A Lightweight and Secure IoT Communication Framework in Content-Centric Network Using Elliptic Curve Cryptography

Sharmistha Adhikari and Sangram Ray

Abstract In the recent era, content-centric network (CCN) is emerging as a future Internet paradigm to leverage scalable content distribution. Similarly, the Internet of things (IoT) is another upcoming technology which integrates as well as manages heterogeneous connected devices over the Internet. Recent literature have shown that IoT architecture can efficiently perform if it is implemented in CCN environment. In addition, considering the openness of the Internet used in IoT communication and limited capacity of IoT devices, security becomes a serious challenge which demands attention of the research community. In this paper, our main objective is to design a secure IoT communication framework that operates in CCN. A certificateless public key infrastructure is designed for our resource-constrained IoT communication framework in CCN. We have incorporated elliptic curve cryptography (ECC), a state-of-the-art lightweight cryptosystem, to ensure security of the proposed scheme. Finally, an in-depth security analysis confirms that the proposed scheme is resistant to various relevant cryptographic attacks.

Keywords Content-centric network (CCN) · Internet of things (IoT)
Elliptic curve cryptography (ECC)

1 Introduction

CCN and its importance: Since 2009, content-centric network (CCN) is envisaged as a future Internet architecture to cope up with the ever-increasing need for information exchange in current technological era. CCN is designed to leverage scalable content distribution over the Internet. In CCN, information/content gets more importance

S. Adhikari (✉) · S. Ray
Department of Computer Science and Engineering,
National Institute of Technology Sikkim, Sikkim 737139, India
e-mail: sharmistha.adhikari@gmail.com

S. Ray
e-mail: sangram.ism@gmail.com

A. Khare et al. (eds.), *Recent Trends in Communication, Computing, and Electronics*, Lecture Notes in Electrical Engineering 524,
https://doi.org/10.1007/978-981-13-2685-1_21

than the host where it is available and security is provided separately on the piece of content [1–4]. CCN uses human-readable hierarchical content naming approach to uniquely identify any content over the network [3]. Thus, CCN replaces the IP-based routing with name-based routing and uses content caching mechanism in intermediate routers to reduce network latency time. CCN router uses a limited buffer called content source (CS) to cache popular content for future use. Routers also maintain a forwarding information base (FIB) and a pending Interest table (PIT) to facilitate name-based routing. The major entities of CCN are content provider, publisher, consumer, and network router. CCN consumer sends Interest packet as a request for content. Any CCN router which receives the Interest checks the availability of the requested content in its CS. If it is available, the router consumes the Interest and sends back the requested content to the respective consumer. Otherwise, the Interest is forwarded to the content provider. Content provider supplies the content to its interface publisher for publishing in CCN. Finally, the requested content is forwarded to the requesting consumer following the reverse Interest path. As CCN enhances smooth delivery of content over the Internet, it can easily support the huge need for data/content exchange in Internet of things (IoT) framework.

IoT and its importance: With the advancement in consumer electronics, wireless communication, and intelligent sensors, IoT is becoming a reality where several heterogeneous things are interconnected through the Internet for information exchange and operated remotely without any human intervention. One of the most significant challenges in IoT framework is security which includes infrastructure-level security, application-level security, general system security, and communication security [5–11]. Now, a brief discussion on IoT communication framework is given here. IoT framework involves three significant entities, namely IoT devices, gateway server, and IoT end users. IoT device includes intelligent sensor devices and actuators which are deployed in the field such as a smart factory, smart home, hospital (in case of smart health care). In IoT network, intelligent sensor devices gather data from the environment and feed them to the gateway server. IoT gateway server manages security of the data communication and gives access of the gathered data to the authorized users who in return may give some commands to the IoT actuators via gateway server.

Background study: CCN as a future Internet architecture is ideally designed to address the emerging technology like IoT and its challenges at its design level [5–11]. In this regard, different researchers [6–10] have focused on IoT as a challenging Internet technology that can be significantly benefitted from CCN. Later, Suarez et al. [11] proposed an IoT management architecture based on information-centric network. Though the authors have addressed the overall IoT management architecture and its different aspects, the paper [11] lacks the cryptographic details of security measures used. In this paper, we have outlined a CCN-based IoT framework and designed a detailed communication security architecture using ECC [12–16].

