A note on "authenticated key agreement protocols for dew-assisted IoT systems"

Zhengjun Cao[1], Lihua Liu[2]

Abstract. We show that the key agreement scheme [J. Supercomput., 78:12093-12113, 2022] is flawed. (1) It neglects the representation of a point over an elliptic curve and the basic requirement for bit-wise XOR, which results in a trivial equality. By the equality, an adversary can recover a target device's identity, which means the scheme fails to keep anonymity. (2) It falsely requires that the central server should share its master secret key with each dew server. (3) The specified certificate is almost nonsensical.
Keywords: Authentication, Key agreement, Dew computing, Certificate, Anonymity

1 Introduction

Dew computing makes use of the capabilities of personal computers along with cloud services in a more reliable manner, which requires that: (1) the local device must be able to provide service without a continuous connection to the Internet, (2) the application must be able to connect to the cloud service and synchronize data. As we see, fog computing mainly involves devices such as routers and sensors in the Internet of Things (IoT), while dew computing mainly involves on-premises computers.

In 2021, Alaoui *et al.* [1] have presented an ECC-based authentication protocol for radio frequency identification (RFID) systems. Shortly after this work, Braeken [2] extended it to a key agreement scheme which could be suitable for dew computing. The scheme has many security features, including mutual authentication between devices and dew servers, confidentiality of the derived secret session key, anonymity, and unlinkability. The anonymity means that the identity of a device cannot be revealed by an adversary. In this note, we show that the scheme is flawed.

2 Review of the key agreement scheme

The scheme involves elliptic curve cryptography (ECC) operations [3]. Let P be a generator of group $\mathbb{G}$ with the prime order q, over the elliptic curve $\mathbb{E}$ defined in the finite field F_p. $h(\cdot)$ is a hash function with output in F_q^*. It has three entities: a central sever (trusted third party, TTP), IoT devices (or sensor nodes), and dew servers. It consists of four phases: initialization, registration, authentication and key agreement, and certificate revocation. In the discussed model, the attacker can eavesdrop on the communication channel, and can also change, replay, insert or delete parts of the message. The

[1]Department of Mathematics, Shanghai University, Shanghai, 200444, China
[2]Department of Mathematics, Shanghai Maritime University, Shanghai, 201306, China.
Email: liulh@shmtu.edu.cn

powerful attacker in possession of a device or a dew server, can abuse its available key material to retrieve information of other entities or to impersonate other entities. The scheme can be described as follows (see Table 1).

Table 1: The Braeken's key agreement scheme

Dew server	TTP	Device
	Set its private/public key pair as (d_{TTP}, Q_{TTP}), s.t., $Q_{TTP} = d_{TTP}P$.	
	For the identity ID_r and certificate $Cert_r$, derive the private/public key pair (d_r, Q_r), $Q_r = d_rP = h(ID_r\|Cert_r)Cert_r + Q_{TTP}$. Set a database DB for revoked identities and certificates $(ID_{rn}, Cert_{rn})_{rn}$	
Store $ID_r, Cert_r, d_r, Q_r, Q_{TTP}$, $(ID_{rn}, Cert_{rn})_{rn}$ in its memory.	$\overset{ID_r, Cert_r, d_r, Q_r, Q_{TTP}}{\underset{(ID_{rn}, Cert_{rn})_{rn}}{\Longleftarrow}}$	
	For the identity ID_n and certificate $Cert_n$, derive the private/public key pair (d_n, Q_n), $Q_n = d_nP = h(ID_n\|Cert_n)Cert_n + Q_{TTP}$	Store $ID_n, Cert_n, d_n, Q_n$,
	$\xRightarrow{ID_n, Cert_n, d_n, Q_n, Q_r, Q_{TTP}}$	Q_r, Q_{TTP} in its memory.
Dew server	Key agreement phase	Device
Pick r_r, r_1, compute $R_r = r_rP, R_1 = r_1P$, $s_r = r_1 - h(R_r\|Q_r\|R_1\|T_i)d_{TTP}$	$\xrightarrow{R_r, Q_r, R_1, T_i, s_r}$	Check $s_rP = R_1 - h(R_r\|Q_r\|R_1\|T_i)Q_{TPP}$. Pick r_n, r, compute $R_n = (r_n + d_n)P$, $C = (ID_n\|Cert_n\|r) \oplus (r_n + d_n)(Q_r + R_r)$,
$(ID_n\|Cert_n\|r) = C \oplus (d_r + r_r)R_n$. If ID_n in DB, stop, else compute $h(ID_n\|Cert_n\|R_r\|R_n\|r) = (h_1\|h_2\|h_3)$, $Q_n = h(ID_n\|Cert_n)Cert_n + Q_{TTP}$. Check $s_nP = R_n - (h_2 \oplus h_3)Q_n$.	$\xleftarrow{R_n, C, s_n}$	$h(ID_n\|Cert_n\|R_r\|R_n\|r) = (h_1\|h_2\|h_3)$, $s_n = (r_n + d_n) - d_n(h_2 \oplus h_3)$. Set the session key $SK = h_3$.
Set the session key $SK = h_3$.	$\xrightarrow{h_1 \oplus h_3}$	If h_1 correct, OK.

