Security Analysis of a Recent Pairing-based Certificateless Authenticated Key Agreement Protocol for Blockchain-based WBANs

Yong-Jin Kim[1], Dok-Jun An[1], Kum-Sok Sin[2], Son-Gyong Kim[3]

[1] *Faculty of Mathematics, KIM IL SUNG University, Pyongyang, 999093, D.P.R of Korea*

[2] *Pyongyang Software JDC, 999093, D.P.R of Korea*

[3] *Institute of Management Practice, Pyongyang, 99903, D. P. R of Korea*

Email: kyj0916@126.com.

Abstract: In this paper, we proposed some vulnerabilities of a recent pairing-based certificateless authenticated key agreement protocol for blockchain-based wireless body area networks (WBAN). According to our analysis, this protocol is insecure against key offset attack (KOA), basic impersonation attack (BIA), and man-in-the-middle attack (MMA) of the malicious key generation center (KGC) administrators. We also found and pointed out some errors in the description of the protocol.

Introduction*:* The WBAN environment is vulnerable to different security and privacy threats as different kinds of attacks, such as leakage of man-in-the-middle, impersonation, and denial-of-service attacks can be mounted by an adversary. Therefore, several certificateless authenticated key agreement (CLAKA) protocols for WBANs based on the blockchains have been proposed and widely used.

In this paper, we analyzed some security vulnerabilities of a pairing-based CLAKA protocol for WBANs proposed by Mwitende [1]. According to our analysis, this protocol is insecure KOA, BIA, and MMA of malicious KGC. We also observed some errors in the protocol description.

Related works: Blake-Wilson proposed the issue of an AKA protocol with shared key confirmation and mentioned the KOA [2].

As Al-Riyami proposed in [3], there are two types of adversaries with different capabilities in the CLAKA protocol. Type I adversaries act as dishonest users whereas type II adversaries act as malicious KGC administrators. A type I adversary does not have access to the master secret key of the KGC, but the adversary can replace the public keys of any entity with a value of his choice. Type II adversaries have access to the master secret key but cannot replace any user's public key.

Not all KGC administrators can always be trusted. If at least one of the KGC administrators has a malicious mind, they can create false assurances, such as forged public keys, to impersonate the users registered in the system, even if they do not know the user's secret key and cannot replace the public key. In this case, the protocol is said to have a trust level of 2. Trust level 3 implies that KGC cannot calculate the user's private key, and if KGC creates fake assurance, it will be exposed [4].

Au et al. proposed the concept of malicious-but-passive KGC, in the sense that the KGC would not actively replace the user public key or corrupt the user secret key [5]. Subsequently,

Ge et al. proposed an e2CK model in which a malicious KGC had stronger attack capabilities. He assumed that the malicious KGC is allowed to replace the public keys of any party, which would be counted as the corruption of one secret, and could also replace the public keys of any party after the test query has been issued [6].

He et al. proposed a new anonymous authentication protocol for WBANs with provable security [7]. Sun, H. et al. proposed strongly secure a pairing-free CLAKA protocol for low-power devices [8]. However, Renu et al. proved that the protocol is vulnerable to the BIA and MMA of malicious KGC [9].

Required security properties: This section describes the security properties of the CLAKA protocol considered in this study [2, 3, 6, 9].

Key Offset Attack Resilience (KOAR): In the KOA, the adversary modifies the message sent by the sender and sends it to the receiver. As a result, the final session keys calculated by both parties are different. For example, an adversary multiplies the sender's ephemeral public key by a random value and then sends it to the receiver. Many AKA protocols that do not verify the final session key are vulnerable to this attack. The difference from MMA is that an adversary does not use any communication security information. This security property implies that no such case should exist.

Basic Impersonation Resilience (BIR): This security property implies that an adversary cannot impersonate a legitimate communication participant without knowing its static private key.

Man-in-the-Middle Attack Resilience (MMAR): The adversary impersonates himself as *A* between participants *A* and *B*, and shares the key with *B*. Of course, participant *A* believes that he shares the key with the legitimate participant *B*. Similarly, the adversary can impersonate himself as *A* and share a key with *B*. This security property implies that no such case should exist.

Security Analysis of Mwitende et al.'s protocol: Mwitende et al. proposed a new CLAKA protocol that can support authenticated key agreements between the nodes and controllers. The first step of the protocol includes a pairing-based CLAKA for WBANs. The second step consists of node authentication and verification. First, we briefly review the first step of the protocol [1].

1) A brief review of the first step of Mwitende's protocol

The scheme can generate a session key between controller *C* and node *N* in the following way.

