A Robust Anonymity Preserving Authentication Protocol for IoT Devices

Aakanksha Tewari[1], B. B. Gupta[2], *Senior Member, IEEE*
National Institute of Technology, Kurukshetra Haryana, India
[1]tewariaakanksha29@gmail.com, [2]gupta.brij@gmail.com

***Abstract*-- In spite of being a promising technology which will make our lives a lot easier we cannot be oblivious to the fact IoT is not safe from online threat and attacks. Thus, along with the growth of IoT we also need to work on its aspects. Taking into account the limited resources that these devices have it is important that the security mechanisms should also be less complex and do not hinder the actual functionality of the device. In this paper, we propose an ECC based lightweight authentication for IoT devices which deploy RFID tags at the physical layer. ECC is a very efficient public key cryptography mechanism as it provides privacy and security with lesser computation overhead. We also present a security and performance analysis to verify the strength of our proposed approach.**

I. INTRODUCTION

Today almost every day to day household object is being connected to the Internet giving rise to the next generation paradigm termed as the Internet of things (IoT). This term was coined by Kevin Ashton in 1999, but it gained attention after 2005 and since then IoT has been evolving very rapidly. IoT is expected to be a major economic contributor in the coming years, it estimated that by the year 2020 the total economic value added by IoT ventures will be 1.9 trillion dollars. The IoT networks are made up of small nodes deployed in bulk, which may or may not be homogeneous, thus different capabilities of IoT devices in same network should also be taken into consideration [1]. In addition to the heterogeneity issues like memory, computation and energy limitations should also be considered while developing IoT applications [2,3].

IoT is one of the most influential domains of the 21st century. It has led to development of smart homes, smart cities, healthcare solutions, wearable technologies etc. One of the most commonly used technology with IoT is RFID sensors [4]. RFID chips can store useful data as well as track and identify other objects. It has several advantages over barcode as it can be processed without being in line of sight, it can also perform some computations and can read more than tags at the same time [5,6].

In this paper, we propose an authentication protocol for IoT devices using ECC, hash functions and a pseudo random number generator (PRNG). The server assigns a temporary key to the tags before authentication session which also updated after every session. The process of authentication is initiated by the tag reader which completes with the tag and reader authenticating each other and the server authenticating both tag and reader. For every session public and private key are generated by using the PRNG and ECC. In RFID authentication protocols privacy is achieved in terms of evaluated notions. These notions guarantee that the adversary cannot retrieve any significant tag information while interacting with tag or the reader, even if it has access to the tag's secret session parameters. Our primary goal is to design an RFID authentication protocol with a strong security mechanism.

In the next section, we discuss some significant protocols. The section 3 gives a detailed description of our proposed work. In the section 4 we discuss the strength of our protocol in terms of the level of security it provides which is followed by section 5 giving a performance analysis of our work. In section 6 we conclude our paper and discuss some future directions for our work.

II. RELATED WORK

The existing literature for authentication protocols for RFID technology comprises of a substantial number of works. Some of these protocols deploy public key cryptography while some others use lightweight mechanisms to protect these devices. The use of public key cryptography exploits a lot of resources on the other hand the symmetric key is not feasible to any network variations. ECC based approaches on the other hand are fully compatible with these devices as it requires a significantly small key to ensure security between the tag and reader or the server [7].

Tulys and batina [8,9] proposed the first authentication protocol for RFID devices using ECC in 2006. In [10] however some of the weaknesses of the protocol were addressed and an improved protocol was proposed. Lee et. al also gave some other ECC based security protocols for RFID tags [11, 12].

In [13] an ECC based mutual authentication approach which was easy to implement due to its simplicity as it did not use any complex hash function or other computations. However, Peter and Herman [14] analyzed the protocol and addressed that it was vulnerable to tracking, spoofing and cloning and does not ensure privacy. Chou [15] proposed an ECC based authentication with one-way hash. However, Farash et. al [16] proved that [15] is vulnerable to impersonation attacks.

