





Article

Context Privacy Preservation for User Validation by Wireless Sensors in the Industrial Metaverse Access System

John Owoicho Odeh ^{1,*}, Xiaolong Yang ^{1,*}, Cosmas Ifeanyi Nwakanma ² and Sahraoui Dhelim ^{3,*}

¹ School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

² ICT-Convergence Research Center, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea; cosmas.ifeanyi@kumoh.ac.kr

³ School of Computer Science, University College Dublin, Belfield, D04 V1W8 Dublin, Ireland

* Correspondence: yangxl@ustb.edu.cn (X.Y.); sahraoui.dhelim@ucd.ie (S.D.)

Abstract: The Industrial Metaverse provides unparalleled prospects for increasing productivity and efficiency across multiple sectors. As wireless sensor networks play an important role in data collection and transmission within this ecosystem, preserving context privacy becomes critical to protecting sensitive information. This paper investigates the issue of context privacy preservation for user validation via AccesSensor in the Industrial Metaverse and presents a technological method to address it. We explore the need for context privacy, look at existing privacy preservation solutions, and propose novel user validation methods that are customized to the Industrial Metaverse's access system. This method is evaluated on time-based efficiency, privacy method and bandwidth utilization. Our method performs better as compared to the DPSensor. Our research seeks to provide insights and recommendations for developing strong privacy protection methods in wireless sensor networks that operate within the Industrial Metaverse ecosystem.

Keywords: context privacy preservation; user validation; sensors; industrial metaverse access system



Citation: Odeh, J.O.; Yang, X.; Nwakanma, C.I.; Dhelim, S. Context Privacy Preservation for User Validation by Wireless Sensors in the Industrial Metaverse Access System. *Algorithms* **2024**, *17*, 225. <https://doi.org/10.3390/a17060225>

Academic Editors: Manolya Kavakli-Thorne and Zhuangzhuang Dai

Received: 2 May 2024
Revised: 14 May 2024
Accepted: 21 May 2024
Published: 23 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Industrial Metaverse depicts the intersection of the physical and digital worlds, resulting in a highly interconnected ecosystem in which virtual and augmented reality interfaces, artificial intelligence, and the Internet of Things (IoT) work together to transform industrial operations [1]. It connects the real world to the virtual world, giving users the feel of the virtual world of augmented (AR) or virtual reality (VR), as shown in Figure 1. In this dynamic environment, wireless sensors serve as the backbone for obtaining real-time data required for operation optimization and safety assurance. However, the widespread deployment of sensors creates serious privacy concerns, particularly about how data are collected, processed, and shared [2].

Context privacy preservation refers to the protection of sensitive contextual information related to user validation activities. In the Industrial Metaverse, where people interact with a variety of physical and digital elements, context privacy is critical to preventing unwanted access, identity theft, and exploitation of personal information [3]. Furthermore, compliance with data protection standards such as GDPR and CCPA requires the deployment of strong privacy safeguards connected with the events, surroundings, or actions related to data gathering [4]. The addition of wireless sensors for user validation within the Industrial Metaverse access system poses both obstacles and opportunities. However, the opportunities outweigh the obstacles. The challenges include guaranteeing safe communication, reducing the danger of data breaches, and addressing user consent and transparency concerns [5]. However, novel technologies like context-aware encryption, dynamic privacy settings, and biometric authentication provide chances to improve privacy while maintaining usability and efficiency [6]. The challenges of preserving context privacy

in wireless sensor networks within the industrial metaverse abound. Examples are data aggregation and fusion. Aggregating data from multiple sensors to derive meaningful insights while preserving the privacy of individual contexts poses a significant challenge. In addition, traditional aggregation techniques may compromise privacy by revealing sensitive information, as the need for real-time processing of sensor data introduces complexities in implementing privacy preservation mechanisms without introducing significant latency or overhead. Dynamic environments characterize industrial settings where contextual factors constantly change. Therefore, adapting privacy preservation techniques to accommodate these changes without sacrificing effectiveness is critical. Ensuring secure data and communication channels between sensors, edge devices, and central processing units is essential to prevent eavesdropping or data interception by malicious entities. Trusted sensors with IP security (IPsec) have to be deployed and utilized within the context privacy framework to guarantee proper user validation.

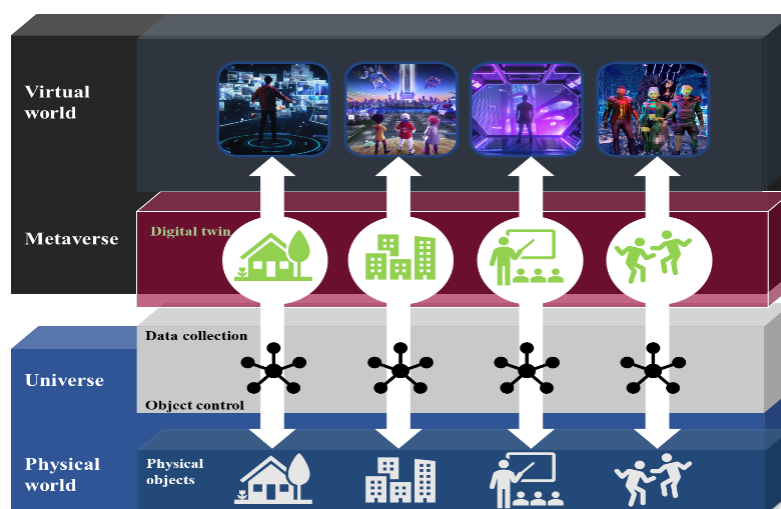


Figure 1. The Metaverse Layout [7].