1) Setup: A KGC takes as input security parameter η and performs as follows.
   - Select a cyclic additive group $\mathbb{G}_1$ of order q, and multiplicative $\mathbb{G}_2$, a generator P of $\mathbb{G}_1$ and bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$.
   - Selects a random master key $z \in Z_q^*$ and compute $P_{pub} = zP$.
   - Selects hash functions: $H_1: \{0,1\}^* \times \mathbb{G}_1 \longrightarrow \mathbb{G}_1, H_2: \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}_2 \times \mathbb{G}_1^3 \longrightarrow \{0,1\}^j$. The list $prs = \{\mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, H_1, H_2, \eta\}$.
2) Partial-Private-Key-Extract: KGC produces the partial private key of participant i as $S_i = zQ_i$ such that $Q_i = H_1(ID_i)$.
3) Set-Secret-Value: For a participant i with an identity ID_i, the algorithm selects a value y_i at random, sets y_i as participant's secret value.
4) Set-Private-Key: It takes as input prs, ID_i, and the secret value y_i, the partial private key S_i to return $PR_i = y_i S_i$.

5) Set-Public-Key: The algorithm takes prs, the secret value y_i and ID_i for the participant to return the public key $X_i = y_i P$, and computes $Y_i = y_i Q_i$
6) Key-Agreement: Algorithm 1 illustrates steps for session key computation.

ALGORITHM 1: Algorithm for CLAKA scheme.

1: ***in:*** ID_i, prs, S_i, Q_i, ***out:*** $SK = H(ID_C, ID_N, T_C, T_N, K)$
2: **User randomly selects** $y_i \in Z_q^*$
3: **Compute** $PR_i = y_i S_i$
4: **Compute** $X_i = y_i P, Y_i = y_i Q_i$
5: // A session key is computed as follows
6: *C* **randomly select** $r_C \in Z_q^*$
7: *C* **Compute** $T_C = r_C P$
8: *C* **send** (ID_C, T_C, Y_C) **to** *N*
9: *N* **randomly select** $r_N \in Z_q^*$ and
10: *N* **Compute** $T_N = r_N P$
11: *N* **send** (ID_N, T_N, Y_N) **to** *C*
12: *N* **compute** $K_{NC} = e\left(T_C + Y_C,\ r_N P_{pub} + PR_N\right)$
13: *C* **compute** $K_{CN} = e\left(T_N + Y_N,\ r_c P_{pub} + PR_C\right)$
14: **if** $K_{CN} = K_{NC} = K$ **then**
15: **Return a session key** $SK = H(ID_C, ID_N, T_C, T_N, K)$
16: **end if**

2) Key Offset Attack

The protocol by Mwitende et al. is vulnerable to KOA. The adversary first intercepts the pairs (ID_C, T_C, Y_C), (ID_N, T_N, Y_N) exchanged between controller *C* and node *N*. The adversary randomly chooses $\alpha \in Z_q^*, \alpha \neq 1$ and computes $T_C' = \alpha T_C, T_N' = \alpha T_N$.

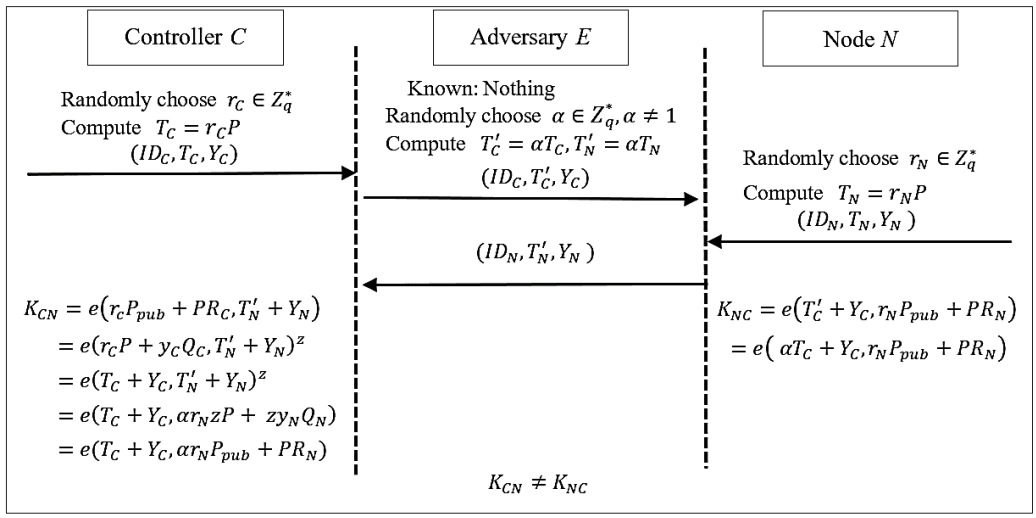

Fig 1 Key Offset Attack against Mwitende et al.'s protocol.