Liu et. Al, [17] proposed an ECC based authentication scheme although Liao and Hsiao [18] found out that the adversary can compromise the secret key used by the protocol to steal information from the tag. They proposed another ECC based authentication protocol which could ensure privacy and anonymity and they also proved that their protocol was able to

978-1-5386-3025-9/18/$31.00 ©2018 IEEE

ensure security against spoofing tracking and cloning. However, Zhao et. al, [19] proved that [18] is also vulnerable to the theft of secret key to steal tag's information and gave a solution to this problem.

In [20] Tan et. al., presented a mutual authentication protocol using three-factor key exchange mechanism to secure the devices. However, the protocol failed to provide security against DoS and replay attacks. Arshad et. al [21] also gave a new ECC based authentication protocol to overcome the vulnerabilities in [20]. But in [22] Lu et. al., proved that [21] was vulnerable to password attack which can lead to tag impersonation.

III. Proposed Solution

A. Preliminaries

Initially *Elliptic Curve on Binary Field* [23]: F_2^m is represented by the equation:

$$y^2 + xy = x^3 + ax^2 + b;$$

The domain parameters of F_2^m are *m, f(x), a, b, P, n* and *h* where *m* is the length of the elements of the finite field which can be represented as a polynomial of degree *m-1*. *f(x)*is an irreducible polynomial of degree *m*. *P* is the prime order generator point ($P \neq 0$) of order n and is the cofactor:

$$h = \#E(F_2^m)/n,$$

where $\#E(F_2^m)$ is the total number of points in the elliptic curve [10].

We are assuming an EC of prime order *q* over F_p. In this case, for a point $R = [r_x, r_y]$, $[R]_x$ maps *R* to r_x mod *q*.

Pseudo random Number Generator: PRNGs are random number generators which work by retaining an internal state which comprises of a key and a seed. Formally PRNG we are using is defined as follows [24]:

Definition: A (base b) pseudo-random sequence generator *G* on seed space *X* is an effective map $G : X \rightarrow \Sigma^{\infty}$ such that for each integer $s \geq 0$, there is an integer $t \geq 0$ such that for all *(N, x)*$\subseteq$X with:

$$G_N(N,x) \neq [G(N,x')]^{n^s},$$

The initial segment of *G(N, x)* of length n^s , is output in time $O(n^t)$.

Thus, from short "seeds" (i.e., of length *n*), that are produced using at most *poly(n)* truly random bits, *G* generates long sequences (i.e., of length n^s), in polynomial time. *G(N, x)* is called the pseudo-random sequence generated by *G* with input or seed *(N, x)*.

B. System Setup

Initially the server determines an elliptic curve equation over field F_q, with *P* as it's generator point with order *n*. The reader selects a random number r_R, which is kept private and calculates a curve point R_R as $R_R = r_R P$ which is public.

For each tag the server selects a random number r_T and verifier as $R_T = r_T P$. Then, corresponding to each tag T_i the values $\{R_T, r_T\}$ are stored into their database. Each device also stores R_R in their memory.

Table I. Notations

Notation	Description
F_2^m	binary field
m	length of the elements of the finite field
f(x)	irreducible polynomial of degree *m*
P	prime order generator point *(P ≠ 0)*
R_T	tag's public key
r_T	tag's private key
R_R	reader's public key
r_R	reader's private key
K_T	temporary key updated after every session
SetKey()	set a new public key
TagInit()	initialize new session for the tag

C. Authentication Phase

The server initially assigns a temporary key K_T to the tag which is updated after every session. The private and public keys for tag and reader are (r_T, R_T) and (r_R, R_R) respectively. The authentication phase requires message exchanges and computations by the tag and the server, which is mentioned in the following steps as:

The reader initiates the authentication session by sending a hello message to the tag.

The reader first sends a *hello* message to the tag to initiate the session. After receiving the message from the reader, the tag generates two random numbers r_1 and r_2 and calculates:

$$R_1 = r_1 P; \text{ and } R_2 = r_2 P$$

It is followed by:

$$X_1 = (K_T + R_1 + R_T).$$

The tag then calculates an authentication parameter:

$$A_1 = | r_T | X_1 |_x P |.$$

Then, $< A_1, R_1, R_2, X_1, K_T >$ are sent to the reader.