2. Related Works

Context privacy preservation techniques for user validation using Wireless Sensor have gained popularity in recent years as IoT devices are being deployed in a variety of areas, including industrial and smart cities. Several studies have investigated various techniques to address the issues associated with maintaining context privacy while verifying users using sensors. Of note, B. Li et al. proposed a system that protects users' privacy by using a hierarchical authentication architecture to confirm users' identities [8]. It ensures the security and integrity of user data by utilizing a blockchain in a named data approach to safeguard communication between authentication servers and sensors. In another work, the system designed provides an ECC-based three-factor user authentication and key agreement system for WSNs in the Internet of Things applications that are context-aware and privacy-preserving [9]. Contextual data, including location and ambient conditions, are used by the technique to improve authentication while protecting user privacy. Homomorphic encryption and secure multiparty computing are two of the cryptographic primitives it uses to offer privacy-preserving authentication without sacrificing security. Using low-tech cryptography approaches, sensors, and authentication servers, they can communicate securely while maintaining user data integrity and confidentiality. In a further stretching of the cryptographic frontiers, a safe and privacy-preserving user authentication mechanism tailored for WSNs in IoT contexts was presented. The approach uses lightweight cryptographic techniques and contextual information to authenticate users while protecting their privacy [10]. It uses group-based authentication and dynamic key management approaches to reduce security risks and increase resilience to attacks like node compromise and eavesdropping. In addition, there is a focus on privacy-preserving au-

thentication in WSN-enabled healthcare applications. The suggested approach assures the confidentiality and integrity of sensitive health data exchanged over the network, as well as secure user authentication [11]. It uses a combination of cryptographic primitives, such as identity-based and attribute-based encryption, to achieve fine-grained access control and privacy preservation in healthcare contexts [12,13]. Despite all the privacy-preserving methods available for other applications, the challenge of user detail exposure during users' validation within the industrial Metaverse access system still exists. Thus, in this paper we aim to:

1. Explore the role of context privacy-preservation technology in ensuring secure and privacy-aware user validation processes within the Industrial Metaverse access system.
2. Propose a sensor-based masked-access fingerprint recognition system for user validation. This will ensure the user's privacy within the access system of the metaverse.
3. Evaluate the AccesSensor time-based efficiency and bandwidth utilization capabilities as compared to the DPSensor's noise addition privacy method.

The rest of the paper is organized as follows: Section 3 describes the proposed technology solutions: Section 4. Elaborates the masked privacy preservation technique for industrial IoT edge network systems. Experiments and analysis are displayed in Section 5 and conclude the paper. Through this research, we endeavor to provide insights and guidelines for the effective implementation of privacy-aware access control systems in the Industrial Metaverse. As far as we know, this is a unique research we have carried out.

3. Proposed Technology Solution

To address the above-mentioned challenge, we use edge computing for privacy preservation, which utilizes edge computing capabilities to perform context-aware processing and aggregation closer to the data source, minimizing the need to transmit sensitive information over the network [14,15]. A physical biometrics device, AccesSensor, with an embedded wireless sensor has the capability for fingerprint recognition system input, as seen in Figure 2.

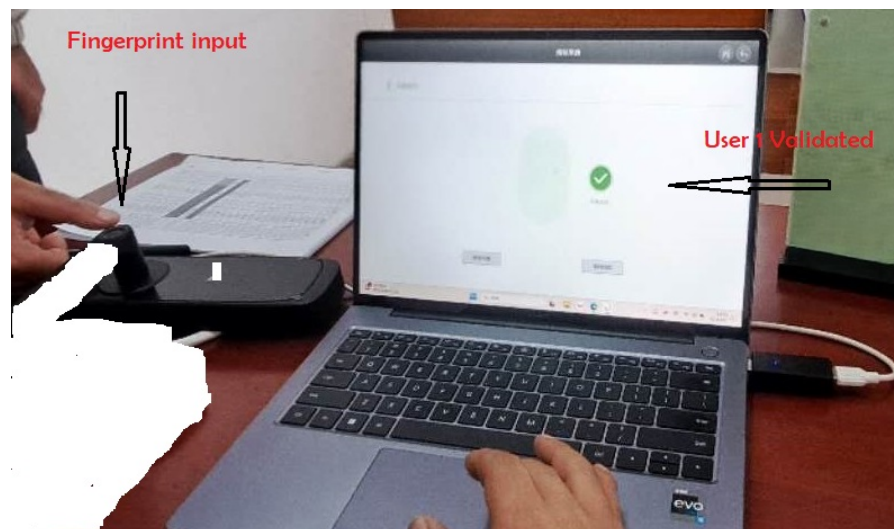


Figure 2. AccesSensor User Validation.

The AccesSensor acquires fingerprint images using biometric sensors [16] as displayed in Figure 3 Preprocess the fingerprint images to enhance quality and remove noise. This involves the filtering technique.

