A note on "HAKECC: highly efficient authentication and key agreement scheme based on ECDH for RFID in IOT environment"

Zhengjun Cao

Abstract. We show that the Nikooghadam-Shahriari-Saeidi authentication and key agreement scheme [J. Inf. Secur. Appl., 76, 103523 (2023)] cannot resist impersonation attack, not as claimed. An adversary can impersonate the RFID reader to cheat the RFID tag. The drawback results from its simple secret key invoking mechanism. We also find it seems difficult to revise the scheme due to the inherent flaw.
Keywords: Authentication, Anonymity, Key agreement, Internet of Things

1 Introduction

The Internet of Things (IoT) is a network of physical devices, which uses a variety of technologies to connect the digital and physical worlds. These devices, such as smart home devices, personal medical devices, can transfer data to one another without human intervention. The security of IoT has attracted much attention. In 2017, Lavanya and Natarajan [1] proposed a lightweight key agreement protocol for IoT based on IKEv2. After that, Parne et al. [2] presented a security enhanced authentication key agreement protocol for IoT enabled LTE/LTE-A networks. Tedeschi et al. [3] discussed a lightweight certificateless key agreement for secure IoT communications. In 2021, Chen et al. [4] put forth a secure blockchain-based group key agreement protocol for IoT. Mahmood, et al. [5] designed a seamless anonymous authentication protocol for mobile edge computing infrastructure. Tomar et al. [6] presented a blockchain-assisted authenticated key agreement scheme for IoT-based healthcare system. Zahednejad et al. [7] investigated a big data based authentication and key agreement scheme for IoT with revocability.

Very recently, Nikooghadam, Shahriari and Saeidi [8] have also presented an authentication and key agreement scheme for Radio Frequency Identification (RFID) in IOT environment. In the considered scenario, there are three entities: RFID tag, RFID reader, and central database. The RFID reader requests access to the tag which forwards the response to the central database. The scheme is designed to meet many security requirements, including authentication, session-key establishment, anonymity, perfect forward secrecy, and resistance to impersonation attack, reply

Department of Mathematics, Shanghai University, Shanghai, 200444, China. caozhj@shu.edu.cn

attack, DoS attacks, etc. In this note, we show that the scheme cannot resist impersonation attack, not as claimed.

2 Review of the Nikooghadam-Shahriari-Saeidi scheme

Let E be an elliptic curve. G is a cyclic additive elliptic curve group with a generator P of the prime order p. $h : \{0,1\}^* \to \{0,1\}^l$ is a hash function. Let t_s be the tag's private key, and $T_s = t_s \cdot P$ be the public key. Let r_s be the reader's private key, and $R_s = r_s \cdot P$ be the public key. $E(\cdot)$ is a symmetric key encryption algorithm and $DEC(\cdot)$ is the symmetric key decryption algorithm. In the setup phase, the reader memory is uploaded with the parameters $\{r_s, R_s, T_s, p\}$, and the tag memory is uploaded with the parameters $\{t_w, T_s, R_s, P\}$. The scheme can be briefly depicted as follows (see Table 1). Its correctness is due to that

$$key_i = t_s \cdot R_i = t_s(r_i \cdot P) = r_i(t_s \cdot P) = r_i \cdot T_s,$$
$$X_i = n_i \cdot R_i = n_i(r_i \cdot P) = r_i(n_i \cdot P) = r_i \cdot N_i = Y_i$$

