve cryptography based mutual authentication schemes were proposed in Wu and Zhou (2011) and Kumari et al. (2020) because of its powerful nature to defence against security breaches. A multi-factor mutual authentication protocol for IoT devices is proposed in Melki et al. (2020), which includes configurable physical unclonable functions (PUF) within IoT devices and channel-based parameters. The PUF value which changes for each session acts as the mutual secret identifier between a pair of users. Additionally, this protocol exploits the random channel characteristics to provide high robustness against different kinds of attacks, while maintaining low complexity.

Machine learning based data sharing approaches

There are several machine learning techniques that have been applied for secure data sharing (Bilogrevic et al., 2016), threat detection (Sreeram & Vuppala, 2019) and workload execution (Saxena & Singh, 2020c, 2020d) in cloud environment. A machine learning based data sharing system is developed in Bilogrevic et al. (2016) that takes data sharing decision based on granularity of request, personal and contextual features. It utilized multi-class classifiers based on support vector machines. It is adaptable to different users behavior and capable of predicting the level and amount of data to be share among users. Li et al. presented an android malware detection system in Li et al. (2018), in which they have identified most significant permissions by pruning at three levels through mining of permission data. Moreover, they have applied machine learning based classifiers to classify different families of malware and benign applications. An ontology based classification method is proposed in Rijvordt et al. (2019) which filters news of user specific domain. Such method can be incorporated to filter the legitimate cloud server login request from the spam or malicious ones. A bio-inspired distributed denial-of-service (D-DOS) attack detection is proposed in Sreeram and Vuppala (2019) to achieve the fast and early detection of the DDOS by HTTP flooding. Moreover, a machine learning based security analysis of authentication protocols is presented in Ma et al. (2018), which could identify three types of authentication attacks. Dodero et al. (2019) have presented a privacy preserving approach by re-engineering cloud applications as linked data-enabled confidentiality pre-servation properties based on the knowledge of architecture of the application. Most of the authentication algorithms (Abbasinezhad-Mood & Nikooghadam, 2018; Mahmood et al., 2018) share identity openly on public channel and have weak registration process, thus, suffers from user's anonymity attack, identity leakage attack etc. The key agreement schemes in

Guo et al. (2018) and Karati et al. (2018) are safe against many known attacks, however, there is no defence against reflection attack. Table 1 shows comparative analysis of proposed and state-of-the-art protocols.

In the light of above works, the contributions of the proposed work are as follows: A novel authentication protocol is developed by integrating cryptography and machine learning approaches to provide multiple security features including resistance against DOS attack, perfect forward secrecy, low communication cost (because of usage of lightweight operations for key agreement session), resistance against impersonation attack, phishing attack, replay attack, ID and password leakage attack, etc. and formal security of proposed protocol is formally verified by using widely accepted security analysis tool.

Preliminaries

Elliptic curve cryptography

Let p, q are large prime numbers. Elliptic Curve (EC) can be shown over a finite field F_p which is defined as $E(a,b): \ y^2 = x^3 + ax + b$, where $4a^3 + 27b^2(modp) \neq 0$ and $(a, b) \in F_p$. The public key is taken as a point on the EC and the private key is a random number. The public key is obtained by multiplying the private key with a generator point or base point G on the curve. The generator point $G \in F_p$ and point O is known as point at infinity, where $E(F_p)$ is the set of all points on E and the subgroup generated by G has a large order q. The various mathematical operations defined in ECC are:

- *Addition of points*: Let $M1$ and $M2$ be two points on EC then $N = M1 + M2$, where line joining the points $M1$ & $M2$ intersects the EC at eN and the reflection of the iN with respects to the X axis is N.
- *Subtraction of Points*: If M2 = -M1, then $M1 + M2 = M1 - M2 = 0$ and the line that joins $M1$ & $M2$ intersects the curve at O.
- *Doubling*: The point doubling on EC can be defined as the addition of point $M1$ to itself, i.e., $2.M1 = M1 + M1$.
- *Scalar Point Multiplication*: Let a scalar $j \in Zq*$, the scalar point multiplication on the EC can be defined as $j.M1 = M1 + M1 + \ldots.. + M1(jtimes)$.

For the given points $M1$ & $M2 = a.M1$ on E, where $a \in Z_q*$, it is computationally intractable to find which is known EC discrete logarithm problem (ECDLP).

Schnorr's signature

Let two large prime numbers are p and q such that, $p \in Z_p$ with order q and the one-way hash function, H: $Z_q \times Z$ in range $[0-2q-1]$. To sign a message m with the private key p, where elliptic curve E, defined over finite field (F_p)

Table 1. Comparison of the proposed protocol with the state-of-the-art approaches.

Security features	Imperson attack resistance	Phishing attack resistance	Providing perfect secrecy	Replay attack resistance	ID and password leakage proof	Low computation cost	Reflection attack resistance	DOS attack resistance
Abbasinezhad-Mood and Nikooghadam (2018)	✓	✓	✓	✓	–	✓	–	–
Karati et al. (2018)	✓	✓	✓	✓	✓	✓	–	–
Mahmood et al. (2016)	✓	✓	–	✓	–	–	–	–
Mahmood et al. (2018)	✓	✓	–	✓	–	✓	–	–
Proposed protocol	✓	✓	✓	✓	✓	✓	✓	✓

such that, $y^2 = x^3 + ax + b$ where $a, b \in F_p, 4a^3 + 27b^2(modp) \neq 0$, $E(Fp)$ is set of all points on curve E. In this propose work, private key is p, public key is y, $y = H(c^p)$, where c is secret identity number. Let $e = Hash(y, t1)$, to sign message m, masked identity $mid_a = Mult\ (aa, e)$, where aa=Hash(ID, $password$, y). The signature is the pair (aa, e) and the elements t, aa are used to verify the signature with message m. The verifier must compute $e = Hash(t, y)$, followed by computation of $xmid_a$ = Mult(aa, e) and thereafter check whether $xmid_a = mid_a$ holds to verify the signature.

CMAP

Consider an organization leverages services of multiple cloud servers where a Trusted Authentication (TA) system is deployed at middleware connected to multiple cloud server. This middleware interface (acting as a software broker or agent) is deployed at trusted centralized server. The TA system integrates cryptography and machine learning capabilities to ensure most secure session set-up among communicating entities. A machine learning based voting classifier is employed for detection and mitigation of any threat or anomaly during session set-up phase as depicted in Figure 3 and the mutual key agreement phase is shown in Figure 4 that utilizes two cryptographic algorithms viz. ECC and Schnorr's signature. Each component is described in detail in the following subsections.

Voting classifier based anomaly detector

Figure 3 shows TA server embedded with voting classifier which is an ensemble machine learning model that trains multiple classifier models in parallel to predict an output (class) with maximum accuracy. There are n classifier models such as $Cl_1, Cl_2, \ldots, Cl_n$ which generates predicted output such as $P_1, P_2, \ldots, P_n$. The historical information from repository as well as live data of session establishment phase (or key agreement) continuously trains the classifiers. This model combines the outcomes of each classifier and predicts the output class based on the highest majority of voting. We have utilized Decision Tree Classifier, K Nearest Neighbors Classifier, Random Forest Classifier and Logistic Regression models and passed their outcomes into voting engine to predict the final output (P_{final}). The predicted output classifies the given input either 'malware' (anomaly detected) or 'benign' (no anomaly detected). When TA server receives session set-up request from an entity (user/data owner), it feeds the key parameters and all the attributes (received from the sender) into voting classifier to predict the probability of any malicious intention behind session set-up. The session establishment is permitted only if the threat probability is zero.