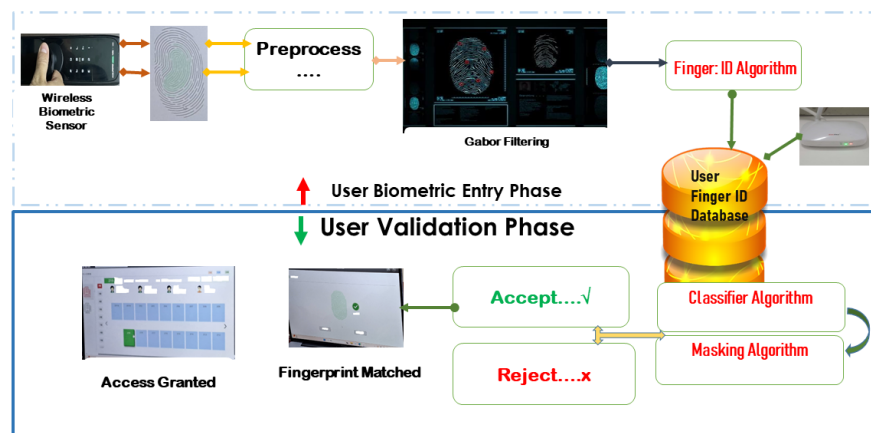


Figure 3. Metaverse User Validation System.

Extract discriminative features from the preprocessed fingerprint images. Common features include ridge orientation and patterns to gather data such as the Finger ID = $f.ID$ [17]. This is then stored in the system's database within the edge network server. Based on demand, the system deploys the Gabor filtering and feature point detection techniques to extract relevant features from the fingerprint images [18]. The system uses the minutiae-based technique, where the features of the minutiae points (ridge endings and bifurcations) are detected. Let $I(x, y)$ be the preprocessed fingerprint image, where (x, y) are pixel coordinates. The equation for feature extraction using the minutiae-based approach can be represented as:

$$F = 1, 2, \dots, n, F = (f_1, f_2, \dots, f_n) \quad (1)$$

where F is the set of extracted features, and f_i = the feature descriptor inputted for a region in the fingerprint image. Extracting discriminative features from preprocessed fingerprint images involves various techniques tailored to each specific feature [19]. There are some methods commonly used to extract, which include ridge orientation, ridge frequency, minutiae points, and texture descriptors (Gabor Filters). In this work, the Gabor filters are used to extract texture features by convolving the fingerprint image with a set of Gabor filter kernels at different orientations and frequencies, capturing texture variations at multiple scales [20].

Matching and Classification

The access system utilizes a deep learning classifier/matching algorithm to compare the extracted features with reference templates stored in a database as displayed in Figure 4 within the edge network [21]. This enables the complete validation of user data to describe their accuracy, or otherwise, of the fingerprint.

Fingerprint matching and classification using minutiae points is a core method in automated fingerprint identification systems (AFIS). The process is explained:

1. **Fingerprint Acquisition:** The procedure begins with capturing an image of the fingerprint using the AccesSensors, which is then transformed into a digital format for processing.
2. **Minutiae Extraction:** The digital fingerprint image is processed to detect and record the orientation of the unique spots of interest on the fingerprint, such as ridge ends.
3. **Fingerprint Matching:** Align fingerprints before matching for accurate comparison. This entails lining up the orientation and position of the two fingerprints A and B as shown in Figure 4.
4. **Point Correspondence:** The minutiae point from the FingerID A is then compared to the finer details of FingerID B. The purpose is to identify corresponding points between the two sets of minutiae, as highlighted in Figure 4. The Deep Learning

technique is then used to calculate or score the similarity of the related minutiae points. The more corresponding minutiae points are discovered and the closer they match, the greater the similarity score. Otherwise, they are deemed mismatched, and access is refused [9,19].



Figure 4. Deep Learning Matching Process.

$$Match(F, F \text{ database}) \quad (2)$$

where $F \text{ database}$ represents the feature vectors of fingerprints in the database. Based on the findings of the matching procedure, it is determined whether the input fingerprint matches any of the fingerprints in the database. This judgment can be binary (match or non-match) [4].

$$Decision = Match(F, F \text{ database}) \quad (3)$$

An accurate result guarantees an access-granted outcome with a 'green' LED indicator light in Figure 5, while a non-accurate result grants a denied access outcome with a 'red' LED indicator light on the sensor [22].

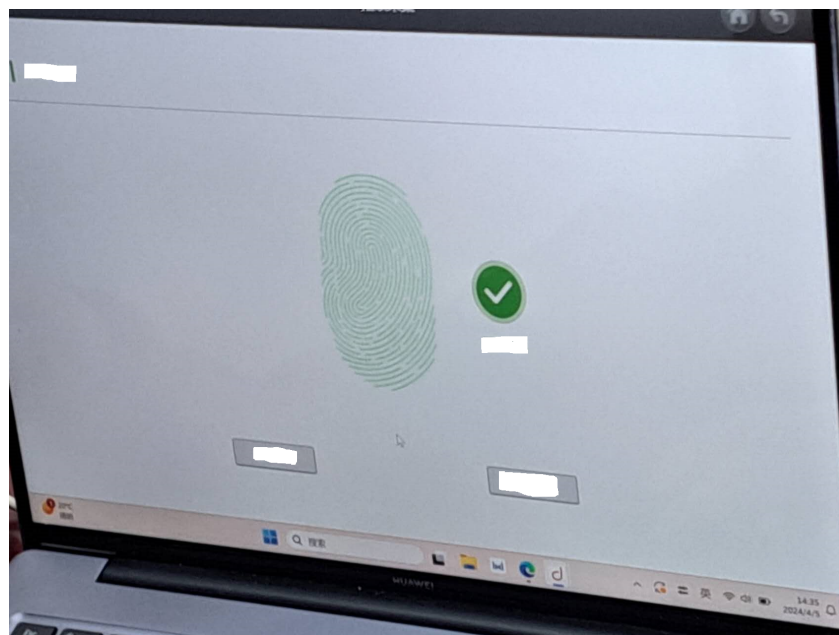


Figure 5. Metaverse User Validation.

