Received: 20 April 2018 | Revised: 25 August 2018 | Accepted: 19 September 2018

DOI: 10.1002/dac.3839

RESEARCH ARTICLE

WILEY

Efficient and secure business model for content centric network using elliptic curve cryptography

Sharmistha Adhikari[1] | Sangram Ray[1] | Gosta P. Biswas[2] | Mohammad S. Obaidat[3,4,5]

[1]Department of Computer Science and Engineering, National Institute of Technology Sikkim, Sikkim 737139, India

[2]Department of Computer Science and Engineering, Indian Institute of Technology (ISM) Dhanbad, Dhanbad 826004, India

[3]Department of ECE, Nazarbayev University, Astana, Kazakhstan

[4]KASIT, University of Jordan, Amman, Jordan

[5]University of Science and Technology Beijing (USTB), Beijing, China

Correspondence
Sangram Ray, Department of Computer Science and Engineering, National Institute of Technology Sikkim, Sikkim-737139, India.
Email: sangram.ism@gmail.com

Summary

Initially, Internet has evolved as a resource sharing model where resources are identified by IP addresses. However, with rapid technological advancement, resources/hardware has become cheap and thus, the need of sharing hardware over Internet is reduced. Moreover, people are using Internet mainly for information exchange and hence, Internet has gradually shifted from resource sharing to information sharing model. To meet the recent growing demand of information exchange, Content Centric Network (CCN) is envisaged as a clean-slate future network architecture which is specially destined for smooth content distribution over Internet. In CCN, content is easily made available using network caching mechanism which is misaligned with the existing business policy of content providers/publishers in IP-based Internet. Hence, the transition from contemporary IP-based Internet to CCN demands attention for redesigning the business policy of the content publishers/providers. In this paper, we have proposed efficient and secure communication protocols for flexible CCN business model to protect the existing business policies of the content publisher while maintaining the salient CCN features like *in-network content caching* and *Interest packet aggregation*. To enhance the efficiency and security, the Elliptic Curve Cryptography (ECC) is used. The proposed ECC-based scheme is analyzed to show that it is resilient to relevant existing cryptographic attacks. The performance analysis in terms of less computation and communication overheads and increased efficiency is given. Moreover, a formal security verification of the proposed scheme is done using widely used AVISPA simulator and BAN logic that shows our scheme is well secured.

KEYWORDS

AVISPA, BAN logic, content centric network, elliptic curve cryptography, in-network caching, interest packet aggregation, mutual authentication, secure business model, session key negotiation

1 | INTRODUCTION

The transition from conventional IP-based Internet to the future Content Centric Network (CCN) paradigm requires redesigning of the existing business policy of the Internet. Before going into the details of CCN business model, the basic idea of current Internet, recent Internet usage scenario, basic issues of conventional IP-based Internet, evolution of CCN and need of a CCN business model are briefly discussed in the following subsections.

1.1 | Evolution of CCN

Contemporary IP-based Internet, a host centric network architecture, was initially designed to share network hardware resources where hardware/hosts are identified by IP addresses. Today, with rapid technological advancement, information is becoming more important than hardware and Internet is mainly used for sharing/exchanging information.[1] According to Cisco Visual Networking Index forecast, global IP traffic will nearly triple by 2020 to reach 194.4 EB per month.[1] Hence, the rate of information exchange is increasing day by day. Moreover, people value Internet for *what* information they get rather than from *where* it is available. Therefore, to meet the ever increasing need of information exchange, Content Centric Network (CCN) is envisaged as a clean-slate future Internet architecture to leverage the ease of information/content distribution.[2-5] In CCN, content gets more importance than the host which provides the content. Here, content is considered as an independently routable unit and is decoupled from its host address to decrease the complexity of point to point content sharing. Moreover, content is uniquely identified by the name of the content over the network and content packets are routed using its unique name. During the transmission, content packets can be cached by the intermediate CCN routers to enhance the easy availability of content as well as to reduce the content response time. In CCN, security is given separately on the piece of content rather than securing the container of the content or the communication between two hosts. Hence, CCN is a future Internet architecture which facilitates easy availability of content to match the growing demand of information exchange.

1.2 | Basic issues

The salient CCN features such as in-network content caching and Interest packet aggregation mechanisms enhance the overall efficiency of CCN but at the same time disturb the fundamental business policy of the content provider/publisher of the conventional IP-based Internet. This is because the content publishers usually earn revenue from their potential consumers through tracking and monetizing content usage. As CCN makes publisher's content available with the network routers that leaves content publishers with unprecedented challenge for tracking content access. Hence, an efficient and flexible CCN business model is required to protect the business interest of content publishers in this new CCN framework.

1.3 | Our contribution

In this paper, we have proposed efficient and secure communication protocols for flexible CCN business model through which the content publisher can track its potential consumers as well as their content usage. The proposed business model not only ensures security of the financial transactions between the consumer and the publisher by using Elliptic Curve Cryptography (ECC) but also minimizes the computation and communication cost and enhances the efficiency of CCN.

1.4 | Paper organization

The remainder of this paper is organized as follows. Section 2 provides a brief background study of the proposed work. The fundamentals of ECC are given in section 3. In section 4 and section 5, proposed CCN architecture and proposed CCN business model are presented respectively. The security analysis is given in section 6. In section 7, formal verification and simulation using AVISPA is done. Section 8 presents the protocol analysis using BAN logic. The performance analysis of the proposed scheme is given in section 9 and finally section 10 concludes the paper.

2 | BACKGROUND STUDY

Initially CCN was envisaged as a new networking paradigm to leverage scalable content distribution with Interest based content retrieval, name based routing, in-network content caching and Interest packet aggregation as salient features.[2-5] Generally, CCN has four types of network entities namely *consumer*, *content provider*, content publisher / publisher and *CCN routers*. In addition, CCN uses two types of packets namely *Interest* packet, generated by consumer for sending content request and *Content* packet, generated by publisher for sending the content. Later on, another type of CCN packet, called *manifest* packet was introduced to communicate access control information.[6] The work paradigm of all the entities using mentioned packets is discussed now. Initially, consumer generates Interest for the respective content he/she needs. The Interest is then forwarded by CCN routers towards the respective content provider/publisher.

Publisher collects the content from the respective content provider and publishes the content packet in the CCN. The nearest router of the publisher then forwards the content packet to the consumer using reverse interest path. In CCN, router performs name based routing similar to conventional IP-based routing using longest prefix match mechanism. However, unlike the conventional network router, CCN router optionally stores content in their limited buffer called content store (CS) for future use. This phenomenon is known as in-network content caching mechanism. Usually CCN router follows popularity based content caching and acts accordingly as discussed now. After receiving an Interest packet, CCN router initially searches the content name in its CS. If the content is available in CS, the router sends the content to the consumer; otherwise it enlists the Interest name in its pending Interest table (PIT) and forwards the Interest according to its forwarding information base (FIB). Moreover, if multiple Interest packets for the same content are received from downstream by a CCN router, it forwards only the first Interest packet upstream towards the respective content provider/publisher and enlists all the Interest requests in its PIT. This phenomenon is called Interest packet aggregation. After receiving the content from the publisher, the router accordingly forwards the content packet to all the consumers.Considering the research aspects of CCN, the existing research work so far mainly focuses on content naming,[7] content caching policy,[8,9] content routing[9] and content security.[10] However, to be widely adopted by the Internet community, the development of business model for CCN publisher is necessary and that is considered as one of the important research aspects to be taken care of. For better understanding, the necessity of the CCN business model is discussed now.

In CCN, the in-network content caching mechanism reduces the response time and enhances the efficiency and easy availability of content whereas Interest packet aggregation reduces the network traffic. However, in both the cases the content publisher becomes unaware about the several accesses of its content and as a result, it remains unacquainted about the consumers' demands and unable to track those consumers.[11] Therefore, considering the existing revenue generation policy of IP-based Internet, the CCN paradigm leaves content publishers with unprecedented challenge in terms of revenue generation. Hence, to leverage the benefits of CCN as well as to restore the realistic business policy of content publishers, we have explored a content provisioning mechanism or business model for CCN. To design secure communication protocols for a flexible CCN business model, it is important to ensure that only the authenticated consumers can access the content of the respective publisher. Moreover, the communication between the consumer and the publisher is usually carried over an insecure channel where authentication is required to ensure privacy and integrity. Hence, a two party mutual authentication between the consumer and the publisher is required where the publisher allows only an authenticated consumer to access its content though the content may be available in network router's cache.

To understand the state-of-the-art authentication protocols and develop an efficient and secure mutual authentication protocol for CCN business model, we have studied several papers on authentication which are briefly discussed here. In 1981, Lamport[12] proposed password based authentication scheme for remote user/server but the scheme is found vulnerable to replay attack.[13] Thereafter, multiple improved authentication and session key negotiation protocols using different cryptosystems were proposed in the timeline. Few researchers have proposed bilinear pairing based authentication schemes[14-16] but it is already known that bilinear pairing has comparable higher computation overhead than ECC based point multiplication operation.[17-19] In 2011, Kalra and Sood[20] have performed a detailed survey on ECC based protocols and mentioned that ECC turns out to be a most efficient and lightweight security measure for authentication between resource constrained client and server. Further, it is found that ECC based authentication protocols for smart devices have several limitations. For example, Wu et al...[21] have ensured user authentication but the scheme is not resilient to server impersonation attack.[22] Further, Abicher et al.[23] and Tian et al.[24] have used public key certificates for mutual authentication that incurs additional overhead for maintaining certificates. Moreover, the ECC-based authentication schemes proposed by Kalra et al[22] and Qi et al.[25] require secure channel in registration phase that incurs additional overheads for the establishment of secure channel. In recent time, many researchers proposed bio-metric based[26-28] and smartcard based[28-33] authentication schemes to provide higher security and robustness but due to the higher maintenance cost of these technologies and security weaknesses, they are not widely accepted as briefly discussed now. In 2014, Chen et al.[31] proposed a smart card based password authentication scheme and claims that the scheme can resist various malicious attacks. However, Jiang et al.[32] found that the scheme proposed by Chen et al.[31] is vulnerable to off-line password guessing attack and accordingly proposed an improved smartcard based authentication scheme. Moreover, due to use of modular exponentiation operations, both the schemes[31,32] have high computation overhead. Later, in 2015, Karuppiah et al.[33] also proposed a novel password and smartcard based remote mutual authentication scheme but due to the use of smartcard, their scheme incurs high computation cost. In 2016, Kumar et al.[34] proposed an improved password and smartcard based remote user authentication scheme which found to be susceptible to replay and session key disclosure attack.[35] Recently, in 2017, Li et al.[36] proposed an ECC and biometric based authentication scheme for IoT environment but due to the use of biometric, the scheme becomes

expensive. In the same year, Karati et al.[37] proposed a new identity based signcryption scheme for authentication but due to the use of bilinear pairing, the scheme incurs high computation overhead. In 2018, Park et al.[38] found few vulnerabilities such as impersonation attack in Qi et al.'s[25] scheme and proposed an improved smartcard based two party authentication and key exchange protocol in mobile environment but their scheme incurs high cost of maintaining smart card and they need a secure channel for registration phase. Hence, due to lack of cost-effective but efficient and secure authentication scheme, the development of the same suitable for CCN business model is needed.

Therefore, our objective is to design secure communication protocols for flexible CCN business model that provides consumer registration, mutual authentication, session key negotiation and consumer's password change option in an efficient but cost effective manner. Our major motivation is to design a widely acceptable and cost effective security solution for CCN business model with the salient CCN features like in-network content caching and Interest packet aggregation.

3 | PRELIMINARIES OF ECC

Elliptic Curve Cryptography (ECC)[39-42] is a state-of-the-art lightweight cryptosystem as it uses smaller key size than other contemporary cryptosystems such as RSA. In addition, ECC uses additive finite group rather than multiplicative group used by RSA. Therefore, additive finite group operations like point addition and point multiplication can be performed more efficiently in ECC over the modular exponentiation operation performed in RSA. Further, ECC attains comparable level of security using only 160-bits key whereas RSA requires 1024-bits key for same level of security. Moreover, as ECC based *Discrete Logarithmic Problem* (ECDLP) does not have any polynomial time algorithm; it is very hard to compromise security in ECC.[43-46] Hence, due to its higher efficiency and security strength, ECC is widely used by many researchers and network security professionals. Now, a brief overview on ECC is given here.[41,42]

Let an elliptic curve E over a prime finite field F_p, denoted as E/F_p, is defined by the following elliptic curve equation:

$$y^2 \bmod p = \left(x^3 + ax + b\right) \bmod p \tag{1}$$

Where p is a prime number; x, y, a, b ϵ F_p and $(4a^3 + 27\,b^2)$ *mod* $p \neq 0$. This *Equation* (1) uses additive elliptic curve group defined as $G_p = \{(x, y)$: x, y $\in F_p$ and (x, y) ϵ $E/F_p\}$ U $\{O\}$, where point O is called *point of infinity*. Point of infinity is the identity element of the additive elliptic curve group, used in ECC such that $P + O = P$ where P is a point on elliptic curve defined by *Equation* (1). Let P and Q be two points on *Equation* (1) and $Q = -P$, then $P + Q = P - P = O$, where it is assumed that the line joining P and $-P$ intersects *Equation* (1) at point O. As the *Equation (1)* uses additive finite elliptic curve group, it supports following operations as discussed below.[39-42]

- **Scalar point multiplication:** A scalar t can be multiplied with an elliptic curve point P on *Equation* (1). It is defined as $tP = P + P + \ldots + P$ (t times), where $t \in_R \mathbb{Z}_p^*$.
- **Point addition:** Addition of two points P and Q on *Equation* (1) is defined as $P + Q = R$ where $P \neq Q$. Here, with respect to x-axis, R is the reflection of intersection point ($-R$), between the line joining P, Q and *Equation* (1).
- **Point doubling:** Adding a point with itself is known as point doubling. Let $2P = P + P = Q$, then with respect to x-axis, Q is the reflection of intersection point ($-Q$), between the tangent line at point P and *Equation* (1).