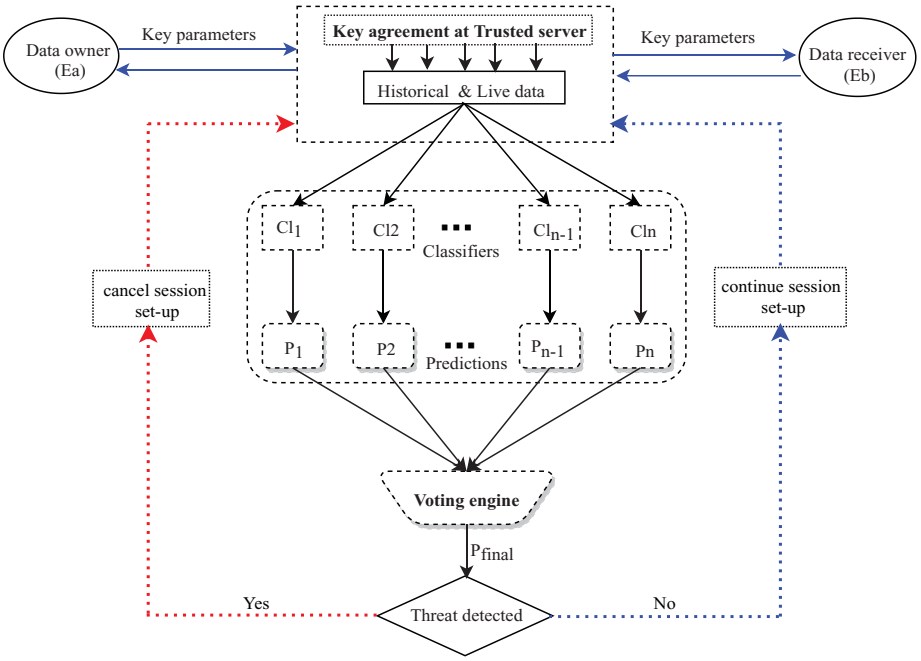


Figure 3. Voting classifier based anomaly detector embedded trusted authentication server system.

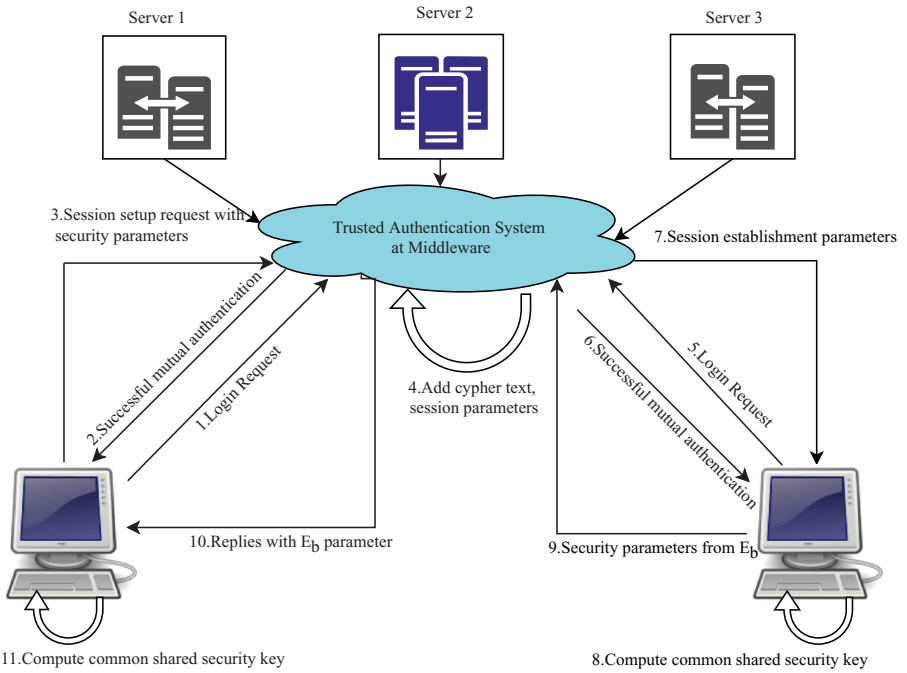


Figure 4. Session set-up and mutual key agreement.

The embedding of voting classifier into trusted authentication system allows multi-factor authentication as it is able to detect *network reputation anomalies* like, Honey pot, IP reputation, Tor network, Blacklisted or New IP addresses etc., *geographic location anomalies* including New or blacklisted location, *device fingerprinting anomalies* such as new device, new or infrequently used operating system log in and *time anomalies* for example, unusual time of session request. According to the detected anomaly, various security actions could be undertaken like one time password, asking security questions, restrict the access level or grant full access to the information.

Mutual key agreement

Figure 4 shows the proposed model for mutual authentication and secure common session key (CSK) computation, which is operable at middleware in the federated cloud server environment. Here, the data owner (E_a) and data receiver (E_b) are entities and TA is Trusted Authentication System deployed at the middleware under federation of multiple cloud servers. Organization has stored all its employee details in the repository at TA.

The complete authentication phase and session key agreements steps are as follows:

1. E_a sends log in request along with some secret identity parameter to TA, which verifies the identity of E_a with the entry of the repository of valid entities and replies with one time password (OTP) after successful authentication of E_a.
2. E_a sends session setup request after entering OTP and usb key log in and session initiation parameters (i.e., secret identities of sender and receiver).
3. TA receives the session establishment parameters such as secret identities of E_a and E_b and nonce etc. TA then prompts voting classifier (discussed in next subsection) for detection of any anomaly or threat. If no threat is detected, it applies modified Schnorr's signature and calculates masked ID's for both the sender and receiver to proceed the session setup, otherwise, cancels the session.
4. Receiver E_b also login at TA using same procedure. TA sends masked ID's and some security parameters to E_b.
5. Then E_b does some computation work and verifies TA as authentic server and calculates common shared session keys (*CSK*) at its device safely. E_b replies TA with some newly created nonce value but not the *CSK* as the response.
6. TA then verifies E_b after doing some computation work and forwards, E_b's response and some other security parameters to E_a.

7. E_a verifies TA and E_b after doing some computation work and accepts the response to calculate its common shared session key *CSK*.

This is how authentication phase is completed and secure session is setup between E_a and E_b via TA and *CSK* is computed successfully without sharing it on the public channel.

Phases of mutual authentication

The proposed authentication scheme works in three phases:

1. Registration phase
2. Login phase
3. Password renewal phase

Registration phase

The entity chooses password (*Pwd*) and submits his unique employee (*ID*) with officially registered mobile number, mail-ID, usb finger print as an usb-key login to the trusted authentication system TA via secure channel. Here, TA calculates secret number for each registered client as follows:

$$c = H(ID||Pwd||PhoneNumber||p||UsbLogin||Email).$$

where p is private key of TA. Then, TA selects large prime number p, and elliptic curve E over finite field F_p as a, b: $y^2 = x^3 + ax + b$ where $a, b \in Fp$ and $4a^3 + 27b^2(modp) \neq 0, E(F_p)$ is set of all points on curve E. Then, TA calculates public key y from private key p as follows: $y = H(c^p)$ and completes the registration by sending public key y and secret number for that entity c to E_a. Moreover, *TA* stores p, c and *UsbLogin* key for every registration in repository for future verification.