Designing algorithms to analyze user fingerprints in real time for context-aware user validation involves several steps. Here, is a high-level overview of the algorithm design process [14].

In addition, the Algorithm 1 describes the process of developing and training a Convolutional Neural Network (CNN) model for fingerprint classification and matching using deep learning techniques. The process is highlighted and explained thus:

Algorithm 1 Deep learning classifiers algorithm

```

1: Imports:
2: from tensorflow.keras import models, layers
3: import numpy as np
4: from tensorflow.keras.utils import to_categorical

5: Define the CNN model creation function:
6: function CREATE_CNN_MODEL(input_fingerID)
7:   model ← models.Sequential([
8:     layers.Conv2D(32, (3, 3), activation='relu',
9:     input_shape=input_fingerID),
10:    layers.MaxPooling2D((2, 2)),
11:    layers.Conv2D(32, (3, 3), activation='relu'),
12:    layers.MaxPooling2D((2, 2)),
13:    layers.Conv2D(64, (3, 3), activation='relu'),
14:    layers.MaxPooling2D((2, 2)),
15:    layers.Flatten(),
16:    layers.Dense(128, activation='relu'),
17:    layers.Dense(num_classes, activation='softmax')
18:   ])
19: return model
20: end function

21: Load and preprocess data:
22: X_train ← np.load('fingerprint_features_train.npy')
23: y_train ← np.load('user_labels_train.npy')
24: input_fingerID ← X_train.shape[1 :]
25: num_classes ← len(np.unique(y_train))
26: X_train ← X_train.astype('float32')/255
27: y_train ← to_categorical(y_train,num_classes)

28: Create and compile the model:
29: model ← create_cnn_model(input_fingerID)
30: model.compile(optimizer='adam', loss='categorical_crossentropy',
31:   metrics=['accuracy'])

32: Train the model:
33: model.fit(X_train,y_train,epochs=10, batch_size=32, validation_split=0.2)

```

1. **Fingerprint Input:** The procedure begins with determining the input shape for the CNN model. This form usually correlates to the dimensions of the input images, such as their height, breadth, and number of channels (for example, grayscale or RGB).
2. **Model Creation:** The method creates a sequential model, a linear stack of layers, from the models. Sequential function.
3. **Convolutional Layers:** The method iterates over multiple convolutional layers. (layers.Conv2D) with increasing filter count (32, 64, and 128) and kernel size (3, 3). To extract features from an input image, each convolutional layer uses a series of filters. The ReLU activation function (*activation = 'relu'*) is used to add non-linearity. After each convolutional layer, add max-pooling layers (layers). MaxPooling2D is used

to downsample feature maps, lowering spatial dimensions and extracting the most significant minutiae point information.

4. Flatten Layer: After convolutional and max-pooling layers, add a flattened layer (layers.Flatten) to turn 2D feature maps into 1D vectors for fully connected neural networks.
5. Data Preparation: Before training the model, the algorithm loads the feature vector of the fingerprint A (X_{train} and y_{train}) from the database (containing fingerprint features and corresponding labels) using NumPy's np.load function.
6. Data Preprocessing: The labels (y_{train}) are converted into one-hot encoded vectors using `tf.keras.utils.to_categorical`, which is necessary for classification tasks.
7. Model Compilation: The model is compiled using the specified optimizer ('User1'), loss function ('categorical_crossentropy'), and evaluation metrics ('accuracy'). Categorical cross-entropy is commonly used for multi-class classification problems, and accuracy is a metric used to evaluate the performance of the model during training.
8. Model Training: The algorithm then trains the model using the training data (X_{train} and y_{train}) for a specified number of users (10), with a batch size of 32 bits. Additionally, it uses 20% of the training data for validation during training (`validation_split = 0.2`).
9. Output: Finally, the algorithm outputs "MatchedFingerID" after the training process completes, indicating that the model is ready for fingerprint classification: Matched or Unmatched.

4. Privacy Preservation

To ensure privacy preservation by securely storing and processing fingerprint data, we implement a masking encryption technique to protect sensitive biometric information during transmission and storage. Apply anonymization or tokenization methods to dissociate user identities from raw fingerprint data while maintaining the ability to perform validation. The biometrics cryptosystem masking algorithm uses a crypto key to obfuscate the details of the fingerprint during the transmission and storage in the database within the edge network server. An unmasking procedure can be performed on a queried Finger: ID from within the database to enable user validation.

$$F1(a, b, c) = A.K.allow(f1, f2, f3, \dots, fn) \quad (4)$$

The privacy protection technique adopts an algorithm, similar to the Paillier cryptosystem algorithm [11] for the generation of private and public twin tokens (N, g) and (λ, μ) that are used for the masking and unmasking of finger ID [23]. This process is shown in Figure 6. The edge server produces random integer z, q , where $z - 1$, and $q - 1$ are prime numbers (set of bits more than 1024). To allow parameters in Equation (4) to have protection parameters Pr, substituting Equation (9) into the protection equation, we have as follows: For a token generation, let p and $q =$ two large prime numbers. These primes are used to generate the modulus n for the cryptosystem [23,24].