3 The flaws in the scheme

The Boolean logic operation XOR, denoted by $\oplus$, is widely used in cryptography which compares two input bits and generates one output bit. If the bits are the same, the result is 0. If the bits are different, the result is 1. When the operator is performed on two strings, they must be of a same bit-length. Otherwise, the shorter string should be stretched by padding some 0s to its left side. In this case, the partial string corresponding to the padding bits is eventually exposed to the adversary.

Though the key agreement scheme is interesting, we find, it neglects the representation of a point over an elliptic curve and the basic requirement for XOR, which results in the following trivial equality.

3.1 A trivial equality

By the expression $Q_n = d_nP = h(ID_n\|Cert_n)Cert_n + Q_{_{TTP}}$, it is easy to find that $Cert_n$ is a point over the underlying elliptic curve. $(r_n + d_n)(Q_r + R_r)$ is also a point over the elliptic curve. Hence,

$$Cert_n \in F_p \times F_p, \qquad (r_n + d_n)(Q_r + R_r) \in F_p \times F_p \tag{1}$$

Without loss of generality, we assume that both binary representations of two points have a same length. In view of that the random nonce $r \in Z_q$, where q is the order of the group $\mathbb{G}$, we find the binary string of concatenation $(Cert_n\|r)$ is longer than that of $(r_n + d_n)(Q_r + R_r)$, i.e.,

$$\text{BitLength}(Cert_n\|r) > \text{BitLength}((r_n + d_n)(Q_r + R_r)) \tag{2}$$

Therefore, in the expression

$$C = (ID_n\|Cert_n\|r) \oplus (r_n + d_n)(Q_r + R_r) \tag{3}$$

the binary representation of ID_n is directly inserted into that of C. Since the parameter C is sent to the dew server via an insecure channel, and can be captured by an outer attacker, the adversary can easily retrieve ID_n from C. Consequently, the equality Eq.(3) is insufficient to blind the device's identity ID_n. Thus, the scheme fails to keep device anonymity. To overcome this shortcoming, it needs to introduce other encryption mechanism to securely transfer the device's identity and certificate.

3.2 A false requirement

Note that the central server's private/public key pair is $(d_{_{TTP}}, Q_{_{TTP}})$, s.t., $Q_{_{TTP}} = d_{_{TTP}}P$, where P is a generator of the group $\mathbb{G}$ over the underlying elliptic curve. In the key agreement phase, the dew server needs to compute

$$s_r = r_1 - h(R_r\|Q_r\|R_1\|T_i)d_{_{TTP}}$$

which means that the central server's private key $d_{_{TTP}}$ should be equally distributed to each dew server. Apparently, this requirement is irrational. To overcome this shortcoming, it can be fixed as follows (see Table 2).