Login and authentication phase

The Figure 5 is showing mutual authentication phase (or login phase).

1. E_a sends login request with secret number ($c\prime$). TA receives it and verifies if $c' \neq$ stored c in the repository, and reject the session. Otherwise, it sends OTP (valid for 5 minutes only) to E_a.
2. E_a enters OTP and usb key, and TA confirms mutual authentication process. E_a, then encrypts sender's ID (E_a) and receivers ID (E_b) as follows:
3. $aa = Hash1(ID_a, t1, y)$
4. $ab = Hash1(ID_b, t1, y)$

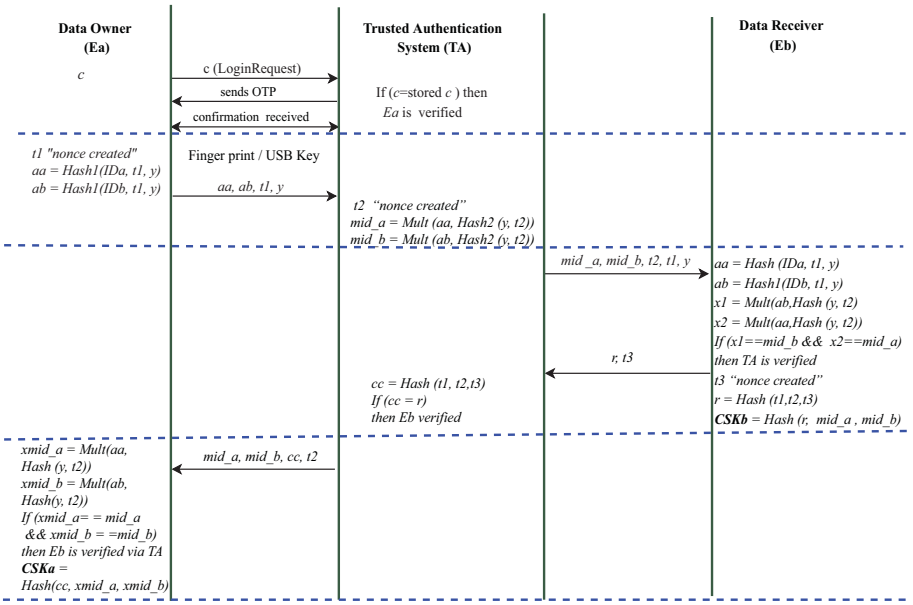


Figure 5. Mutual Authentication phase.

5. E_a then sends aa, ab and nonce $t1$ (freshly created time stamp that can never be repeated) to TA.
6. TA then calculates masked identities for E_a and E_b to setup current session, by applying Schnorr's signature as follows:
7. $mid\_a = Mult(aa, Hash2(y, xt1))$
8. $mid\_b = Mult(ab, Hash2(y, t2))$
9. TA sends mid_a, mid_b, $t1$, $t2$, aa, ab to E_b. Then, E_b applies modified Schnorr's signature verifying scheme to compute $x1$ and $x2$ as follows:
10. $x1 = Mult(ab, Hash(y, t2)$
11. $x2 = Mult(aa, Hash(y, t1))$
12. If ($x1 \neq mid\_b$ and $x2 \neq mid\_a$) then terminate the session, otherwise, TA is verified as trusted server and E_b generates new nonce value $t3$ and computes hashed nonce value r and common shared session key CSK.
13. $r = Hash(Hash(t1, t2), t3)$
14. $csk = Hash(r, mid\_a, mid\_b)$
15. E_b replies TA with security parameters $r$and $t3$ TA receives r and new nonce $t3$, TA computes cc to verify E_b as follows:
16. $cc = Hash(Hash(t1, t2), t3)$
17. Using its previously known nonce value (i.e., $t1$ and $t2$) and newly received nonce $t3$. If $cc \neq r$ then E_b is not a valid user and reject the session, otherwise, sends r and $t2$ (nonce created by TA) to E_b.
18. E_a receives masked ID (mid_a and mid_b) from TA and verifies TA and E_b as follows:
19. $xmid\_a = Mult(aa, Hash(y, t1))$

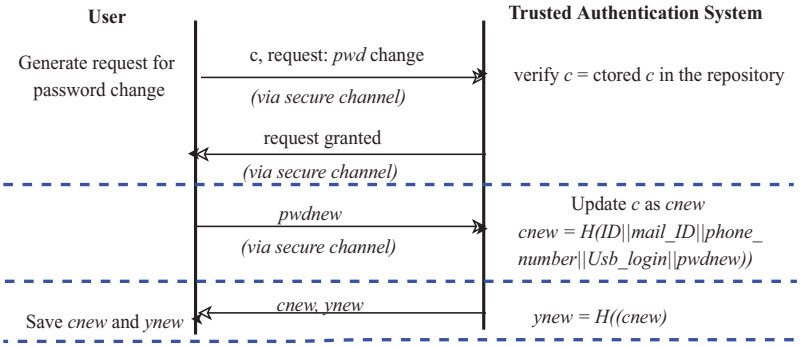


Figure 6. Password renewal phase.

20. $xmid\_b = Mult(ab, Hash(y, t2))$
21. If $xmid\_a \neq mid\_a$ and $xmid\_b \neq mid\_b$ then terminate the session, otherwise, calculate common shared session key (CSK) as follows:
22. $csk = Hash(cc, mid\_a, mid\_b)$

Password renewal phase

It is mandatory for every client to renew his password atleast once in a specific time period like, a week, after two weeks or month etc. But password can be changed anytime at the request of user. Since secret number c depends on password Pwd along with ID, $mail_ID$ etc., if the Pwd changes, c, automatically changes, hence, secret number is also renewed every week or as per the choice of user. Here, Figure 6 depicts password change phase.

Security analysis

Theorem 6.1. *The common session key is computed by E_a is identical to the common session key derived by E_b.*

Proof.

$$\begin{aligned} CSK_b &= r + mid\_a + mid\_b = Hash(t1 + t2 + t3) + x1 + x2 \\ &= Hash(t1 + t2 + t3) + Mult(aa; Hash(y + t2)) + Mult(ab; Hash(y + t2)) \\ &= Hash(t1 + t2 + t3) + (Hash(ID_a + t1 + y) : Hash(y + t2)) \\ &\quad + (Hash(ID_b + t1 + y) : Hash(y + t2)) \\ &= cc + Mult(aa; Hash(y + t2)) + Mult(ab; Hash(y + t2)) \\ &= cc + xmid\_a + xmid\_b \\ &= r + mid\_a + mid\_b \\ &= CSK_a \text{ because } cc == r \text{ as verified by TA.} \end{aligned}$$

□

Theorem 6.2. *The proposed protocol is semantically secure against online data sharing security threats including insider attack, ID and password leakage attack, client impersonation attack, third party impersonation attack, session key computation attack, replay attack, forward secrecy attack, man-in-the-middle attack, user's anonymity attack and denial-of-service attack.*

