# Security and Privacy Analysis of Recently Proposed ECC-Based RFID Authentication Schemes

Atakan Arslan[a,*], Muhammed Ali Bingöl[b]

[a] *TÜBİTAK BİLGEM-UEKAE, Gebze, Kocaeli, Turkey*
[b] *De Montfort University University, Cyber Technology Institute, Leicester, UK*

**Abstract**

Elliptic Curve Cryptography (ECC) has been popularly used in RFID authentication protocols to efficiently overcome many security and privacy issues. Even if the strong cryptography primitives of ECC are utilised in the authentication protocols, the schemes are alas far from providing security and privacy properties as desired level. In this paper, we analyze four up-to-minute ECC based RFID authentication schemes proposed by Gasbi et al., Benssalah et al., Kumar et al., and Agrahari and Varma. The authors claim that their schemes provide prominent and important security and privacy requirements. However, we have shown some crucial vulnerabilities of the schemes against their allegations. We attack to Gasbi et al.'s protocol by using transmitted messages in insecure channel and exploiting the message relations which points a specific tag, and show that the scheme does not provide tag anonymity/untraceability, forward and backward security and the scheme has performance problems. Moreover, we demonstrate that Kumar et al., and Agrahari and Varma's schemes do not achieve forward and backward security because the schemes are not designed to eliminate the advantage of an adversary obtaining full knowledge of a tag from by attack definition. We also show that Benssalah et al.'s scheme suffers from tag anonymity/untraceability, forward and backward security when the pseudonym of a tag is transmitted in insecure channel somehow without updating.

*Keywords:* Security, Privacy, ECC, RFID, Authentication Protocols

## 1. Introduction

Radio Frequency Identification (RFID) is a technology that has been preferred in many different application areas for a long time and is used to identify entities and to wirelessly exchange information [1]. Internet of Things (IoT) is another promising technology providing the interconnection of all things and

*Corresponding author. E-mail: atknarsln@gmail.com

with the advancement of IoT, RFID applications have been becoming pervasive more than before [2]. Indeed, both technologies support and develops each others.

Nowadays, Elliptic Curve Cryptography (ECC) based RFID authentication protocols have been mostly used to efficiently overcome the security and privacy issues [3, 4, 5, 6, 7] in application areas of IoT. On the other hand, numerous protocols alas suffers from efficiency, security and privacy vulnerabilities [8, 9, 4, 10].

In this paper, we show an attack against one of the novel ECC-based RFID authentication protocols recently proposed by Gabsi et al. [3]. We argue that their scheme (called GK21) does not satisfy anonymity, forward, backward secrecy, and GK21 is vulnerable to position tracking and server impersonation attacks. Moreover, the scheme might suffer from scalability problems. Second, we analyse Benssalah et al.'s scheme [11] (called BS21) and show that the scheme does not provide tag anonymity, untraceability, forward and backward privacy. Third, we demonstrate that Agrahari and Varma's, and Kumar et al.'s protocols (called AV21 and KB21, respectively) both do not satisfy forward and backward privacy.

In what follows, the organization of this paper is as follows. In Section 2, adversary model will be introduced. In Section 3 - 6, the brief description, security an privacy analysis of GK21, BSD21, AV21 and KB21 will be given. Finally, Section 7 will present the conclusion.

## 2. Adversary Model

In our analysis, we assume that the adversary ***Adv*** can perform active and passive attacks against to an RFID scheme in polynomial-time. We also assume that the channel between the reader and tag is insecure and ***Adv*** can threat the scheme by eavesdropping, recording, intercepting, replaying, relaying, blocking and modifying session messages of the scheme. In addition to this, ***Adv*** is able to utilize rough devices (i.e. reader or tag) to start sessions with one of the parties. In fact, ***Adv*** can take whole control of the insecure channel between the tag and the server.

***Adv*** intends to breach anonymity, confidentiality, integrity, availability, forward/backward privacy, of an RFID scheme, besides ***Adv*** attempts to perform the existing well-known attacks such as man-in-the-middle, replay, impersonation, position tracking, tag cloning, key compromising, etc. Moreover, we assume that ***Adv*** can observe the results of a protocol session, and deduce whether the session is successfully accomplished or terminated with a failure.

