Journal of Information Hiding and Multimedia Signal Processing ©2024 ISSN 2073-4212
Ubiquitous International Volume **15**, Number **3**, September **2024**

Lightweight WSN Anonymous Authentication Protocol Based on Daublon Filter

Zhi-Hao Yin*, Qi-Dong Ling

Xuzhou College of Industrial Technology, Xuzhou 221140, P. R. China
Email: liehu85@163.com, qidong66@163.com
*Corresponding author: Zhi-Hao Yin

Received June, 2024, revised July, 2024, accepted July, 2024.

Abstract. *With the rapid development of Internet of Things (IoT) technology, the security of Wireless Sensor Networks (WSN) has attracted increasing attention from researchers. WSN contains a lot of private data, so it is necessary to ensure that legitimate users can access sensor nodes and prevent malicious users from eavesdropping or tampering with data. In this paper, a lightweight anonymous authentication protocol based on multiple Bloom filters is proposed to solve the resource limitations of sensor nodes in WSN in computing power, storage and bandwidth. Firstly, Bloom filter is assigned to each sensor node in the initial stage to prevent the transmission of fraudulent query data. In the registration stage, the user generates the key parameters related to biometric features through the fuzzy extractor, and registers the identity information to the gateway using a pseudonym. In the authentication stage, the sensor node uses the fuzzy extractor to verify the user's identity through biometrics, and establishes a session key with the authorized user. The security analysis results show that the proposed protocol is anonymous and authentic, and can resist replay attacks, man-in-the-middle attacks and online password guessing attacks.*
Keywords: WSN; lightweight; anonymous authentication protocol; bloom filter

1. **Introduction.** In recent years, with the rapid development of IoT technology, the security of WSNs has been widely concerned by researchers. WSNs contain a large amount of private data, and any sensor node within the network is required to transmit data. Malicious users can not only eavesdrop on the data information of the sensor nodes, but also even impersonate the legitimate nodes to tamper with the data. Therefore, how to ensure legitimate users' access to sensor nodes is the primary issue in the current IoT field [1, 2]. Given the resource-constrained nature of sensor nodes in terms of computing power, storage, and bandwidth, many authentication protocols based on digital signatures and certificates cannot be directly applied to WSNs [3]. Traditional user authentication systems based on cryptographic theory are not only of high computational complexity, but also easily cracked by dictionary attacks. Compared with traditional cryptographic based systems, authentication protocols based on biometric processes have higher reliability and security [4]. Nam et al. [5] proposed an ECC based authentication protocol for WSN users, but it is not able to resist impersonation attacks and replay attacks. Xue et al. [6] designed a mutual authentication and timestamping based authentication system for WSN external users, which is susceptible to replay attacks. system which is vulnerable to replay attacks. Ren et al. [7] proposed a batch WSN user authentication protocol by applying Bloom filters to save multiple user identities and public keys but the user's identities are vulnerable to leakage. Banerjee et al. [8] proposed a biometrics

based user authentication method for WSNs but this method is not able to defend against man-in-the-middle attacks. Rehman et al. [9] proposed a real-time data-based user sharing authentication system for WSNs but it is computationally intensive. Temirlan and Li [10] proposed a fuzzy extractor based user authentication technique for WSNs which is resistant to known plaintext attacks. Lee et al. [11] proposed an anonymous user authentication mechanism for WSNs for for real-time information access in WSNs. However, this mechanism does not consider the anonymity of the sensor nodes. Khalid et al. [12] proposed an external user authentication scheme for WSN using PHOTON family of hash functions and elliptic curves, but the practicality is not good. Shao et al. [13] constructed a lightweight user authentication system for WSN based on PUF, but the security is not good. Sakthivel and Vidhya [14] Constructed an authentication scheme for WSNs in IoT environment based on the trust model, but the identity information of WSNs is not anonymous and needs to consume more computation and communication costs. It is clear from the analysis of existing research that the existing WSN user authentication methods have problems such as easy leakage of identity information and high computation volume. To address the above problems, this paper proposes a lightweight sensor network anonymous authentication protocol based on multiple Bloom filters. Firstly, Bloom filters are assigned to each sensor node in the initial stage to prevent fraudulent query data transmission. Then, in the registration phase users generate secret parameters related to their biometrics using fuzzy extractor and register their identity information with the gateway through pseudonyms. Finally, in the authentication phase, the sensor uses the fuzzy extractor to verify the user's identity through biometrics and establishes a session key with the authorised user. The security analysis results show that the proposed protocol is anonymous and authentic and is resistant to replay attacks, man-in-the-middle attacks, online password guessing attacks, etc. In addition, it is compared with five other protocols in terms of performance. The results show that the proposed protocol has low computational overhead and communication overhead.

2. **Preparatory knowledge.**

2.1. **Computationally difficult problems with elliptic curves.**

Definition 1: Elliptic Curve Discrete Logarithm Problem (ECDLP) [15]. Suppose $E_p(a,b)$ is an elliptic curve on $GF(p)$, given two points $P \in E_p(a,b)$ and $Q = k \times P \in E_p(a,b)$, for positive integers $k \in \mathbb{Z}_p^*$, where $Q = k \times P$ denotes a scalar multiplication such that the point P on the elliptic curve $E_p(a,b)$ adds to itself k times. Then, the ECDLP is to determine k given P and Q.

Definition 2: Elliptic Curve Deterministic Diffie-Hellman Problem (ECDDHP) [16]. Suppose $P \in E_p(a,b)$ is a point on an elliptic curve, and given a quaternion $(P, x \times P, y \times P, z \times P)$, the ECDDHP has to determine whether $z = x \times y$ or $\bar{z}$ is a mean, where $x, y, z \in \mathbb{Z}_p^*$.

2.2. **Bloom Filter.** The Bloom Filter (BF) [17] has the advantage of high query efficiency and small memory footprint to provide relevant membership queries quickly. The BF is a vector of β bits $\boldsymbol{V}$ that represents a set consisting of a components by applying γ secure hash functions $\boldsymbol{S}$. It utilizes probabilistic data structures to support set-association queries, including false positives $f^2 = (1 - e^{-\frac{\gamma \times m}{\beta}})^\gamma$.

A Bloom filter $\boldsymbol{B}$ consists of the algorithm $\boldsymbol{B} = (\text{BFGen}, \text{BFUpdate}, \text{BFCheck})$ defined as follows:

(1) BFGen(β, γ): has two input integers $\beta, \gamma \in \mathbb{N}$, which first samples γ generic hash functions $h^1, h^2, \ldots, h^\gamma$, where $h_j : \mathcal{X}_{U_j} \to [\beta]$. Define $H = (h^i)_{i\in\gamma}$ and $V = 0^\beta$ (i.e., V is a β bit array with all bits set to 0), and then output (H, V).

(2) BFUpdate($H, V, \mathcal{X}_U$): Given $H = (h^i)_{i\in\gamma}$, $V \in \{0,1\}^\beta$, $U_i \in \mathcal{U}$, the algorithm defines the update state V' by first assigning $V' = V$. Then, $V'[\eta]$ denotes the η-th bit of V', sets $V'[h^j(\mathcal{X}_U)] = 1$ for all $j \in [\gamma]$, and finally returns V'.

(3) BFCheck($H, V, \mathcal{X}_U$): Given $H = (h^i)_{i\in\gamma}$, $V \in \{0,1\}^\beta$, $U_i \in \mathcal{U}$, $V[\eta]$ denotes the η-th bit of V, $\mathcal{X}_U \subseteq \mathcal{K}$, the algorithm returns $\mathcal{X}_U \in \mathcal{K}$.