Table 2: The revised key agreement phase

Dew server: $\{d_r\}$	Key agreement phase	Device: $\{d_n\}$
Pick r_r, r_1, compute $R_r = r_rP, R_1 = r_1P$, $s_r = r_1 - h(R_r\|Q_r\|R_1\|T_i)d_r$	$\xrightarrow{R_r, Q_r, R_1, T_i, s_r}$	Check $s_rP = R_1 - h(R_r\|Q_r\|R_1\|T_i)Q_r$. Pick r_n, r, compute $R_n = (r_n + d_n)P$, $C = (ID_n\|Cert_n\|r) \oplus (r_n + d_n)(Q_r + R_r)$,
$(ID_n\|Cert_n\|r) = C \oplus (d_r + r_r)R_n$. If ID_n in DB, stop, else compute $h(ID_n\|Cert_n\|R_r\|R_n\|r) = (h_1\|h_2\|h_3)$, $Q_n = h(ID_n\|Cert_n)Cert_n + Q_{_{TTP}}$. Check $s_nP = R_n - (h_2 \oplus h_3)Q_n$.	$\xleftarrow{R_n, C, s_n}$	$h(ID_n\|Cert_n\|R_r\|R_n\|r) = (h_1\|h_2\|h_3)$, $s_n = (r_n + d_n) - d_n(h_2 \oplus h_3)$. Set the session key $SK = h_3$.
Set the session key $SK = h_3$.	$\xrightarrow{h_1 \oplus h_3}$	If h_1 correct, OK.

Correctness. Since $Q_r = d_r P, Q_n = d_n P$, we have

$$s_r P = (r_1 - h(R_r \| Q_r \| R_1 \| T_i) d_r) P = r_1 P - h(R_r \| Q_r \| R_1 \| T_i) d_r P = R_1 - h(R_r \| Q_r \| R_1 \| T_i) Q_r$$
$$(d_r + r_r) R_n = (d_r + r_r)(r_n + d_n) P = (r_n + d_n)(d_r P + r_r P) = (r_n + d_n)(Q_r + R_r)$$
$$s_n P = ((r_n + d_n) - d_n(h_2 \oplus h_3)) P = (r_n + d_n) P - (h_2 \oplus h_3) d_n P = R_n - (h_2 \oplus h_3) Q_n$$

3.3 The nonsensical certificate

To generate the secret keys d_r, d_n for the dew server and the device, respectively, the TTP has to express the certificates $Cert_r = aP$ and $Cert_n = bP$ for some integers a, b, such that

$$Q_r = d_r P = h(ID_r \| Cert_r) Cert_r + Q_{TTP},$$
$$Q_n = d_n P = h(ID_n \| Cert_n) Cert_n + Q_{TTP}.$$

Namely, $d_r = ah(ID_r \| Cert_r) + d_{TTP}, d_n = bh(ID_n \| Cert_n) + d_{TTP}$.

Given $Cert_r$ and P, it is impossible to compute a such that $Cert_r = aP$, which is just the discrete logarithm problem over the elliptic curve. So, the TTP needs to first select a nonce $a \in F_q^*$ and then assign $Cert_r \leftarrow aP$. Notice that a certificate is a string stating that particular facts are true. But $Cert_r$ is randomly generated, and is almost nonsensical.

4 Conclusion

We show that the Braeken's key agreement scheme is flawed. The findings in this note could be helpful for the future work on designing such schemes for dew computing scenario.

References

[1] H. Alaoui and *et al.* , A highly efficient ECC-based authentication protocol for RFID. J. Sensors, 8876766:1–8876766:16 (2021)

[2] A. Braeken, Authenticated key agreement protocols for dew-assisted IoT systems. J. Supercomput. 78(10), 12093–12113 (2022)

[3] D. Hankerson, S. Vanstone, A. Menezes, Guide to elliptic curve cryptography. Springer New York, USA (2006)