Our contribution: In this paper, we have addressed an IoT communication framework which operates in CCN environment. As security of IoT communication is an important challenge, we have proposed an ECC-based efficient and secure architecture for IoT communication. Considering the limited resource of IoT devices, a certificateless

ECC-based security architecture is designed that uses identity-based private keys for users and IoT devices. Moreover, a thorough security analysis is done to ensure that the proposed scheme is well protected from relevant cryptographic attacks.

Paper organization: The rest of the paper is organized as follows. In Sect. 2, the proposed scheme is presented and its security analysis is discussed in Sect. 3. Finally, Sect. 4 concludes the paper.

2 Proposed Scheme

In this section, we have designed a lightweight and secure IoT communication framework in CCN considering security of IoT communication as a major concern. The proposed IoT communication framework is depicted in Fig. 1 where the IoT gateway server is connected with the IoT devices and IoT end users using CCN framework. The connection is usually achieved through wireless connectivity where gateway server acts as a trusted authentication server (AS) and handles the security measures of the IoT communication system.

In the proposed scheme, we have used Interest, Content, and Manifest packets for communication among user, AS, and IoT device. The general structure of these packet types is shown in Fig. 2. In addition, general structures of two databases maintained by AS, namely IoT device database and IoT user database, are given in Fig. 3.

"Name prefix space" column of both the databases can have values for all the CCN namespaces involved in the IoT framework. The name prefix spaces are IoT_dev, IoT_user, IoT_admin, and IoT_as. The "commands" column of IoT device database has values like increase, decrease, print. The "device access permission" column stores the IoT device names for which the user has access. Similarly, "commands permitted" column lists the permitted commands which are used by a user to a particular IoT device. In our scheme, hierarchical CCN name is used for the IoT devices and end users. The example of CCN naming approach for the proposed IoT framework is shown in Fig. 4.

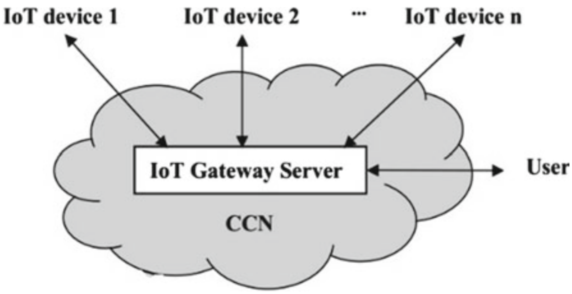


Fig. 1 Proposed IoT communication framework in CCN

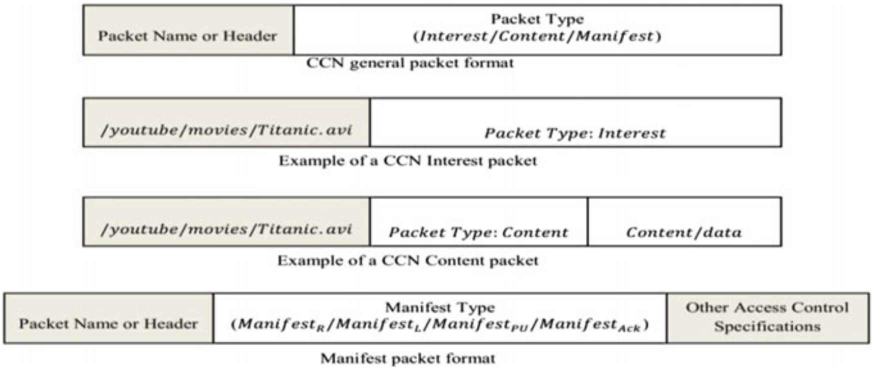
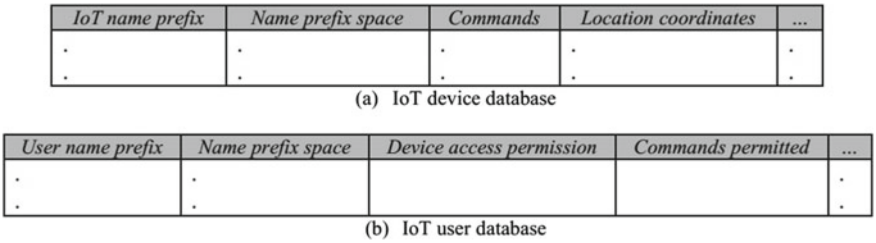