2.3. **Fuzzy Extractor.** Fuzzy Extractor (FE) [18] takes biometrics as input and outputs two random numbers. Using a given biometric input b, it extracts an almost random string μ [19]. FE mainly consists of generation algorithm (Gen) and regeneration algorithm (Rep).

(1) Gen: Input $b \in \mathcal{B}$, compute $R = \mathrm{Ext}(b)$ and $P = \mathrm{SS}(b)$, where R and P represent the values computed using the safe catch and output functions in the safe stalk algorithm, respectively.

(2) Rep: Enter b' and P. If $\mathrm{dis}(b, b') < t$, calculate $b = \mathrm{SS.Rec}(b', P)$, $R = \mathrm{Ext}(b)$, where b and b' are successive samples of the same random source, respectively. The string P and the key R are generated by the Gen algorithm in FE, only P needs to be saved. In the Rep algorithm, the random number b' is entered again, and the initial key R can be recovered by P if the errors of b' and b are within the specified threshold.

3. System Models and Authentication Protocol Constructs.

3.1. **System Model.** In this paper, we assume that a large-scale spatially distributed wireless sensor network in an IoT environment consists of a fixed Gateway Node (GW), a large number of WSN Nodes (WN), and an externally accessed user U_i. The system model is shown in Figure 1 and the main notations used in the protocol are shown in Table 1. The purpose of this model is to provide secure and reliable information services to authorized users U_i.

(1) GW: responsible for distributing keys for WN nodes and authenticating the authenticity of WN node identities. In addition, the GW needs to verify the registration information of the external access user U_i and authorize its legitimacy.

(2) WN: responsible for establishing session keys with external access users and authenticating their identity information; responsible for receiving keys distributed by GW and verifying their identity.

(3) External access user U_i: responsible for establishing secure session keys with WN and mutual authentication; registering real identity information with GW and mutual authentication.

For WN nodes, the communication cost is much higher than the computational cost. In this paper, we propose a fuzzy extractor-based anonymous authentication protocol for WSNs based on the above system model. It authenticates the user when querying the WN node itself and prevents unauthorized query information from flooding from the WN node to the GW node in order to overcome the brute force attack of the WN node.

3.2. **Protocol description.**

3.2.1. *Initialisation phase.* Before an external user accesses a sensor node, the GW node performs the following operations in offline mode. The GW generates and selects a number of secret and public parameters for each user U_i and for each deployed sensor node WN_j over a reliable communication channel. Since WSNs have limited battery and voltage resources, WNs are divided into different clusters.

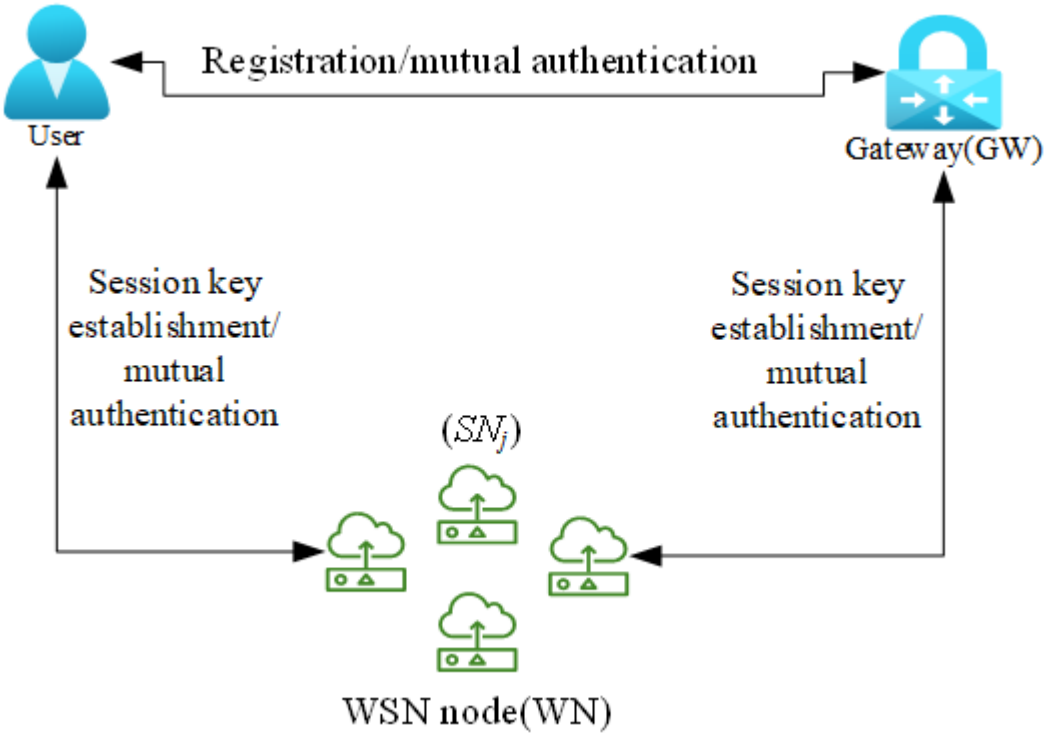

FIGURE 1. System modelling

TABLE 1. Definition of main symbols

Symbols	Description of definitions
U_i	The i-th authenticated user in the WSN
ID_{U_i}	Identities of U_i
B_i	The secret biological credentials of U_i
WN_j	The j-th sensor node in the WSN
PW_{U_i}	Passwords of U_i
ID_{WN_j}	Identities of SN_j
SC_i	Smart card of U_i
GW	Gateway
$E_p(a,b)$	Elliptic curves over finite fields F_p and $4a^3+27b^2 \neq 0 \pmod p$
V_{WN}	Bloom filter vector
$Gen(.)$	Fuzzy extractor generation function
$Rep(.)$	Regeneration function of fuzzy extractor
R_{U_i} and K_{U_i}	Public key and private key of U_i

(1) GW uses a backtracking algorithm to assign a BF vector V_{WN_j} to each sensor node WN_j in the cluster such that two neighbouring sensor nodes do not receive the same vector V_{WN_j} ($V_{WN_i} \neq V_{WN_j} \neq V_{WN_k}$, where $WN_1, \ldots, WN_j$ and WN_k belong to the same cluster C_l). We assume that the total number of sensor nodes in this cluster is k.

(2) GW takes a generating point P on the elliptic curve $E_p(a,b)$. Then computes the public-private key pairs $(R_{U_1}, R_{U_1} \times P), (R_{U_2}, R_{U_2} \times P), \ldots, (R_{U_\alpha}, R_{U_\alpha} \times P)$, where $R_{U_i} \in \mathbb{Z}_q^*(\forall i \in [1,\alpha])$.

(3) GW creates a set of vectors $\{V_{WN_1}, \ldots, V_{WN_j}, V_{WN_k}\}$, where $V_{WN_\theta} = \{v^1_{WN_\theta}, \ldots, v^\beta_{WN_\theta}\}$. For any $\theta \in [1,k]$, $H_{WN_\theta} = \{h^1_{WN_\theta}, \ldots, h^\gamma_{WN_\theta}\}$ can be obtained by computing a different

set of hash functions $h_{WN_1}, h_{WN_2}, \ldots, h_{WN_k}$. Here, the length of the vector V_{WN_θ} is β and the number of elements in H_{WN_θ} is γ.

$$V_{WN_\theta}^{\eta} = \begin{cases} 1, \forall \lambda \in [1,\gamma], \eta \in [1,\beta], h_{WN_\theta}^{\lambda}(R_{U_i} \times P) = \eta \\ 0, otherwise \end{cases} \tag{1}$$

(4) GW stores the public-private key pairs $R_{WN_\theta}, K_{WN_\theta} = R_{WN_\theta} \times P$ for $V_{WN_\theta}, H_{WN_\theta}$ and WN_0 in its internal control unit.