In addition to ECDLP, *Elliptic Curve Factorization Problem* (ECFP), *Computational Diffie-Hellman Problem* (CDHP) and *Decisional Diffie-Hellman Problem* (DDHP) also do not have any polynomial time algorithm[47-49] and that makes ECC based security solutions very hard to compromise. Therefore, these security hardnesses, higher efficiency, smaller key size, less computation, communication and storage cost of ECC have motivated us to use ECC based security solution for designing the proposed business model for CCN.

4 | PROPOSED CCN ARCHITECTURE

As mentioned in the literature,[2-10] in CCN, publisher works as the interface of the actual content provider which stores/generates the content. Multiple content providers may operate under one publisher. After collecting the content from a content provider, the publisher performs the content encryption, packetization and content dissemination operations.

The content is encrypted using a content key for preventing unauthorized access. The content key is a secret key which is distinct for every content published by a publisher. When content is first time published, the publisher randomly generates a content key, encrypts the content using it and stores it in a database against the respective content name. For accessing the content, a consumer needs to decrypt the content and invariably needs the content key from the publisher. During the transmission, the encrypted content may be optionally cached by any intermediate CCN routers. Usually, CCN routers use popularity based content caching mechanism. In such mechanism, the router measures the popularity of a content by considering the number of Interest packet enlisted in its PIT, for a particular content. So, the content may be available in the CCN router's cache, but to access the content the consumer needs to get the content key from the respective publisher. The publisher uses encryption to send the content key in a secure way to the respective consumer. In our scheme, the content key is never cached by the CCN routers. In case, we enable CCN routers to cache the content key, the router has to do the content key encryption separately for each consumer. In such circumstance, efficiency will increase along with router's overhead but the security will significantly decrease. Now, a brief workflow of proposed CCN architecture is discussed below:

In the proposed CCN architecture, initially, a consumer requests for content by sending *Interest* packet. The Interest is then forwarded by the CCN routers and finally reaches the respective content publisher. The publisher collects the requested content from the respective content provider. After receiving the content from the content provider, the publisher encrypts it with a distinct secret content key. The publisher also encrypts the secret content key with another shared secret key which is negotiated between the publisher and the respective consumer. Then, the publisher packetizes both the encrypted content and encrypted content key and publishes in CCN. Finally, the encrypted content and encrypted content key are forwarded by the CCN routers using reverse Interest path and sent to the respective consumer. During transmission, CCN intermediate routers optionally store the encrypted content part in their CS. Later, if the same content is requested by any other consumer, the router finds the content in its CS and sends a request for content key to the respective publisher who originally publishes the particular content. After receiving the content key request from the router, the publisher sends only the encrypted content key to the requesting router who holds the content. Then the router combines both the encrypted content and encrypted content key together and sends to the consumer. The basic workflow of the proposed CCN architecture is depicted in Figure 1 for better understanding.

In the proposed scheme, we use three types of packets namely Interest packet for sending content request, Content packet for sending content and Manifest packet for sending metadata of the communication. In this scheme, two types of Interest packet are used namely, $Interest_C$ and $Interest_{K_C}$. $Interest_C$ is generated by the consumer for sending content request and $Interest_{K_C}$ is generated by the intermediate CCN router which has the requested content in its CS and used for sending content key request to the publisher. Manifest packet is generally used for decoupling the content from its metadata such as access control specification, payload, etc. We use Manifest packet to exchange the access control specification such as algorithms, hash function, cryptographic parameters, acknowledgement etc. between the consumer and the publisher. Here, specifically, types of Manifest are used to send registration request ($Manifest_R$), login request ($Manifest_L$), acknowledgement ($Manifest_{AckP}$ or $Manifest_{AckC}$), password change request ($Manifest_P$) and secret content key ($Manifest_{K_C}$). However, Manifest packets are never cached by the CCN routers. A general format of different packet structures used in the proposed scheme is given in Figure 2.

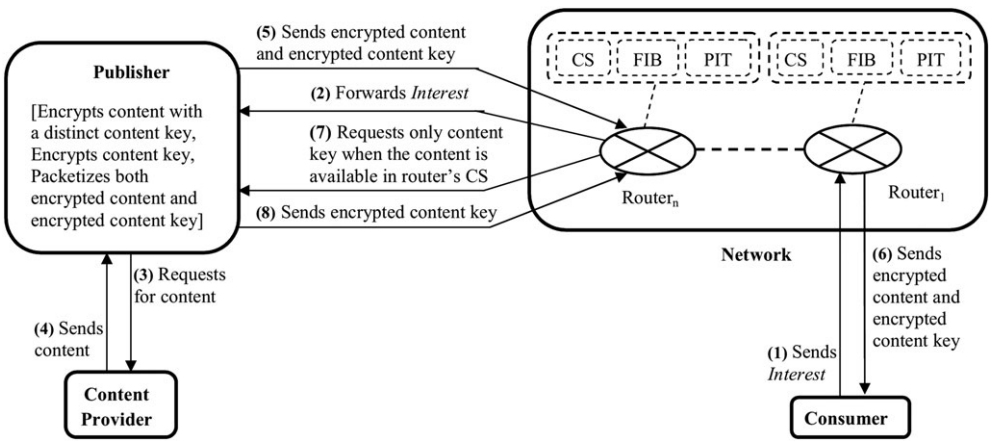

FIGURE 1 Proposed workflow of CCN architecture

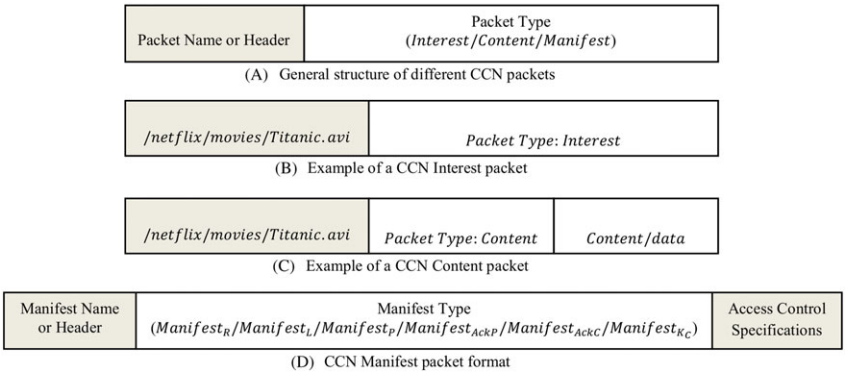

FIGURE 2 General format of different CCN packet structures

5 | PROPOSED CCN BUSINESS MODEL

In CCN, the available/requested content is divided in two categories namely *general content* and *exclusive content*. General content is available free of cost but for accessing the exclusive content, the consumer has to pay the subscription fee to the publisher. In the proposed scheme, publisher maintains two databases namely *CCN content database* and *CCN consumer registration database* to store content details and consumer details respectively. The structures of these two databases are shown in Figure 3 and Figure 4 respectively.

The CCN content database stores content name, content type, secret content key generated by the publisher, content provider's name, *popularity-based counter* (PCT) and other related data. Generally, content providers periodically advertise their content name and publisher enlists the content name with the content provider in the content database. The 'content type' attribute in CCN content database takes any of the two values namely general content and exclusive content.

On the other hand, CCN consumer registration database stores consumer's ID, subscription type, consumer's secret password, *hit-based counter* (HCT) and other related information such as content request history etc. The 'subscription type' attribute in CCN consumer registration database may take values as: 0 – for no subscription, 1- for pay per content, 2 – for monthly subscription, 3 – for yearly subscription and 4 – for hit based subscription. In case of a consumer who doesn't subscribe to the publisher, his/her subscription type is 0 and the consumer's secret password attribute is null. With the subscription type 0, the consumer can only access general content. In case of a consumer who accesses any general content, the HCT counter of the CCN consumer registration database is not updated i.e. the HCT counter is updated only when any exclusive content is successfully delivered to the respective consumer. In case of 1, 2 and 3 subscription type, the HCT counter increases with each content access but in case of subscription type 4, HCT takes a maximum value as specified and decreases with each content access. Moreover, for 2 and 3 subscription type, the subscription has to be renewed after the end of the month and year respectively but in case of hit based subscription, it has to renew after HCT becomes zero. The 'content request history' attribute of the CCN consumer registration database stores the pattern of content usage of the particular consumer that can be further used to predict the future request or to send recommendations for other content.

Now a brief workflow of the proposed business model is discussed here. In our scheme, when an Interest for content request comes to a content provider, it is attended by the interface publisher. The publisher searches the content name in its CCN content database and follows any of the cases discussed below.

Case 1: If the requested content is a general type content i.e. available free of cost, then the publisher follows *model-1*, discussed in *subsection* 5.1, for general content provisioning. In brief, the publisher collects the respective content from the content provider listed in its CCN content database. If the content is already requested previously, then the secret content key K_C for the respective content is stored in the CCN content database. If the content is requested

Content Name	Content Type	Secret Content Key	Content Provider	PCT	

FIGURE 3 CCN content database

Consumer's ID	Subscription Type	Consumer's Secret Password	HCT	Content Request History	

FIGURE 4 CCN consumer registration database

for the first time then the publisher randomly generates K_C and stores in the CCN content database for future use. After getting K_C, the publisher encrypts the content using K_C, packetizes the encrypted content and sends to the consumer. The publisher also sends the K_C to the respective consumer in a secure way. The publisher also updates the CCN consumer registration database with the content request history of the consumer that may be used in future for analyzing the demand or for recommending another content.

Case 2: On the other hand, if the content is an exclusive content i.e. paid content, then the publisher searches the CCN consumer registration database for the particular consumer ID. If the consumer is a registered consumer, then he/she has a subscription type and the secret password, stored in the CCN consumer registration database. Initially, the consumer sends a login request with the Interest for the particular content and the publisher follows *model-2*, discussed in *subsection* 5.2, for exclusive content provisioning. After receiving the request, the publisher authenticates the consumer and after successful authentication, the publisher follows the same procedure as in *case 1* and accordingly delivers the content to the consumer. After successful content delivery, the publisher updates the CCN consumer registration database and specially, the HCT counter. However, if it is found that the login request is not attached with the Interest, then the publisher simply rejects the content request. Otherwise, if the content request is for exclusive content and the consumer is not a registered consumer, then the publisher sends a response Manifest requesting the consumer to subscribe for the exclusive content.

Thus, the publisher performs all the required work for content delivery and revenue generation. More importantly, the publisher handles the business policy of the content providers and keeps track of the potential consumers who access their content. The publisher also tracks the amount of content usage and usage pattern of the consumers.

TABLE 1 Notations and their meaning

Symbols	Meaning
P	Publisher
CM	Consumer
E_X/D_X	Encryption/decryption using secret key X
CA_X	Public key certificate of X
K_C	Secret content key for encrypting content C
K_{CM}	Secret key between publisher and consumer
$h(\_)$	A secure one-way hash function such as SHA-1
ID_P	Identity of publisher
ID_{CM}	Identity of consumer
$\|\|$	Concatenation
F_p	A finite field over prime p
$E_p(a, b)$	An elliptic curve over Fp
G	Generator of the cyclic group on $E_p(a, b)$ with order n where $G(x, y) \in \mathbb{Z}_p^*$
(r_p, PU_p)	Private/public key pair of publisher where$PU_p = r_p . G$
(r_{CM}, PU_{CM})	Private/public key pair of consumer where$PU_{CM} = r_{CM} . G$
$Interest_C$	Interest (content request) from consumer
$Interest_{KC}$	Interest (content key request) from CCN router
$Manifest_{AckP}$	Manifest for acknowledgement from publisher
$Manifest_{AckC}$	Manifest for acknowledgement from consumer
$Manifest_R$	Manifest for registration request
$Manifest_L$	Manifest for login request
$Manifest_P$	Manifest for password change request
$Manifest_{Kc}$	Manifest for encrypted secret content key

In this paper, efficient and secure communication protocols for a flexible CCN business model is proposed where the publisher tracks its potential consumers, their content usage pattern and amout of content usage by a hit based content provisioning method and thereby earns revenue from consumers. In addition, the publisher also monitors the popularity of its content and monetize accordingly. As stated earlier, publisher has two types of business provisions: *case 1* and *case 2* which are presented in detail as *model-1* and *model-2* respectively in the following subsections where the following notations, given in Table 1, are used. The publisher can follow both the models for general as well as exclusive content provisioning simultaneously and that gives business flexibility in the proposed scheme. Moreover, in both the business models, a popularity-based counter PCT is used to count the hits for a particular content and a hit-based counter HCT is used to count the hits of a particular consumer. PCT and HCT are maintained to track the popularity of a content and the number of the particular consumer's login respectively.