On receiving tag's response, the reader sends $< K_T >$ to the server. The server first verifies the key K_T from the database. If match is not found then the process is aborted immediately. Else, if the match is found then, server sends a message to the reader notify successful authentication so that reader can proceed.

The reader now calculates:

$$R_T' = X_1 - (K_T + R_1)$$

It is followed by

$$r_T = P^{-1}(R_T).$$

To verify that the values are correct the reader calculates:

$$r_T' = A_1 (r_1 + r_T + K_T [P^{-1}])^{-1}.$$

If $r_T' = r_T$ the tag is legitimate. The reader then calculates:

$$X_2 = (R_2 + X_1 + R_R)$$

and an authentication parameter:

$$A_2 = | r_R | r_2 R_T |_x |.$$

Then, $< X_1, A_2 >$ are sent to the tag.

On receiving reader's response the tag then calculates:

$$R_R' = X_2 - (R_2 + X_1)$$

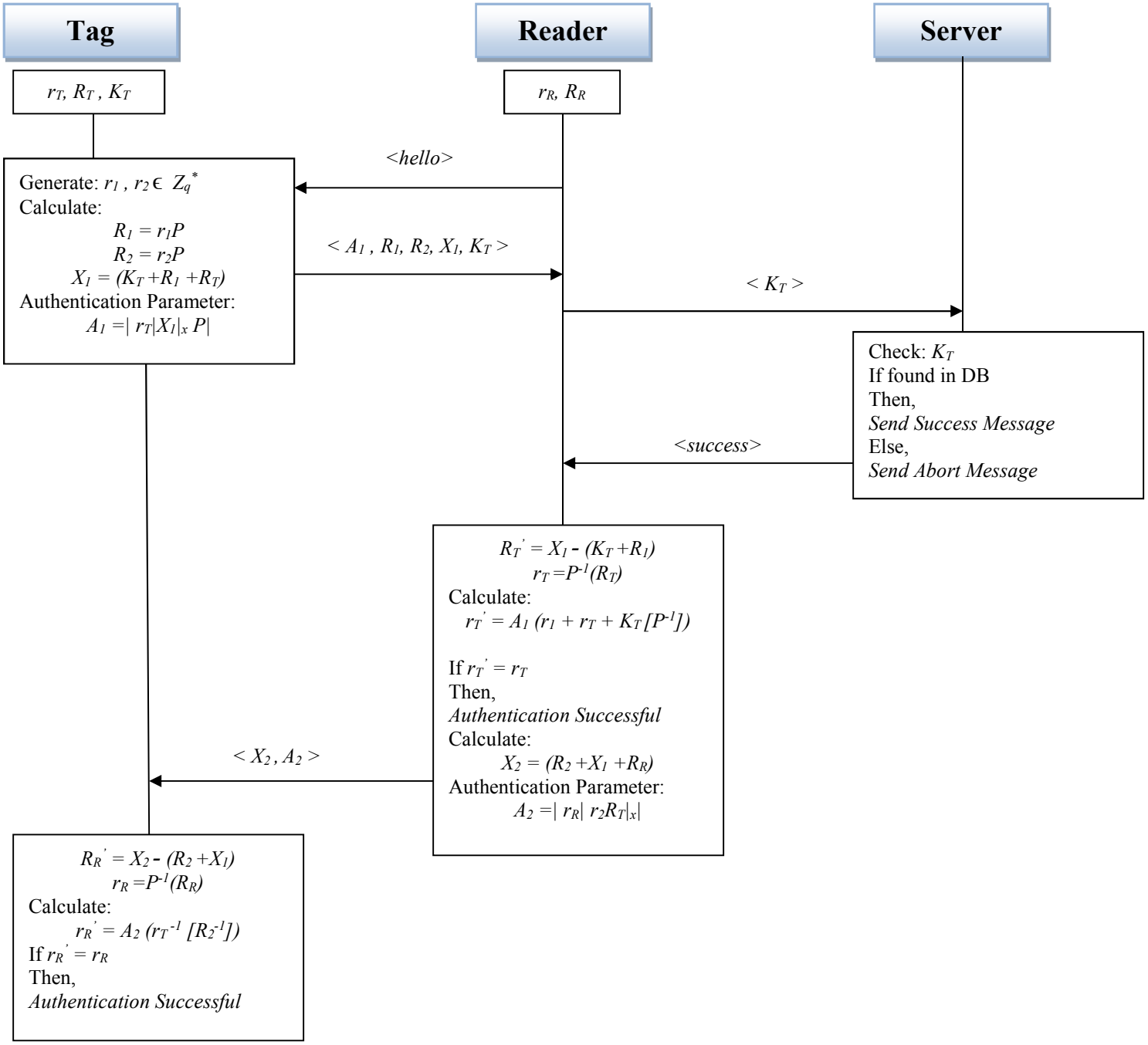

Figure 1. Flowgraph of the protocol

It is followed by

$$r_R = P^{-1}(R_R).$$

Now, if $r_R' = r_R$, then the authentication is successful.

IV. Security Analysis

In this section we give a brief security analysis of our protocol to show that it ensures privacy and is able to defend against various attacks. We provide two cases here, by which we will show that our protocol is secure.