Proof. The proposed mutual authentication scheme resists above security attacks as follows:

1. **Secure against insider attack**: The registration phase is completely secure due to usage of more sophisticated credentials input like mobile number, *ID*, *UsbLogin* along with email and password. During every session setup, user enters unique secret number *c* as a substitute for all the credentials, during each session login, without this secret number *c* and *UsbLogin* no other insider person can get access to the authenticated user account.
2. **Secure against ID and password leakage**: The proposed protocol is perfectly secure against both ID and password leakage, as both ID and password are secure under encrypted hash function by the user in the form of a secret bit string, $aa = Hash1(ID, pwd, P)$. Therefore, ID and password are never shared openly during session setup. Hence, safe from ID and password leakage.
3. **Client impersonation attack**: In this type of attack, attacker forges a valid login request and pretend as legitimate user, using previously eaves dropped messages but in proposed protocol, attacker cannot compute *aa*, *ab*, *c* and *UsbLogin* simultaneously. Moreover, if attacker anyhow obtains user ID, mail ID, phone number and extracts secret bit string c, he or she cannot get *UsbLogin* and calculate *aa* & *ab* which are protected under one way hash function. Hence, the proposed protocol is completely free from client impersonation attack.
4. **Leak of verifier attack**: Attacker cannot forge TA system during login verification as attacker cannot enter secret number *c* and *UsbLogin* which is used for verification and the secret string is also protected by hash function. There occurs proper mutual authentication at each step of session, to compromise each step of verification is impossible. Since, the valid users identity is protected under bit string *aa* & *ab* and further protected by *mid_a* and *mid_b* by the trusted authentication system, the exact identity can never get reveal. Hence, the attacker cannot qualify for the stage of mutual authentication verification.
5. **Fake registration attack**: Attacker cannot register himself as a valid user without having employee user ID, *UsbLogin* (allotted by the organization) and registered phone number (officially recorded by organization). Therefore, any outsider attacker can never get access and register himself as valid user in TA system.

6. **Users anonymity attack**: In login phase, user submits secretely secured ID in the form of $aa = Hash1(ID_a, t1, y)$ & $ab = Hash2(ID_b, t1, y)$ as substitute for real identities. Moreover, authentication and session key agreement of proposed protocol is based on the masked identities like $mid\_a = Mult(aa, Hash2(y, t2))$ & $mid\_b = Mult(ab, Hash2(y, t2))$ which are dynamic identities and freshly created during each session setup, the attacker can never distinguish various different sessions corresponding to a specific user. Therefore, our proposed protocol is completely free from user's anonymity attack.
7. **Session key computation attack**: In our proposed protocol, common shared security key (*CSK*) is calculated by data owner and data receiver after complete mutual authentication and verification separately at their own place, but it is never shared on transmission channel. Here, *CSK* is composed of mid_a, mid_b, $t1$, $t2$ & $t3$, all of which are dynamically and uniquely created during each session setup and can never be reused. Hence, it is completely resilient to session key computation attack.
8. **Replay attack resilience**: In replay attack, adversary eaves drops a valid message by tracking some previous session and then resends it with optimum caution in order to retrieve confidential information from the authenticated server. However, in the proposed scheme when attacker resends any previous message TA system generates fresh nonce $t2$ and compute:
9. $t = Hash(t1, t2)$ and also masked ID, that are created using nonce $t1$ & $t2$. Thereafter, this message will be discarded by the TA as the nonce value can never be repeated.
10. **Perfect forward secrecy**: Common shared secured key is freshly prepared for each session which can never be reused again. Moreover, the attacker can never verify the log in phase because of the perfectly secure log in session. Hence, the proposed protocol withstand perfect forward secrecy attack.
11. **Resist man-in-the-middle attack**: In the proposed protocol scheme, proper mutual authentication occurs at each step among data owner(E_a) and data receiver (E_b) entity. If any attacker in the middle modifies any value in the message, it is discarded and the session is terminated, because of the failure of mutual verification, that is mandatory at each successive step of the protocol.
12. **Denial-of-service**: The voting classifier employed at the trusted authentication server verifies the session set-up request coming from legitimate user and is capable of identifying any anomalous network traffic. Henceforth, any attempt for DOS or distributed DOS is captured at initial state and prevented.

□

Algorithm and formal verification of proposed mutual authentication protocol

The operational summary of CMAP is depicted in Algorithm 1, where step 1 initializes entities E_a, TA and E_b, key parameters required for session establishment. Step 2 transfers parameters from E_a to TA, which calls Classifier (i.e., Algorithm 2) in step 3 to predict the status of threat. The Classifier calls different machine learning based predictors including, SVM, KNN, Decision Tree, Random Forest etc, denoted as $\{P_1, P_2, \ldots, P_n\}$. Steps 4–9 govern the session authentication and set-up phase based on the outcome of the estimated threat status.

1 Initialize user-profiles of E_a and E_b and key parameters $\{ID_a, ID_b, aa, ab, csk, mid\_a, mid\_b, y, c\}$ for mutual authentication set-up;
2 E_a sends session set-up parameters $\{k_1, k_2, ..., k_N\}$ to TA ;
3 $Threat_{status} \Leftarrow$ Classifier$(k_1, k_2, ..., k_N)$;
4 **if** $(Threat_{status} = 1)$ **then**
5 | terminate session set-up;
6 **else**
7 | E_b receives authenticated parameters $\{k^*_1, k^*_2, ..., k^*_N\}$ from TA;
8 | The session set-up completes by exchanging required parameters among E_a, E_b and TA and generating common session keys CSK_a and CSK_b using steps mentioned in section 5.3.2.;
9 **end**

1 **for** *each* $C_i : i \in \{1, 2, ..., n\}$ **do**
2 | $P_i \Leftarrow C_i(k_1, k_2, ..., k_N)$;
3 **end**
4 $P_{final} \Leftarrow$ Voting engine $(P_1, P_2, ..., P_n)$;
5 **return** P_{final};

The authentication phase of the protocol is formally verified through implementation, on ProVerif security analysis tool. The ProVerif is an automatic security protocol verifier which is based on representation of cryptographic protocol using horn clauses. It can handle different cryptographic primitives like shared and public key, hash function specified as equations. It also deals with unbounded number of sessions in parallel and it is able to prove secrecy (attacker cannot obtain secret), proper authentication etc.