## 3. Analysis of Gabsi et al.'s Protocol

In this section, we first briefly describe Gabsi et al.'s protocol (GK21) [3] and present our analysis in terms of efficiency, security and privacy.

*3.1. Protocol Description*

We show the overview of Gabsi et al.'s scheme in Figure 1 and present the related notations in Table 1.

Table 1: Gabsi et al.'s Protocol Notations [3]

| | |
|---|---|
| $P$ | Elliptic curve base point |
| $s$ | Secret key of Server/Reader |
| $P_s$ | Public key of Server/Reader |
| $t$ | Secret key of Tag |
| $P_t$ | Public key of Tag |
| $sP_t$ | Identifier of Tag |
| $N$ | Number of Valid Tags |

Gabsi et al.'s protocol has two phases such as setup phase and authentication phase. Firstly, in setup phase, all cryptographic keys are generated. Server randomly picks a secret key $s$ and computes its public key $P_s = sP$. Similarly, a tags generates a random value $t$ as its secret key and calculates its public key $P_t = tP$. Secondly, the tag and server share public keys with each other. At the end of the phase, the tag has $t, P_t, P_s, P$ parameters and the server stores $s, P_s, P_t, P$ parameters.

| ***Reader*** $\{s, P_s, P_t, P\}$ | | ***Tag*** $\{t, P_t, P_s, P\}$ |
|---|---|---|
| Generates $r_1 \epsilon Z_n^*$<br>Computes $R_1 = r_1 P$ | | |
| | $\xrightarrow{R_1}$ | Generates $r_2 \epsilon Z_n^*$<br>Computes $R_2 = r_2 P$ |
| | $\xleftarrow{R_2, M_1}$ | $M_1 = tR_1 + r_2 P_s$ |
| Checks $M_1 \stackrel{?}{=} r_1 P_t + sR_2$<br>If yes, it authenticates the tag,<br>Calculates $x = s + r_1$<br>$M_2 = xP_t + sR_2$<br>*else*<br>$\perp$ | | |
| | $\xrightarrow{M_2}$ | Calculates $x^{'} = t + r_2$<br>$M_2^{'} = x^{'} P_s + tR_1$<br>Checks $M_2^{'} \stackrel{?}{=} M_2$<br>If yes, it authenticates the reader<br>*else*<br>$\perp$ |

Figure 1: Gabsi et al.'s Protocol[3]

The authentication phase starts with sending $R_1 = r_1 P$ by the reader to the tag, where $r_1$ is ephemerally generated random value. Then, the tag picks a random value and computes $R_2 = r_2 P$ and $M_1 = tR_1 + r_2 P_s$. The tag responses with transmitting $R_2$ and $M_1$ messages to the reader. After the reader receives

them, the reader validates $M_1$ message. If the validation is successful, the reader authenticates the tag and calculates $x = s + r_1$ and $M_2 = xP_t + sR_2$. Else, the reader stops the protocol.

Later on, the reader sends $M_2$ to the tag for mutual authentication. When the tag gets $M_2$, the tag checks its validity by calculating $x^{'} = t + r_2$ and $M_2^{'} = x^{'}P_s + tR_1$. If $M_2^{'} = M_2$, the tag also authenticates the reader. Otherwise, the tag terminates the protocol.

### *3.2. Security and Privacy Analysis*

Gabsi et al. claim that their scheme provides confidentiality, anonymity, forward security properties, and resistance against impersonation and position tracking attacks. On the contrary, we show below that their protocol does not satisfy the security requirements that they have asserted.

#### *3.2.1. Anonymity*

Gabsi et al. assert that their scheme fulfills tag anonymity in their security analysis because of the randomly generating or computing the messages transmitted in each protocol session. Although they ground their statement to the randomness, they have not notice that there might be relation between the messages causes security vulnerabilities.

In our opinion, an adversary ***Adv*** can divulge a certain identifier that specifically points a tag by using the relation of the transmitted messages during only one protocol session. Once ***Adv*** has the identifier, ***Adv*** ruins the tag anonymity.

According to us, ***Adv*** can compute $sP_t$, as a tag identifier by using session messages $R_1, R_2, M_1, M_2$. $s$ and $P_t$ remain unchanged so $sP_t$ always points a specific tag and causes to breaking anonymity property.