3.2.2. *User registration phase.* Before accessing the WN, the user needs to register the relevant information with the GW. Firstly, U_i selects their identity ID_{U_i}, password PW_{U_i}, and identifies their biometrics B_i on the reader of the particular terminal and then performs the following steps as shown in Figure 2.

(1) U_i generates function $Gen()$ using a fuzzy extractor and computes confidential and copy parameters as $(\sigma, \tau) = Gen(B_i)$. U_i generates two random numbers $a, b \in \mathbb{Z}_q^*$, computes $D = aQ$, $C = bP$, $PID_{U_i} = C \oplus ID_{U_i}$, and uses the hash function to compute $IPB_i = h(ID_{U_i} \parallel PW_{U_i} \parallel \sigma)$ and transmits $\langle D, PID_{U_i}, IPB_i \rangle$ to the GW over a secure channel.

(2) After receiving $\langle D, PID_{U_i}, IPB_i \rangle$, GW distributes the public-private key pair $(R_{U_i}, K_{U_i} = R_{U_i} \times P)$ for each user U_i, computes $C^* = sD$ and $ID_{U_i} = PID_{U_i} \oplus C^*$. Then GW performs the summation computation $HK_{U_i} = IPB_i \oplus R_{U_i}$, $A_{U_i} = IPB_i \oplus R_{U_i}$, $B_{U_i} = h(ID_{U_i} \parallel IPB_i \parallel R_{U_i})$ and W_{U_i}. Subsequently, the GW transmits $\langle h(\cdot), A_{U_i}, B_{U_i}, W_{U_i}, K_{U_i} \rangle$ to U_i over a secure channel.

$$HK'_{U_i} = \sum\nolimits_{\lambda=1}^{\gamma} h_{WN_1}^{\lambda}(K_{U_i}) \oplus \cdots \oplus \sum\nolimits_{\lambda=1}^{\gamma} h_{WN_k}^{\lambda}(K_{U_i}) = \oplus_{\theta=1}^{k} \left[\sum\nolimits_{\lambda=1}^{\gamma} h_{WN_\theta}^{\lambda}(K_{U_i}) \right] \tag{2}$$

$$W_{U_i} = h(ID_{U_i} \parallel R_{U_i} || IPB_i) \oplus h(ID_{U_i} \parallel HK_{U_i}) \tag{3}$$

(3) After getting $\langle h(\cdot), A_{U_i}, B_{U_i}, W_{U_i}, K_{U_i} \rangle$, U_i generates its own signing public-private key pair $(PR_{U_i} \in \mathbb{Z}_q^*, PK_{U_i} = PR_{U_i} \times P)$ and calculates C_{U_i}, A_{U_i}. Finally, U_i saves $\{h(\cdot), Gen(\cdot), Rep(\cdot), T, P, A_{U_i}, B_{U_i}, C_{U_i}, W_{U_i}, K_{U_i}, PK_{U_i}, \tau_i\}$ in the smart card SC_i.

$$C_{U_i} = PR_{U_i} \oplus h(\sigma_i \parallel PW_{U_i} || ID_{U_i} || R_{U_i}) \tag{4}$$

$$A'_{U_i} = (A'_{U_i} \oplus IPB_i) \oplus h(\sigma_i \parallel PW_{U_i}) = R_{U_i} \oplus h(\sigma_i \parallel PW_{U_i}) \tag{5}$$

3.2.3. *User authentication and session key establishment.* nIn this section, we apply the regeneration algorithm of fuzzy extractor $Rep(\cdot)$ to authenticate the user U_i by the biometric B_i entered during login. The Bloom filter vector $\{V_{WN_1}, V_{WN_2}, ..., V_{WN_k}\}$ of the sensor node $\{WN_1, WN_2, ..., WN_k\}$ is used to authenticate the user U_i, respectively. Then, the session key is established between the sensor node WN_j and the user U_i using ECDH based algorithm as shown in Figure 3.

Step 1: U_i inserts SC_i into the card reader of the specific terminal and generates 2 random numbers $a, b \in Z_q^*$. Calculate $D' = aQ = (D_1, D_2)$, $C = bP$, $PID_{U_i} = C \oplus ID_{U_i}$. Then present the pseudonym PID_{U_i} and the password PW_{U_i}, and mark its biometrics B'_i. SC_i uses the regeneration function $Rep(\cdot)$ to compute the secret parameters $\sigma_i = Rep(B'_i, \tau_i)$, and to use the hash function $h(\cdot)$ to calculate $IPB_i = h(ID_{U_i} \parallel PW_{U_i} \parallel \sigma_i)$, $R_{U_i} = A_{U_i} \oplus h(\sigma_i \parallel PW_{U_i})$, $B'_{U_i} = h(ID_{U_i} \parallel IPB_i \parallel R_{U_i})$. To ensure the accuracy of confidential credentials such as identities, biometric information, and secret parameters, SC_i needs to verify that $B'_{U_i} = B_{U_i}$ holds.

If not, the phase ends. Otherwise, SC_i computes $PR_{U_i} = C_{U_i} \oplus h(\sigma_i \parallel PW_{U_i} \parallel ID_{U_i} \parallel R_{U_i})$, $h(ID_{U_i} \parallel HK_{U_i}) \oplus h(ID_{U_i} \parallel R_{U_i} \parallel IPB_i)$, generates the current timestamp T_{U_i}, and computes m_1 and m_2. Next, U_i constructs the following message M_1, and sends M_1 to WN_j.