To analyze the security of different security primitives used in the protocol, queries are framed. If the query is satisfied and returns true it means the respective security primitive is secure and (cannot be attacked).If certain security element is not safe, the query returns false and the ProVerif can reconstruct the executing trace of protocol that falsifies the desired property (steps of attacks can be traced). The implementation of proposed mutual authentication scheme is given in Appendix A. To formally verify the safety of ephemeral secret values like private keys, public keys, common

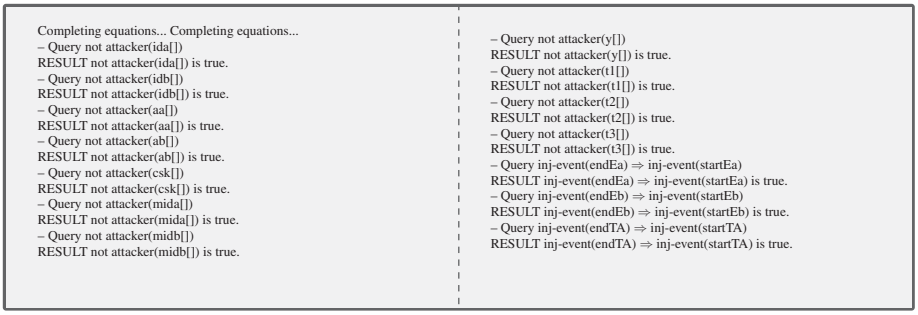


Figure 7. Results obtained on ProVerif tool.

shared keys, id and passwords, queries are executed and corresponding results are shown in Figure 7. The phrase 'RESULT not attacker' indicates that all the variables used during mutual authentication, including *IDa*, *IDb*, *aa*, *ab*, *CSK*, mid_a, mid_b, y, nonce values: $\{t1, t2, t3\}$ along with the generated queries and events are safe from security breaches. Hence, the results obtained verify the complete security of CMAP.

Performance analysis

The simulation experiments are executed on a server machine assembled with two Intel® Xeon® Silver 4114 CPU with 40 core processor and 2.20 GHz clock speed. The computation machine is deployed with 64-bit Ubuntu 16.04 LTS, having main memory of 128 GB. The proposed work is implemented in Python 3.7. To validate the performance of voting classifier, we have utilized benchmark dataset from Canadian Institute for Cybersecurity datasets (CICD) (Arash & Kadir, 2017) which provides wide range of datasets for research on cybersecurity. This dataset includes 635 records each of 8115 attributes, overall there were 160 malware, 473 benign and rest binary or undetermined class of http requests. Figure 8 shows the count of all three class types obtained from CICD network malware dataset.

Figures 9 and 10 show the comparison of different confusion matrices and classification accuracy, respectively, achieved for various classifiers where, SVM classifier shows least performance among all. The Logistic Regression (LR) and K-Nearest Neighbor (KNN) achieved good accuracy over SVM. The Decision Tree (DT) outperforms SVM, KNN and LR by 1–1.5% in terms of accuracy of malware prediction. The Random Forest or Bagging Classifier (BC) which is an ensemble of number of similar decision trees shows equivalent accuracy up to 99.61%, as given by Voting Classifier which combines the classification accuracy of LR and KNN classifiers.

The performance of the proposed mutual authentication protocol is evaluated in terms of communication cost with respect to time and space

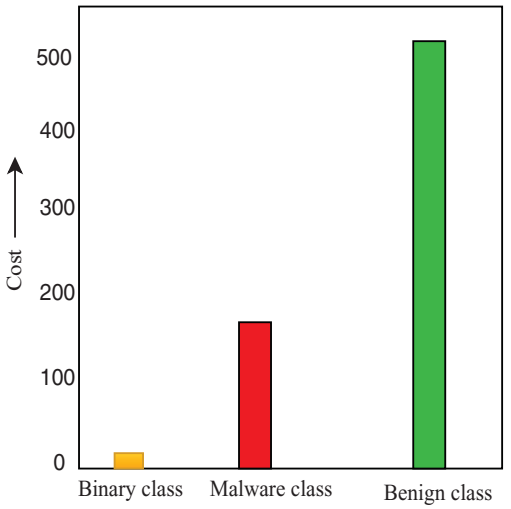
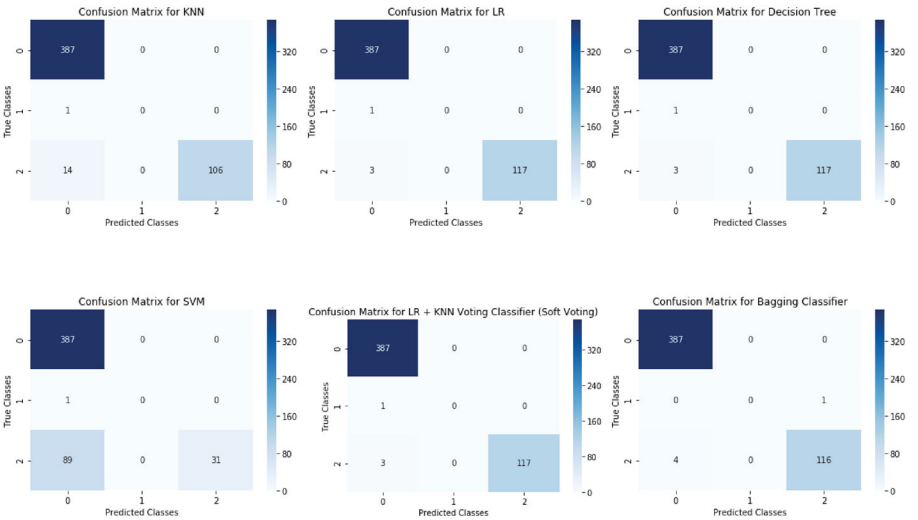


Figure 8. Malware, Benign and other class-type counter plot for CICD network malware dataset.



(d) Support Vector Machine (e) Voting Classifier (f) Bagging Classifier

Figure 9. Confusion matrices obtained for various classifiers. Malware, Benign and other class-type counter plot for CICD network malware dataset. (a) KNeighbors. (b) Logistic Regression. (c) Decision Tree. (d) Support Vector Machine. (e) Voting Classifier. (f) Bagging Classifier.

consumption during key agreement. Additionally, time and space operational cost of different ECC curves are also compared to select the most suitable one to develop an effective mutual authentication protocol.

Computation cost

Computational time complexity of the proposed scheme and analogous schemes are calculated by identifying the primitive operations and their

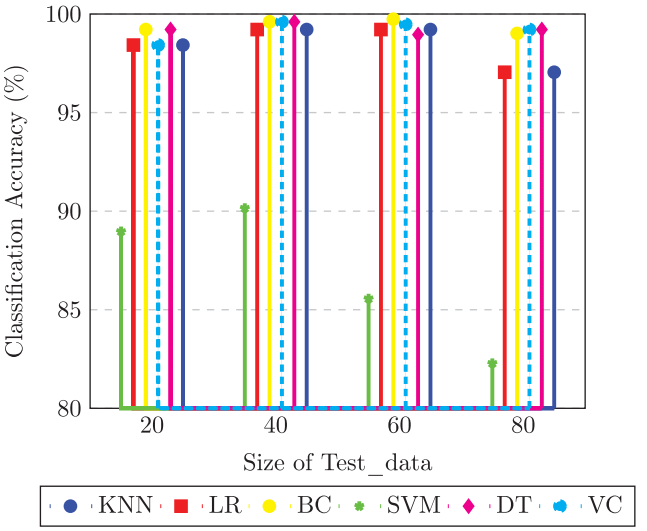


Figure 10. Comparison of accuracy for different classifiers.

Table 2. Comparison of computational cost of proposed against existing authentication schemes.