5.1 | Model – 1

In this subsection, the main focus is on the general / free of cost content provisioning where the content is available free of cost and publisher tracks their potential consumers and their active content usage pattern. Although, in this model the publisher doesn't earn revenue directly from the content usage but makes profit by using the CCN content database and CCN consumer registration database for further business analytics such as future market prediction, demand analysis etc. The publisher stores the consumer's identity, their usage pattern/content request history etc. in the CCN consumer registration database through which the publisher can analyze the future content demand as well as can send future content recommendations. The CCN consumer registration database can also be used for advertisements etc. according to the consumers' usage pattern to earn revenue from the advertisers. The databases can also be used for sending recommendations for similar types of exclusive contents in which the consumer may be interested and thereby increasing the scope of business provisions for the publisher. Moreover, the PCT counter of the CCN content database may be used to measure the popularity of a particular content that helps to decide the cost of the respective content for conversion to exclusive content. The proposed *model-1* for general content provisioning is depicted in Figure 5 and step-wise discussed below where $X \rightarrow Y : M$ means the sender X sends message M to the receiver Y.

Step 1. Consumer → Publisher: $\{ID_{CM}, CA_{CM}, Interest_C\}$

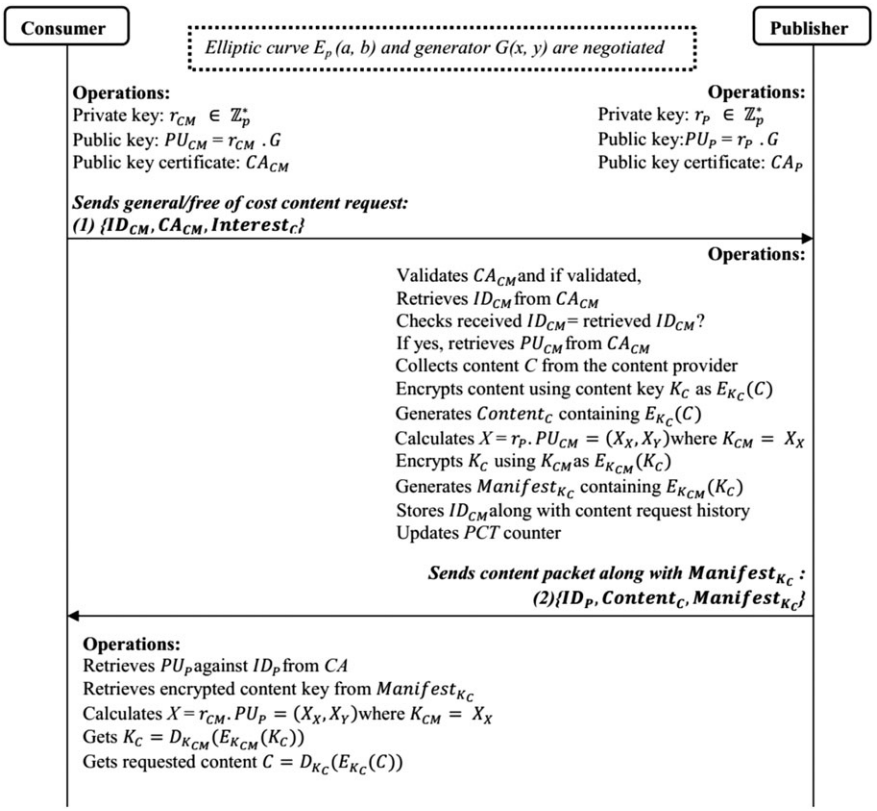

FIGURE 5 Model - 1 for general content provisioning

Initially, consumer generates $Interest_C$ and forwards it in CCN along with its identity ID_{CM} and public key certificate CA_{CM}.

Step 2. Publisher → Consumer: $\{ID_P, Content_C, Manifest_{K_C}\}$

After receiving the content request, the publisher searches content name in the CCN content database. If the content is a general type content i.e. available free of cost, the publisher follows *model-1*. Initially, the publisher validates CA_{CM} and if validated, retrieves ID_{CM} from it and checks whether received ID_{CM} = retrieved ID_{CM}? If yes, retrieves PU_{CM} from CA_{CM}. After that, the publisher collects the requested content C from the respective content provider and subsequently encrypts the content using the corresponding secret content key K_C. The publisher also generates $Content_C$ packet containing the encrypted content $E_{K_C}(C)$. Now the publisher calculates K_{CM}, the shared secret between the publisher and the consumer, as: $X = r_P.PU_{CM} = (X_X, X_Y)$ where $K_{CM} = X_X$. After deriving K_{CM}, the publisher encrypts K_C using K_{CM} as: $E_{K_{CM}}(K_C)$ and generates $Manifest_{K_C}$ packet containing the encrypted content key. The publisher stores/updates the respective consumer's identity ID_{CM} in its CCN consumer registration database along with its content request history and other related details. The publisher also updates the PCT counter of the requested content in the CCN content database. Finally, the publisher sends its identity ID_P, content packet $Content_C$ and Manifest packet $Manifest_{K_C}$ which contains the secret content key, to the respective consumer.

It is to be noted that during transmission, the $Content_C$ packet may be cached by the intermediate CCN routers. Hence, if next time a similar content request comes to the router who previously cached the content, the router sends the response $Content_C$ packet to the respective consumer. In addition, the router generates an content key request $Interest_{K_C}$ from the original $Interest_C$ and sends to the publisher who generates the content. After receiving the content key request $Interest_{K_C}$, the publisher searches its CCN content database and gets the corresponding secret content key K_C. Then the publisher follows previously mentioned procedure given in Figure 5 and sends only $Manifest_{K_C}$ packet along with ID_P to the respective router (who sends $Interest_{K_C}$) as: $\{ID_P, Manifest_{K_C}\}$ in *step 2*. After receiving the $Manifest_{K_C}$, the respective router combines the $Manifest_{K_C}$ with the $Content_C$ available in its CS and sends to the respective consumer as stated earlier. Thus, though the consumer gets the required content from the nearest CCN router's CS, the content key, which is required to decrypt the content, is received only from the original publisher that makes the publisher able to track the use of content by the consumer.

5.2 | Model – 2

In this subsection, the main focus is to keep provision for accessing the exclusive/paid content where publisher earns revenue by delivering exclusive content to its consumers on the basis of respective paid subscription types. Initially, a consumer's registration procedure is performed by the publisher before the delivery of any content / content key. After verifying all the required credentials, the publisher registers a consumer under any of the subscription types (1, 2, 3 and 4) with the payment of appropriate charges, if any, as mentioned in 3rd paragraph of *section* 5. After the successful registration, the consumer gets a secret password PW_{CM} from the publisher that is stored in the CCN consumer registration database along with consumer's identity ID_{CM}, subscription type and other related details as shown in Figure 4.

In case of monthly (subscription type – 2) or yearly (subscription type – 3) or hit based subscription (subscription type – 4), the registration is successfully completed against certain amount of payment and the consumer gets a secret password PW_{CM}. On the other hand, in pay per content policy (subscription type – 1), the consumer has to initially register to get the secret password PW_{CM} but has to pay at the time of login to the publisher for accessing exclusive content. For hit based subscription (subscription type – 4), the number of hit (content access) is monitored by updating HCT counter in the CCN consumer registration database as stated in 3rd paragraph of *section* 5. In this case, a maximum number of hit (HCT value) is specified by the respective publisher that decreases with each content access. Finally, when the HCT counter decreases to zero, the consumer's password becomes invalid i.e. the subscription ends. On the other hand, for monthly or yearly subscription (subscription type – 2 or 3), the consumer's secret password becomes invalid i.e. the subscription ends after the end of the month or year.

Note that, when a consumer searches a content in the application layer, he/she may get several options of availability by different content publishers. If the required content is an exclusive type content, the consumer sends Interest request $Interest_C$ to the publisher with whom the consumer has valid registration. In case, the required content is not

available from the publisher where the consumer is registered, the consumer can access the required exclusive content from another publisher by pay per content basis (subscription type – 1).

Each time when a consumer wants to access an exclusive content with valid registration; he/she has to send an authentication request along with the $Interest_C$ to the respective publisher. The registered consumer's content request is processed by the publisher with higher priority than the unregistered consumer's request. The consumer's authentication request is validated by the publisher using the consumer's secret password PW_{CM}. This validation procedure follows the proposed *ECC-based mutual authentication and session key negotiation protocol* as depicted in Figure 7. To provide sufficient security, the *model-2* includes a remote mutual authentication scheme which is divided into three following sub-sections namely – (1) *ECC-based consumer registration protocol*, (2) *ECC-based mutual authentication and session key negotiation protocol* and (3) *ECC-based consumer's password change protocol* with step-wise descriptions where $X \rightarrow Y : M$ means the sender X sends message M to the receiver Y.

5.2.1 | ECC-based consumer registration protocol

Any consumer who wants to access exclusive content has to register to the respective publisher. The registration procedure is shown in Figure 6 and described below.

Step 1. Consumer → Publisher: $\{ID_{CM}, CA_{CM}, E_{K_{CM}}(ID_{CM}||n_1), Manifest_R\}$

Initially, the consumer retrieves the public key PU_P of the publisher against ID_P from the certificate authority (CA). Now the consumer calculates K_{CM}, a contributory shared secret key between the consumer and the respective publisher as: $X = r_{CM}.PU_P = (X_X, X_Y)$ where $K_{CM} = X_X$. Then, the consumer generates a random nonce n_1, concatenates it with identity ID_{CM} and encrypts the concatenated message using K_{CM}. The consumer also generates a registration request $Manifest_R$ which contains the payment details in case of monthly, yearly and hit based subscription as shown in Figure 2*, (D)*. Finally, the consumer sends the $Manifest_R$ to the publisher along with its identity ID_{CM}, public key certificate CA_{CM} and $E_{K_{CM}}(ID_{CM}||n_1)$.

Step 2. Publisher → Consumer: $\{ID_P, E_{K_{CM}}(PW_{CM}), h(PW_{CM}||n_1), Manifest_{AckP}\}$

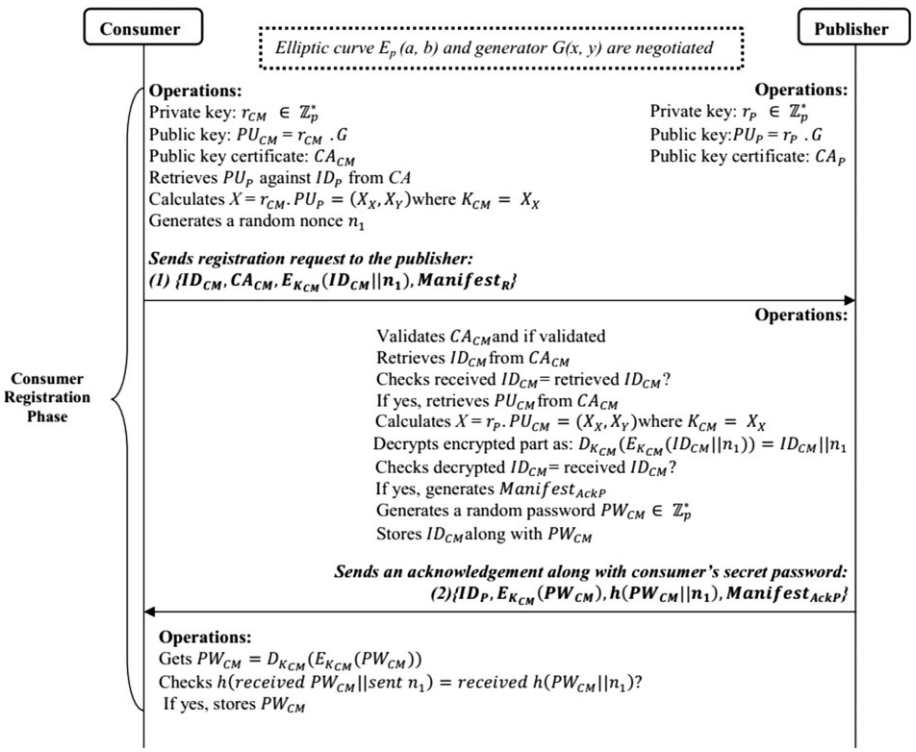

FIGURE 6 ECC-based consumer registration protocol

After receiving the registration request, the publisher validates CA_{CM} and if validated retrieves ID_{CM} from it. Checks received ID_{CM} = retrieved ID_{CM}? If yes, retrieves PU_{CM} from CA_{CM} and calculates K_{CM}, the same shared secret between the consumer and the publisher as: $X = r_P.PU_{CM} = (X_X, X_Y)$ where $K_{CM} = X_X$. Then, the publisher decrypts the encrypted part as: $D_{K_{CM}}(E_{K_{CM}}(ID_{CM}||n_1)) = ID_{CM}||n_1$ and checks *decrypted* ID_{CM} = *received* ID_{CM}? If yes, the publisher generates a random password $PW_{CM} \in \mathbb{Z}_p^*$ and stores ID_{CM} alongwith the consumer's secret password PW_{CM} in its CCN consumer registration database. Now, the publisher generates a $Manifest_{AckP}$, the acknowledgement of the completion of registration. The publisher also concatenates the received nonce n_1 with PW_{CM} and makes a hash digest of the concatenated message as: $h\ (PW_{CM}||n_1)$. Finally, the publisher encrypts PW_{CM} with K_{CM} as: $E_{K_{CM}}(PW_{CM})$ and sends to the consumer along with its identity ID_P, the hash digest and $Manifest_{AckP}$.