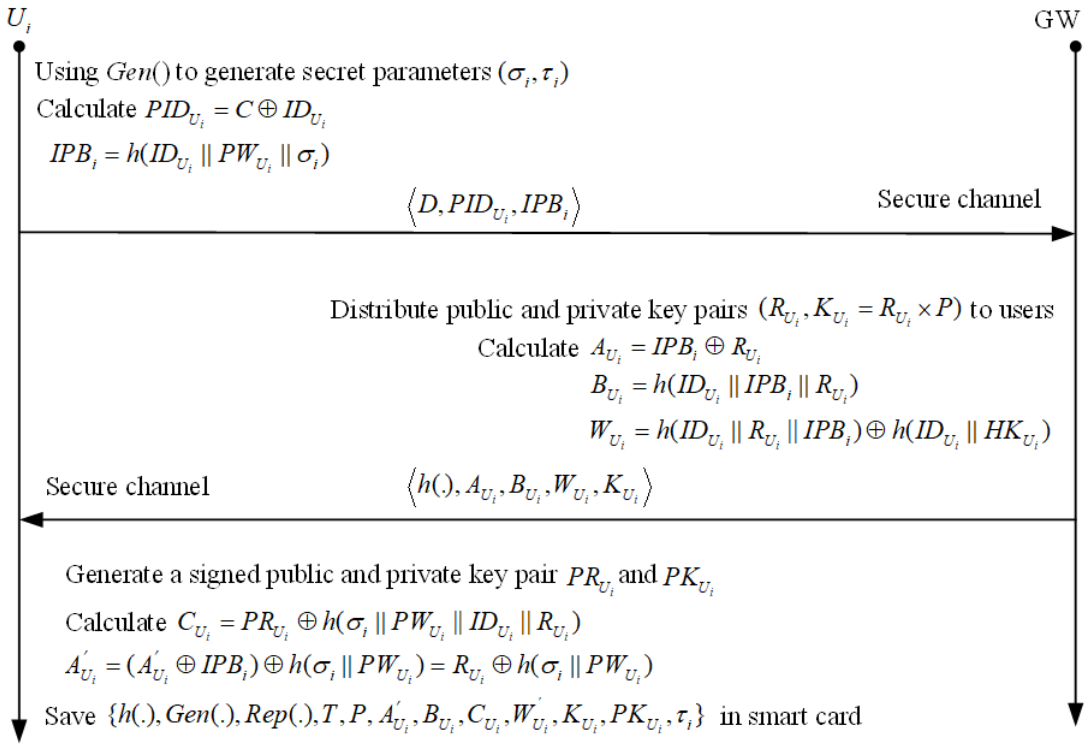

FIGURE 2. User registration process

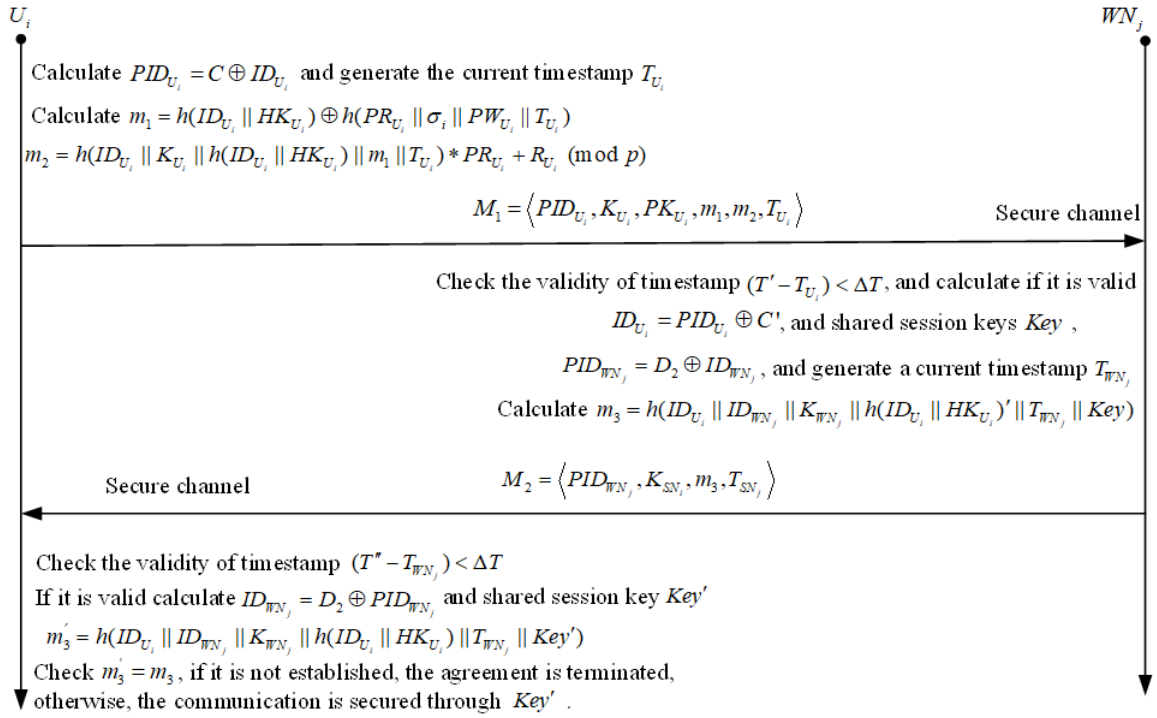

FIGURE 3. Authentication and session key establishment process

$$m_1 = h(ID_{U_i} \parallel HK_{U_i}) \oplus h(PR_{U_i} \parallel \sigma_i \parallel PW_{U_i} \parallel T_{U_i}) \tag{6}$$

$$m_2 = h(ID_{U_i} \parallel K_{U_i} \parallel h(ID_{U_i} \parallel HK_{U_i}) \parallel m_1 \parallel T_{U_i}) * PR_{U_i} + R_{U_i} \pmod p \quad (7)$$

$$M_1 = \langle PID_{U_i}, K_{U_i}, PK_{U_i}, m_1, m_2, T_{U_i} \rangle \quad (8)$$

Step 2: After receiving M_1, WN_j first checks the validity of the timestamp $(T' - T_{U_i}) < \Delta T$ to prevent replay attacks and DOS attacks, where T' is the time the message was received, and ΔT is the maximum transmission delay associated with M_1.

If the timestamp is invalid, the authentication procedure stops and indicates the presence of a security attack. Otherwise, WN_j verifies the legitimacy of the U_i identity through the BF, using Equation (1). If U_i is an illegal user, WN_j rejects the authentication request from U_i. Otherwise, WN_j computes $C' = s'D_1, ID_{U_i} = PID_{U_i} \oplus C'$, $PID_{WN_j} = D_2 \oplus ID_{WN_j}$.

Verify that $m_2 \times P = h(ID_{U_i} \parallel ID_{WN_j} \parallel K_{U_i} \parallel h(ID_{U_i} \parallel HK_{U_i}) \parallel m_1 \parallel T_{U_i}) \times PK_{U_i} + K_{U_i}$. If the equation is valid, WN_j generates the current timestamp T_{WN_j} and computes the session key Key shared with U_i, m_3, and gets the message M_2.

$$Key = h(T_{U_i} \parallel T_{SN_j} \parallel (R_{SN_j} \times K_{U_j}) \parallel h(ID_{U_i} \parallel HK_{U_i}) \parallel h(PR_{U_i} \parallel \sigma_i \parallel PW_{U_i} \parallel T_{U_i})) \quad (9)$$

$$m_3 = h(ID_{U_i} \parallel ID_{WN_j} \parallel K_{WN_j} \parallel h(ID_{U_i} \parallel HK_{U_i}) \parallel T_{WN_j} \parallel Key) \quad (10)$$

$$M_2 = \langle PID_{WN_j}, K_{SN_j}, m_3, T_{SN_j} \rangle \quad (11)$$

Finally, WN_j sends M_2 to U_i over the open channel.

Step 3: After receiving M_2, U_i first checks the validity of the timestamp $(T' - T_{WN_j}) < \Delta T$, where T' is the time, the message was received. If the timestamp is invalid, the protocol is stopped, and a security vulnerability is indicated. Otherwise, U_i computes $ID_{WN_j} = D_2 \oplus PID_{WN_j}$ and the session keys Key' and m_3' shared with WN_j. To confirm the legitimacy of WN_j, U_i checks $m_3' = m_3$ and if it is not valid, the protocol is terminated, otherwise U_i establishes the session key Key' with WN_j to ensure secure communication.

BF based user authentication ensures the security of communication by providing real-time authentication, which indicates that there is no authentication delay in transmitting the information. Therefore, it is difficult for an attacker to implement a large number of flooding attacks in the system.

3.2.4. *User credentials update.* In some cases, registered users of U_i may need to update their credentials, such as passwords or biometrics, to ensure that they cannot be brute-force attacked by an attacker. The steps to allow U_i to update its credentials are as follows.

(1) U_i inserts the smart card SC_i into the reader, then enters the identity information ID_{U_i} and the current password PW_{U_i} and records the current biometric information B_i on the sensor of the specific terminal. The secret parameters σ_i, IPB_i, R_{U_i} are calculated using the regeneration function $Rep(.)$ in SC_i and B'_{U_i} is calculated using the hash function $h(.)$. In order to ensure the accuracy of the identity credentials, SC_i verifies the legitimacy of $B'_{U_i} = B_{U_i}$, and if it is not legitimate, the phase terminates immediately. Otherwise, SC_i calculates PR_{U_i}.