Key agreement authentication schemes	Operations involved	Computational cost
Mahmood et al. (2016)	$6\,Exp + 4Encr/Decr + 2H + 2HMAC$	23.2
Mahmood et al. (2018)	$10Mul + 4Add + 8H$	22.4
Abbasinezhad-Mood and Nikooghadam (2018)	$8H + 4Add + 8H$	17.9
Karati et al. (2018)	$16H + 10Add + Bi + Exp + Encr/Decr + 4R$	30.09
Propose protocol	$14H + 16Mul$	13.4

frequencies in the respective schemes. Table 2 depicts comparative computational cost of the various authentication schemes in terms of summing up the frequency of execution time of various primitive operations. Here, the time of execution of each primitive operation is taken as arithmetic mean of thousand executions of each primitive operation (excluding standard deviation) as presented in Kilinc and Yanik (2014). The various time notations involved in presenting comparison are listed as under:

- *H*-time incurred during one-way hash. (0.0023 ms)
- *M*ul-time incurred during ECC scalar point multiplication. (2.223 ms)
- *A*dd-time incurred during ECC scalar point addition. (0.0288 ms)
- *E*xp-time incurred during exponentiation. (3.85 ms)
- *B*i-time incurred during bilinear pairing. (0.5811 ms)
- *E*ncr-time incurred during Encryption. (3.85 ms)
- *D*ecr-time incurred during Decryption. (3.85 ms)
- *HMAC*-time incurred during Hash based message authentication code operation. (0.0046 ms)
- *R*-time incurred in random number generation. (0.539 ms)

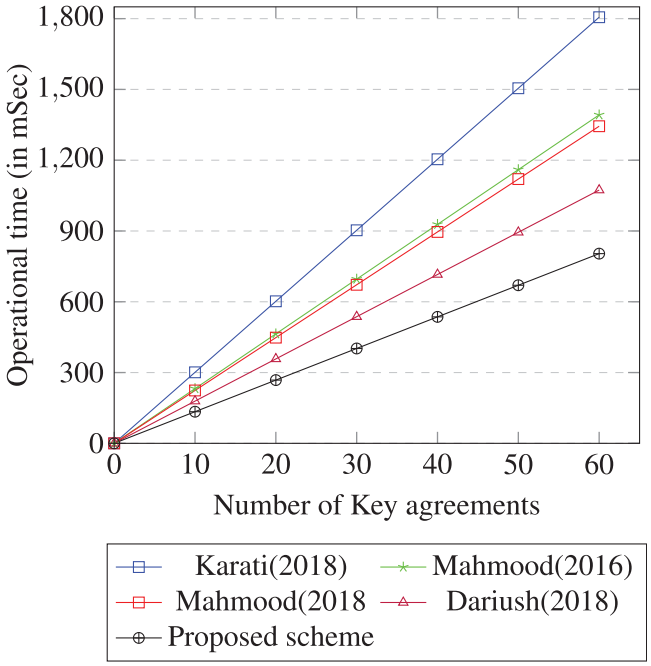


Figure 11. Graph showing operational time comparison proposed scheme v/s other existing schemes.

The evaluation results shown in Table 2 concludes that proposed scheme takes 13.4 msec for single key agreement, which is least among all the analogous authentication protocols. This is due to the simplicity and discard of expensive primitive operations used in existing mutual authentication schemes (Abbasinezhad-Mood & Nikooghadam, 2018; Karati et al., 2018; Mahmood et al., 2016; 2018) during key agreement in the proposed scheme. Hence, the proposed protocol implies required and optimum level of security using lighter weight primitive operations.

Figures 11 and 12 show comparative time and space computational cost of numerous key agreements of proposed and existing protocols, which analyses that proposed scheme has least space and time complexity among all.

The proposed protocol provides perfect forward secrecy, low communication cost because of appropriate usage of ECC algorithm and Schnorr's signature during registration and login phase. Most of the existing authentication protocol do not support all the security features together, for instance, Abbasinezhad-Mood & Nikooghadam (2018) and Mahmood et al. (2016, 2018) lack ID and password leakage proof, Wu and Zhou (2011) do not provide any security analysis proof, Abbasinezhad-Mood & Nikooghadam (2018), Fouda et al. (2011), Karati et al. (2018), Mahmood et al. (2016, 2018), and Sule et al. (2012) have comparatively high

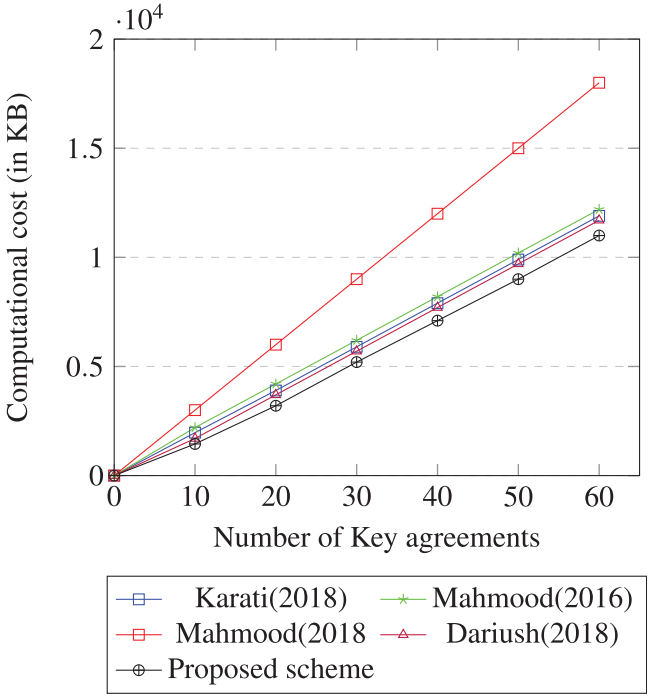


Figure 12. Graph showing computational cost (in KB) comparison proposed scheme v/s other existing schemes.

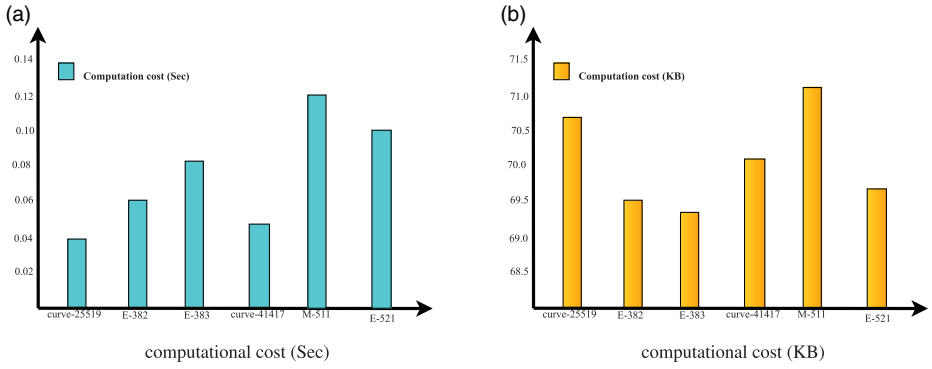


Figure 13. Comparison of computational cost of different ECC curves. (a) Computational cost (sec). (b) Computational cost (KB).

communication cost. Moreover, the performance efficiency of proposed protocol is presented in terms of primitive metrics like computation cost (in sec).

Figure 13 shows the comparative analysis of various elliptic cryptography curves including Curve-25519, E-382, E-383, Curve-41417, M-511 and E-521, in terms of computation cost in seconds and space consumption in kilobytes (KB) while converting private key into public key.