Let lambda (λ): The Carmichael function, also known as the least common multiple of $p - 1$ and $q - 1$. It is used in various calculations within the Paillier cryptosystem [23], particularly for generating public and private tokens.

$$\lambda = LCM(z - 1, q - 1) \quad (5)$$

Let the plaintext message x be transformed using $L(x)$ before being masked, set

$$L(x) = \frac{(x - 1)}{N}, \text{ where, } N = zq \quad (6)$$

To obfuscate the matched fingerprint, select a token

$$g \in (Z * N^2) \quad (7)$$

Then, let $g = \text{LIG}$, the learning integer generator
 Let the public and private pair token

$$(N, g) \text{ and } (\lambda, \mu) \tag{8}$$

Choose a random integer r from the set of invertible elements modulo N , denoted as $Z * N$, such that $\text{gcd}(r, N) = 1$.

Let $Pr = \text{mask}$

$$Pr = \text{LIG}^n * r^N \text{ mod } N + F1(r^N \text{ mod } N) + F1(r^N \text{ mod } N) + F3(r^N \text{ mod } N). \tag{9}$$

where $\Delta = r^N \text{ mod } N$.

Let the sum of all random values and modulus = 1

$$Pr = \text{LIG}(F1, 2, 3) * \sum_1^i (r^N \text{ mod } N)(F1, 2, 3) F1, 2, 3 = \Delta a, b, c * \text{LIG}(F1, 2, 3 \dots n) \tag{10}$$

Then the ciphertext mask is calculated as follows:

$$Pr(a, b, c) = \Delta(a, b, c) \cdot \text{LIG}(F((a, b, c))) \tag{11}$$

Summarily, token generation involves selecting two large primes and computing public and private keys using a generator g . Masking consists of selecting a random value r and masking the message with the public token (N, g) .

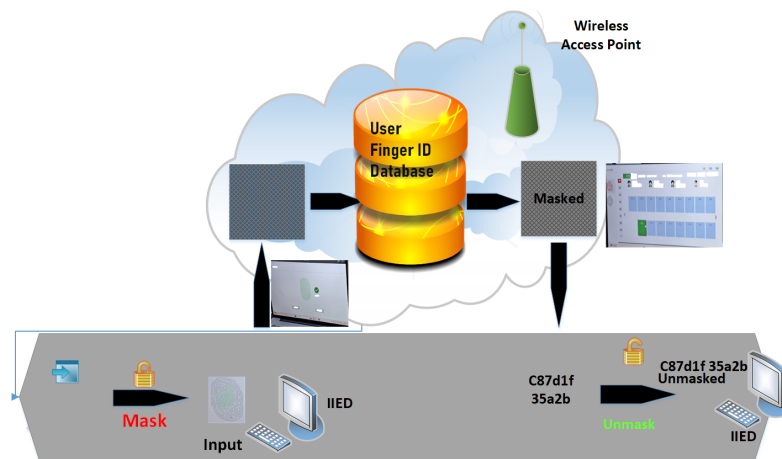


Figure 6. Masking Procedure.

Testing and Evaluation

The experiment setup is made up of a biometric sensor, AccesSensor, used as a fingerprint input device for users of the industrial Metaverse system. A display device, a laptop of specification processor: Intel Core i7-1165G7 Processor, memory: 16 GB DDR4 2666 Mhz Memory, storage: 1TB PCIe NVMe M.2 SSD solid state drive internal storage, with connectivity to a wireless network and the server/database for validation, classification, matching and storage of FingerID, based on the deep learning classification of minutiae points for feature vector. The process for user validation includes the User Biometric entry phase and the User validation phase as displayed in Figure 3. The outcome of the timely validation process of User fingerID is Unmatched (Reject) or matched (Accept), for which user access to the Metaverse platform is declined or granted, respectively. In conducting extensive testing and evaluation of the algorithm using benchmark fingerprints from the database and real-world scenarios, and displaying the masked user interface, as shown in Figure 7.

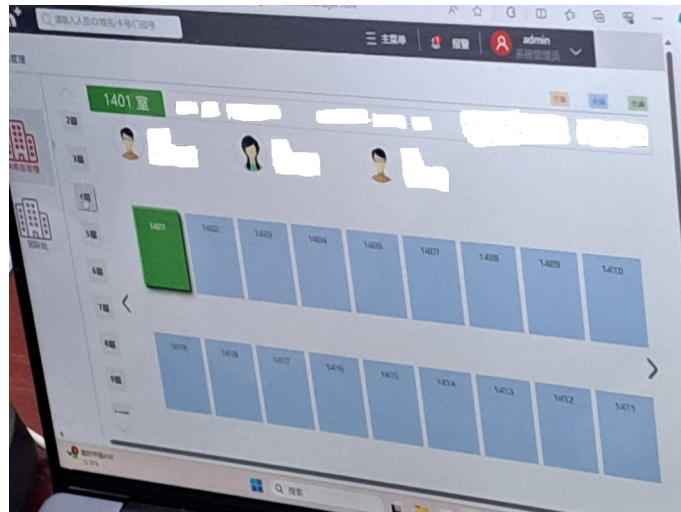


Figure 7. Masked User Interface.

