Secure Biometric Authentication with Dynamic Symmetric Key Generation in IoT Systems

Sabina Szymoniak[1][0000−0003−1148−5691] and Mariusz Kubanek[1][0000−0001−9651−9525]

Department of Computer Science, Czestochowa University of Technology, Czestochowa, Dabrowskiego 69, Poland
{sabina.szymoniak,mariusz.kubanek}@icis.pcz.pl

Abstract. Expanding networks of interconnected devices pose novel security issues, particularly in commonplace IoT settings. This study presents a biometric authentication system that generates encryption keys on demand from facial characteristics, thereby obviating the need for permanent storage or external key servers. This technique facilitates rapid user authentication and secure messaging by generating session keys locally, making the system suitable for devices with constrained resources and fluctuating connectivity. The system demonstrated reliability and efficiency in actual tests, providing robust security against key duplication and brute-force attacks. Performance simulations demonstrate that the system operates reliably, even under substantial user and session loads. The findings indicate that biometric key generation may serve as a beneficial alternative to conventional security approaches in smart homes, healthcare, and industrial IoT applications.

Keywords: Symmetric key generation · Identity verification · Internet of Things · Pervasive computing.

1 Introduction

The Internet of Things (IoT) is transforming homes, workplaces, and public spaces by connecting everyday devices for tasks such as lighting control, health monitoring, industrial supervision, and traffic management. However, this growing connectivity also raises serious security and privacy concerns [22, 6, 15, 19, 12, 8]. In IoT systems, user authentication and encryption are essential. Biometric methods, such as facial recognition, are increasingly used instead of traditional passwords, but key storage and exchange remain challenging, especially in distributed and resource-constrained environments [21, 4].

To address these issues, we propose a local facial-feature-based method for generating one-time encryption keys. The entire process is performed on-device, without transmitting sensitive data or storing it long term. Using YOLOv11, the system derives a unique session key from selected facial points, and both the user identity and encryption key are generated anew for each session. We evaluate the method on diverse images and devices, and the results show that it is fast,

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

practical, and resistant to unauthorised access in typical IoT scenarios. This paper presents the assumptions of the communication and verification system, describes the biometric key generation mechanism, and reports the experimental results. We also compare the proposed method with existing approaches and discuss its practical use in modern IoT environments.

The rest of this work is structured as follows. Section 2 reviews the relevant literature and situates our methodology within the contemporary research landscape on biometric authentication and IoT security. Section 3 delineates the design assumptions of our system, encompassing its architecture, biometric verification methodology, and dynamic symmetric-key generation process. Section 4 presents and analyses the outcomes of our assessments, including security evaluations and scalability simulations. Section 5 examines the practical consequences, limitations, and prospective applications of the solution. Ultimately, Section 6 summarises the findings and delineates potential research directions.

2 Related Works

Symmetric keys are used to secure communications by encrypting and decrypting data. To ensure security, these keys must be sufficiently long and random, making them difficult for attackers to infer. Moreover, communication systems employing symmetric-key generation must protect keys from loss, duplication, or unauthorised access. In many cases, hardware random number generators play a key role in mitigating vulnerabilities associated with software-based generators used in key derivation [23].

Common symmetric-key generation methods are based on Password-Based Key Derivation Functions, such as PBKDF2 [3] and bcrypt [24]. PBKDF2 combines a password with a salt, a random string that significantly improves security and reduces vulnerability to dictionary attacks [2]. These methods often use standard hash functions, such as SHA-1 [20], SHA-256 [18], or HMAC [26], to produce keys, coupled with adjustable computational cost and salt to strengthen resistance to brute-force attacks. Algorithms such as Yarrow, Fortuna, and ANSI X9.31 extend the flexibility of key generation, allowing variable key lengths [14, 1]. Elliptic Curve Cryptography typically uses symmetric keys derived from operations on points of an elliptic curve, whereas algorithms such as Diffie-Hellman and the Digital Signature Algorithm produce secure outputs of varying lengths. Symmetric encryption methods, such as AES, employ techniques to extend short keys into longer ones while maintaining robust security [13, 7].