Packet Name or Header
Packet Type
(*Interest*/*Content*/*Manifest*)
CCN general packet format
/youtube/movies/Titanic.avi
Packet Type: *Interest*
Example of a CCN Interest packet
/youtube/movies/Titanic.avi
Packet Type: *Content*
Content/*data*
Example of a CCN Content packet
Packet Name or Header
Manifest Type
$(Manifest_R/Manifest_L/Manifest_{PU}/Manifest_{Ack})$
Other Access Control Specifications
Manifest packet format

Fig. 2 General structure of different CCN packets



IoT name prefix	*Name prefix space*	*Commands*	*Location coordinates*	...
.	.	.	.	.
.	.	.	.	.

(a) IoT device database

User name prefix	*Name prefix space*	*Device access permission*	*Commands permitted*	...
.	.			.
.	.			.

(b) IoT user database

Fig. 3 General structure of IoT device database and IoT user database

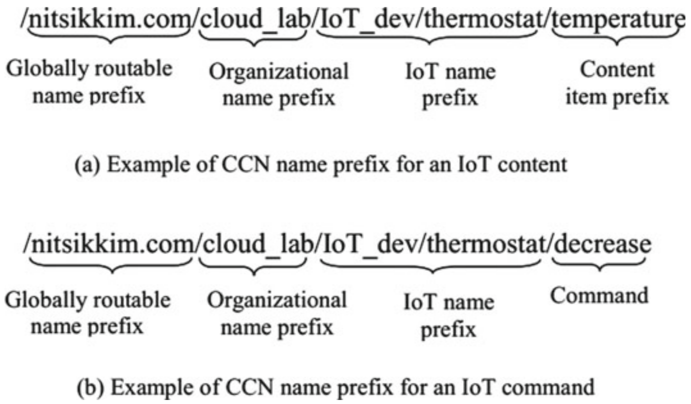

/nitsikkim.com/cloud_lab/IoT_dev/thermostat/temperature
Globally routable name prefix
Organizational name prefix
IoT name prefix
Content item prefix
(a) Example of CCN name prefix for an IoT content
/nitsikkim.com/cloud_lab/IoT_dev/thermostat/decrease
Globally routable name prefix
Organizational name prefix
IoT name prefix
Command
(b) Example of CCN name prefix for an IoT command

Fig. 4 Example of CCN naming approach used for proposed IoT framework

Multiple end users may access an IoT framework but depending on their authorization, specific access is allowed by the gateway/AS. We have used Manifest packet

[4] to share access control information between the user and the AS. Manifest is also used by user to send commands to the IoT devices via AS. The proposed security framework is divided into three phases, namely (1) system initialization phase, (2) registration phase, and (3) authentication and session key negotiation phase. The details of these three phases are briefly described in the following subsections.

2.1 System Initialization

In this phase, the AS selects all the security parameters and publishes them in the respective CCN where the IoT framework operates. AS selects an elliptic curve E with prime order p and generator G and two secure one-way hash function h and h_1 where h_1: $\{0, 1\}^* \Rightarrow \mathbb{Z}_p^*$. AS also chooses a random secret $s \in \mathbb{Z}_p^*$ and calculates its public key $PU_{AS} = s \cdot G$. Finally, AS publishes the security parameters as $\{E, p, G, h, h_1, PU_{AS}\}$.