(2) SC_i asks U_i to provide a new password and a new biometric. Then U_i enters the new password, presents PW_{U_i}, and identifies the new biometric information and presents B_i^n at the specific terminal sensor. Subsequently, SC_i computes $(\sigma_i^n, \tau_i^n) = Gen(B_i^n)$, $IPB_i^n = h(ID_{U_i} \parallel PW_{U_i}^n \parallel \sigma_i^n)$, $C_{U_i}^n$, $A_{U_i}^n$, $B_{U_i}^n$ and $W_{U_i}^n$ using a fuzzy extractor.

$$C_{U_i}^n = PR_{U_i} \oplus h(\sigma_i^n \parallel PW_{U_i} \parallel ID_{U_i} \parallel R_{U_i}) \tag{12}$$

$$A_{U_i}^n = R_{U_i} \oplus h(\sigma_i^n \parallel PW_{U_i}) \tag{13}$$

$$B_{U_i}^n = h(ID_{U_i} \parallel IPB_i^n \parallel R_{U_i}) \tag{14}$$

$$W_{U_i}^n = h(ID_{U_i} \parallel IPB_i^n) \oplus h(HK_{U_i}) \tag{15}$$

(3) Finally, SC_i utilises $\{A_{U_i}^n, B_{U_i}^n, C_{U_i}^n, W_{U_i}^n, \tau_i^n\}$ in its storage to update the information $\{A_{U_i}, B_{U_i}, C_{U_i}, W_{U_i}, \tau_i\}$ so that SC_i contains the credentials $\{h(.), Gen(.), Rep(.), T, P, A_{U_i}, B_{U_i}, C_{U_i}, W_{U_i},$ of the user U_i.

3.2.5. *Dynamic node joining.* Due to the vulnerability of sensor nodes in WSNs to the surrounding environment and capture by attackers, even some sensors stop working due to power depletion. Therefore, new sensor nodes need to be added to ensure the proper functioning of the WSN.

(1) In order to deploy a new sensor node, the GW first assigns a Bloom vector V_{WN_j} to WN_j within a cluster using the Bloom filter and performs the third step of 2.2.1.

(2) GW generates a public-private key pair $(R_{WN_j}^n, K_{WN_j}^n = (R_{WN_j}^n \times P))$ for WN_j and stores the vector $V_{WN_j}^n$, the set $H_{WN_j}^n$, and the public-private key pair $(R_{WN_j}^n, K_{WN_j}^n)$ in the storage unit.

4. Protocol correctness and security analysis.

4.1. **Correctness analysis.** In this paper, we focus on the correctness analysis of whether U_i and WN_j obtain the same session key. WN_j establishes the session key Key shared with U_i in the authentication and key negotiation phase as shown in Equation (9). Similarly, U_i still establishes the session key Key' shared with WN_j.

$$Key' = h(T_{U_i} \parallel T_{WN_j} \parallel (R_{U_i} \times K_{WN_j}) \parallel h(ID_{U_i} \parallel HK_{U_i}) \parallel h(PR_{U_i} \parallel \sigma_i \parallel PW_{U_i} \parallel T_{U_i})) \tag{16}$$

In order to prove $Key = Key'$, it is only necessary to prove $h(ID_{U_i} \parallel HK_{U_i})' = h(ID_{U_i} \parallel HK_{U_i})$ and $R_{U_i} \times K_{WN_j} = R_{WN_j} \times K_{U_i}$.

WN_j calculates $\forall \eta \in [1, \beta], \forall \theta \in [1, k]$ and if $v_{WN_\theta}^\eta = 1$ then $h_{WN_\theta}^\lambda(K_{U_i}) = x, x \in [1, \beta]$.

Subsequently, WN_j computes $h(ID_{U_i} \parallel HK_{U_i})' = h(ID_{U_i} \parallel \oplus_{\theta=1}^k [\sum_{\lambda=1}^\gamma h_{WN_\theta}^\lambda(K_{U_i})])$, where $WN_\theta \in C_I, \forall \theta \in [1, k]$, which indicates $h(ID_{U_i} \parallel HK_{U_i})' = h(ID_{U_i} \parallel HK_{U_i})$. In addition, it can be obtained:

$$R_{U_i} \times K_{WN_j} = R_{U_i} \times (R_{WN_j} \times P) = R_{WN_j} \times (R_{U_i} \times P) = R_{WN_j} \times K_{U_i} \tag{17}$$

As a result, $Key = Key'$, U_i and WN_j both establish the same session key.

4.2. **Security Analyses.** (1) The protocol is anonymous and untraceable. Suppose adversary A intercepts $\langle D, PID_{U_i}, IPB_{U_i} \rangle$, M_1, and M_2. $\langle D, PID_{U_i}, IPB_{U_i} \rangle$ and PID_{U_i} in M_1 is the pseudo-identity of U_i, and parameter D mainly generates random number a. The adversary needs to compute $D = aQ$ and $C = bP$ in order to get the real identity ID_{U_i} corresponding to PID_{U_i}. The PID_{WN_j} in M_2 is also a pseudo-identity. A needs to obtain the random number D in order to calculate ID_{WN_j}, therefore, the explicit identities of U_i and WN_j are not involved in $\langle D, PID_{U_i}, IPB_{U_i} \rangle$, M_1 and M_2, and the other parameters used in the calculation process are all related to the selected random

numbers, so the messages in the whole communication process achieve anonymity and untraceability.

(2) The protocol is resistant to impersonation attacks. The value of HK_{U_i} can only be generated by the user U_i during the authentication phase. Therefore, validation of $h(ID_{U_i} \parallel HK_{U_i})' = h(ID_{U_i} \parallel \bigoplus_{\theta=1}^{k} \left[\sum_{\lambda=1}^{\gamma} h_{WN_\theta}^{\lambda}(K_{U_i})\right])$ ensures the integrity and authenticity of M_1 and avoids forgery attacks and impersonation attacks by the user.

(3) The protocol is resistant to replay attacks. Suppose A eavesdrops on the message M_1 and forwards this message to WN_j. However, since WN_j verifies the timestamp T_{U_i}, the message will be recognised as a replay message. Similarly, if A eavesdrops on the message M_2 and resends it to U_i, then since U_i validates the timestamp T_{WN_j}, it will also be recognised as a replay message. Therefore, the proposed protocol can resist replay attacks because of the validity of the timestamp.

(4) The protocol is resistant to man-in-the-middle attacks. Suppose adversary A eavesdrops on M_1 and finds out $ID_{U_i}, ID_{WN_j}, K_{U_i}, m_1, m_2, T_{U_i}$ and generates the current timestamp T_A. However, A is unable to access R_{U_i}, PR_{U_i} and $h(ID_{U_i} \parallel HK_{U_i})$ in U_i. Without knowing PW_{U_i} and σ_i, A cannot modify the messages M_1, m_1, and m_2. Moreover, it is computationally infeasible for A to modify the messages M_2 or m_3 because A does not have the keys R_{WN_j}, PR_{U_i} and $h(ID_{U_i} \parallel HK_{U_i})'$. Therefore, the proposed protocol can defend against man-in-the-middle attacks.

(5) The protocol is resistant to internal privilege attacks. Although GW is a fully trusted node in the WSN, GW may act as an insider attacker A. Suppose that during the user registration phase, A obtains the registration information $\{ID_{U_i}, IPB_{U_i}\}$ sent to GW by the legitimate user U_i. Furthermore, suppose A obtains the smart card SC_{U_i} after the registration process is completed, but according to (4), A does not have B_i and PW_{U_i}, so cannot obtain $PW_{U_i}, \sigma_i, R_{U_i}, PR_{U_i}$, and $h(ID_{U_i} \parallel HK_{U_i})$, so the protocol is resistant to an insider privilege attack.

(6) The protocol is resistant to online password guessing attacks. Suppose A intercepts the messages M_1 and M_2, but the legitimate user's password PW_{U_i}, biometric key σ_i, private keys R_{U_i} and PR_{U_i} are secret, so A is not able to obtain any secret parameters of U_i.