After receiving, the consumer decrypts the encrypted password and gets the password PW_{CM} as $D_{K_{CM}}(E_{K_{CM}}(PW_{CM})) = PW_{CM}$. Now, the consumer checks *h (received* PW_{CM}*||sent* n_1*) = received h (*$PW_{CM}||n_1$*)*? If yes, the consumer stores the secret password PW_{CM} through which the registered consumer can authenticate himself/herself to the publisher during login phase.

5.2.2 | ECC-based mutual authentication and session key negotiation protocol

In order to access any exclusive content of the publisher, a registered consumer has to login to the publisher using his/her login pair (ID_{CM}, PW_{CM}). The step-wise login procedure is shown in Figure 7 and described below.

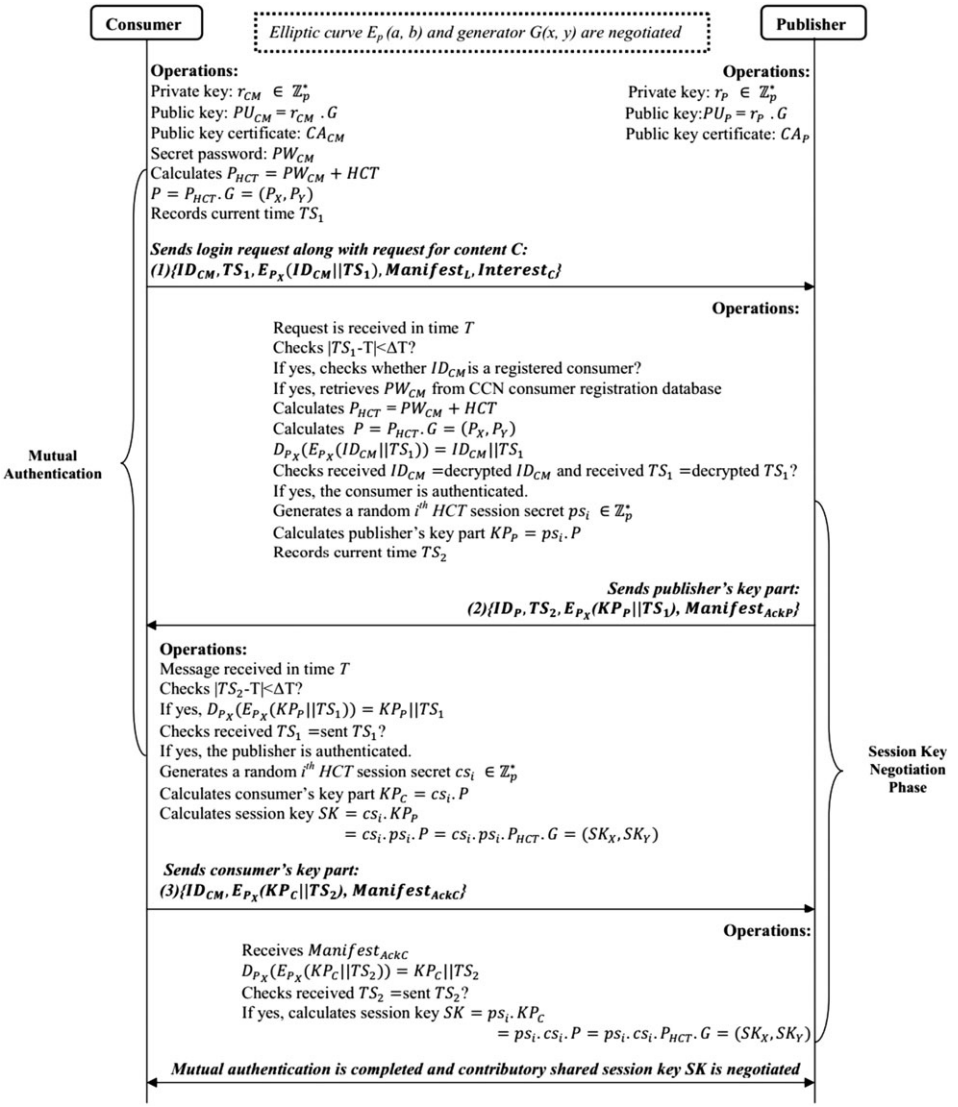

FIGURE 7 ECC-based mutual authentication and session key negotiation protocol

Step 1. Consumer → Publisher: $\{ID_{CM},\ TS_1,\ E_{P_X}(ID_{CM}||TS_1),\ Manifest_L,\ Interest_C\}$

Initially, the consumer enters his/her secret password PW_{CM} and accordingly a secret key P is generated as: $P = P_{HCT}$. $G = (P_X,\ P_Y)$ where, $P_{HCT} = PW_{CM} + HCT$. Here, as stated earlier, HCT is a hit-based counter in the CCN consumer registration database which counts the number of exclusive content access by the respective consumer. In case of monthly, yearly and pay per content subscription, HCT initially starts from 1 and after each successful receiving of an exclusive content it is increased by 1. In case of hit based subscription, HCT starts from a fixed number, decided by the publisher and after each successful receiving of an exclusive content it is decreased by 1. So, the maximum number of login in one particular subscription can be restricted and monitored by both the consumer and publisher. Now, the consumer records current time stamp TS_1, concatenates it with ID_{CM}, encrypts the concatenated message using P_X, the x-coordinate of the calculated secret key P, as: $E_{P_X}(ID_{CM}\|TS_1)$. Finally, the consumer sends his identity ID_{CM}, time stamp $TS1$, the encrypted message $E_{P_X}(ID_{CM}\|TS_1)$ and login request $Manifest_L$ along with the exclusive content request $Interest_C$ to the publisher. In case of pay per content subscription, $Manifest_L$ contains the payment details of the particular exclusive content.

Step 2. Publisher → Consumer: $\{ID_P, TS_2, E_{P_X}(KP_P\|TS_1), Manifest_{AckP}\}$

After receiving the login request from the consumer in time T, the publisher initially checks $|TS_1 - T| < \Delta T$? If yes, checks whether ID_{CM} is a registered consumer i.e. the ID_{CM} is present in the CCN consumer registration database? If yes, the publisher retrieves PW_{CM} from CCN consumer registration database and calculates the secret key P, using corresponding consumer's secret password PW_{CM} as: $P = P_{HCT}$. $G = (P_X,\ P_Y)$ where, $P_{HCT} = PW_{CM} + HCT$. Now, the publisher uses P_X, the x-coordinate of the calculated secret key P, to decrypt the received encrypted message as: $D_{P_X}(E_{P_X}(ID_{CM}\|TS_1))$ and gets ID_{CM} and TS_1. Then, the publisher checks *received* ID_{CM} = *decrypted* ID_{CM} and *received* TS_1 = *decrypted* TS_1? If yes, the consumer is authenticated. Then, the publisher selects a random i^{th} HCT session secret $ps_i \in \mathbb{Z}_p^*$ and accordingly calculates $KP_P = ps_i$. P; concatenates KP_P with the received time stamp TS_1 and encrypts the concatenated message using P_X as $E_{P_X}(KP_P\|TS_1)$. Finally, the publisher records the current time stamp TS_2 and sends to the consumer along with the identity ID_P, encrypted key part $E_{P_X}(KP_P\|TS_1)$ and $Manifest_{AckP}$, the acknowledgement of login, to the consumer.

Step 3. Consumer → Publisher: $\{ID_{CM},\ E_{P_X}(KP_C||TS_2),\ Manifest_{AckC}\}$

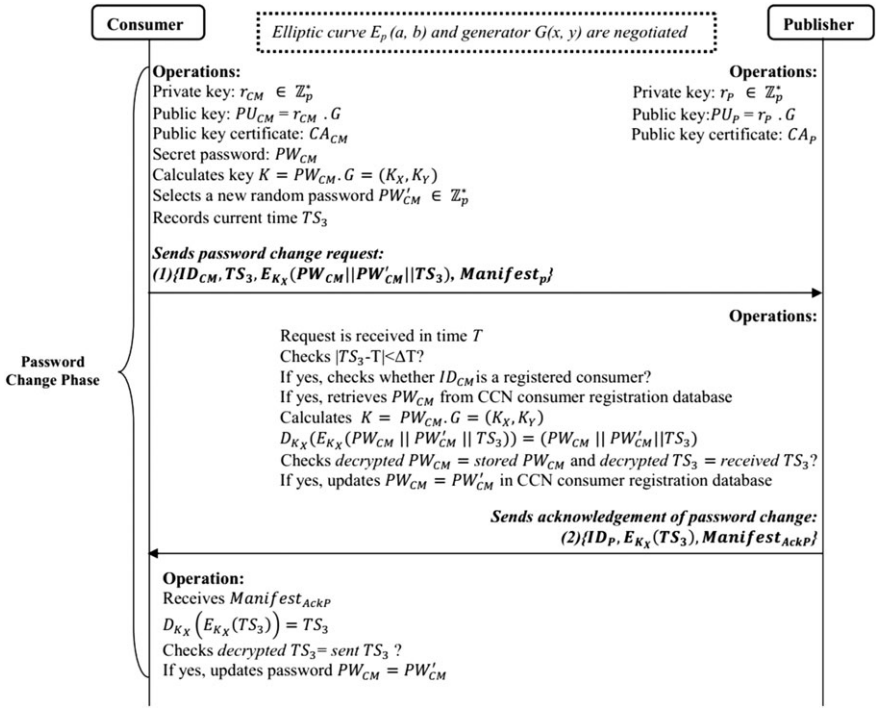

FIGURE 8 ECC-based consumer's password change protocol

After receiving the acknowledgement of login in time T, the consumer initially checks $|TS_2 - T| < \Delta T$? If yes, decrypts the encrypted key part using previously calculated secret key P_X as: $D_{P_X}(E_{P_X}(KP_P \| TS_1)) = KP_P \| TS_1$and gets KP_P and TS_1. Now, the consumer checks *received* TS_1 = *sent* TS_1? If yes, the publisher is authenticated and thus, the mutual authentication is completed. Now, the consumer selects a random i^{th} HCT session secret $cs_i \in \mathbb{Z}_p^*$. The consumer also calculates its key part $KP_C = cs_i.\ P$; concatenates KP_C with the received time stamp TS_2 and encrypts the concatenated message using P_X as $E_{P_X}(KP_C \| TS_2)$. Then, the consumer calculates the contributory shared session key $SK = cs_i.\ KP_P = cs_i.\ ps_i.\ P = cs_i.\ ps_i.\ P_{HCT}.\ G = (SK_X,\ SK_Y)$ and the SK_X will be kept secret and used for secure transmission of the exculsive content. Finally, the consumer sends the encrypted key part along with the identity ID_{CM} and $Manifest_{AckC}$, the acknowledgement of receiving publisher's key part, to the publisher for negotiating the contributory shared session key SK_X.

After receiving the consumer's key part, the publisher decrypts it using P_X as: $D_{P_X}(E_{P_X}(KP_C \| TS_2)) = KP_C \| TS_2$ and gets the consumer's key part KP_C and time stamp TS_2. Now, the publisher checks *received* TS_2 = *sent* TS_2? If yes, the publisher calculates the session key $SK = ps_i.\ KP_C = ps_i.\ cs_i.\ P = ps_i.\ cs_i.\ P_{HCT}.\ G = (SK_X,\ SK_Y)$ i.e. same SK_X is generated and will be used for secure transmission of the exculsive content. Once SK is calculated by both the consumer and the publisher, the transfer of all the content including the secret content key K_C is symmetrically encrypted/decrypted using SK_X of the session key SK.

5.2.3 | ECC-based Consumer's password change protocol

A registered consumer's password must be changed periodically to prevent any password guessing attack that increases the security strength of the proposed scheme. The step-wise password change procedure is shown in Figure 8 and described below.