Table 1: The Nikooghadam-Shahriari-Saeidi key agreement scheme

Tag: $\{t_s, T_s, R_s, P\}$		Reader: $\{r_s, R_s, T_s, P\}$
		Pick $a_i, r_i \in F_p$, and a timestamp T_1.
		Compute $R_i = r_i \cdot P$, $Q_i = h(r_s \| a_i)$,
Check the timestamp T_1.	$\xleftarrow[\text{[open channel]}]{R_i,\ B_i,\ T_1}$	$key_i = r_i \cdot T_s$, $V_i = h(Q_i \| a_i \| key_i)$,
If true, compute $key_i = t_s \cdot R_i$,		$B_i = E_{key_i}(V_i \| Q_i \| a_i)$.
$DEC_{key_i}(B_i) = (V_i \| Q_i \| a_i)$.		
Check if $V_i = h(Q_i \| a_i \| key_i)$.		
If so, pick $n_i \in F_p$ and timestamp T_2,	$\xrightarrow{N_i,\ D_i,\ T_2}$	Check the timestamp T_2.
compute $N_i = n_i \cdot P$, $X_i = n_i \cdot R_i$,		If true, compute $Y_i = r_i \cdot N_i$.
$D_i = h(V_i \| R_i \| a_i \| Q_i \| X_i)$.		Check $D_i = h(V_i \| R_i \| a_i \| Q_i \| Y_i)$.
		If so, pick the timestamp T_3,
		compute $F_i = h(a_i \| Q_i \| Y_i)$,
Check the timestamp T_4.	$\xleftarrow{F_i,\ T_3}$	$SK = h(Q_i \| key_i \| Y_i)$.
If true, check $F_i = h(a_i \| Q_i \| X_i)$.		
If so, compute $SK = h(Q_i \| key_i \| X_i)$.		

3 Insecurity against impersonation attack

Though the Nikooghadam-Shahriari-Saeidi scheme is interesting, we find it is insecure against impersonation attack. As for this property, it argues that (see page 5, Ref.[8]):

> *Let us assume an attacker has access to R_i and N_i because of an insecure channel. If attackers want to create a tampered version of D_i without the reader realizing this, an attacker requires access to the V_i, a_i, Q_i, and Y_i parameters. However, they require the key_i to the last four parameters. Access(ing) to key_i, (one) needs (to) access to r_i, but an attacker does not have access to r_i based on the ECDLP theorem. Also, (an) attacker requires access to the t_s to calculate key_i; therefore, such an attack is impossible for (the) attacker.*

The simple argument is not sound. In fact, the reader's secret key r_s is simply invoked to compute the hash value $Q_i = h(r_s\|a_i)$. Besides, the reader's public key R_s is not used. The inherent relationship $R_s = r_s \cdot P$ is not utilized at all. That means the tag has no means of authenticating the reader.

An adversary who knows the tag's public key T_s and public parameter P can impersonate the reader to cheat the the tag. In fact, the adversary only needs to do as follows (see Table 2, for comparison, we redraw the table). In this case, there is no way for the tag to discriminate the hash values $h(\beta\|a_i)$ and $h(r_s\|a_i)$, which is really generated by invoking the secret key r_s.

Table 2: An impersonation attack against the Nikooghadam-Shahriari-Saeidi scheme

Tag: $\{t_s, T_s, P\}$		Adversary: $\{T_s, P\}$
		Pick $a_i, r_i, \beta \in F_p$, and a timestamp T_1.
		Compute $R_i = r_i \cdot P$, $Q_i = h(\beta\|a_i)$,
Check the timestamp T_1.	$\xleftarrow[\text{[open channel]}]{R_i,\ B_i,\ T_1}$	$key_i = r_i \cdot T_s$, $V_i = h(Q_i\|a_i\|key_i)$,
If true, compute $key_i = t_s \cdot R_i$,		$B_i = E_{key_i}(V_i\|Q_i\|a_i)$.
$DEC_{key_i}(B_i) = (V_i\|Q_i\|a_i)$.		
Check if $V_i = h(Q_i\|a_i\|key_i)$.		
If so, pick $n_i \in F_p$ and timestamp T_2,	$\xrightarrow{N_i,\ D_i,\ T_2}$	Check the timestamp T_2.
compute $N_i = n_i \cdot P$, $X_i = n_i \cdot R_i$,		If true, compute $Y_i = r_i \cdot N_i$.
$D_i = h(V_i\|R_i\|a_i\|Q_i\|X_i)$.		Check $D_i = h(V_i\|R_i\|a_i\|Q_i\|Y_i)$.
		If so, pick the timestamp T_3,
		compute $F_i = h(a_i\|Q_i\|\|Y_i)$,
Check the timestamp T_4.	$\xleftarrow{F_i,\ T_3}$	$SK = h(Q_i\|key_i\|Y_i)$.
If true, check $F_i = h(a_i\|Q_i\|X_i)$.		
If so, compute $SK = h(Q_i\|key_i\|X_i)$.		