Case I: (Secrecy of R_T and R_R) The values R_T and R_R are only known to the tag and the reader and when the tag sends the authentication parameter A_1 to the reader it encrypts the value of R_T with $X_1 = (K_T + R_1 + R_T)$ which is known to the tag only. Thus, if the attacker obtains A_1, they cannot be able to get R_T from it. Similar is the case with R_R which is sent by the reader to the tag encrypted as $X_2 = (R_2 + X_1 + R_R)$.

Case II: (Freshness of A_1 and A_2) The values A_1 and A_2 are fresh in every session which are obtained by encrypting the public keys with freshly generated points as: $A_1 = | r_T | X_1 |_x P |$ and $A_2 = | r_R | r_2 R_T |_x P |$ in every session.

Case III: (Correctness)

Proof: Since, $r_T' = A_1 (r_1 + r_T + K_T [P^{-1}])^{-1}$

$= [r_T | X_1|_x P] (r_1 + r_T + K_T [P^{-1}])^{-1}$

$= [r_T | X_1|_x P] [P^{-1}] (R_1 + R_T + K_T)^{-1}$

$= [r_T | X_1|_x] (R_1 + R_T + K_T)^{-1} [P P^{-1}]$

$= r_T X_1 (R_1 + R_T + K_T)^{-1}$ {Since, $X_1 = (K_T + R_1 + R_T)$}

$= r_T X_1 (X_1)^{-1} = r_T$

Similarly, $r_R' = A_2 (r_T^{-1} [R_2^{-1}])$

$= A_2 (r_T^{-1} [r_2 P]^{-1})$

$= A_2 (r_T^{-1} r_2^{-1} P^{-1})$

$= A_2 (r_2^{-1} [r_T P]^{-1})$

$= A_2 (r_2^{-1} R_T^{-1})$

$= (r_R r_2 R_T) (r_2^{-1} R_T^{-1})$ {Since, $A_2 = | r_R | r_2 R_T |_x |$}

$= r_R$

Case IV: (Game-based Security Evaluation) In this case we assume that an adversary A is able to break the security of our protocol with some non-negligible probability.