We evaluate performance metrics such as efficiency, precision, recall, and processing time under varying conditions. Then validate the algorithm's robustness against spoofing attacks, noisy input data, and changes in contextual factors. The algorithm executes a good solution of biometrics cryptosystem of masking [25], as displayed in Algorithm 2.

Algorithm 2 Biometrics Cryptosystem Masking Algorithm

```

function MASK_KEY(key, mask)
2:   if len(token) ≠ len(mask) then
       raise ValueError("Key and mask must have the same length")
4:   end if
       masked_key ← empty byte array
6:   for k, m in zip(key, mask) do
       masked_key.append(k ⊕ m)
8:   end for
       return masked_key
10: end function
function UNMASK_TOKEN(masked_token, mask)
12: if len(masked_token) ≠ len(mask) then
       raise ValueError("Masked token and mask must have the same length")
14: end if
       key ← empty byte array
16: for mk, m in zip(masked_token, mask) do
       key.append(mk ⊕ m)
18: end for
       return token
20: end function
function MAIN
22: key ← b'VerySecrettoken123'
       mask ← os.random(len(token))
24: masked_token ← mask_token(token, mask)
       unmasked_token ← unmask_key(masked_key, mask)
26: print("Original token:", token)
       print("Masked token:", masked_token)
28: print("Unmasked token:", unmasked_token)
       end function
30: Masked ← FingerID

```

5. Experiment and Result

In this section, we describe the experiments and outcomes. This includes evaluation metrics and the compared methods. The results of the experiments are then presented, along with the analysis drawn from them.

Result and Analysis

The result from Table 1 clearly shows the various times to match a fingerprint from the sensor database. This reflects the time-based efficiency of the system. It was discovered that it took user 1 about 3 s to match, then validated by the AccesSensor. Conversely, it took User 1 approximately 4 s to be matched and validated by the DPSensor [26]. However, this time difference tends to be equal, when User 3 and User 4 fingerprints are validated at (10 s, 10.1 s) and (15 s, 15.2 s), respectively, as seen in Figure 8. This indicates that our method efficiently uses less time to execute its task.

Table 1. Comparison of Time-based Efficiency.

User	AccesSensor		DPSensor	
	Time to Match (s)	Time to Mask (s)	Time to Match (s)	Time to AddNoise (s)
User 1	3	4	4	6
User 2	7	9	7.2	11.5
User 3	10	15	10.1	16.7
User 4	15	19	15.2	19.7
User 5	18	22	18.4	23.6
User 6	21	26	22	22.6
User 7	27	33	27.8	27.9
User 8	32	38	32.9	39
User 9	37	41	38	43
User 10	38	41.6	38.5	43.7

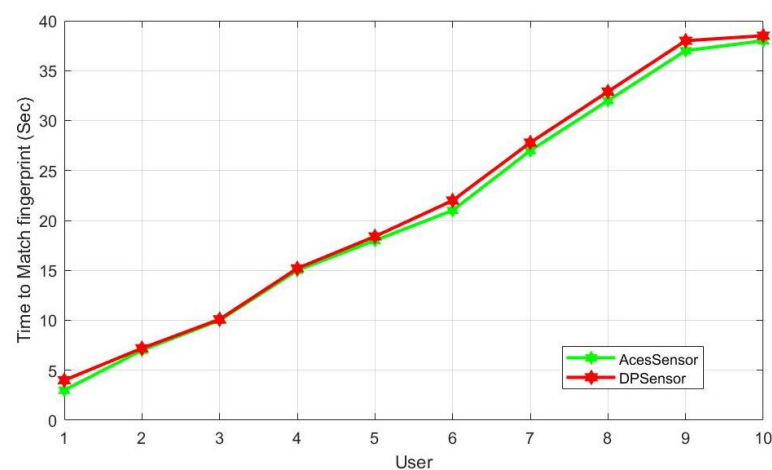


Figure 8. Time-based Efficiency of Sensors.

Furthermore, the outcome of the masking process as a privacy preservation measure by the AccesSensor, in contrast to the noise addition mechanism of the DPSensor is depicted in Figure 9. The validated finger ID of User 2 took our method, 9 s to be masked while the noise addition method of the DPSensor took 11.5 s to complete this privacy preservation process. Thus, this shows that our masking method is faster and more efficient in user validation than the compared method.

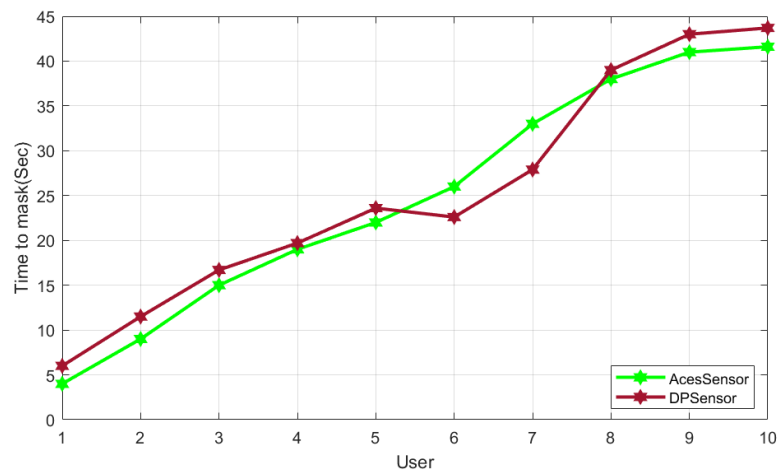


Figure 9. Efficiency of Privacy Method of Sensors.