***Adv*** obtains $sP_t$ by eavesdropping a protocol session and recording the session messages $R_1, R_2, M_1$ and $M_2$. Then, ***Adv*** calculates $M_2 - M_1$ and obtains $sP_t$ as below.

$$\begin{aligned} M_2 - M_1 &= (xP_t + sR_2) - (tR_1 + r_2P_s) \\ &= ((s + r_1)\,tP + sr_2P) - (tr_1P + r_2sP) \\ &= stP + r_1tP + sr_2P - tr_1P - r_2sP \\ &= stP = sP_t \end{aligned}$$

After revealing of the parameter $sP_t$, ***Adv*** certainly distinguishes the tag whenever it communicates with a valid reader.

#### *3.2.2. Position Tracking*

Gabsi et al. state that their scheme is resistant against position tracking attack due to the fact that their scheme does not unveil the identity of a tag in their scheme. We have presented above that the schemes does not overcome anonymity and reader impersonation immunity. From the point of these weakness, we claim that the schemes is not also resilience to position tracking attack.

An adversary ***Adv*** interrogates a valid tag $T_t$ whose private key $t$ and obtains $sP_t$. Later on, whenever ***Adv*** meets the tag $T_t$, she is able to deceive the tag by using $sP_t$ and observing the output of the protocol. Thus, ***Adv*** is able to track the position of the tag $T_t$. Actually, ***Adv*** can trace all tags and draw location history map for them, by collecting their identifiers $sP_t$ as a list. Wherever ***Adv*** encounters a tag, she checks the list. If she does not find the identifier in her list, she only adds the new identifier to her list.

*3.2.3. Server/Reader Impersonation*

Gabsi et al. state that their scheme is resistant against reader impersonation attack. They present reader impersonation attack as server spoofing attack in their paper. They define that if a protocol prevents server spoofing attack, an adversary ***Adv*** never impersonate a legitimate server or, in other words, deceive a valid tag. In this work, we prefer to use reader impersonation notion instead of reader spoofing.

Gabsi et al. claim that ***Adv*** cannot generate a genuine response message $M_2$ when ***Adv*** interrogates a legitimate tag and receives messages $R_2$ and $M_1$. However, we argue that ***Adv*** can impersonate a valid server by using the knowledge of $sP_t$.

We show above that ***Adv*** can obtain $sP_t$, where identifies a specific tag. $sP_t$ depends to the secret keys of the reader and the tag. $sP_t$ only changes whenever the secret keys must be renewed.

***Adv*** can generate a legitimate message $M_2$ after receiving $R_2$ and $M_1$ by using $sP_t$. ***Adv*** calculates $M_2 = M_1 + sP_t$ as a response message. Therefore, ***Adv*** cheats the tag by pretending as a legal reader.

*3.2.4. Forward Security*

Gabsi et al. claim that their scheme provides forward security property. Forward secrecy implies indistinguishability of a tag in its previous transactions, even if the internal information including all secrets privileges is known by an adversary [12]. It can be clearly seen that if a scheme does not have anonymity property, it cannot achieve forward secrecy any more. In such schemes, an attacker does not need the internal knowledge of a tag to breach forward security.

We demonstrate above analysis that ***Adv*** can destroy anonymity of a tag and trace it so ***Adv*** can distinguish a tag by using its past recorded session messages. Hence, Gabsi et al.'s scheme does not accomplish forward security requirements.

*3.2.5. Backward Security*

Gabsi et al. do not mention the backward secrecy (or forward untraceability) property in their paper but we present our privacy analysis in terms on backward secrecy because of its importance for RFID authentication protocols [4, 13] Similarly to forward security attack, ***Adv*** can distinguish and trace a tag by using its future recorded session messages, even if the internal information including all secrets privileges is obtained. Thus, Gabsi et al.'s scheme does not satisfy backward secrecy.

### *3.3. Performance Analysis*

We also reconsider the computational cost of the scheme and we point out that the scheme is much heavier that they claim because of computation cost in the tag authentication process. In our opinion, searching time of a tag identity in the server database takes much time and makes the scheme cumbersome. In time, the scheme might encounter scalability issues with increasing number of registered tags in the system.