Step 1. Consumer → Publisher: $\left\{ID_{CM}, TS_3, E_{K_X}\left(PW_{CM} \| PW'_{CM} \| TS_3\right), Manifest_P\right\}$

Initially, the consumer generates a $Manifest_P$ as given in Figure 2 for sending password change request to the publisher. Now, the consumer randomly selects a new password PW'_{CM}, calculates a secret key K_X using the consumer's old password PW_{CM} as: $K = PW_{CM}.G = (K_X,\ K_Y)$, concatenates the old password PW_{CM} with the new password PW'_{CM} and current timestamp TS_3, encrypts the concatenated message using K_X and finally sends its identity ID_{CM}, timestamp TS_3 and encrypted message $E_{K_X}\left(PW_{CM} \| PW'_{CM} \| TS_3\right)$ along with the $Manifest_P$ to the publisher as a password change request.

Step 2. Publisher → Consumer: $\{ID_P, E_{K_X}(TS_3), Manifest_{AckP}\}$

The publisher receives the password change request in time T and checks $|TS_3 - T| < \Delta T$? If yes, the publisher checks whether the consumer's ID_{CM} is present in the CCN consumer registration database? If so, the publisher retrieves PW_{CM} from the database and calculates the secret key K_X as: $K = PW_{CM}.G = (K_X,\ K_Y)$. Now, the publisher decrypts the received encrypted message using K_X as: $D_{K_X}\left(E_{K_X}\left(PW_{CM} \| PW'_{CM} \| TS_3\right)\right) = \left(PW_{CM} \| PW'_{CM} \| TS_3\right)$ and gets the consumers' old password, new password and the timestamp TS_3. Then, the publisher checks *decrypted* PW_{CM} = *stored* PW_{CM} and *decrypted* TS_3 = *received* TS_3? If both are yes, the publisher updates the consumer's old password PW_{CM} with new password PW'_{CM} in the CCN consumer registration database and finally sends the $Manifest_{AckP}$, the acknowledgement of the password change, to the consumer along with $E_{K_X}(TS_3)$, the encrypted received TS_3 using K_X.

After receiving the $Manifest_{AckP}$ from the publisher, the consumer initially decrypts the encrypted message as: $D_{K_X}(E_{K_X}(TS_3)) = TS_3$ and checks *decrypted* TS_3 = *sent* TS_3? If yes, the consumer is ensured about the updatation of its password to PW'_{CM}.

6 | SECURITY ANALYSIS

In this section, an in-depth security analysis of all the proposed protocols is done to show that all of them are well secured against relevant cryptographic attacks.

6.1 | Confidentiality

Confidentiality is one of the major concerns for any network security protocol where the data communication must remain secret between the sender and the receiver. As the medium of communication is an insecure channel, an intruder can access any information traveling between the consumer and the publisher. Hence, in the *model-1* of the proposed scheme, the content is encrypted using secret content key K_C and K_C is sent after encryption using another secret key K_{CM}. Moreover, in the proposed ECC-based mutual authentication and session key negotiation protocol, all the sensitive information such as consumer's key part KP_C and publisher's key part KP_P are encrypted before transmission. Similarly, in the proposed ECC-based consumer's password change protocol, consumer's old password PW_{CM} and new password PW'_{CM} are also encrypted using secret password. Therefore, the confidentiality of the shared data is preserved in our scheme.

6.2 | Mutual authentication

Mutual authentication is an important network security parameter where sender and the receiver authenticate each other. In the *step 1* of the *model-1* of the proposed protocol, the publisher validates the consumer's public key certificate CA_{CM} to ensure that the consumer is genuine. On the other hand, the consumer validates the authenticity of the publisher from CA. Moreover, the content key K_C is encrypted by the publisher using a contributory shared secret key K_{CM} which can be negotiated only by the respective consumer and the publisher. In the proposed ECC-based mutual authentication and session key negotiation protocol, initially, the consumer sends encrypted ID_{CM} and TS_1 using secret key P_X as: $E_{P_X}(ID_{CM}\|TS_1)$. The secret key P_X is calculated as: $P = P_{HCT}$. $G = (P_X, P_Y)$ where, $P_{HCT} = PW_{CM} + HCT$. Here, P_X is calculated using consumer's secret password PW_{CM} which is known to the respective consumer and the publisher. Moreover, hit-based counter HCT is counted by the respective consumer and publisher. Hence, P_X is a secret between the consumer and the publisher. After receiving $E_{P_X}(ID_{CM}\|TS_1)$ from *step 1*, the publisher decrypts it using its own P_X. Then the publisher checks *decrypted* ID_{CM} = *received* ID_{CM} and *decrypted* TS_1 = *received* TS_1? If yes, consumer is authenticated. On the other hand, the publisher encrypts the concatenated key part KP_P and received TS_1 using P_X and sends to the consumer in *step 2*. After receiving the encrypted key part from publisher, the consumer decrypts it using his own P_X and gets the key part KP_P and TS_1. The consumer checks *received* TS_1 = *sent* TS_1? If yes, then the publisher is authenticated. Thus, both the publisher and consumer authenticate each other.

Similarly, in the proposed ECC-based consumer's password change protocol, the communication between the consumer and the publisher is encrypted using secret key K_X which is calculated as: $K{=}PW_{CM}.G = (K_X, K_Y)$. As, PW_{CM} is a secret between the consumer and the publisher, if any mismatch found by any party, the authentication process is terminated. Hence, in both of these two proposed protocols, mutual authentication between the consumer and the publisher is ensured.

6.3 | Replay attack resilience / information freshness

Replay attack means any attacker captures legal network packets from one session and resends them in another session or at a later time and thereby impersonates himself as a legal user. Usually nonce is used to prevent the replay attack; however, use of timestamp not only ensures replay attack resilience but preserves the information freshness. In the ECC-based consumer registration protocol, nonce n_1 is used to prevent replay attack. In addition, in the proposed ECC-based mutual authentication and session key negotiation protocol, timestamp TS_1 is used and sent after encryption using the secret key P_X. After receiving in time T, the publisher initially checks $|TS_1 - T| < \Delta T$? If yes, it ensures no network delay. The publisher also checks *received* TS_1 = *decrypted* TS_1? If yes, it ensures the information freshness as well as prevents the replay attack. Moreover, in *step 3*, the consumer receives $E_{P_X}(KP_P\|TS_1)$ which ensures that the key part KP_P is sent by the respective publisher only. As well as, in *step 4*, when the publisher receives $E_{P_X}(KP_C\|TS_2)$, it becomes ensured that the key part KP_C is sent by the respective consumer only. Similarly, in the proposed ECC-based consumer's password change protocol, the use of timestamp TS_3 and encrypted timestamp prevents the replay attack. Thus, the proposed scheme is free from the replay attack.

6.4 | Man-in-the-middle attack resilience

Man-in-the middle attack means during communication between two parties an intruder may come in between and captures the communicating messages, modifies it and sends them for its own benefit. Thus, the intruder sets up secure communication with both the parties while the two end-parties believe that they are communicating between themselves only. In the proposed ECC-based mutual authentication and session key negotiation protocol, the session key parts KP_P and KP_C are communicated between the consumer and the publisher in an encrypted form using secret key P_X which is calculated as: $P = P_{HCT}$. $G = (P_X, P_Y)$ where $P_{HCT} = PW_{CM} + HCT$. Here, PW_{CM} is consumer's secret password and is a secret only between the respective consumer and the publisher. Moreover, HCT is known to and monitored by the respective consumer and publisher only. Hence, any attacker cannot access P_X and thereby unable to launch a man-in-the middle attack.

Similarly, in case of ECC-based consumer's password change protocol, the sensitive data are communicated between the consumer and the publisher after encryption using a secret key K_X which is calculated as: $K = PW_{CM}$. $G = (K_X, K_Y)$ where PW_{CM} is the secret between the respective consumer and the publisher. Hence, the man-in-the-middle attack is prevented. Thus, the proposed scheme is free from the man-in-the-middle attack.

6.5 | Impersonation attack resilience

Impersonation attack is a serious network threat in which an attacker impersonates himself as an authorized/valid consumer to the publisher or vice versa. In our scheme, initially, the publisher registers a consumer after validating its public key certificate CA_{CM} and provides a password PW_{CM} which is kept secret. Later in the subsequent phases/protocols, the password is used to authenticate the consumer. The consumer also uses PW_{CM} to calculate the secret key P_X which is used to encrypt all the confidential data for communication. Similarly, the respective publisher also calculates P_X using the same secret password PW_{CM} stored in its database. Since, PW_{CM} is a secret between the respective consumer and the publisher, nobody else cannot access PW_{CM} and thereby cannot impersonate either the consumer or the publisher. Hence, the proposed scheme successfully prevents the impersonation attack.

6.6 | Perfect forward secrecy

Perfect forward secrecy means even if the long term key becomes known at a point of time, the already negotiated session key before that time remains secure. In our proposed scheme, even if the consumer's password PW_{CM} is compromised to an intruder, the secret session key SK remains unknown because SK is calculated in consumer side as $SK = cs_i$. $KP_P = cs_i$. ps_i. $P = cs_i$. ps_i. P_{HCT}. $G = (SK_X, SK_Y)$ and in publisher's side as $SK = ps_i$. $KP_C = ps_i$. cs_i. $P = ps_i$. cs_i. P_{HCT}. $G = (SK_X, SK_Y)$. Hence SK is not only dependent on PW_{CM} but it depends on two random secrets ps_i and cs_i which cannot be compromised due to the computational problem of ECDLP as described in *step 2* and *step 3* in *subsection* 5.2.2. Hence, in the proposed scheme, perfect forward secrecy is maintained.

6.7 | Known session key attack resilience

A protocol is vulnerable to known session key attack if the knowledge of the session key in earlier session reveals the session keys of later sessions. In our scheme the session key SK is calculated as: $SK = cs_i$. $KP_P = ps_i$. KP_C where cs_i and ps_i are randomly generated in each session by the consumer and the publisher respectively. As both the session secrets are changed in each session, knowing one session key does not reveal the other.

6.8 | Brute force attack resilience

The proposed scheme is resilient to brute force attack / offline password guessing attack because the adversary has no way of guessing the secret session key SK since the session key SK is calculated as: $SK = cs_i$. $KP_P = ps_i$. KP_C where $KP_P = ps_i$. P, $KP_C = cs_i$. P, $P = P_{HCT}$. G and $P_{HCT} = PW_{CM} + HCT$. Here, the contributory key parts KP_P and KP_C, two points on elliptic curve, are communicated between each other in an encrypted form and SK is dependent on three secret numbers ps_i, cs_i and PW_{CM} which are randomly generated from $\mathbb{Z}_p^*$. Since the proposed scheme uses three random numbers in establishing the shared secret SK and according to,[50] it can be concluded that the proposed scheme is well secured as a shared secret with only one random number is assumed to be compromised. Moreover, SK is a point on the

elliptic curve thus, due to the hardness of ECDLP, CDHP and DDHP, it is impossible for the adversary to guess *SK* in polynomial time.

7 | FORMAL VERIFICATION AND SIMULATION USING AVISPA

In this section, a formal security analysis of the proposed protocols is done using the well known AVISPA simulator. AVISPA (Automated Validation of Internet Security Protocol and Applications)[51-53] is a virtual protocol simulation tool which detects whether any security protocol is safe or unsafe in a viable network. AVISPA uses HLPSL (High Level Protocol Specification Language) for simulation of network security protocols. HLPSL is a role based language in which each active participant involved in a communication protocol is presented as a basic role. Each role is independent from the other and presents some initial information as parameters which are needed for communication with other role over.the channel. In AVISPA, the channel is represented as standard *Dolev-Yao* (DY) intruder model which means the intruder has full control over the channel and he/she knows all the public keys and other pre-negotiated algorithms like hash operation, encryption/decryption etc. Initially, the protocol written in HLPSL is translated into a lower level format called *Intermediate Format* (IF) by a translator called *HLPSL2IF*. Then the intermediate format of the protocol is fed into one of the four back-end modules which are implemented using formal methods and theoretical axioms. The four back-end modules of AVISPA are – (1) *OFMC* – on-the-fly Model-Checker, (2) *CL-AtSe* – Constraint Logic based Attack Searcher, (3) *SATMC* – SAT-based Model-Checker, and (4) *TA4SP* – Tree Automata-based Protocol Analyzer.[51-53] All these back-ends are used to provide protocol falsification and, bounded and unbounded verification.

All the proposed protocols are analyzed using both OFMC and CL-AtSe AVISPA back-ends where the output format of the protocol simulation represents either the *safe* state or *unsafe* state.[54,55] The formal analysis of each proposed protocol is beneficial in detecting design flaws which will be very difficult and expensive to detect after the real life deployment. For each protocol, two basic roles *consumer* and *publisher* are presented in HLPSL that are played by *C* and *S* respectively. To ensure secrecy/confidentiality and authentication, HLPSL uses few predicates namely SECRET, WITNESS and REQUEST. The SECRET predicate is used for confidentiality and ensures that the transmitted value is secret between two concerned parties. The WITNESS and REQUEST predicates are used for authentication where a party witnessing a variable means that it is sending that variable to the receiver only and its value is fresh, and a party requesting a variable means that it wants to be assured about the freshness of the variable received and the variable has been sent by the sender in the live session.