In terms of bandwidth utilization, edge network devices as well as IoT devices are displayed in Table 2. The AccesSensor matched and granted access to users 1–4, utilizing lower bandwidths of 0.12, 0.5, 0.9, and 3.087 Mbps, respectively, to obfuscate the validated fingerprints. Conversely, the DPSensor utilized a higher bandwidth of 0.15, 0.8, 3.4, and 3.8, respectively, to match, validate and carry out noise addition as a privacy measure. For users 5 to 10, the result is different as it requires more for noise addition. This clearly shows the AccesSensor’s edge over the DPSensor as represented in Figure 10.

Table 2. Comparison of AcesSensor and DPSensor Bandwidth (BW).

User	AcesSensor—BW	DPSensor—BW
User 1	0.15	0.12
User 2	0.8	0.5
User 3	3.4	0.9
User 4	3.8	3.087
User 5	3.95	3.669
User 6	3.99	3.816
User 7	4	3.879
User 8	3.991	3.983
User 9	4.2	4.023
User 10	4.3	4.086

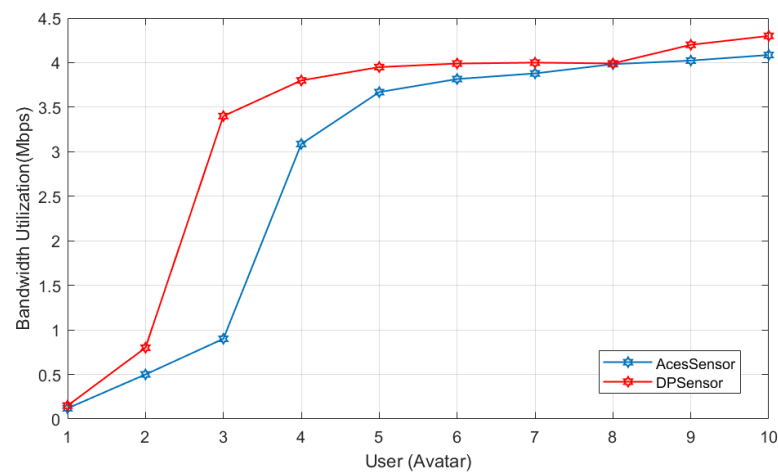


Figure 10. Bandwidth Utilization for User Validation.

6. Conclusions

Preserving context privacy in wireless sensor networks within the Industrial Metaverse is essential for maintaining trust, security, and compliance with regulatory requirements. By addressing the unique challenges posed by this dynamic ecosystem and adopting innovative technological solutions, organizations can harness the full potential of wireless sensor networks while safeguarding sensitive information. In this research, we proposed a novel user validation method that is implementable within the Industrial Metaverse's access system. Also, the evaluation of time-based efficiency, privacy method, and bandwidth utilization is provided. The AccesSensor is used to input and validate users utilizing the deep learning classifier method Decision = Match (F, Fdatabase). We further masked the validated fingerprint, thereby providing a good recommendation for strong privacy protection methods in sensor networks. This method is evaluated using its efficiency and bandwidth utilization metrics, it was discovered that it is time efficient and utilizes a lower bandwidth compared to the PDSensor. Further research and innovations are crucial to advancing context privacy preservation technologies and ensuring the sustainable development of the Industrial Metaverse ecosystem.

Author Contributions: Conceptualization, J.O.O., X.Y.; methodology, J.O.O.; software, J.O.O., X.Y.; validation, J.O.O., X.Y.; formal analysis, J.O.O.; investigation, J.O.O., X.Y., C.I.N., S.D.; data curation, J.O.O.; original draft preparation, J.O.O., X.Y., C.I.N.; writing—review and editing, J.O.O., X.Y., C.I.N., S.D.; visualization, J.O.O., X.Y., C.I.N.; supervision, X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (NSFC) under Grants 61971033.