2.2 Registration Phase

Initially, all the users and the IoT devices need to register to the AS in order to be included in the IoT framework. The registration phase is conducted through a secure channel where the user and IoT device get their private keys from AS depending on their CCN name prefix which is unique and treated as identity. The identity of any user or IoT device is verified by the AS at the time of registration. Any IoT device is registered at the time of its deployment in the network, and its private key gets hardcoded in the respective device through the registration procedure. The private key of the IoT device is used for symmetric encryption of the transmitted data between the IoT device and AS. In the registration process, user provides its identity ID_U to the AS and gets its private key $s \cdot h_1(ID_U)$. Similarly, an IoT device with identity ID_D gets its private key $s \cdot h_1(ID_D)$.

2.3 Authentication and Session Key Negotiation Phase

The user needs to login to AS in order to access IoT framework for which he/she is registered. After successful mutual authentication between the user and AS, a contributory secret session key is negotiated. The session key is changed in each session/user login to ensure security. Now, the detailed procedure is depicted in Fig. 5 and discussed stepwise where $X \rightarrow Y : M$ means sender X sends message M to receiver Y. Here, E/D_X means symmetric encryption/decryption using key X. In addition, dot operator (.) and concatenation operator (||) are used for ECC point multiplication and message concatenation, respectively.