4 Further discussions

As we see, the value $key_i = r_i \cdot T_s$ is used as a symmetric key for the encryption algorithm $E(\cdot)$ and decryption algorithm $DEC(\cdot)$, i.e.,

$$B_i = E_{key_i}(V_i \| Q_i \| a_i), \quad DEC_{key_i}(B_i) = (V_i \| Q_i \| a_i)$$

But the value is not suitable for the use because it is only a point over the underlying elliptic curve. Usually, one needs to convert the point into a random string with fixed length by hashing. Namely, set the symmetric key as $key_i = h(r_i \cdot T_s)$.

Note that the process

$$V_i \| Q_i \| a_i \xrightarrow{E_{key_i}} B_i \xrightarrow{DEC_{key_i}} V_i \| Q_i \| a_i$$

is a common encryption-decryption paradigm. Its confidentiality depends on the privacy of key_i. Generally, the final session key $SK = h(Q_i \| key_i \| X_i)$ is also used as a secret key for a common encryption-decryption paradigm. That means it becomes a simple repetitive process by exchanging key_i for SK. Naturally speaking, the Nikooghadam-Shahriari-Saeidi scheme is a variation of the general public key encryption. In view of this fact, we do not think it is necessary to revise the scheme.

5 Conclusion

We show that the Nikooghadam-Shahriari-Saeidi authentication and key agreement scheme is flawed. It seems difficult to revise the scheme because of its simple secret-key invoking mechanism. The findings in this note could be helpful for the future work on designing such schemes.

References

[1] M. Lavanya, V. Natarajan: Lightweight key agreement protocol for IoT based on IKEv2. *Comput. Electr. Eng.*, 64, 580-594 (2017)

[2] B. L. Parne, S. Gupta, N. S. Chaudhari: PSE-AKA: Performance and security enhanced authentication key agreement protocol for IoT enabled LTE/LTE-A networks. *Peer-to-Peer Netw. Appl.*, 12(5), 1156-1177 (2019)

[3] P. Tedeschi, S. Sciancalepore, A. Eliyan, R. D. Pietro: LiKe: Lightweight Certificateless Key Agreement for Secure IoT Communications. *IEEE Internet Things J.*, 7(1), 621-638 (2020)

[4] C. M. Chen, X. Deng, W. Gan, J. Chen, SK H. Islam: A secure blockchain-based group key agreement protocol for IoT. *J. Supercomput.*, 77(8), 9046-9068 (2021)

[5] K. Mahmood, M. F. Ayub, S. Z. Hassan, Z. Hassan, Z. Lv, S. A. Chaudhry: A seamless anonymous authentication protocol for mobile edge computing infrastructure. *Computer Communications*, 186, 12-21 (2022)

[6] A. Tomar, N. Gupta, D. Rani, S. Tripathi: Blockchain-assisted authenticated key agreement scheme for IoT-based healthcare system. *Internet Things*, 23, 100849 (2023)

[7] B. Zahednejad, T. Huang, S. Kosari, X. Ren: A Lightweight, Secure Big Data-Based Authentication and Key-Agreement Scheme for IoT with Revocability. *Int. J. Intell. Syst.*, 1-19 (2023)

[8] M. Nikooghadam, H. R. Shahriari, S. T. Saeidi: HAKECC: highly efficient authentication and key agreement scheme based on ECDH for RFID in IOT environment. *J. Inf. Secur. Appl.*, 76: 103523 (2023)