Next, the modified pair (ID_C, T_C', Y_C), (ID_N, T_N', Y_N) is sent to *N* and *C* respectively. Then

$$K_{CN} = e\left(r_c P_{pub} + PR_C, T_N' + Y_N\right)$$

$$= e(r_C P + y_C Q_C, T'_N + Y_N)^z$$
$$= e(T_C + Y_C, T'_N + Y_N)^z$$
$$= e(T_C + Y_C, \alpha r_N z P + z y_N Q_N)$$
$$= e(T_C + Y_C, \alpha r_N P_{pub} + PR_N)$$
$$K_{NC} = e\left(T'_C + Y_C, r_N P_{pub} + PR_N\right)$$
$$= e\left(\alpha T_C + Y_C, r_N P_{pub} + PR_N\right),$$

so

$$K_{CN} \neq K_{NC}.$$

The adversary replaces the ephemeral public key with another key and sends it to the receiver. However, the receiver cannot detect whether the ephemeral public key has been replaced by an adversary; therefore, the KOA can easily succeed (see Fig 1).

The KOA is effective in energy-constrained applications, such as WBANs. The attacker can repeat this attack and completely consume energy resources by preventing the node and controller from sharing a key (see Fig 2).

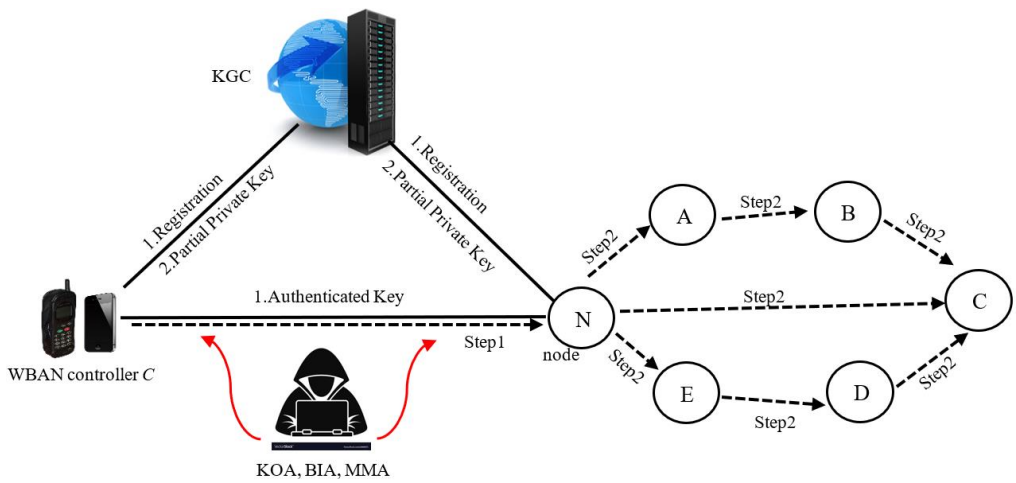

Fig 2 System model and KOA, BIA, and MMA.

3) Basic Impersonation Attack by malicious KGC

The protocol proposed by Mwitende is insecure against the BIA of malicious KGCs. The malicious administrator does not know the private secret value y_N of node N that is registered in the system. However, he knows node N's partial private key S_N. Therefore, malicious KGC randomly chooses $y'_N \in Z_q^*, r'_N \in Z_q^*$ and calculates $Y'_N = y'_N Q_N$, $T'_N = r'_N P$ and $PR'_N = y'_N S_N$. Next, the malicious KGC sends (ID_N, T'_N, Y'_N) to the controller C (see Fig 3).

The controller C thinks it was sent by the node N and calculates $K_{CN} = e\left(r_C P_{pub} + PR_C, T'_N + Y'_N\right)$, $SK_{CN} = H_2(ID_C, ID_N, T_C, T'_N, K_{CN})$. Meanwhile, the malicious KGC administrator impersonates node N and calculates $K_{NC} = e\left(T_C + Y_C,\ r'_N P_{pub} + PR'_N\right) = e\left(T_C + Y_C,\ r'_N P_{pub} + y'_N S_N\right) = e(T_C + Y_C,\ z(r'_N P + y'_N Q_N) = e(T_C + Y_C,\ r'_N P + y'_N Q_N)^z = e(T_C + Y_C,\ T'_N + Y'_N)^z = e(r_C z P + y_C z Q_C,\ T'_N + Y'_N) = e(r_C P_{pub} + PR_C,\ T'_N + Y'_N)$ so, $SK_{NC} = H_2(ID_C, ID_N, T_C, T'_N, K_{NC})$. In this way, the malicious KGC can impersonate itself as N without knowing N's secret value y_N, because $\left(T_C + Y_C,\ r'_N P_{pub} + PR'_N\right) = e\left(r_C P_{pub} + PR_C,\ T'_N + Y'_N\right) = K_{CN}$. Finally, $SK_{NC} = SK_{CN}$.