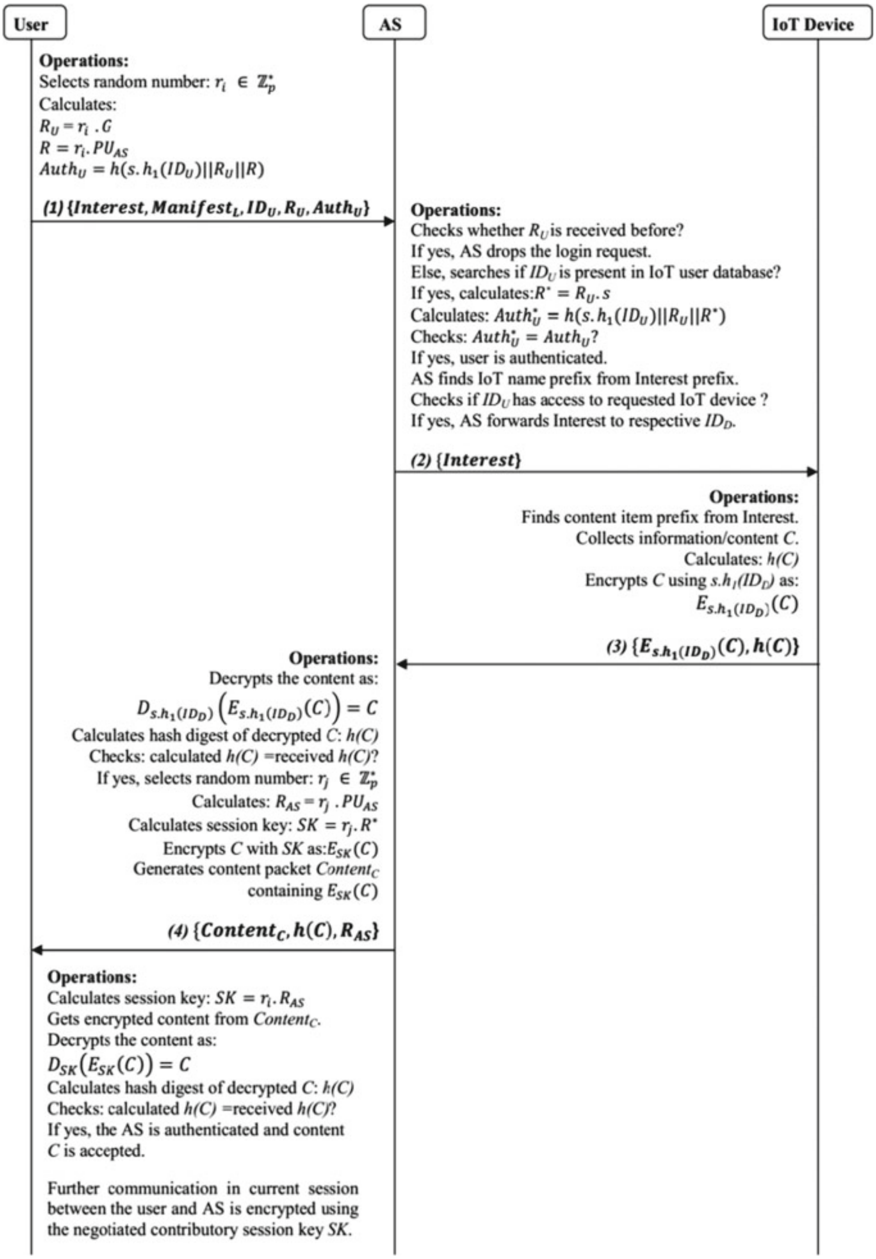

User
AS
IoT Device
Operations:
Selects random number: $r_i \in \mathbb{Z}_p^*$
Calculates:
$R_U = r_i \,.\, G$
$R = r_i . PU_{AS}$
$Auth_U = h(s.h_1(ID_U)||R_U||R)$
(1) $\{Interest, Manifest_L, ID_U, R_U, Auth_U\}$
Operations:
Checks whether R_U is received before?
If yes, AS drops the login request.
Else, searches if ID_U is present in IoT user database?
If yes, calculates: $R^* = R_U . s$
Calculates: $Auth_U^* = h(s.h_1(ID_U)||R_U||R^*)$
Checks: $Auth_U^* = Auth_U$?
If yes, user is authenticated.
AS finds IoT name prefix from Interest prefix.
Checks if ID_U has access to requested IoT device ?
If yes, AS forwards Interest to respective ID_D.
(2) $\{Interest\}$
Operations:
Finds content item prefix from Interest.
Collects information/content C.
Calculates: $h(C)$
Encrypts C using $s.h_1(ID_D)$ as:
$E_{s.h_1(ID_D)}(C)$
(3) $\{E_{s.h_1(ID_D)}(C), h(C)\}$
Operations:
Decrypts the content as:
$D_{s.h_1(ID_D)}\left(E_{s.h_1(ID_D)}(C)\right) = C$
Calculates hash digest of decrypted C: $h(C)$
Checks: calculated $h(C)$ =received $h(C)$?
If yes, selects random number: $r_j \in \mathbb{Z}_p^*$
Calculates: $R_{AS} = r_j \,.\, PU_{AS}$
Calculates session key: $SK = r_j . R^*$
Encrypts C with SK as: $E_{SK}(C)$
Generates content packet $Content_C$
containing $E_{SK}(C)$
(4) $\{Content_C, h(C), R_{AS}\}$
Operations:
Calculates session key: $SK = r_i . R_{AS}$
Gets encrypted content from $Content_C$.
Decrypts the content as:
$D_{SK}(E_{SK}(C)) = C$
Calculates hash digest of decrypted C: $h(C)$
Checks: calculated $h(C)$ =received $h(C)$?
If yes, the AS is authenticated and content
C is accepted.
Further communication in current session
between the user and AS is encrypted using
the negotiated contributory session key SK.

Fig. 5 Mutual authentication and session key negotiation between user and AS

Step 1: User ➔AS: $\{Interest, Manifest_L, ID_U, R_U, Auth_U\}$
Initially, the user selects a random number $r_i \in \mathbb{Z}_p^*$ and calculates $R_U = r_i \cdot G$, $R = r_i \cdot PU_{AS}$, and $Auth_U = h(s \cdot h_1(ID_U)||R_U||R)$. Finally, the user sends an Interest, requesting the content, a *Manifest*$_L$ for sending access control information regarding login process, along with identity ID_U, R_U, and authentication parameter $Auth_U$ to the AS. Here, Interest name prefix signifies the requested content name as shown in Fig. 4a.

Step 2: AS ➔ IoT Device: {***Interest***}
After receiving the authentication request from *step 1*, the AS initially checks whether R_U is received before. If yes, AS drops the login request. Else, searches if ID_U is present in IoT user database. If yes, AS calculates $R^* = R_U \cdot s$, $Auth_U^* = h(s \cdot h_1(ID_U)||R_U||R^*)$ and checks $Auth_U^* = Auth_U$? If yes, the user is authenticated; otherwise, AS drops the session. After successful authentication of the user, AS finds IoT name prefix from received Interest prefix and checks if ID_U has access to requested IoT device. If yes, AS forwards the Interest to the respective IoT device (ID_D).