Conclusion

A novel mutual authentication protocol based on cryptography and machine learning is proposed for secure data sharing. This protocol can resist security attacks including user anonymity attack, middle-man-attack, reflection attack, replay attack, Denial-of-service and many more, because user's real identity and common shared session keys are never shared directly on the public network. Moreover, security analysis of this mutual authentication scheme has been done using ProVerif security analysis tool. The results verify that proposed protocol is lightweight in terms of computation cost and safe from many possible security attacks occur during online data sharing under multi-cloud environment.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Disclosure statement

The authors declare that they have no conflict of interest.

Funding

This research work is financially supported by National Institute of Technology Kurukshetra, India.

ORCID

Ashutosh Kumar Singh http://orcid.org/0000-0002-8053-5050
Deepika Saxena http://orcid.org/0000-0002-9689-6387

References

Abbasinezhad-Mood, D., & Nikooghadam, M. (2018). Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications. *Future Generation Computer Systems*, *84*, 47–57. https://doi.org/10.1016/j.future.2018.02.034

Arash, H. A., & Kadir, F. H. A. (2017). *Towards a network-based framework for android malware detection and characterization* [Paper presentation]. The proceeding of the 15th International Conference on Privacy, Security and Trust. PST.

Armerding, T. (2018). *The 18 biggest data breaches of the 21st century*. https://www.csoonline.com

Balaji, N. A., Sukumar, R., & Parvathy, M. (2019). Enhanced dual authentication and key management scheme for data authentication in vehicular ad hoc network. *Computers & Electrical Engineering*, *76*, 94–110. https://doi.org/10.1016/j.compeleceng.2019.03.007

Bilogrevic, I., Huguenin, K., Agir, B., Jadliwala, M., Gazaki, M., & Hubaux, J.-P. (2016). A machine-learning based approach to privacy-aware information-sharing in mobile social networks. *Pervasive and Mobile Computing*, *25*, 125–142. https://doi.org/10.1016/j.pmcj.2015.01.006

Dodero, J. M., Rodriguez-Garcia, M., Ruiz-Rube, I., & Palomo-Duarte, M. (2019). Privacy-preserving reengineering of model-view-controller application architectures using linked data. *Journal of Web Engineering*, *18*(7), 695–728. https://doi.org/10.13052/jwe1540-9589.1875

Fan, C-I., Chan, Y-C., & Zhang, Z-K. (2005). Robust remote authentication scheme with smart cards. *Computers & Security*, *24*(8), 619–628.

Fouda, M. M., Fadlullah, Z. M., & Kato, N. (2011). A lightweight message authentication scheme for smart grid communications. *IEEE Transactions on Smart Grid*, *2*(4), 675–685. https://doi.org/10.1109/TSG.2011.2160661

Guo, C., Luo, N., Bhuiyan, M. Z. A., Jie, Y., Chen, Y., Feng, B., & Alam, M. (2018). Key-aggregate authentication cryptosystem for data sharing in dynamic cloud storage. *Future Generation Computer Systems*, *84*, 190–199. https://doi.org/10.1016/j.future.2017.07.038

Hwang, M-S., Chong, S-K., & Chen, T-Y. (2010). Dos-resistant ID-based password authentication scheme using smart cards. *Journal of Systems and Software*, *83*(1), 163–172.

Hwang, M-S., & Li, L-H. (2000). A new remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *46*(1), 28–30.

Kalra, S., & Sood, S. K. (2015). Secure authentication scheme for IoT and cloud servers. *Pervasive and Mobile Computing*, *24*, 210–223. https://doi.org/10.1016/j.pmcj.2015.08.001

Karati, A., Amin, R., & Islam, S. H. (2018). Provably secure and lightweight identity-based authenticated data sharing protocol for cyber-physical cloud environment. *IEEE Transactions on Cloud Computing*.

Kilinc, H. H., & Yanik, T. (2014). A survey of sip authentication and key agreement schemes. *IEEE Communications Surveys & Tutorials*, *16*(2), 1005–1023. https://doi.org/10.1109/SURV.2013.091513.00050

Krombholz, K., Hobel, H., Huber, M., & Weippl, E. (2015). Advanced social engineering attacks. *Journal of Information Security and Applications*, *22*, 113–122. https://doi.org/10.1016/j.jisa.2014.09.005

Kumar, V., Ahmad, M., & Kumari, A. (2019). A secure elliptic curve cryptography based mutual authentication protocol for cloud-assisted TMIS. *Telematics and Informatics*, *38*, 100–117. https://doi.org/10.1016/j.tele.2018.09.001

Kumari, A., Jangirala, S., Abbasi, M. Y., Kumar, V., & Alam, M. (2020). Eseap: ECC based secure and efficient mutual authentication protocol using smart card. *Journal of Information Security and Applications*, *51*, 102443. https://doi.org/10.1016/j.jisa.2019.102443

Li, H., Yang, C., & Liu, J. (2019). A novel security media cloud framework. *Computers & Electrical Engineering*, *74*, 605–615. https://doi.org/10.1016/j.compeleceng.2018.07.022

Li, J., Sun, L., Yan, Q., Li, Z., Srisa-An, W., & Ye, H. (2018). Significant permission identification for machine-learning-based android malware detection. *IEEE Transactions on Industrial Informatics*, *14*(7), 3216–3225. https://doi.org/10.1109/TII.2017.2789219

Li, W., Xuelian, L., & Gao, J. (2019). Design of secure authenticated key management protocol for cloud computing environments. *IEEE Transactions on Dependable and Secure Computing*.

Li, X., Xiong, Y., Ma, J., & Wang, W. (2012). An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. *Journal of Network and Computer Applications*, *35*(2), 763–769. https://doi.org/10.1016/j.jnca.2011.11.009

Lim, S. Y., Kiah, M. M., & Ang, T. F. (2017). Security issues and future challenges of cloud service authentication. *Acta Polytechnica Hungarica*, *14*(2), 69–89.

Liu, Z. P., Wu, F. L., Shang, K. Y., & Chai, W. L. (2013). C-mas: The cloud mutual authentication scheme. *Advanced Materials Research*, *756–759*, 3209–3214. https://doi.org/10.4028/www.scientific.net/AMR.756-759.3209

Ma, Z., Liu, Y., & Wang, Z. (2018). A machine learning-based scheme for the security analysis of authentication and key agreement protocols. *Neural Computing and Applications*, 1–13.

Mahmood, K., Ashraf Chaudhry, S., Naqvi, H., Shon, T., & Farooq Ahmad, H. (2016). A lightweight message authentication scheme for smart grid communications in power sector. *Computers & Electrical Engineering*, *52*, 114–124. https://doi.org/10.1016/j.compeleceng.2016.02.017

Mahmood, K., Chaudhry, S. A., Naqvi, H., Kumari, S., Li, X., & Sangaiah, A. K. (2018). An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. *Future Generation Computer Systems*, *81*, 557–565. https://doi.org/10.1016/j.future.2017.05.002

Melki, R., Noura, H. N., & Chehab, A. (2020). Lightweight multi-factor mutual authentication protocol for IoT devices. *International Journal of Information Security*, *19*(6), 679–616. https://doi.org/10.1007/s10207-019-00484-5

Nimmy, K., & Sethumadhavan, M. (2014). *Novel mutual authentication protocol for cloud computing using secret sharing and steganography* [Paper presentation]. ICADIWT (pp. 101–106). IEEE.