It is clear that the malicious KGC administrator does not replace the (ID_A, T_N, Y_N) sent by node N to controller C with (ID_N, T'_N, Y'_N). The malicious KGC creates a (ID_N, T'_N, Y'_N) himself and sends it to C.

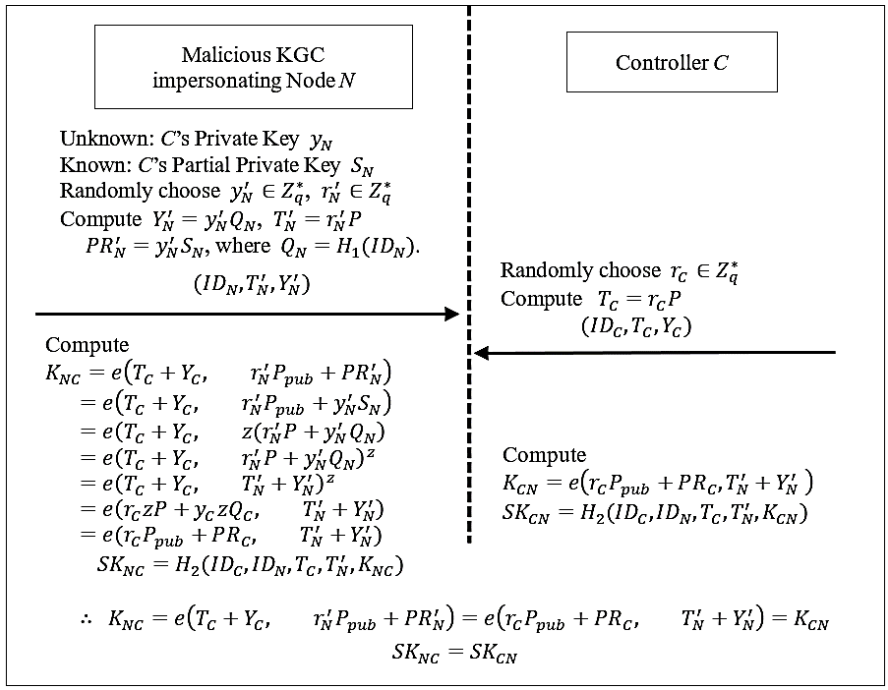

Fig 3 Basic Impersonation Attack by malicious KGC.

4) Man-in-the-Middle Attack by malicious KGC

If the BIA discussed above is executed between controller C and node N, respectively, the malicious KGC administrator E can share the session key with C and N (see Fig 4).