Gabsi et al. evaluate and compare their protocol design with existing schemes in terms of communication cost, computational cost and storage cost. However, they do not consider the searching time of a tag in the server database which directly affects the computational cost. When the searching time becomes longer, the computations costs increases.

In Gabsi et al.'s scheme, the searching time depends on the number tags within the RFID system and the searching complexity of the scheme is $O(N)$, where $N$ denotes the number of tags registered in the server database. It means that the server consumes $O(N)$ elliptic curve scalar multiplication time to search the identity of tag $P_t$ inside its storage to check the message $M_1$ for each authentication process. In fact, this searching time is much higher than the computation time that are considered in Gabsi et al.'s paper. In their scheme, the searching time will linearly increase with the number of the registered tags. This might cause scalability problems in time.

## 4. Analysis of Benssalah et al.'s Protocol

In this section, we present the Benssalah et al.'s protocol (BS21) [11] in a nutshell. Then, we show the security analysis of their scheme.

### *4.1. Protocol Description*

Briefly, the scheme consists of two phases such as authentication phase and updating phase. In the authentication phase, both entities authenticate each other based on using ECC. The scheme is depicted in Figure 2 and the detailed information about the scheme is found in [11].

Table 2: Benssalah et al.'s Protocol Notations [11]

| | |
|---|---|
| $P$ | elliptic curve base point |
| $r_1, r_2$ | Random numbers |
| $x_S$ | secret key of the Server |
| $P_S$ | public key of the server $P_{S=}x_S P$ |
| $x_t$ | secret key of the tag |
| $ID_S$ | pseudonym of the tag |
| $ID_S^{old}$ | old pseudonym of the tag |
| $ID_S^{new}$ | new pseudonym of the tag |
| $h$ | Hash function |
| $\{.\}_x$ | the x-coordinate of the given point. |

| Reader/server (DB) $\{Server(x_S), Tag(x_t, ID_S)\}$ | Insecure channel | Tag $\{x_t, ID_S, P, P_S\}$ |
|---|---|---|
| Generates: $r_1 \in Z_n^*$ | $\xrightarrow{\texttt{Query}, r_1}$ | Generates $r_2 \in Z_n^*$<br>Computes $R_2 = r_2 P_S$<br>$R_3 = r_2 P$<br>$R_4 = x_t + h(\{R_2\}_x \|\| \{R_3\}_x \|\| r_1)$ |
| Calculates:<br>$R_2^* = x_S R_3$<br>$x_t = R_4 - h(\{R_2^*\}_x \|\| \{R_3\}_x \|\| r_1)$<br>$\langle ID_S, x_t \rangle$ `authenticates the tag` | $\xleftarrow{R_3, R_4, ID_S}$ | |
| Computes:<br>$R_5 = h(x_t \|\| \{R_2\}_x \|\| r_1 \|\| R_4)$ | $\xrightarrow{R_5}$ | $R_5^* = h(x_t \|\| \{R_2\}_x \|\| r_1 \|\| R_4)$<br>The server is authenticated if the equality holds |
| Updating phase: $ID_S$ | | Updating phase: $ID_S$ |

Figure 2: BSD21 Scheme [11].

After mutual authentication is succeeded, the server and the tag execute the updating process as below.

The tag:

$$ID_S^* = h\left(\{R_2\}_x \|\| ID_S \|\| r_1 \|\| R_4\right)$$
$$ID_S \leftarrow ID_S^*$$

The server:

If $ID_S^{old}$ is received:

$$ID_S^{new} = h\left(\{R_2\}_x \|\| ID_S^{old} \|\| r_1 \|\| R_4\right)$$

Else, If $ID_S^{new}$ is received:

$$ID_S^{old} = ID_S^{new}$$
$$ID_S^{new} = h\left(\{R_2\}_x \|\| ID_S^{new} \|\| r_1 \|\| R_4\right)$$

### *4.2. Security and Privacy Analysis*

We argue that Benssalah et al.'s protocol [11] does not provide forward security, tag untraceability and anonymity properties, opposing they claimed.

#### *4.2.1. Anonymity/Untraceability*

In their scheme, the tag openly transmits its pseudo-identity $ID_S$ (called pseudonym) in insecure channel and updates pseudo-randomly this identity if the tag authenticates the server. Otherwise, the tag uses the same identity for the next session. We state that this is a privacy weakness which provides an advantage to the adversary to threaten the tag privacy.