We assume that the tag (i.e., T_1) to be used by the adversary is selected during the game initialization phase. The adversary A is initialized as follows:

- Set a new public key $X = SetKey( )$ for the tag T_1.
- A now simulates *TagInit()* query for T_1 to initialize new session for the tag. For the i^{th} iteration:
- First execution of *TagInit()* returns:

$$R_2 = r_2 P, \text{ where } r_2 \in Z_q^*$$

- Then, it tries to guess the value of R_1 from the above data.
- The second execution of *TagInit()* is as follows:
  Set Authentication Parameter:

$$\{A_1 = | r_T | X_1 |_x P |\}_i$$

  Return:

$$\{X_1 = (K_T + R_1 + R_T)\}_i$$

- During the second phase:
  Result(): computes A_1, X_1 and store (A_1, X_1).
- The iterations are now backtracked until the value of (R_1, R_2) is obtained from the call.
- From the values (R_1, R_2) return (r_1', r_2').
- Calculate: r_T using: $r_T' = A_1 ([X_1] [P'])^{-1}$

For the above game simulation the set up will not be able to correctly guess (r_1', r_2') due to the Discrete logarithm problem (DLP) which states that:
Assuming P to be the generator of a group G_q and let R be an element belonging to G_q. The DLP is to obtain an integer value $r : R = rP$, which is a hard problem.
Thus, our protocol is secure due to the notion of DLP.

Ensuring privacy: Our protocol provides mutual authentication as both the server and the tag are able to ensure the other one is legitimate via the authentication parameters which are fresh in every session. By case I and II we can also ensure the confidentiality and anonymity of our protocol. Even if the attacker obtains the values of R_T and R_R from the previous session due to case II forward secrecy remain as the information cannot be tracked back.

Security against various attacks: The secret keys that are computed initially are stored in the database and are not updated during any other computation thus making it possible to execute the authentication mechanism between the server and the device any time. The prevention from updating this data also prevents the DoS attacks. Also, from case II (Freshness of A_1 and A_2) we can ensure our protocol is secure from tracking, replay and de-synchronization attacks.

V. Performance Analysis

Due to the low-resource and constrained environments in which the IoT devices have to perform, the performance of the protocol is also an important concern. It is necessary to take into account the cost of computations using the random number generation and point additions and communication. The random number generation increases the security of our protocol and reduces the storage overhead. We assume the random number to 128-bits and the EC points to be 224 bits in size.

A. Storage Cost

The device requires to store P, R_R, K_T and the values (r_T,R_T) are stored in the database which needs total memory = [224 + 224 + 224+ 128+ 128] = 928 bits.

B. Communication and Computation Cost

The total cost involved in the message transmission at each step can be given as:

- After receiving <*hello*> the tag generates a random number (128 bits) and send the message < A_1 , R_1, R_2, X_1, K_T >. The total cost here is [128+128+128+224+244+224+244] = 1280 bits.
- After the reader verifies the tag and sends <A_2, X_2> to the

tag it now extracts and compares R_R. There is no other operation or random number generation involved in this step.

VI. Conclusion and Future Work

The Internet of things architecture is still evolving with new opportunities every day, however, it is also vulnerable to many security threats. Thus, security protocols are necessary in order to ensure the success of these devices. In this paper, we have presented an authentication protocol for IoT devices using ECC and random number generations. The use of ECC makes our protocol more scalable and provides better security with lower resources as compared to other public key cryptography techniques. Our protocol is fairly simple and can be easily implemented. We have also performed a security and performance analysis of our protocol in subsequent section. Our protocol ensures mutual authentication, confidentiality, non-tracking and forward secrecy. Our protocol can be extended for use in other low-resource systems also. Further we aim strengthen our approach to secure it from DDoS and spam attacks which target and infect IoT devices and use them as botnet.

Acknowledgment

This research work is being funded by Department of Electronic and Information technology (DeitY), Ministry of Communications and IT, Government of India.

References

[1] G. L. Atzori, A. Iera, G. Morabito, The Internet of Things: a survey, Computer Networks 54 (2010) 2787–2805.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol.29, no.7, pp. 1645–1660, 2013.

[3] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," Computer, vol.44, no.9, pp.51-58, 2011.

[4] P. Najera, J. Lopez, and R. Roman, "Real-time location and inpatient care systems based on passive RFID," J. Netw. Comput. Appl., vol. 34, no. 3, pp. 980–989, 2011.