Rijvordt, W., Hogenboom, F., & Frasincar, F. (2019). Ontology-driven news classification with aethalides. *Journal of Web Engineering*, *18*(7), 627–654. https://doi.org/10.13052/jwe1540-9589.1873

Saxena, D., & Saxena, S. (2015). *Highly advanced cloudlet scheduling algorithm based on particle swarm optimization* [Paper presentation]. 2015 Eighth International Conference on Contemporary Computing (IC3) (pp. 111–116). IEEE. https://doi.org/10.1109/IC3.2015.7346663

Saxena, D., & Singh, A. K. (2020a). Energy aware resource efficient-(eare) server consolidation framework for cloud datacenter. In D. Saxena & A. K. Singh (Eds.), *Advances in communication and computational technology* (pp. 1455–1464). Springer.

Saxena, D., & Singh, A. K. (2020b). Security embedded dynamic resource allocation model for cloud data centre. *Electronics Letters*, *56*(20), 1062–1065. https://doi.org/10.1049/el.2020.1736

Saxena, D., & Singh, A. K. (2020c). Auto-adaptive learning-based workload forecasting in dynamic cloud environment. *International Journal of Computers and Applications*, 1–11. https://doi.org/10.1080/1206212X.2020.1830245

Saxena, D., & Singh, A. K. (2020d). A proactive autoscaling and energy-efficient vm allocation framework using online multi-resource neural network for cloud data center. *Neurocomputing*, *426*, 248–264. https://doi.org/10.1016/j.neucom.2020.08.076

Saxena, D., Chauhan, R., & Kait, R. (2016). Dynamic fair priority optimization task scheduling algorithm in cloud computing: Concepts and implementations. *International Journal of Computer Network and Information Security*, *8*(2), 41–48. https://doi.org/10.5815/ijcnis.2016.02.05

Saxena, D., Vaisla, K. S., Rauthan, M. S. (2018). *Abstract model of trusted and secure middleware framework for multi-cloud environment* [Paper presentation]. International Conference on Advanced Informatics for Computing Research (pp. 469–479). Springer.

Saxena, S., & Saxena, D. (2015). *Ewsa: An enriched workflow scheduling algorithm in cloud computing* [Paper presentation]. 2015 International Conference on Computing, Communication and Security (ICCCS) (pp. 1–5). IEEE. https://doi.org/10.1109/CCCS.2015.7374202

Sood, S-K., Sarje, A-K., & Singh, K.A. (2011). Secure dynamic identity based authentication protocol for multi-server architecture. *Journal of Network and Computer Applications*, *34*(2), 609–618.

Sreeram, I., & Vuppala, V. P. K. (2019). Http flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm. *Applied Computing and Informatics*, *15*(1), 59–66. https://doi.org/10.1016/j.aci.2017.10.003

Sule, R., Katti, R. S., & Kavasseri, R. G. (2012). *A variable length fast message authentication code for secure communication in smart grids* [Paper presentation]. 2012 IEEE Power and Energy Society General Meeting (pp. 1–6). IEEE.

Vincent, O. R., Folorunso, O., & Akinde, A. (2010). Improving e-payment security using elliptic curve cryptosystem. *Electronic Commerce Research*, *10*(1), 27–41. https://doi.org/10.1007/s10660-010-9047-z

Wu, D., & Zhou, C. (2011). Fault-tolerant and scalable key management for smart grid. *IEEE Transactions on Smart Grid*, *2*(2), 375–381. https://doi.org/10.1109/TSG.2011.2120634

Xia, J., & Wang, Y. (2012). Secure key distribution for the smart grid. *IEEE Transactions on Smart Grid*, *3*(3), 1437–1443. https://doi.org/10.1109/TSG.2012.2199141

Appendix A.

ProVerif security analysis implementation code

```
(*variables and function declaration*)
free c: channel.
type ID.
type key.
type nonce.
type point.
free ida: ID[private].
free idb: ID[private].
free y: point[private].
free aa: bitstring[private].
free ab: bitstring[private].
free mida: bitstring[private].
free midb: bitstring[private].
free t1: nonce[private].
free t2: nonce[private].
free t3: nonce[private].
free csk: key[private].
fun Hash1(ID,key,point): bitstring
fun Hash2(point, nonce): bitstring
fun Hash3(nonce,bitstring, bitstring):key
fun Hash4(nonce, nonce): nonce
fun Hash5(bitstring): bitstring
```

```
fun Mult(bitstring, bitstring): bitstring. equation forall xx: bitstring, yy: bitstring; Mult(xx,yy)= Mult(yy,xx)
(*Event and Query declaration*)
event endEa.
event startEb.
event endEb.
event startTA.
event endTA.
query attacker(ida).
query attacker(idb).
query attacker(aa).
query attacker(ab).
query attacker(csk).
query attacker(mida).
query attacker(midb).
query attacker(y).
query attacker(t1).
query attacker(t2).
query attacker(t3).
query inj-event(endEa) ⇒ inj-event(startEa).
query inj-event(endEb) ⇒ inj-event(startEb).
query inj-event(endTA) ⇒ inj-event(startTA).
(*TA*)
let pTA = event startTA;
in(c,(xida: bitstring, xidb: ID, xt1: nonce));
let mida = Mult(aa, Hash2(y,xt1))in
new t: nonce;
new t2: nonce;
let t = Hash4(xt1, t2)in
let midb = Mult(ab, Hash2(y,t2))in
out(c,(mida, midb, t2, xt1, Hash5(ab), Hash5(aa)));
in(c,(k: nonce, xt3: nonce));
new cc: nonce;
let cc = Hash4(Hash4(xt1, t2), xt3)in
if (k = cc) then
out(c,(k, t2));
out(c,(mida, midb));
0.
(*Ea*)
let pEa = new t1: nonce;
new pa: key;
let aa = Hash1(ida, pa, p)in
let ab = Hash1(idb, pa, p)in
event startEa;
out(c,(Hash5(aa), Hash5(ab), t1));
in(c, (xk: nonce, xxt2: nonce));
new xmida: bitstring;
new xmidb: bitstring;
let xmida = Mult(aa, Hash2(y, t1))in
```

```
let xmidb = Mult(ab, Hash2(y, xxt2))in
in(c,(mida: bitstring, midb: bitstring));
if ((mida = xmida) && (midb = xmidb)) then
let csk = Hash3(xk, mida, midb)in
event endEa else 0.
(*Eb*)
let pEb = event startEb;
in(c,(xmida: bitstring, xmidb: bitstring, xt2: nonce, xxt1: nonce,xab: bitstring, xxaa:
bitstring)); let x1 = Mult(xab, Hash2(y,xt2))in
if (x1 = xmidb) then
let x2 = Mult(xxaa, Hash2(y,xxt1))in
if (x2 = xmida) then
new t3: nonce;
new r: nonce;
let r = Hash4(Hash4(xxt1, xt2), t3)in
let csk = Hash3(r, xmida, xmidb)in
out(c,(r, t3)); event endEb else 0.
process ((!pEa)|||(!pTA)|||(!pEb))
```