It is possible that $ID_S$ might not be updated and the same identity would be used for several times in the future sessions because of some reasons such as the channel distortion, the failure in calculations or attacking the scheme.

***Adv*** can block or change the last message $R_5$ sent by the server for every session and record a pseudonym list $L_{ID}$. ***Adv*** eavesdrops a protocol session, she look the current pseudonym $ID_S$ up in the list $L_{ID}$. If ***Adv*** finds the

identifier in her list, she links the current session with the past one. It means that ***Adv*** can destroy anonymity of the tag and trace the tag both in past and future sessions. Else ***Adv*** does not find the tag in her list, she updates her list by adding the new pseudonym.

With increasing in number of sessions dealt by ***Adv***, she will be successful with higher probability. To sum up, ***Adv*** is able to ruin the anonymity of the tag and trace it. Thus, the scheme does not provide tag anonymity.

***Adv*** violates the location privacy of a tag by the same attack mentioned above. ***Adv*** can trace the tag whose pseudonym are not properly updated in the protocol sessions.

*4.2.2. Forward and Backward Security*

If a scheme does not provide anonymity and untraceability, a tag can be trackable by ***Adv*** with using the past and future protocol transactions. Even if ***Adv*** does not need to obtain the internal knowledge of the tag.

*4.3. Some Notes On The Scheme*

We provide a simple solution to fix the weakness of Benssalah et al.'s protocol. In our opinion, a tag should never transmit its pseudonym $ID_S$ since the reader does not need the message $ID_S$ to authenticate the tag. The reader can obtain the secret of the tag $x_t$ by using the messages $R_3$ and $R_4$ and checks it in its database.

## 5. Analysis of Agrahari and Varma's Protocol

In this section, we present the Agrahari and Varma's protocol (AV21)[14] in a nutshell. Then, we show the security analysis of their scheme.

*5.1. Protocol Description*

The scheme includes two phases: initialization and authentication phases. In the initialization phase, the server assigns the system parameters to the tag and reader. In the authentication phase, the tag and the reader mutually authenticate each other by using ECC, encoding and decoding algorithms. The protocol is shown in Figure 3 and for detailed information, please see [14].