Step 3: IoT Device ➔AS:$\{\boldsymbol{E_{s \cdot h_1(ID_D)}(C), h(C)}\}$
After receiving the Interest from *step 2*, the IoT device finds content item prefix from Interest name prefix. Then, it collects information/content C from the environment. It calculates $h(C)$ and also encrypts C using its private key $s \cdot h_1(ID_D)$ as $E_{s.h_1(ID_D)}(C)$. Finally, the IoT device sends the encrypted content to AS along with the content hash digest $h(C)$.

Step 4: AS ➔User:$\{\boldsymbol{Content_C, h(C), R_{AS}}\}$
After receiving in *step 3*, AS decrypts the content as $D_{s \cdot h_1(ID_D)}\big(E_{s \cdot h_1(ID_D)}(C)\big)$ and gets requested content C. Then, AS calculates hash digest of decrypted Cas $h(C)$, and checks if calculated $h(C)$ = received $h(C)$. If yes, selects a random number $r_j \in \mathbb{Z}_p^*$ and calculates $R_{AS} = r_j \cdot PU_{AS}$ and contributory session key $SK = r_j \cdot R^*$. Finally, AS encrypts C using *SK*as $E_{SK}(C)$, generates content packet *Content*$_C$ with the encrypted content, and sends *Content*$_C$, *h*(C) and key part R_{AS} to the user.

After receiving the message in *step 4*, the user calculates contributory session key $SK = r_i \cdot R_{AS}$. User gets encrypted content from *Content*$_C$ packet and decrypts as $D_{SK}(E_{SK}(C))$ to get C. The user also calculates hash digest of decrypted C as $h(C)$ and checks if calculated $h(C)$ = received $h(C)$. If yes, the AS is authenticated, *SK* is negotiated, and the received content C is accepted. Any further communication in the current session between the user and AS is encrypted using the negotiated contributory session key *SK*.

3 Security Analysis

In this section, several relevant cryptographic attacks are analyzed to show that our scheme is well protected.

3.1 Mutual Authentication

Mutual authentication is an important parameter of any security framework that is being taken care of in our scheme. In the proposed scheme, the user sends $Auth_U$ which is dependent on user's secret key and random value r_i and can be verified only by the AS. Accordingly, AS authenticates the user. Similarly, AS sends encrypted content, content hash digest $h(C)$, and key part R_{AS}. Upon calculating SK using R_{AS}, the user decrypts the content, calculates its hash digest, and verifies received $h(C)$ with calculated $h(C)$. If verified, the user authenticates AS. Thus, the mutual authentication is completed.

3.2 Confidentiality

Confidentiality property is maintained in our scheme as none of the secret keys, session key SK, requested content, or authentication parameter $Auth_U$ travels openly rather either they are hashed or encrypted.

3.3 Replay Attack Resilience

In our scheme, replay attack by an intruder is successfully prevented as in *step 2* of the authentication phase, AS checks whether R_U is received before. If yes, AS drops the login request. Here, value of R_U is dependent on a random value r_i which is changed in each session. Hence, any repetition in the value of R_U is detected by the AS.

3.4 Man-in-the-Middle Attack Resilience

As mutual authentication and confidentiality are maintained in the proposed scheme, any attempt of message modification or replay by an intruder will be identified by the user or AS. Hence, man-in-the-middle attack is prevented.

3.5 Perfect Forward Secrecy

Perfect forward secrecy is a property which ensures that even if the long-term keys are compromised at a point in time, the sessions before that time are still secure. As the proposed scheme uses random values r_i and r_j to calculate SK in each session, even if user's long-term key $(s \cdot h_1(ID_U))$ becomes compromised nobody will be able to compute SKs before that time.