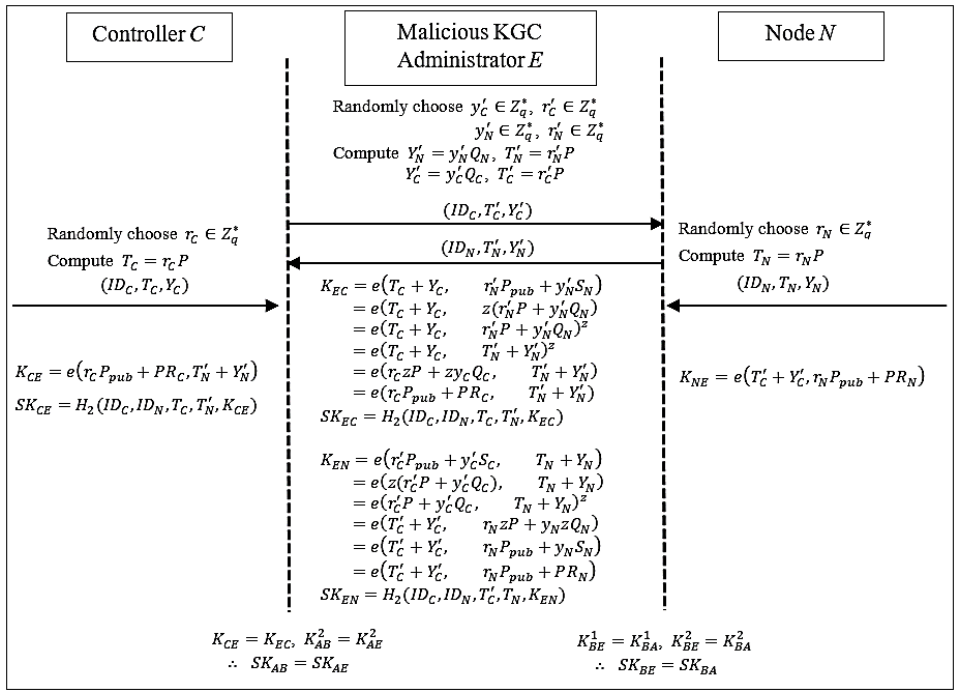

Fig 4 Man-In-The-Middle Attack by malicious KGC.

5) Errors of notation in Mwitende et al.'s protocol

There are some errors in section 5.1 of the protocol [1].

1) In the algorithm 1 and table 3, it should be described as $K_{NC} = e(r_N P_{pub} + PR_N, T_C + Y_C)$, not $K_{NC} = e(T_C + Y_C,\ r_N P_{pub} + PR_N)$.
2) The public key $X_i, i \in \{C, N\}$ was not used at all in the key agreement or the ring signing stages.
3) In the setup, the second hash function is $H_2: \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}_2 \times \mathbb{G}_1^3 \longrightarrow \{0,1\}^j$, so the number of inputs should be 6. However, in algorithm 1 and table 3 the number of parameters is 5.
4) In table 3, it should be described as (ID_C, T_C, Y_C) and (ID_N, T_N, Y_N), not (ID_C, T_C) and (ID_N, T_N). If Y_C and Y_N are not exchanged, they cannot be used in generating key SK.

***Conclusion**:* According to our analysis, the protocol of Mwitende et al. has trust level 2, because the protocol is insecure against KOA, malicious KGC's BIA, and MMA by the malicious KGC. The reason Mwitende's protocol is vulnerable against malicious KGC's BIA and MMA is that the partial private key $Q_i = H_1(ID_i)$ of the participant has been generated using only the participant's ID_i. KOAR can be easily achieved by using a hash function to verify the integrity of the shared key. If these causes are well considered, it will be possible to easily design a lightweight protocol with trust level 3 that can withstand the attacks considered above while considering the characteristics of WBANs. In the future, we will study more about it.

References

1. Mwitende, G. et al.: Certificateless authenticated key agreement for blockchain-based WBANs, J. Syst. Arch., 110 (2020) 101777, available at: https://doi.org/10.1016/j.sysarc.2020.101777
2 Blake-wilson, S. et al.: Key agreement protocols and their security analysis, Proc. Crypt. Cod, Cirencester, UK, 1997, in LNCS, vol. 1335, Springer, pp. 30–45, available at: https://doi.org/10.1007/BFb0024447
3. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography, Proc. ASIACRYPT 2003, Taipei, Taiwan, 2003, in LNCS, vol.2894, Springer, pp. 452–473, available at: https://doi.org/10.1007/978-3-540-40061-5_29
4. Girault, M.: Self-certified public keys, Proc EUROCRYPT 1991, Brighton, UK, in LNCS, vol. 547, Springer, pp. 490–497, available at: https://doi.org/10.1007/3-540-46416-6_42
5. Au, M.H. et al.: Malicious KGC Attacks in Certificateless Cryptography, Proc. of the 2nd ACM symposium on Information, computer and communications security, Singapore, 2007, in ASIACCS '07, pp. 302–311, available at: https://dl.acm.org/doi/10.1145/1229285.1266997
6. Lippold, G. et al.: Strongly Secure Certificateless Key Agreement, Proc. Third International Conference (Pairing 2009), Palo Alto, CA, USA, 2009, in LNCS, vol. 5671, Springer, pp. 206–230, available at: https://doi.org/10.1007/978-3-642-03298-1_14
7. He, D. et al.: Anonymous authentication for wireless body area networks with provable security, IEEE systems journal, 11 (2017), no. 4, pp. 2560-2601, available at: https://doi.org/10.1109/JSYST.2016.2544805
8. Sun, H. et al.: A strongly secure pairing-free certificateless authenticated key agreement protocol for low-power devices, Inf. Technol. Control. (42) 2013, 113–123, available at: https://doi.org/10.1007/s11432-015-5303-0
9. Daniel, R.M. et al.: An efficient eCK secure certificateless authenticated key agreement scheme with security against public key replacement attacks, J. Inf. Sec. Appl., 47 (2019), 156–172, available at: http://doi.org/10.1016/j.jisa.2019.05.003