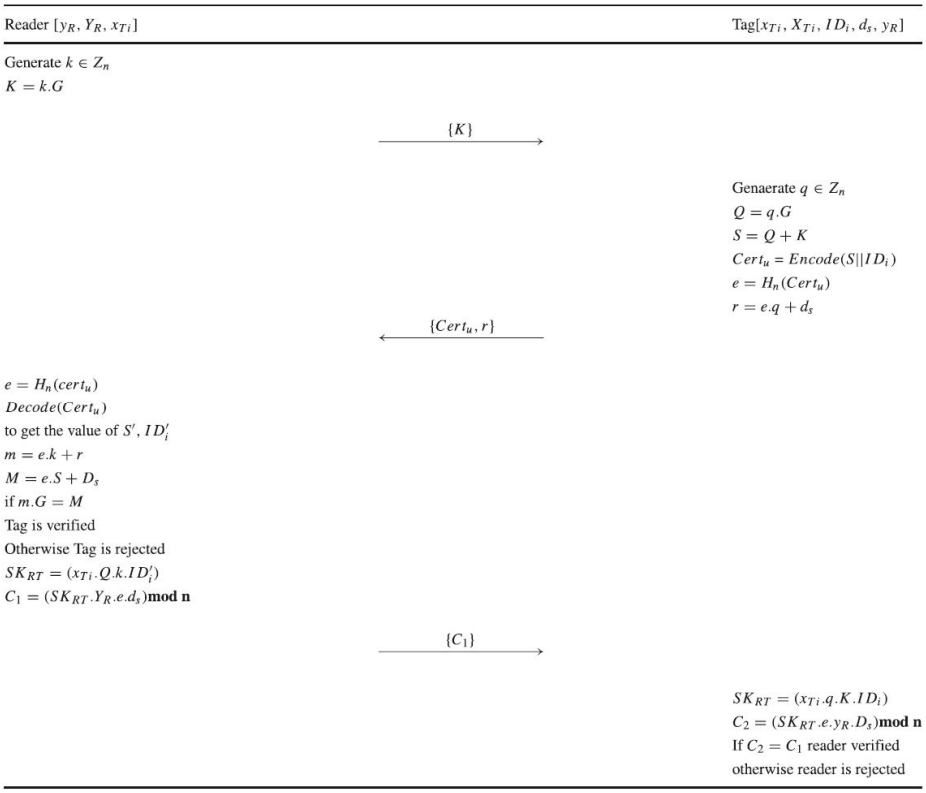


Figure 3: Agrahari and Varma's Scheme [14].

### *5.2. Security and Privacy Analysis*

Our analysis focuses on the encoding and decoding functions considered in Agrahari and Varma's protocol description. We claim that their scheme does not satisfy neither forward secrecy nor backward secrecy.

The authors only mention that an adversary cannot decode an encoded message $Cert_u$ but they do not define and explain the encoding algorithm or give an example refers its type. In their scheme, the tag encodes a session randomized value $S$ and their own identity $ID_i$ and calculates $Cert_u = Encode\,(S||ID_i)$. The tag sends $Cert_u$ to the reader. Then, the reader obtains $S$ and $ID_i$ by decoding the message $Cert_u$.

We deduce the following conclusion from the usage of the encoding and decoding algorithms/functions in the scheme.

- $Decode\,(Encode\,(S||ID_i)) = S||ID_i$
- All tags in the system utilize the same encoding algorithm.
- The algorithms are not based on any key. They are keyless algorithms.

- The security level of the algorithms only depend on their secrecy and complexity.

*5.2.1. Forward Security*

By the definition of forward security notion, the adversary can access the full internal knowledge of the tag (including its private key, etc.). It means that the adversary will reveal the encoding mechanism so ***Adv*** can figure out how to decoding algorithm works and obtain the identity of a tag, too.

***Adv*** records all the messages transmitted in protocol sessions until corrupting a tag or getting the internal knowledge of the tag. Let ***Adv*** acquires the identity $ID_{T1}$ of the tag $T1$.

After having ability to decode an encoded messages, ***Adv*** can reveal the $S$ and $ID_i$ for each session by computing $Decode\,(Cert_u) = S||ID_i$. ***Adv*** compares the output identities with the obtained identity of tag, $ID_i \stackrel{?}{=} ID_{T1}$. Whenever ***Adv*** reaches the equality, she discloses the session of $T1$. By using this attack, she can distinguish the tag and trace it among the past protocol transactions. Therefore, the scheme does not provide forward security.

*5.2.2. Backward Security*

Similarly the above explanation, After getting the internal knowledge of the tag $T1$ and having ability to decode an encoded messages, ***Adv*** collects the future messages transmitted between the server and tags and calculates $Decode\,(Cert_u) = S||ID_i$ and makes the comparison $ID_i \stackrel{?}{=} ID_{T1}$.

Whenever ***Adv*** finds the equality, she divulges the related session of $T1$. By using this attack, she can distinguish the tag and trace it among the future protocol transactions. Therefore, the scheme does not provide backward security.

## 6. Analysis of Kumar et al.'s Protocol

In this section, we present Kumar et al.'s protocol (KB21)[10] in a nutshell. Then, we show the security analysis of their scheme.

*6.1. Protocol Description*

Kumar et al.'s authentication protocol consists of two phases: (i) setup phase and (ii) authentication phase. In the setup phase, elliptic curve parameters are defined. The private and public key pairs are generated and stored in both server and tag.

Table 3: Notations of Kumar et al.'s Protocol [10]

| | |
|---|---|
| $G$ | elliptic curve base point |
| $y_R$ | secret key of the reader |
| $Y_R$ | public key of the reader $Y_{R=}y_RG$ |
| $x_{T_i}$ | secret key of the tag |
| $X_{T_i}$ | secret key of the tag $X_{T_i} = x_{T_i}G$ |
| $d_S$ | secret key of the server |
| $D_S$ | public key of the reader $D_{S=}d_SG$ |
| $ID_i$ | identifier of the tag |

In the authentication phase, the server interrogates the tag by sending random EC point to the tag. The tag responses $Ath_T$ and $R_1$ messages to the server. Later, the server authenticates the tag by controlling the messages and sends $Ath_s$ messages back to the tag. Finally, the tag checks the message and authenticates the server, too. The scheme is illustrated in Figure 4 and the scheme notations are also given in Table 3. The more information about the scheme is found in [10].