(7) This protocol protects against Event Stream Language (ESL) attacks. A session key share key is established by WN_j with the legitimate user U_i. On the other hand, U_i also establishes a session key key' shared with WN_j. The session key consists of a temporary key and a long-term key. Without long-term keys like PW_{U_i}, R_{U_i}, R_{WN_j}, σ_i, and PR_{U_i}, adversary A cannot compute $Key = key'$. Moreover, in order to derive $R_{U_i} \times K_{WN_j} = R_{WN_j} \times K_{U_i}$, A needs to solve the ECDDHP hard problem. Therefore, unless both the temporary and long-term keys are broken by the adversary, $Key = key'$ will not be broken. Therefore, the proposed protocol is resistant to ESL attacks.

4.3. **Simulation experiment analysis.** In order to evaluate the performance of the proposed authentication protocol, this paper establishes a user node and gateway node on a PC computer configured with Intel (R) Core (TM) 2 Quad CPU Q8300, @2.50 Hz processor, Windows 7 OS, and 2GB RAM. Since the configuration of the user device and gateway node is higher than that of the sensor node, the MicaZ is used as the sensor node which is configured with 128K bytes of ROM, 8-bit Atmega128L Atmel processor, 2 AA batteries, 512K bytes of EEPROM, nesC programming language, and TinyOS as the operating system.

In order to verify the efficiency of the proposed protocol, this paper compares the proposed protocol with the protocols in references [7], [10], [12], [19], and [20]. For the convenience of description, the protocol in [7] is named TFUA, the protocol in [10] is

named FELU, the protocol in [12] is named IBKA, the protocol in [19] is named MBSU, and the protocol in [20] is named ALSA.

4.4. **Analysis of computational overhead.** According to the conclusion in the literature [5], the time consumption of performing the hash function of the Secure Hash Algorithm (SHA-1) in the computer system is 0.5 ms for T_H. The time consumption of the ECC dot product operation is 40.8 ms for T_M. The time consumption of the Fuzzy Extractor operation for biometric authentication is 0.5ms for T_{FE}. The time consumption of encrypting or decrypting with AES-128 symmetric key is 8.7 ms for the MicaZ sensor node. 128 symmetric key encryption at T_{ENC} or decryption at T_{S_DEC} is 8.7 ms. The time consumption for AES-128 symmetric key encryption at T_{S_ENC} or decryption at T_{S_DEC} at the MicaZ sensor node is 5.05ms, and the energy consumption is 121.7 μJ. The time consumption for performing the SHA-1 hash function at T_{S_H} is 3.63ms, and the energy consumption is 87.12 μJ. The time consumption for the dot product operation at T_{S_M} for ECC is 72 ms, and the energy consumption is 2880 μJ. A comparison of the computational overheads of different authentication protocols is shown in Table 2.

Table 2. Comparison of computational overhead of different authentication protocols

Protocols	**Computational overhead**		
	U_i	WN_j	GW
TFUA	$T_{FE} + 4T_M + 2T_{ENC} + 8T_H$	$2T_{S\_DEC} + 3T_{S_M} + 6T_{S_H}$	$3T_M + 11T_H$
FELU	$T_{FE} + 5T_M + 2T_{ENC} + 12T_H$	$T_{S\_DEC} + 2T_{S_M} + 2T_{S_H}$	$2T_{ENC} + 2T_M$
IBKA	$7T_H + 2T_M + T_{FE}$	$2T_{S_H} + 2T_{S_M}$	$T_{ENC} + 12T_H$
MBSU	$2T_H + 3T_M + 2T_{ENC} + T_{FE}$	$T_{S\_DEC} + 2T_{S_M} + 4T_{S_H}$	$2T_M + 16T_H$
ALSA	$4T_H + 2T_M + T_{ENC} + T_{FE}$	$3T_{S_M} + 8T_{S_H}$	$T_M + 16T_H$
OURS	$8T_H + T_M + 2T_{FE}$	$4T_{S_H} + T_{S_M}$	$7T_H$

As can be seen from Table 1, the computational overhead of the proposed protocol OURS is 45.8 ms, 86.52 ms and 3.5 ms for user, sensor node and gateway node, respectively. Compared to the comparison protocols, the computational overhead of the proposed protocol is less. This is due to the fact that the OURS protocol rejects false messages in advance using multiple Bloom filters in the pre-preparation phase. Users use a fuzzy extractor to generate biometric features as identifiers of user identity in the registration and authentication phases, which greatly reduces the computational complexity. The TFUA, FELU, MBSU, and ALSA protocols use a large number of ECC calculations and complex cryptographic algorithms in the registration and authentication phases, which results in a much larger computational overhead for the authentication protocols than that of the OURS protocols. The computational time consumption of the IBKA protocol is only next to that of the OURS, which achieves the authentication of the user through two encryption and decryption operations in the authentication phase, resulting in a higher computational overhead than OURS.

4.5. **Communication overhead and storage cost analysis.** In order to evaluate the communication overhead, it is necessary to calculate the number of message bits required by different protocols in the authentication and key negotiation phases. Assume that the length of the sensor identity message is 32 bits, the length of the user's or GW's identity message is 160 bits, and the length of the random number is 160 bits. By applying SHA-1, the hash output is 160 bits, the ECC operation is 320 bits and the timestamp is 32 bits. In OURS protocol, the number of bits required for the messages

$M_1 = \langle ID_{U_i}, ID_{WN_j}, K_{U_i}, m_1, m_2, T_{U_i} \rangle$ and $M_2 = \langle ID_{U_i}, ID_{WN_j}, K_{WN_j}, m_3, T_{WN_j} \rangle$ are $(160+32+320+320+160+160+32) = 1184$ bits and $(160+32+320+160+32) = 704$ bits respectively, and hence the total communication overhead is 1888 bits.

Comparison of the communication overhead of OURS protocol with the other protocols is shown in Figure 4, where OURS protocol has the lowest communication overhead, which is reduced by 54.8%, 34.7%, 6.9%, 26.5%, and 18.3% compared to TFUA, FELU, IBKA, MBSU, and ALSA protocols, respectively. This reduction is due to the fact that the GWs and sensor nodes in the OURS protocol authenticate the users through multiple Bloom filters, which reduces the communication overhead.

Figure 5 shows the storage cost comparison between the proposed OURS protocol and other existing protocols. The storage cost of OURS protocol is 352 bits for sensors and 320 bits for users, which is lower. TFUA, FEL, and MBSU protocols require more cryptographic primitives to be stored in registration and authentication phases and hence the storage cost is higher. IBKA and ALSA protocols reduce the storage cost in authentication phase through hash function and biometrics for authentication, which reduces the storage cost. OURS protocol reduces the storage cost by replacing the traditional cryptographic algorithms through Bloom filter and fuzzy extractor.