[5] D. Ranasinghe, Q. Sheng, and S. Zeadally, Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks. New York, NY, USA: Springer, 2010.

[6] Y. Hung, "The study of adopting RFID technology in medical institute with the perspectives of cost benefit," Ph.D. dissertation, Dept. Comput. Sci. Inform. Eng., Fu Jen Catholic Univ., New Taipei City, Taiwan, 2011.

[7] S. Kalra, S.K. Sood, "Secure authentication scheme for IoT and cloud servers", in Special Issue on Secure Ubiquitous Computing, Pervasive and Mobile Computing Elsevier, Vol 24 pp. 210-223, 2015.

[8] Tuyls P, Batina L (2006) RFID-tags for anti-counterfeiting. In: Topics in Cryptology (CT-RSA'06), LNCS 3860, pp 115–131.

[9] Batina L, Guajardo J, Kerins T, Mentens N, Tuyls P, Verbauwhede I (2007) Public-key cryptography for RFID-tags. In: Fifth annual IEEE international conference on pervasive computing and communications workshops, 2007. (PerCom Workshops'07), pp 217–222.

[10] Lee YK, Batina L, Verbauwhede I (2008) EC-RAC (ECDLP based randomized access control): provably secure RFID authentication protocol. In: IEEE international conference on RFID, pp 97–104.

[11] Y. Lee, L. Batina, D. Singelee and I. Verbauwhede "Low-cost untraceable authentication protocols for RFID" Proc. 3rd ACM Conf. Wireless Netw. Secur. (WiSec'10), pp. 55-64, 2010.

[12] Y. Lee, I. Batina, I. Verbauwhede, Untraceable RFID authentication protocols: revision of EC-RAC. In IEEE International Conference on RFID 2009. IEEE: Orlando,FL,USA, 178–185, 2009.

[13] Y. Liao, C. Hsiao, A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol, Ad Hoc Networks, 2013, doi: 10.1016/j.adhoc.2013.02.004.

[14] R. Peeters, J. Hermans, Attack on Liao and Hsiao's Secure ECC-based RFID Authentication Scheme integrated with ID-Verifier Transfer Protocol. Cryptology ePrint Archive, Report 2013/399, 2013.

[15] J. Chou, "An efficient mutual authentication RFID scheme based on elliptic curve cryptography," J. Supercomput., vol. 70, no. 1, pp. 75–94, 2014.

[16] M. Farash, "Cryptanalysis and improvement of an efficient mutual authentication RFID scheme based on elliptic curve cryptography," J. Supercomput., 2014, doi: 10.1007/s11227-014-1272–0.

[17] Y. Liu, X. Qin, and C. Wang, "A lightweight RFID authentication protocol based on elliptic curve cryptography," J. Comput., vol. 8, no. 11, pp. 2880–2887, 2013.

[18] Y. Liao and C. Hsiao, "A secure ECC-based RFID authentication scheme using hybrid protocols," in Advances in Intelligent Systems and Applications. Berlin, Germany: Springer-Verlag, 2013, pp. 1–13.

[19] Z. Zhao, "A secure RFID authentication protocol for healthcare environments using elliptic curve cryptosystem," J. Med. Syst., vol. 38, no. 5, 2014, doi: 10.1007/s10916-014-0046-9.

[20] Tan, Z., A user anonymity preserving three-factor authentication scheme for telecare medicine information systems. J. Med. Syst. 38(3):1–9, 2014.

[21] Arshad, H., and Nikooghadam, M., Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems. J. Med. Syst. 38(12):1–12, 2014.

[22] Lu, Y., Li, L., Peng, H., Yang, Y., An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem. J. Med. Syst. 39(3):32, 2015. doi:10.1007/s10916-015-0221-7.

[23] Anoop MS, " Elliptic Curve Cryptography - An implementation guide," May 2007.

[24] L. Blum, M. Blum, and M. Shub, "A Simple Unpredictable Pseudo-Random Number Generator", SIAM Journal on Computing, vol. 15, no. 2, pp. 364–383, 1986.