Apart from the basic roles, there are two additional compulsory roles namely *session* and *environment*. In the role *session*, both the basic roles (consumer and publisher) are instantiated by providing concrete arguments. The role *session* also contains composition of basic roles and global constants such as symmetric_key, protocol_id etc. The role *environment*, a top- level role, contains the security goals of the protocol and the composition of roles where the intruder *i* acts as a role of a legitimate user. Mainly it is used to detect the – (1) parallel session attack/replay attack by executing two sessions between the consumer and the publisher simultaneously, and (2) man-in-the-middle attack by initiating two sessions between consumer and intruder, and between intruder and publisher.

The simulation of all the proposed protocols is done using SPAN,[56] a security protocol animator for AVISPA, and shows that all are safe. However, due to the space limitation, the roles and simulation results of the proposed ECC-based mutual authentication and session key negotiation protocol, and ECC-based consumer's password change protocol are given in following sub-sections.

7.1 | Simulation result of ECC-based mutual authentication and session key negotiation protocol

The two basic roles: *consumer* and *publisher* and two additional roles: *session* and *environment* for the proposed ECC-based mutual authentication and session key negotiation protocol are given in Figure 9. The *goal* section of role *environment* ensures the secrecy/confidentiality of KP_C and KP_P which are the key parts, shared by the consumer and the publisher respectively, to negotiate a shared session key *SK*. The goal section also ensures mutual authentication between two parties by ensuring the freshness of TS_1 and TS_2. The simulation result of the protocol executed in OFMC back-end and CL-AtSe back-end is given in Figure 10 with the output “safe”.

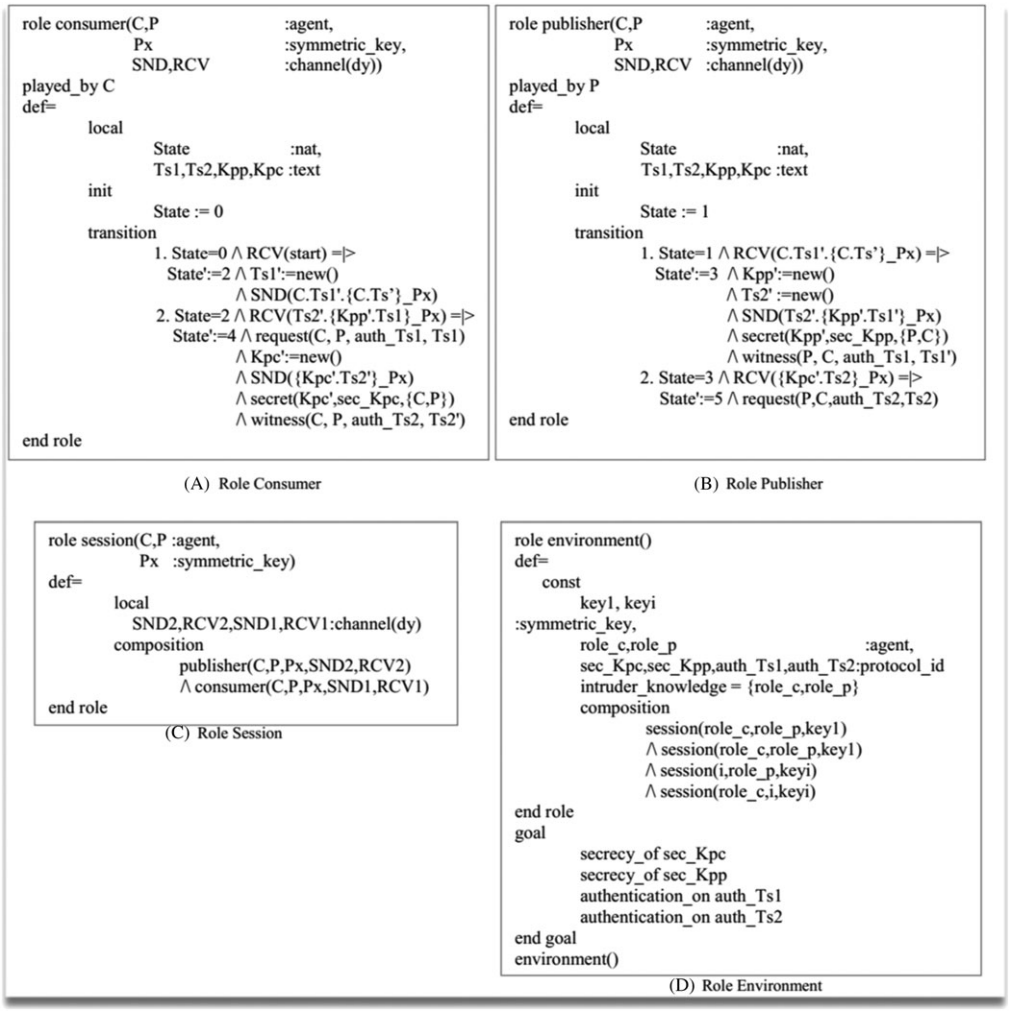

FIGURE 9 Roles of the authentication phase in AVISPA

7.2 | Simulation result of ECC-based Consumer's password change protocol

The two basic roles: *consumer* and *publisher* and two additional roles: *session* and *environment* for the proposed ECC-based consumer's password change protocol are given in Figure 11. The goal section of role *environment* ensures the secrecy/confidentiality of PW_{CM} and PW'_{CM}, and mutual authentication between two parties on the freshness of TS_3. The simulation result of the protocol executed in OFMC back-end and CL-AtSe back-end is given in Figure 12 with the output "safe".

8 | FORMAL VERIFICATION USING BAN LOGIC

In this section, a formal verification of the proposed protocol is done using the popular BAN logic. BAN logic[57] was first time proposed in 1990 and very soon became a prominent tool for proving the correctness of the authentication protocols. BAN logic is formalized on the many-sorted model logic and focused on the beliefs of the communicating parties involved in the authentication protocol. Now, we will discuss the logical notations, postulates used in BAN logic and the analysis of the proposed authentication protocol in the following subsections.

8.1 | BAN logical notations

The usual BAN logical notations are presented in the following Table 2.

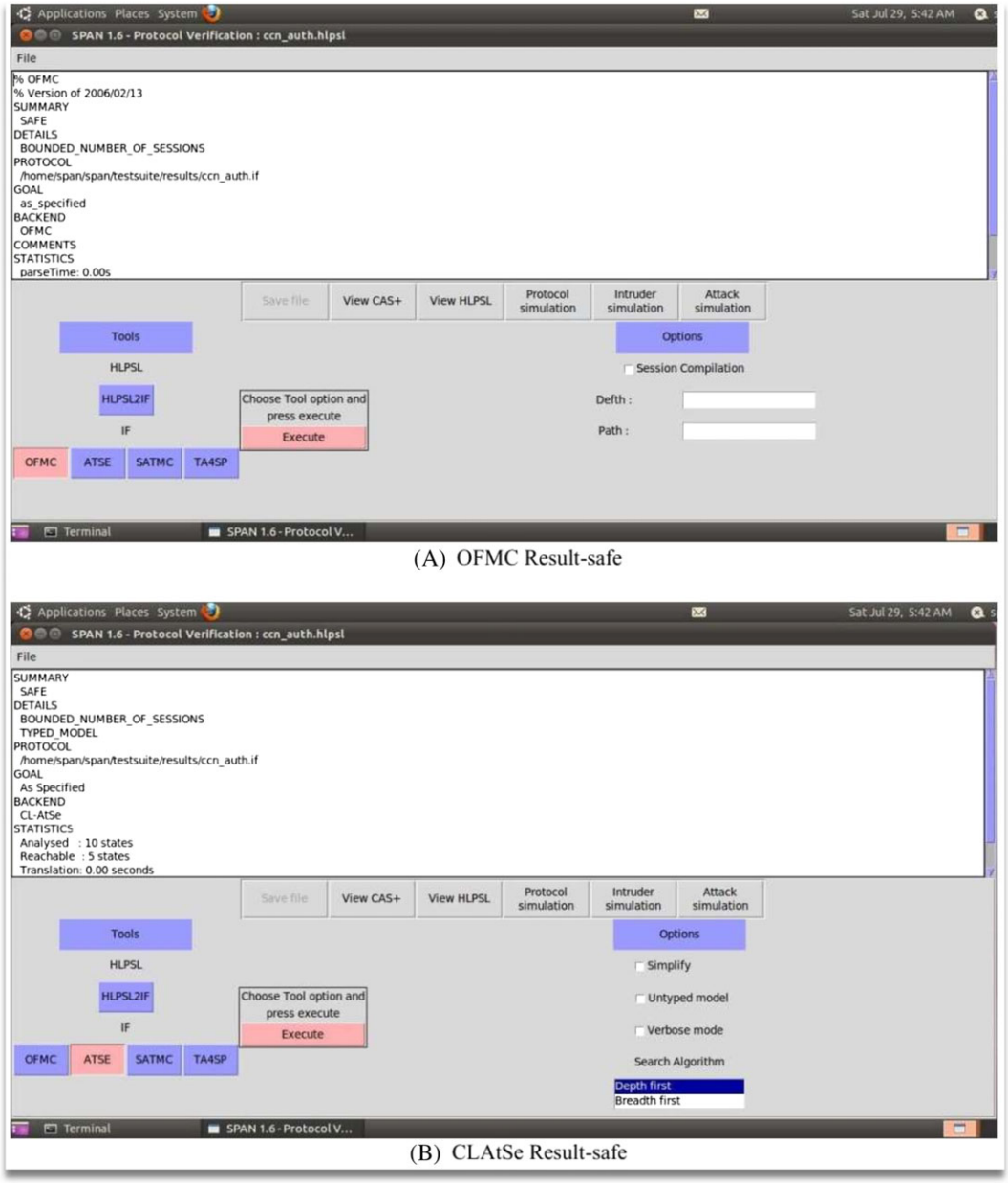

FIGURE 10 Protocol simulation results of authentication phase in AVISPA

8.2 | BAN logical postulates

BAN logic has many logical postulates but among them only five rules are used in this paper and presented below.

R1: **Message Meaning Rule**

R1: $\frac{P\left|\equiv C\overset{K}{\leftrightarrow}P,\ P\triangleleft\{X\}_K\right.}{P|\equiv C|\sim X}$, states that if P believes K is shared between C and P, and P also sees X encrypted by K, then P believes C once said X.

R2: **Freshness Rule**

R2: $\frac{P|\equiv\ \#(X)}{P|\equiv\ \#(X,Y)}$, states that if P believes X is fresh then P believes that the entire formula (X, Y) is also fresh.

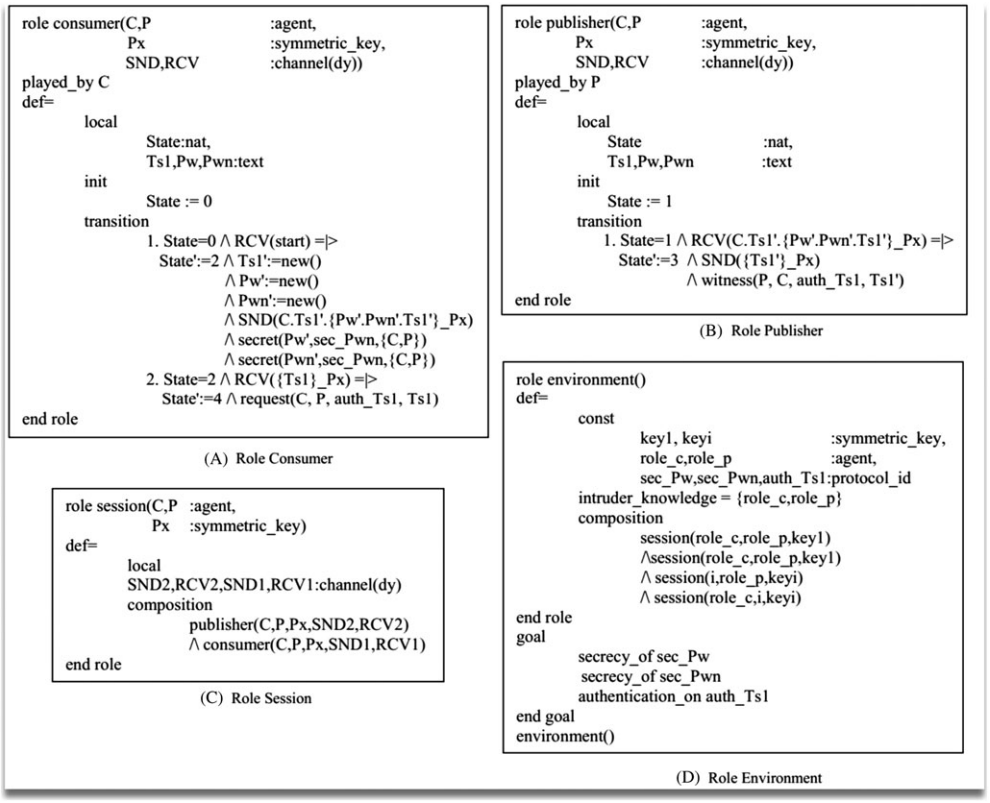

FIGURE 11 Roles of the password change phase in AVISPA

R3: **Nonce-verification Rule**

R3: $\frac{P|\equiv\ \#(X),\ P|\equiv C|\sim X}{P|\equiv C|\equiv X}$, states that if P believes X is fresh and C once said X, then P believes that C believes X.