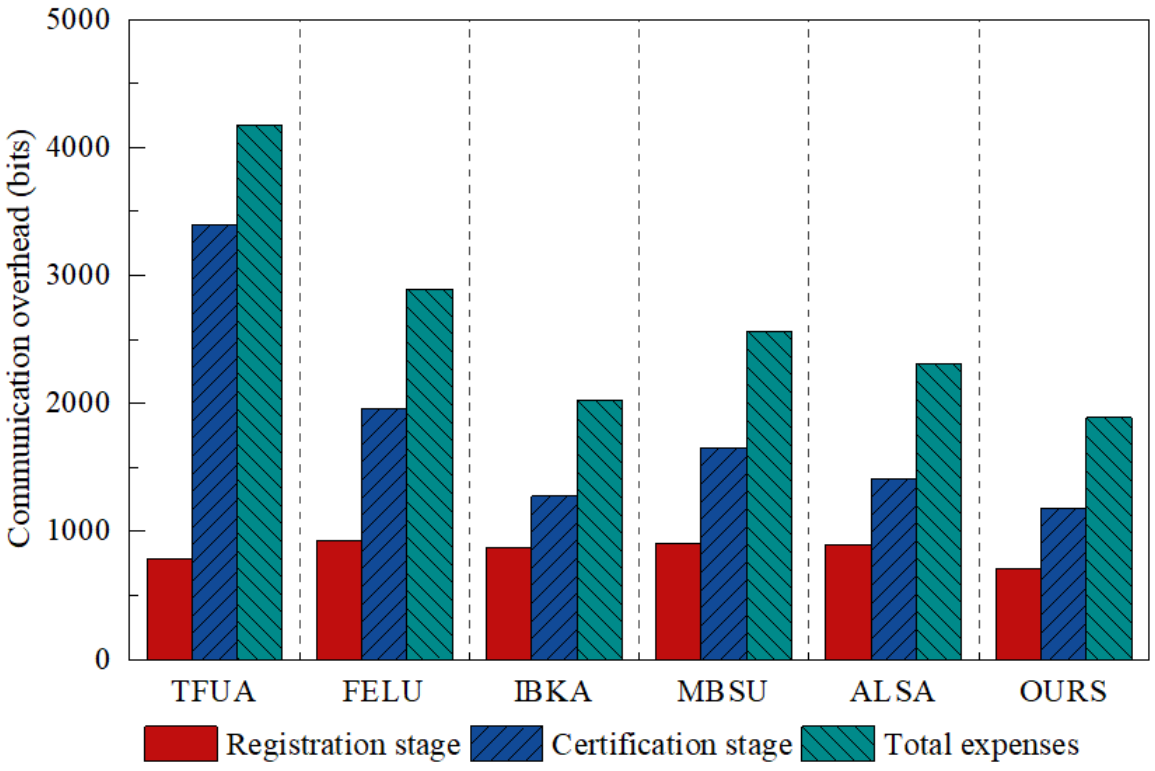

FIGURE 4. Comparison of communication overheads of different WSN authentication protocols

4.6. **Average delay.** From the literature [13], the average delay of messages in a sensor network is shown in Equation (18):

$$delay = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{n=1}^{N}\left(T_{\text{cream}}^{m-n} + T_{\text{transmission}}^{m-n-l} + T_{\text{authentication}}^{n-m-l}\right) \tag{18}$$

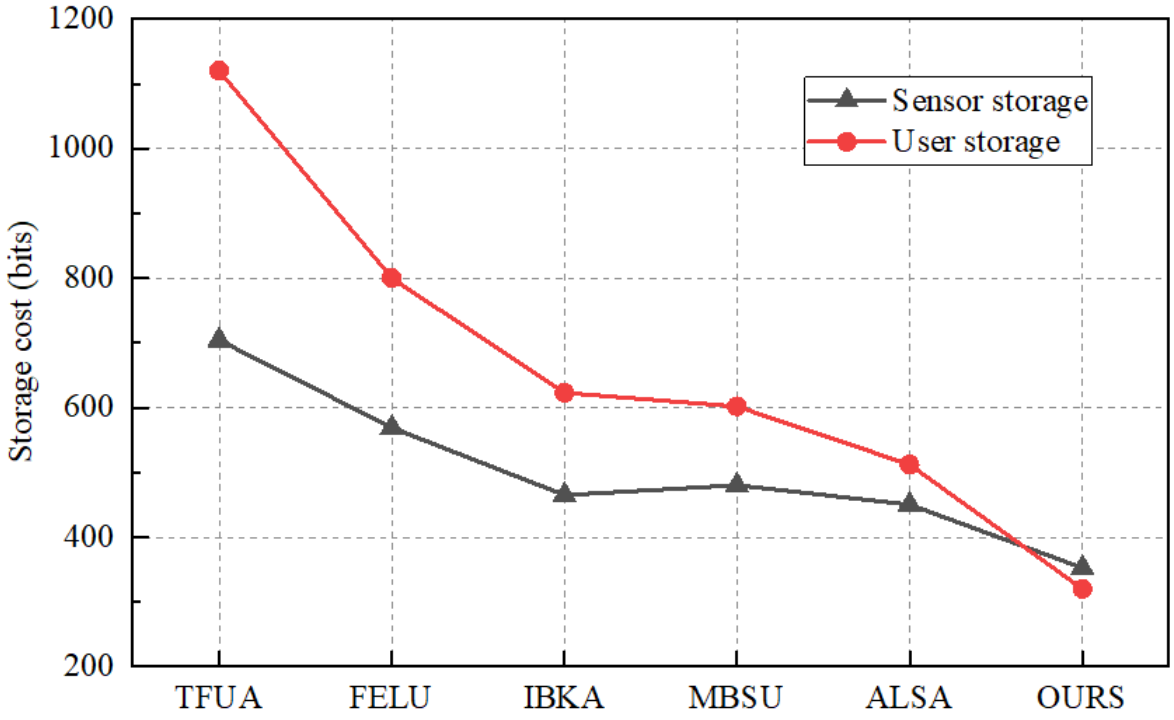

FIGURE 5. Comparison of Storage Costs for Different WSN Authentication Protocols

where M is the number of communicating entities in the WSN, n is the number of sent messages, T_{cream}^{m-n} is the time consumed by the user or the sensor node for generating the message, $T_{\text{transmission}}^{m-n-l}$ is the time consumed by the entity m for transmitting the message to the entity l, and $T_{\text{authentication}}^{n-m-l}$ is the time consumed by the entity l for authenticating the message transmitted by the entity m.

Figure 6 show the effect of the number of sensors on the average delay, respectively, and the communication in this paper's protocol is mainly between users and sensor nodes in the authentication phase. It can be seen that the number of sensor nodes is proportional to the average latency of communication, and when the number of sensors is 40, the average latency of the OURS protocol is 77.6%, 68.7%, 59.9%, 49.8% and 39.6% lower than that of the TFUA, FELU, MBSU, ALSA, and IBKA protocols, respectively. In OURS protocol, a complete authentication and session key negotiation process for an illegal user takes time consuming 23.95ms, of which authentication consumes 22ms and session key negotiation portion takes 1.95ms, so the time for session key negotiation is less than the time for authentication. In addition, the OURS protocol only needs to perform simple different-or operation and power operation during the authentication process, which greatly reduces the communication time consumption.

5. **Conclusion.** Identity authentication is one of the important means to ensure users' secure communication. However, in the WSN environment, there are many security threats, which makes the design of authentication protocol complicated. Aiming at the problem that node identity information in WSN is easy to be faked and has a large amount of calculation, this paper proposes a lightweight anonymous authentication protocol for sensor networks based on fuzzy extractor.