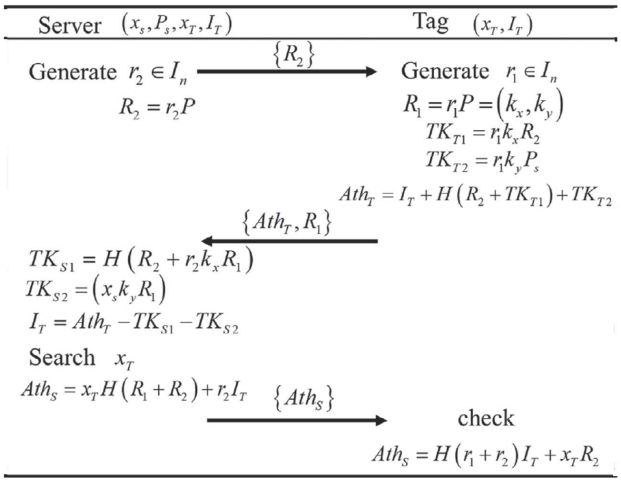


Figure 4: Kumar et al.'s Scheme [10].

### *6.2. Security and Privacy Analysis*

Kumar et al. claim that their scheme provides forward security. However we show that the scheme has a privacy vulnerability that the adversary destroys both forward and backward security.

#### *6.2.1. Forward Security*

The authors present an informal security analysis in their paper and they say in the proof of this property that the adversary cannot determine and trace the tag by recording the messages $R_1, R_2$ and $Ath_T$, although the adversary

gets security key of tag $x_T$ and $I_T$. However, we prove below that the adversary distinguish the tag whose keys are captured by recording and checking the past session messages $Ath_s$.

The adversary collects all the previous session messages. Whenever she obtains $x_T$ and $I_T$ keys of a tag, she calculates $x_T H(R_1, R_2) + x_T R_2$ by using the related session messages $R_1$ and $R_2$. Then, the adversary compares the calculation output with the message of $Ath_S$ of the related session, $Ath_s \stackrel{?}{=} x_T H(R_1, R_2) + x_T R_2$. When she finds the equality, she distinguishes the session of the tag and trace the tag. Therefore, she definitely destroys the forward security of the scheme.

*6.2.2. Backward Security*

Kumar et al. assert that their scheme achieves the backward security property. We prove below that the scheme is also vulnerable to backward security attacks.

Similarly to the mentioned explanation of our analysis in Section 6.2.1 that the adversary is able to find the equality $Ath_s \stackrel{?}{=} x_T H(R_1, R_2) + x_T R_2$ by using future messages of the sessions. Because $R_1$ and $R_2$ messages are randomly chosen for each independent session and they are openly transmitted. The adversary successfully distinguish the future transactions of the tag and trace the tag like aforementioned before.

## 7. Conclusions

We analyse the novel ECC based RFID authentication protocols [3, 11, 14, 10] which are proposed to mitigate the existing security and privacy issues. We show that they have security and privacy weaknesses as opposition of their claim, in our informal analysis. We present the summarization of our analysis in Table 4, where + denotes the violated properties of the protocols.

Table 4: The Summarization of Our Analysis

| | GK21[3] | BS21[11] | AV21[14] | KB21[10] |
|---|---|---|---|---|
| Tag Anonymity/ Untraceability | + | + | | |
| Forward Privacy | + | + | + | + |
| Bacward Privacy | + | | + | + |
| Reader Impersonation | + | | | |
| Scalability | + | | | |

According to our analysis, all schemes suffers from forward privacy issue and similarly all of them except BS21 [11] do not achieve backward privacy, too. Furthermore, GK21 [3] and BS21 [11] does not provide tag anonymity and location privacy. GK21 is also not immunity against to the reader impersonation attack and the scheme have scalability problem.