R4: **Decomposition Rule**

$R4$: $\frac{P|\equiv C|\equiv (X,Y)}{P|\equiv C|\equiv X}$, states if P believes that C believes *(X, Y)*, then P believes that C believes X.

R5: **Jurisdiction Rule**

$R5$: $\frac{P|\equiv C|\equiv X, P|\equiv C=\ >X}{P|\equiv X}$, states if P believes that C believes X and C has control/jurisdiction over X, then P believes X.

8.3 | Analysis of the proposed authentication protocol using BAN logic

In this model, the consumer and the publisher are considered as the principals C and P respectively. The shared pre-secret key between C and P is P_X. TS_1 and TS_2 are the timestamp used in the communication. KP_P and KP_C are the contributory key parts shared by P and C respectively to negotiate a session key between them. Now, the idealized form of our protocol according to BAN logic, establishment of security goals, initial assumptions and proof of security goals are discussed below.

A. **Idealized Form**

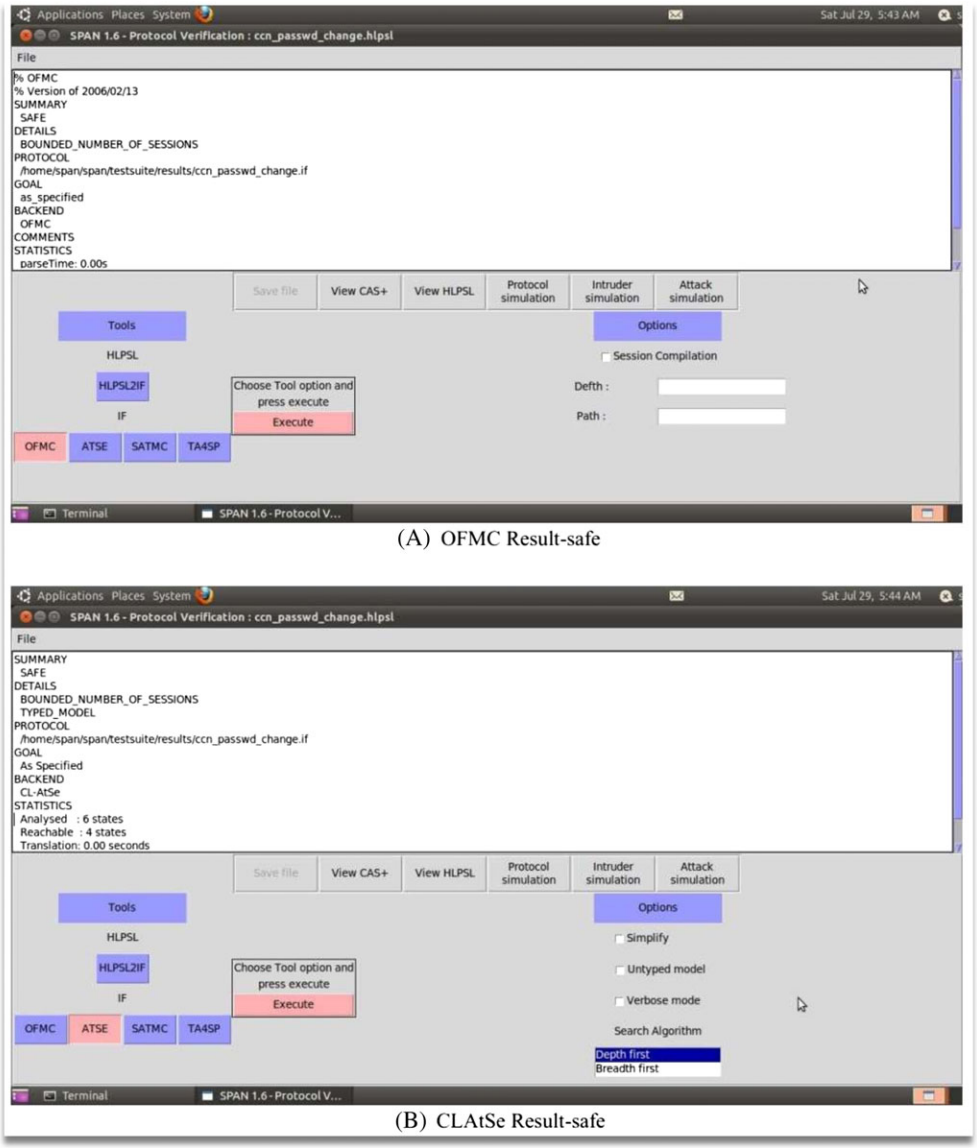

FIGURE 12 Protocol simulation results of password change phase in AVISPA

$$M\ 1\text{: } C \rightarrow P\text{: } TS_1, \quad \{C, TS_1\}_{P_X}$$
$$M\ 2\text{: } P \rightarrow C\text{: } TS_2, \quad \{TS_1, KP_P\}_{P_X}$$
$$M\ 3\text{: } C \rightarrow P\text{: } \{TS_2, KP_C\}_{P_X}$$

TABLE 2 BAN logical notations and their meanings

Notations	Meanings
C, P	Principals
$C\mid \equiv X$	C believes X
$C \lhd X$	C sees X
$C\mid\sim X$	C once said X
$C = > X$	C controls X
$\#(X)$	X is fresh
$C \overset{K}{\leftrightarrow} P$	K is shared key between C and P
$\{X\}_K$	X is encrypted by K

B. **Establishment of Security Goals**

G 1: $C \mid \equiv KP_P$
G 2: $P \mid \equiv KP_C$

C. **Initial Assumptions**

A 1: $C \mid \equiv \#(TS_1)$
A 2: $P \mid \equiv \#(TS_2)$
A 3: $C \mid \equiv C \overset{Px}{\leftrightarrow} P$
A 4: $P \mid \equiv C \overset{Px}{\leftrightarrow} P$
A 5: $C \mid \equiv P => KP_P$
A 6: $P \mid \equiv C => KP_C$

D. **Protocol Analysis**

To establish our security goals, we consider the following steps given in *step 1* to *step 12*.

Step 1. Initially, from message *M 2*, we get statement *S 1* as: $S1 : C \lhd \{TS_1, KP_p\}px$
Step 2. Now, in accordance with initial assumption *A 3* and statement *S 1*, we apply message meaning rule (*R1*) and get statement *S 2* as: $S\,2 : C \mid \equiv P \mid \sim(TS_1, KP_P)$.
Step 3. From initial assumption *A 1* and message *M 2*, we apply freshness rule (*R2*) and get statement *S 3* as: $S\,3 : C \mid \equiv \#(TS_1, KP_P)$.
Step 4. Now, from statement *S 2* and *S 3*, we apply nonce-verification rule (*R3*) and get statement *S 4* as: $S\,4 : C \mid \equiv P \mid \equiv (TS_1, KP_P)$.
Step 5. On statement *S 4*, we apply decomposition rule (*R4*) and get statement *S 5* as: $S\,5 : C \mid \equiv P \mid \equiv KP_P$.
Step 6. Finally, in accordance with our initial assumption *A 5* and statement *S 5*, we apply jurisdiction rule (*R5*) and get statement *S 6* as: $S\,6 : C \mid \equiv KP_P$ **(Goal *G 1*)**.
Step 7. $S7 : \mathrm{P} \lhd \{TS_2, KP_c\}_{px}$ From message *M 3*, we get statement *S 7* as:
Step 8. Now, in accordance with the initial assumption *A 4* and statement *S 7*, we apply message meaning rule (*R1*) to get statement *S 8* as: $S\,8 : P \mid \equiv C \mid \sim(TS_2, KP_C)$.
Step 9. From initial assumption *A 2* and message *M 3*, we apply freshness rule (*R2*) and get statement *S 9* as: $S\,9 : P \mid \equiv \#(TS_2, KP_C)$.
Step 10. Now, from statement *S 8* and *S 9*, we apply nonce-verification rule (*R3*) and get statement *S 10* as: $S\,10 : P \mid \equiv C \mid \equiv (TS_2, KP_C)$.
Step 11. On *S 10*, we apply decomposition rule (*R4*) and get statement *S 11* as: $S\,11 : P \mid \equiv C \mid \equiv KP_C$.
Step 12. In accordance with our initial assumption *A 6* and statement *S 11*, we apply jurisdiction rule (*R5*) and get statement *S 12* as: $S\,12 : P \mid \equiv KP_C$ **(Goal *G 2*)**.

Hence, through BAN logic, our security goals *G 1* and *G 2* are established and it is proved that the proposed proptocol achieves mutual authentication between *C* and *P*.

9 | PERFORMANCE ANALYSIS

In this section, the performance analysis of the proposed scheme is done in terms of computation and communication overhead as presented below.

9.1 | Computation cost

In the proposed scheme, the ECC is used to design all the communication protocols. It is well established that due to use of smaller key size (160-bits) to provide same level of security compared to other public key cryptosystems such as RSA

(1024 bits), ECC incurs low computation, communication and storage cost.[39-45] Moreover, due to the use of additive elliptic curve group, the operation such as scalar point multiplication becomes more efficient and cost effective than the modular exponentiation operation used in multiplicative group. As introduced in,[58] one-way hash operation is very fast, time for symmetric encryption/decryption is at least 100 times faster than asymmetric encryption/decryption, and time for elliptic curve point multiplication is much faster than modular exponentiation. For better understanding, the approximate time estimation of different cryptographic operations in milliseconds is considered from Kilinc et al...[59] and listed in Table 3. As discussed in Kilinc et al.,[59] the approximate running times of various cryptographic operations are calculated on a PC with Intel Pentium Dual CPU E2200 2.20GHz processor, 2048 MB of RAM and the Ubuntu 12.04.1 LTS 32bit operating system.

Thus, the computation cost (approximately estimated time in ms) of the proposed ECC-based mutual authentication and session key negotiation protocol is compared with other related existing schemes and given in Table 4 which shows that our scheme has low computation overhead (same as Qi et al.[25] and Park et al[38]) for both the mutual authentication and negotiation of the session key.

9.2 | Communication cost

In the proposed mutual authentication protocol, the consumer's identity ID_{CM}, publisher's identity ID_P, timestamps TS_1 and TS_2, symmetric encryption block size (according to AES algorithm approved by NIST in December, 2001), $Manifest_L$, $Manifest_{AckP}$ and $Manifest_{AckC}$ are assumed to be 128-bits long. Hence, in the proposed scheme, the total communication cost for the exchange of eight mutual authentication parameters is: $8 \times 128 = 1024$-bits. Further, the block size generated through one-way hash operation and RSA based modular exponentiation operation used in other related authentication schemes are assumed to be 128-bits and 1024-bits long, respectively. Considering these assumptions, the comparison of communication cost of the proposed ECC-based mutual authentication protocol with other related existing schemes[22,25,31-33,38] is given in Table 5 which shows that our scheme has low communication overhead for both the mutual authentication and negotiation of the session key.

Although Table 4 shows our scheme has same computation overhead as Qi et al.'s[25] and Park et al.'s[38] schemes, Table 5 shows our scheme has less communication overhead than Qi et al.'s[25] and Park et al.'s[38] schemes, and that justifies the importance of our work.