The main contributions of this agreement are as follows:

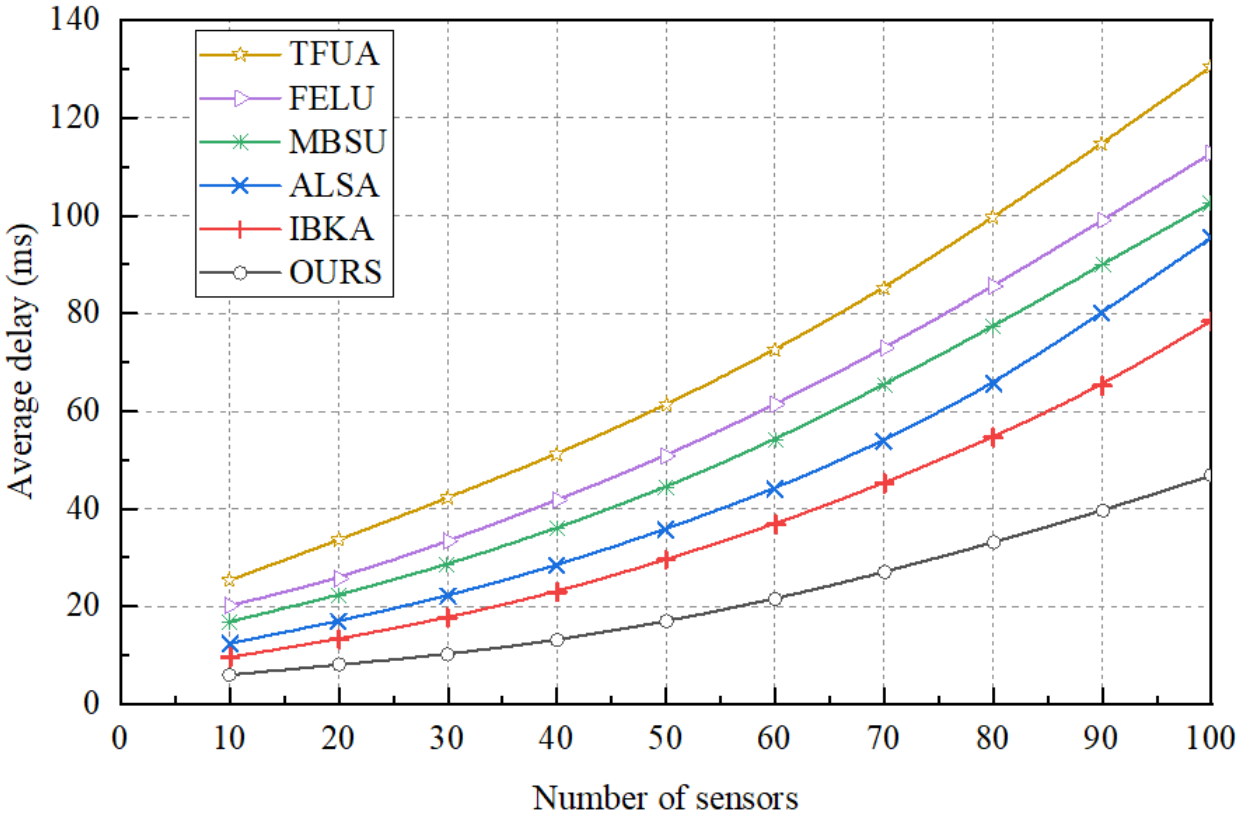

FIGURE 6. Comparison of average latency with different number of sensors

(1) **Initial stage:** Bloom filter is assigned to each sensor node, which effectively prevents the spread of false information and ensures the authenticity of the message.

(2) **Registration stage:** Users use fuzzy extractor to generate secret parameters related to their biological characteristics, and register identity information with the gateway in combination with pseudonyms, thus realizing the anonymity of identity.

(3) **Authentication stage:** The sensor node uses the fuzzy extractor to verify the user's identity and establishes a session key with the authorized user through biometric features, thus ensuring the security of communication.

Security and performance analysis show that the proposed protocol has the following advantages:

(1) **Low computation overhead:** The computation overhead on user, sensor node, and gateway node is 45.8ms, 86.52ms, and 3.5ms respectively, which is significantly lower compared to existing protocols.

(2) **Low storage cost:** The storage cost for sensor nodes and users is 352 bits and 320 bits respectively, which is lower than other protocols.

(3) **Short average delay:** In the case of 40 sensors, the average delay is 77.6%, 68.7%, 59.9%, 49.8%, and 39.6% lower than the comparison protocols, and the authentication process only requires simple different-or-operation and power operation, which effectively reduces communication time consumption.

To sum up, the proposed anonymous authentication protocol for lightweight sensor networks based on fuzzy extractor can not only provide high anonymity, but also have low communication overhead, storage cost, and average delay, which is very suitable for WSN environment and provides effective security protection for users.

REFERENCES

[1] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the Internet of Things: perspectives and challenges," *Wireless Networks*, vol. 20, no. 8, pp. 2481-2501, 2014.

[2] N. Park, H. Hu, and Q. Jin, "Security and Privacy Mechanisms for Sensor Middleware and Application in Internet of Things (IoT)," *International Journal of Distributed Sensor Networks*, vol. 12, no. 1, 2965438, 2016.

[3] International Journal of Sports Medicine, vol. 39, no. 05, 2018/04/30, 2018.

[4] J. Moon, D. Lee, Y. Lee, and D. Won, "Improving Biometric-Based Authentication Schemes with Smart Card Revocation/Reissue for Wireless Sensor Networks," *Sensors*, vol. 17, no. 5, 940, 2017.

[5] J. Nam, M. Kim, J. Paik, Y. Lee, and D. Won, "A Provably-Secure ECC-Based Authentication Scheme for Wireless Sensor Networks," *Sensors*, vol. 14, no. 11, pp. 21023-21044, 2014.

[6] K. Xue, C. Ma, P. Hong, and R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 316-323, 2013.

[7] Q. Jiang, J. Ma, X. Lu, and Y. Tian, "An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1070-1081, 2014.

[8] S. Banerjee, C. Chunka, S. Sen, and R. S. Goswami, "An Enhanced and Secure Biometric Based User Authentication Scheme in Wireless Sensor Networks Using Smart Cards," *Wireless Personal Communications*, vol. 107, no. 1, pp. 243-270, 2019.

[9] M. A. U. Rehman, R. Ullah, B.-S. Kim, B. Nour, and S. Mastorakis, "CCIC-WSN: An Architecture for Single-Channel Cluster-Based Information-Centric Wireless Sensor Networks," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7661-7675, 2021.

[10] I. Temirlan, and Y. Li, "ECC-based User Authentication Scheme for Wireless Sensor Networks," *International Journal of Engineering Research & Science*, vol. 3, no. 6, pp. 21-28, 2017.

[11] H. Lee, D. Kang, J. Ryu, D. Won, H. Kim, and Y. Lee, "A three-factor anonymous user authentication scheme for Internet of Things environments," *Journal of Information Security and Applications*, vol. 52, 102494, 2020.

[12] B. Khalid, K. N. Qureshi, K. Z. Ghafoor, and G. Jeon, "An improved biometric based user authentication and key agreement scheme for intelligent sensor based wireless communication," *Microprocessors and Microsystems*, vol. 96, 104722, 2023.

[13] X. Shao, Y. Guo, and Y. Guo, "A PUF-based anonymous authentication protocol for wireless medical sensor networks," *Wireless Networks*, vol. 28, no. 8, pp. 3753-3770, 2022.

[14] S. Sakthivel, and G. Vidhya, "A Trust-Based Access Control Mechanism for Intra-Sensor Network Communication in Internet of Things," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 3147-3153, 2020.

[15] S. D. Galbraith, and P. Gaudry, "Recent progress on the elliptic curve discrete logarithm problem," *Designs, Codes and Cryptography*, vol. 78, no. 1, pp. 51-72, 2015.

[16] W. Castryck, M. Houben, F. Vercauteren, and B. Wesolowski, "On the decisional Diffie–Hellman problem for class group actions on oriented elliptic curves," *Research in Number Theory*, vol. 8, no. 4, 329, 2022.

[17] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz, "Theory and Practice of Bloom Filters for Distributed Systems," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 131-155, 2012.

[18] B. Fuller, X. Meng, and L. Reyzin, "Computational fuzzy extractors," *Information and Computation*, vol. 275, 104602, 2020.

[19] A. Jabbari, and J. B. Mohasefi, "User-sensor mutual authenticated key establishment scheme for critical applications in wireless sensor networks," *Wireless Networks*, vol. 27, no. 1, pp. 227-248, 2020.

[20] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. A. Doostari, "A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT," *Computer Networks*, vol. 177, 107333, 2020.