3.6 Known Session Key Attack Resilience

Known session key attack means the knowledge of one session key reveals other session keys of different sessions. This is successfully prevented in our scheme because SK is calculated by using random values r_i and r_j which are changed in every session.

3.7 Brute Force Attack Resilience

Our scheme is resilient to brute force attack as an intruder cannot guess SK. Strength of SK is dependent on three secret random numbers r_i, r_j, and s from $\mathbb{Z}_p^*$. Moreover, SK is a point on the elliptic curve. Hence, based on the security strength of elliptic curve discrete logarithmic problem, it is impossible to guess SK in polynomial time.

4 Conclusion

In this paper, a CCN-based IoT communication framework is proposed. Considering the openness of wireless connectivity and limited capacity of resource-constrained IoT devices, we have presented a lightweight security framework for CCN-based IoT communication using ECC. As ECC uses efficient point multiplication operation and smaller key size (160-bits) than other public key cryptosystems such as RSA (1024-bits) to provide same level of security, our scheme incurs low computation and communication overheads. Moreover, as identity-based private keys are used for user and IoT devices, the overheads of public key certificate generation, verification, management, etc., are eliminated. Finally, an in-depth security analysis ensures that the proposed scheme is resilient against relevant cryptographic attacks.

Acknowledgments This work is financially supported by Visvesvaraya Ph.D. Scheme, Ministry of Electronics and Information Technology, Government of India.

References

1. Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., Braynard, R. L. (2009). Networking named content. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies* (pp. 1–12). ACM.
2. Smetters, D., Jacobson, V. (2009). *Securing network content* (pp. 2009–01). Technical report, PARC.
3. Mahadevan, P. (2014). *CCNx 1.0 Tutorial*. Technical report, PARC.
4. Kuriharay, J., Uzun, E., Wood, C. A. (2015). An encryption-based access control framework for content-centric networking. In *2015 IFIP Networking Conference* (*IFIP Networking*) (pp. 1–9). IEEE.
5. Li, S., Da Xu, L. (2017). *Securing the internet of things*. Syngress.
6. Corujo, D., Aguiar, R. L., Vidal, I., Garcia-Reinoso, J. (2012). A named data networking flexible framework for management communications. *IEEE Communications Magazine*, *50*(12).
7. François, J., Cholez, T., Engel, T. (2013). CCN traffic optimization for IoT. In *2013 Fourth International Conference on the Network of the Future* (*NOF*) (pp. 1–5). IEEE..
8. Baccelli, E., Mehlis, C., Hahm, O., Schmidt, T. C., Wählisch, M. (2014). Information centric networking in the IoT: experiments with NDN in the wild. In *Proceedings of the 1st ACM Conference on Information-Centric Networking* (pp. 77–86). ACM.
9. Quevedo, J., Corujo, D., Aguiar, R. (2014). A case for ICN usage in IoT environments. In *Global Communications Conference* (*GLOBECOM*) (pp. 2770–2775). IEEE.
10. Ahlgren, B., Lindgren, A., Wu, Y. (2016). Experimental Feasibility Study of CCN-lite on Contiki Motes for IoT Data Streams. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking* (pp. 221–222). ACM.
11. Suarez, J., Quevedo, J., Vidal, I., Corujo, D., Garcia-Reinoso, J., & Aguiar, R. L. (2016). A secure IoT management architecture based on Information-Centric Networking. *Journal of Network and Computer Applications, 63,* 190–204.
12. Hankerson, D., Menezes, A. J., Vanstone, S. (2006). *Guide to elliptic curve cryptography*. Springer Science and Business Media.
13. Stallings, W. (2006). *Cryptography and network security: principles and practices*. Pearson Education India.
14. Miller, V. S. (1985). Use of elliptic curves in cryptography. In: *Conference on the Theory and Application of Cryptographic Techniques* (pp. 417–426). Berlin, Heidelberg: Springer.
15. Koblitz, N. (1987). *Elliptic Curve Cryptosystems. Mathematics of Computation, 48*(177), 203–209.
16. Lauter, K. (2004). The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless Communications, 11*(1), 62–67.