9.3 | Overall efficiency

The security strength and computation and communication overhead analysis are done by inspecting each technique in detail and thus, a comparative study on overall efficiency between the proposed scheme and other existing

TABLE 3 Approximate execution time (in milliseconds) of different cryptographic operations[59]

Notations	Description	Approx. Execution time (in ms)
T_h	Time for one-way hash operation	0.002
T_{PM}	Time for point multiplication	2.226
$T_{E/D}$	Time for symmetric encryption/decryption	0.004
T_{ME}	Time for modular exponentiation	3.850

TABLE 4 Computation cost comparison of the proposed scheme with other related works[22,25,31-33,38]

Schemes	Mutual authentication and session key negotiation	Total time (in ms)
Proposed scheme	$6T_{PM} + 6T_{E/D}$	**13.38**
Kalra et al...[22]	$9T_h + 8T_{PM}$ (only for authentication)	17.82
Qi et al.[25]	$6T_{PM} + 10T_h$	13.38
Chen et al.[31]	$5T_h + 3T_{ME} + 3T_{PM}$	18.23
Jiang et al.[32]	$5T_h + 5T_{ME} + 1T_{PM}$	21.48
Karuppiah et al[33]	$8T_{ME} + 2T_h$	30.80
Park et al.[38]	$12T_h + 6T_{PM}$	13.38

TABLE 5 Comparison of communication cost of the proposed ECC-based mutual authentication and session key negotiation scheme with other related works[22,25,31-33,38]

Schemes	Communication cost for mutual authentication (in number of bits)
Kalra et al.[22]	1280
Qi et al.[25]	1056
Chen et al.[31]	1792
Jiang et al.[32]	1792
Karuppiah et al.[33]	3968
Park et al.[38]	1376
Proposed scheme	**1024**

TABLE 6 Security comparison of the proposed scheme with other related works[22,25,31-33,38]

Security attributes	Schemes						
	22	25	31	32	33	38	Our scheme
Mutual authentication	No	Yes	Yes	Yes	Yes	Yes	**Yes**
Session key negotiation	No	Yes	Yes	Yes	No	Yes	**Yes**
Session key security	Yes	Yes	Yes	Yes	No	Yes	**Yes**
Forward secrecy	Yes	Yes	No	No	Yes	Yes	**Yes**
Impersonation attack resilience	No	Yes	Yes	Yes	Yes	Yes	**Yes**
Replay attack resilience	Yes	Yes	Yes	Yes	Yes	Yes	**Yes**
Man-in-the-middle attack resilience	Yes	Yes	Yes	Yes	Yes	Yes	**Yes**
Fully operates on insecure channel	No	No	No	No	No	No	**Yes**
Offline password guessing resilience	No	No	No	No	Yes	Yes	**Yes**

schemes[22,25,31-33,38] is summarized in Table 6. It can be noted from the above comparisons that the proposed scheme outperforms the existing schemes in terms of less computation and communication costs with higher security as well as greater overall efficiency. On the other hand, the proposed scheme can be compatible with the existing IP-based network infrastructure by incorporating certain modification in the network layer. The modification may include the replacement of IP-based routing with name based routing and incorporation of the network caching mechanism. Moreover, the incremental CCN LAN deployment in the existing IP-based Internet can be done by using a CCN gateway node for packet conversion.

10 | CONCLUSION

A flexible business model for content centric network with its security measures is proposed in this paper to preserve the business interests of content publishers/providers that seems to be first time proposed. In this scheme, two different types of business models namely *model-1* and *model-2* are designed that can be run simultaneously by the publisher for general and exclusive content provisioning. The security of the models are ensured using ECC-based protocols in which consumer registration, mutual authentication, session key negotiation and consumer's password change provisions are provided. All the proposed protocols are securely operated in insecure channel and are mathematically analyzed to show the strong resilience against relevant cryptographic attacks. Moreover, all the proposed protocols are formally verified using well accepted AVISPA simulator and BAN logic and found well secured. Finally, the performance analysis shows that the proposed scheme is efficient and thus, it preserves the business policy of CCN in terms of revenue generation.

ACKNOWLEDGEMENT

This publication is an outcome of the R&D work undertaken as project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation. We are also thankful to the anonymous reviewers and the editors of this journal for their insightful comments and suggestions that makes this article a valuable contribution.

ORCID


Sangram Ray http://orcid.org/0000-0002-1920-4172

REFERENCES

1. *Cisco Visual Networking Index: Forecast and Methodology*, pp.2012–2017, May 29, 2013.
2. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard, RL. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies,* ACM. 2009. pp. 1-12.
3. Lixia Z, Alexander A, Jeffrey B, et al. Named data networking. *ACM SIGCOMM Comp Commun Rev*. 2014;44(32):66-73.
4. Smetters D, Jacobson V. *Securing network content*. Technical report, PARC 2009.
5. Mahadevan P. *CCNx 1.0 Tutorial*. PARC, Tech. Rep. 2014.
6. Kurihara J, Uzun E, Wood CA. An encryption-based access control framework for content-centric networking. In *IFIP Networking Conference (IFIP Networking)*. 2015 pp. 1-9.
7. Ghodsi A, Koponen T, Rajahalme J, Sarolahti P, Shenker S. Naming in content-oriented architectures. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*. 2011. pp. 1-6.
8. Zhang G, Li Y, Lin T. Caching in information centric networking: a survey. *Comput Netw*. 2013;57(16):3128-3141.
9. Bari MF, Chowdhury S, Ahmed R. A survey of naming and routing in information-centric networks [J]. *IEEE Commun Mag*. 2012;50(12):44-53.
10. Zhang X, Chang K, Xiong H. Towards name-based trust and security for content-centric network [C], In Proceedings of 19th IEEE International Conference on Network Protocols (ICNP) 2011. pp. 1-6.
11. Qiao X, Chen J, Tan W, Dustdar S. Service provisioning in content-centric networking: challenges, opportunities, and promising directions. *IEEE Internet Comput*. 2016;20(2):26-33.
12. Lamport L. Password authentication with insecure communication. *Commun ACM*. 1981;24(11):770-772.
13. Mitchell CJ, Chen L. Comments on the S/KEY user authentication scheme. *ACM SIGOPS Oper Syst Rev*. 1996;30(4):12-16.
14. Amin R. Cryptanalysis and efficient dynamic id based remote user authentication scheme in multi-server environment using smart card. *IJ Netw Secur*. 2016;18(1):172-181.
15. Amin R, Biswas G. Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment. *Wireless Pers Commun*. 2015;84(1):439-462.
16. Irshad A, Sher M, Nawaz O, Chaudhry SA, Khan I, Kumari S. A secure and provable multi-server authenticated key agreement for TMIS based on Amin et al. scheme. *Multimed Tools Appl*. 2016;1-27. https://doi.org/10.1007/s11042-016-3921-1
17. Mohit P, Amin R, Karati A, Biswas G, Khan MK. A standard mutual authentication protocol for cloud computing based health care system. *J Med Syst*. 2017;41(4):50.
18. He D. An efficient remote user authentication and key agreement protocol for mobile client–server environment from pairings. *Ad Hoc Netw*. 2012;10(6):1009-1016.
19. Amin R, Islam SH, Biswas G, Khan MK, Kumar N. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Gener Comput Syst*. 2016;80:483-495. https://doi.org/10.1016/j.future.2016.05.032
20. Kalra S, Sood SK. Elliptic curve cryptography: survey and its security applications. In *Proceedings of the International Conference on Advances in Computing and Artificial Intelligence*. 2011. pp. 102-106.
21. Wu ST, Chiu JH, Chieu BC. ID-based remote authentication with smart cards on open distributed system from elliptic curve cryptography. In: IEEE International Conference on Electro Information Technology. 2005 pp 5.
22. Kalra S, Sood SK. Secure authentication scheme for IoT and cloud servers. *Pervasive Mobile Comput*. 2015;24:210-223.
23. Abi-Char PE, Mhamed A, El-Hassan B. A fast and secure elliptic curve based authenticated key agreement protocol for low power mobile communications. In: The International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST'07). (2007). pp 235–240.

24. Tian X, Wong DS, Zhu RW. Analysis and improvement of an authenticated key exchange protocol for sensor networks. *IEEE Commun Lett*. 2005;9(11):970-972.
25. Qi M, Chen J. An efficient two-party authentication key exchange protocol for mobile environment. *Int J Commun Syst*. 2017;2017:8.
26. Chandrakar P, Om H. A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC. *Comput Commun*. 2017;42:765.
27. Chaudhry SA, Naqvi H, Farash MS, Shon T, Sher M. An improved and robust biometrics-based three factor authentication scheme for multiserver environments. *J Supercomput*. 2015;1-17. https://doi.org/10.1007/s11227-015-1601-y
28. Lu Y, Li L, Peng H, Yang Y. A biometrics and smart cards-based authentication scheme for multi-server environments, Secur. *Commun Netw*. 2015;8(17):3219-3228.
29. Chaturvedi A, Das AK, Mishra D, Mukhopadhyay S. Design of a secure smart card-based multi-server authentication scheme. *J Inf Security Appl*. 2016;30:64-80.
30. Wei J, Liu W, Hu X. Cryptanalysis and improvement of a robust smart card authentication scheme for multi-server architecture. *Wireless Pers Commun*. 2014;77(3):2255-2269.
31. Chen BL, Kuo WC, Wuu LC. Robust smart-card-based remote user password authentication scheme. *Int J Commun Syst*. 2014;27(2):377-389.
32. Jiang Q, Ma J, Li G, Li X. Improvement of robust smart-card-based password authentication scheme. *Int J Commun Syst*. 2015;28(2):383-393.
33. Karuppiah M, Saravanan R. Cryptanalysis and an improvement of new remote mutual authentication scheme using smart cards. *J Discret Math Sci Cryptogr*. 2015;18(5):623-649.
34. Kumar R, Amin R, Karati A, Biswas GP. Secure remote login scheme with password and smart card update facilities. In *Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA)*. Springer. 2016 Pp. 495-505.
35. Karuppiah M, Pradhan A, Kumari S, Amin R, Rajkumar S, Kumar R. Security on "Secure Remote Login Scheme with Password and Smart Card Update Facilities". In *International Conference on Mathematics and Computing*. 2017 pp. 26-33.
36. Li X, Niu J, Bhuiyan MZA, Wu F, Karuppiah M, Kumari S. A robust ECC based provable secure authentication protocol with privacy protection for industrial internet of things. *IEEE Trans Ind Inf*. 2017;14:3599-3609.
37. Karati A, Islam SH, Biswas GP, Bhuiyan MZA, Vijayakumar P, Karuppiah M. Provably secure identity-based signcryption scheme for crowdsourced industrial internet of things environments. *IEEE Internet Things J*. 2017;5:2904-2914.
38. Park K, Park Y, Park Y, Das AK. 2PAKEP: provably secure and efficient two-party authenticated key exchange protocol for Mobile environment. *IEEE Access*. 2018;6:30225-30241.
39. Hankerson D, Menezes AJ, Vanstone S. *Guide to elliptic curve cryptography*. Heidelberg: Springer Science and Business Media; 2006.
40. Stallings W. *Cryptography and network security: principles and practices*. 4th ed. Upper Saddle River: Prentice Hall; 2009:420-430.
41. Miller VS. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*. Springer Berlin Heidelberg. 1985. pp. 417-426.
42. Koblitz N. Elliptic curve cryptosystem. *J Math Comput*. 1987;48(177):203-209.
43. Gupta V, Gupta S, Chang S, Stebila D. Performance analysis of elliptic curve cryptography for SSL. In *Proceedings of the 1st ACM workshop on Wireless security*. ACM. 2002. pp. 87-94.
44. Gura N, Patel A, Wander A, Eberle H, Shantz SC. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer Berlin Heidelberg. 2004. pp. 119-132.
45. Lauter K. The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless Commun*. 2004;11(1):62-67.
46. Ray S, Biswas GP, Dasgupta M. Secure multi-purpose Mobile-banking using elliptic curve cryptography. *Wireless Pers Commun*. 2016;90(3):1331-1354.
47. Diffie W, Hellman M. New directions in cryptography. *IEEE Trans Inf Theory*. 1976;22(6):644-654.
48. Ray S, Biswas GP. An ECC based public key infrastructure usable for mobile applications. In *Proceedings of the second international conference on computational science, engineering and information technology*. ACM. 2012. pp. 562-568.
49. Ray S, Biswas GP. Establishment of ECC-based initial secrecy usable for IKE implementation. In *Proc. of World Congress on Expert Systems (WCE)*. Vol(1) 2012 pp.6.
50. Biswas GP. Establishment of authenticated secret session keys using digital signature standard. *Inf Security J*. 2011;20(1):9-16.
51. AVISPA. Automated validation of Internet security protocols and applications. http://www.avispa-project.org/ (accessed July 2017).
52. HLPSL Tutorial. *A Beginner's Guide to Modelling and Analysing Internet Security Protocols* by AVISPA team. 2016 pp. 3. June 30.
53. Armando A, Basin D, Boichut Y, et al. The AVISPA tool for the automated validation of internet security protocols and applications. In *International conference on computer aided verification*. Springer, Berlin, Heidelberg. 2005. pp. 281-285.
54. Basin D, Mödersheim S, Vigano L. OFMC: a symbolic model checker for security protocols. *Int J Inf Security*. 2005;4(3):181-208.

55. Turuani M. The CL-Atse protocol analyser. In *International Conference on Rewriting Techniques and Applications*. Springer, Berlin, Heidelberg. 2006. pp. 277-286.

56. SPAN. A Security Protocol Animator for AVISPA. http://people.irisa.fr/Thomas.Genet/span/ (accessed July 2017).

57. Burrows M, Abadi M, Needham RM. A logic of authentication. *Proc R Soc Lond A*. 1989;426(1871):233-271.

58. Schneier B. *Applied Cryptography, Protocols, Algorithms, and Source Code*. second ed. Wiley; 1996.

59. Kilinc HH, Yanik T. A survey of SIP authentication and key agreement schemes. *IEEE Commun Surv Tutorials*. 2014;16(2):1005-1023.

How to cite this article: Adhikari S, Ray S, Biswas GP, Obaidat MS. Efficient and secure business model for content centric network using elliptic curve cryptography. *Int J Commun Syst*. 2019;32:e3839. https://doi.org/10.1002/dac.3839