In addition to this, we evaluate GK21 [3] in terms of efficiency and exhibit that the scheme has a higher computational cost than they considered. In GK21, the server consumes much more time while performing tag authentication because of looking up the valid tag in its database. Also, this inefficiency enables that GK21 could suffer from scalability troubles in time.

## References


[1] C. Munoz-Ausecha, J. Ruiz-Rosero, G. Ramirez-Gonzalez, RFID Applications and Security Review, Computation 9 (2021) 69.

[2] H. Landaluce, L. Arjona, A. Perallos, F. Falcone, I. Angulo, F. Muralter, A Review of IoT Sensing applications and Challenges Using RFID and Wireless Sensor Networks, Sensors 20 (2020) 2495.

[3] S. Gabsi, Y. Kortli, V. Beroulle, Y. Kieffer, A. Alasiry, B. Hamdi, Novel ECC-Based RFID Mutual Authentication Protocol for Emerging IoT Applications, IEEE Access 9 (2021) 130895–130913. doi:10.1109/ACCESS.2021.3112554.

[4] A. Arslan, S. A. Çolak, S. Ertürk, A Secure and Privacy Friendly ECC Based RFID Authentication Protocol for Practical Applications, Wireless Personal Communications (2021) 1–39. doi:https://doi.org/10.1007/s11277-021-08552-7.

[5] S. Izza, M. Benssalah, K. Drouiche, An Enhanced Scalable and Secure RFID Authentication Protocol for WBAN Within An IoT Environment, Journal of Information Security and Applications 58 (2021) 102705. URL: https://www.sciencedirect.com/science/article/pii/S2214212620308516. doi:https://doi.org/10.1016/j.jisa.2020.102705.

[6] M. Naeem, S. A. Chaudhry, K. Mahmood, M. Karuppiah, S. Kumari, A Scalable and Secure RFID Mutual Aauthentication Protocol Using ECC for Internet of Things, International Journal of Communication Systems 33 (2020) e3906. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.3906. doi:https://doi.org/10.1002/dac.3906. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/dac.3906, e3906 dac.3906.

[7] V. Kumar, M. Ahmad, D. Mishra, S. Kumari, M. K. Khan, RSEAP: RFID Based Secure and Efficient Authentication Protocol for Vehicular Cloud Computing, Vehicular Communications 22 (2020) 100213. URL: https://www.sciencedirect.com/science/article/pii/S2214209619302608. doi:https://doi.org/10.1016/j.vehcom.2019.100213.

[8] S. Gabsi, V. Beroulle, Y. Kieffer, H. M. Dao, Y. Kortli, B. Hamdi, Survey: Vulnerability Analysis of Low-Cost ECC-Based RFID Protocols against Wireless and Side-Channel Attacks, Sensors 21 (2021) 5824.

[9] M. Safkhani, C. Camara, P. Peris-Lopez, N. Bagheri, RSEAP2: An Enhanced Version Of RSEAP, An RFID Based Authentication Protocol for Vehicular Cloud Computing, Vehicular Communications 28 (2021) 100311. URL: https://www.sciencedirect.com/science/article/pii/S2214209620300826. doi:https://doi.org/10.1016/j.vehcom.2020.100311.

[10] S. Kumar, H. Banka, B. Kaushik, S. Sharma, A Review and Analysis of Secure And Lightweight ECC-Based RFID Authentication Protocol For Internet of Vehicles, Transactions on Emerging Telecommunications Technologies (2021) e4354.

[11] M. Benssalah, I. Sarah, K. Drouiche, An Efficient RFID Authentication Scheme Based on Elliptic Curve Cryptography for Internet of Things, Wireless Personal Communications 117 (2021) 2513–2539.

[12] M. Burmester, J. Munilla, A Flyweight RFID Authentication Protocol, in: Workshop on RFID Security, Citeseer, 2009.

[13] A. Arslan, M. A. Bingöl, Cryptanalysis of Izza et al.'s Protocol: An Enhanced Scalable and Secure RFID Authentication Protocol for WBAN Within An IoT Environment, Cryptology ePrint Archive, Report 2021/519, 2021. https://ia.cr/2021/519.

[14] A. K. Agrahari, S. Varma, A Provably Secure RFID Authentication Protocol Based on ECQV For The Medical İnternet of Things, Peer-to-Peer Networking and Applications 14 (2021) 1277–1289.