Data Availability Statement: Data available on request due to privacy restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Sooyeon, S.; Taekyoung, K. A Privacy-Preserving Authentication, Authorization, and Key Agreement Scheme for Wireless Sensor Networks in 5G-Integrated Internet of Things. *IEEE Access* **2020**, *8*, 67555–67571. [CrossRef]
- Qi, M.; Chen, J. Secure authenticated key exchange for WSNs in IoT applications. *J. Supercomput.* **2021**, *77*, 13897–13910. [CrossRef]
- Xie, Q.; Li, K.; Tan, X.; Han, L.; Tang, W.; Hu, B. A secure and privacy-preserving authentication protocol for wireless sensor networks in smart city. *EURASIP J. Wirel. Commun. Netw.* **2021**, *2021*, 119. [CrossRef]
- Odeh, J.O.; Yang, X.; Nwakanma, C.I.; Sahraoui, D. Asynchronous Privacy-Preservation Federated Learning Method for Mobile Edge Network in Industrial Internet of Things Ecosystem. *Electronics* **2024**, *13*, 1610. [CrossRef]
- Zheng, Z.; Li, T.; Li, B.; Chai, X.; Song, W.; Chen, N.; Zhou, Y.; Lin, Y.; Li, R. Industrial Metaverse: Connotation, Features, Technologies, Applications and Challenges. In *Methods and Applications for Modeling and Simulation of Complex Systems. AsiaSim 2022*; Springer: Singapore, 2022; pp. 239–263. [CrossRef]
- Sanders, C. The Data-Driven Future of Industries. Available online: <https://www.microsoft.com/en-us/industry/blog/manufacturing-and-mobility/2023/02/13/industrial-metaverse-the-data-driven-future-of-industries/> (accessed on 1 January 2020).
- International Telecommunication Union. Cyber risks, threats, and harms in the metaverse. *FG-MV* **2023**, *6*. Available online: <https://www.itu.int/en/ITU-T/focusgroups/mv/Pages/deliverables.aspx> (accessed on 1 January 2020).
- Li, B.; Ma, M. An Advanced Hierarchical Identity-Based Security Mechanism by Blockchain in Named Data Networking. *J. Netw. Syst. Manag.* **2022**, *31*, 13. [CrossRef]
- Huang, W. ECC-based three-factor authentication and key agreement scheme for wireless sensor networks. *Sci. Rep.* **2024**, *14*, 1787. [CrossRef] [PubMed]
- ITU-Telecommunication. M.3080:Framework of Artificial Intelligence-Enhanced Enhanced Telecom Operation and Management (AITOM); 2021. Available online: <https://www.itu.int/rec/T-REC-M.3080> (accessed on 1 January 2020).
- Blockchain Research Lab. Avatars: Shaping Digital Identity in the Metaverse. 2023. Available online: <https://www.blockchainresearchlab.org/wp-content/uploads/2020/05/Avatars-Shaping-Digital-Identity-in-the-Metaverse-Report-March-2023-Blockchain-Research-Lab.pdf> (accessed on 1 January 2020).
- Yaqoob, I.; Salah, K.; Jayaraman, R.; Omar, M. Metaverse applications in smart cities: Enabling technologies, opportunities, challenges, and future directions. *Internet Things* **2023**, *23*, 100884. [CrossRef]
- Thakur, S.S.; Bandyopadhyay, S.; Datta, D. *Artificial Intelligence and the Metaverse: Present and Future Aspects*; Springer: Cham, Switzerland, 2023; Volume 123, pp. 169–184. [CrossRef]

14. Enrique, C.; Sánchez, I.M.; David, L.; Lopez-Caudana, E. The Metaverse and complex thinking: Opportunities, experiences, and future lines of research. *Front. Educ.* **2023**, *8*, 1–13. [[CrossRef](#)]
15. Shi, L.; Xiong, T.; Cui, G.; Pan, M.; Cheng, N.; Wu, X. Multi-scale inputs and context-aware aggregation network for stereo matching. *Multimed. Tools Appl.* **2024**, 1–24. [[CrossRef](#)]
16. Schultz, A.D.; Kasnevich, D.A. IP in the Metaverse: An Overview for Retailers in a New Landscape. 2022. Available online: <https://www.huntonak.com/hunton-retail-law-resource/ip-in-the-metaverse-an-overview-for-retailers-in-a-new-landscape> (accessed on 1 January 2020).
17. Quach, S.; Thaichon, P.; Martin, K.D.; Weaven, S.; Palmatier, R.W. Digital technologies: Tensions in privacy and data. *J. Acad. Mark. Sci.* **2022**, *50*, 1299–1323. [[CrossRef](#)] [[PubMed](#)]
18. Seow, J.W.; Lim, M.K.; Phan, R.C.W.; Liu, J.K. A comprehensive overview of Deepfake: Generation, detection, datasets, and opportunities. *Neurocomputing* **2022**, *513*, 351–371. [[CrossRef](#)]
19. Ge, S.; Bai, C.; Liu, Y.; Liu, Y.; Zhao, T. Deep and discriminative feature learning for fingerprint classification. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (IC), Chengdu, China, 13–16 December 2017; pp. 1942–1946. [[CrossRef](#)]
20. Imamura, A.; Arizumi, N. Gabor filter incorporated CNN for compression. *arXiv* **2021**, arXiv:2110.15644.
21. Editorial, M. Deep Learning Models for Classification: A Comprehensive Guide. 2023. Available online: <https://metana.io/blog/deep-learning-models-for-classification-a-comprehensive-guide/> (accessed on 1 January 2020).
22. imabhijith25. Ultrasonic Sensor and LED Using Arduino. Available online: <https://www.instructables.com/Ultrasonic-Sensor-and-LED-Using-Arduino/> (accessed on 1 January 2020).
23. Clark, M. Fingerprint Reference Point Detection and Feature Extraction. 2021. Available online: <https://www.bayometric.com/fingerprint-reference-point-detection-and-feature-extraction/> (accessed on 1 January 2020).
24. University of Maryland, Baltimore County. Paillier’s Homomorphic Cryptosystem (Java Implementation). 2024. Available online: <https://redirect.cs.umbc.edu/~kunliu1/research/Paillier.html> (accessed on 1 January 2020).
25. Kaur, P.; Kumar, N. Biometric Cryptosystem with Deep Learning: A New Frontier in Security. In Proceedings of the 2023 International Conference in Advances in Power, Signal, and Information Technology (APSIT), Bhubaneswar, India, 9–11 June 2023; pp. 739–744. [[CrossRef](#)]
26. Liu, C.; Zhi, Z.; Zhao, W.; He, Z. Research on Fingerprint Image Differential Privacy Protection Publishing Method Based on Wavelet Transform and Singular Value Decomposition Technology. *IEEE Access* **2024**, *12*, 28417–28436. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.