Table 1 summarises key characteristics of representative symmetric key generation methods from the literature. Unlike approaches relying on classical hash functions, genetic algorithms, or complex mathematical operations, our method is distinguished by its simplicity, lightweight implementation, and minimal resource requirements. It does not depend on computationally intensive processing or the storage of sensitive data, making it well-suited for IoT devices with limited resources. The main advantages include high entropy, strong resistance to brute-force attacks, the ability to operate entirely locally, and independence from

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

Table 1: Comparison of symmetric key generation methods

Method	**Features**
[5]	Dynamic key generation in wireless networks based on information theory. Keys derived from natural randomness in data transmission (for example, shared frames). Continuous refresh via XOR operation. High-entropy, quantum-resistant, scalable, and energy-efficient. Designed for IEEE 802.11 and IoT environments.
[25]	Symmetric key generation using genetic algorithms for IoT devices with limited computational resources. Novel evaluation function prevents weak keys. Focuses on differences between key and source data for greater randomness. Medium–high complexity, moderate brute-force resistance. Requires device-specific tuning.
[17]	Combines symmetric cryptography with polynomial operations and fuzzy logic. Designed for 5G networks. Uses differential calculus and modified Newton's theorem. High computational complexity, strong security, faster encryption/decryption. Limited applicability to low-power IoT devices.
Ours	Lightweight implementation without intensive computation or sensitive data storage. High entropy (0.9), very high brute-force resistance ($> 10^{18}$ years). Fully local operation, no internet or certification servers required. Adapted for low-power IoT devices. Biometric-based key generation from facial features.

continuous internet connectivity or external certification infrastructure, which together enhance security and privacy in pervasive environments.

Furthermore, we compare our method with a similar approach [22]. That method uses a static triangle formed by the left-eye corner, the right-eye corner, and the chin to generate symmetric keys, while our approach enables dynamic selection of three biometric parameters from a larger set: pupils, eye corners, nose tip, mouth corners, and chin. This flexibility increases entropy by allowing a distinct reference-point configuration for each communication session.

A further improvement is the use of YOLOv11 instead of YOLOv4 for face detection. YOLOv11 offers higher accuracy, faster inference, and lower computational cost, while remaining robust to challenging lighting conditions and varying head tilts—key requirements in IoT settings where users adopt different postures. Our method also applies more advanced biometric processing techniques, which increase complexity and make reverse-engineering of keys much more difficult, even if an attacker has partial information about the input data. Table 2 clearly shows that our solution achieves higher entropy, stronger security testing, and lower latency, while still being suitable for resource-constrained devices.

The flexible selection of biometric parameters and the use of YOLOv11 make the system practical for real-world IoT environments, including smart homes, office buildings, medical facilities, industrial settings, and access control systems. Local, distributed authentication and key generation align with current trends in IoT and pervasive computing, providing a scalable, secure, and lightweight alternative to the prior triangular-coordinate method.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

Table 2: Comparison of our method and the approach from [22].

Aspect	**Method from [22]**	**Our method**
Detection model	YOLOv4 + MobileNetV3	YOLOv11 + VGGFace
Biometric parameters	Fixed triangle (eyes + chin)	Flexible selection (3 of more than 6 parameters)
Key length	Unspecified	192 bits (24 characters)
Inference time	5–7 ms	2.4 ms
Rounds	10 simple	10 advanced
Security testing	Basic	NIST + Hashcat + Entropy
Entropy	Unmeasured	0.9/1.0

3 Assumptions of Secure Biometric Authentication with Dynamic Symmetric Key Generation in IoT Systems

User authentication is an important component in securely generating and maintaining symmetric keys, a process that can be challenging due to privacy and data security considerations. If keys are generated in an easily guessable manner, others may be able to decrypt the data, access it, or even modify it. It is essential to use strong pseudorandom number generators that make the generated numbers difficult to predict. It is also important to consider where keys are stored, since unsecured devices can leak data. As a result, we propose a novel, secure method for IoT systems to use biometric authentication with dynamically generated symmetric keys. Figure 1 presents a schematic diagram of the complete system architecture. Part A illustrates the biometric verification module. Part B illustrates the communication flow for the symmetric key generation process.

The proposed method can be applied in IoT settings with cameras, such as hospitals. The server and the user are the two main parts of the system's operational model. The server is responsible for verifying identities and controlling access. It has a separate application for managing users and the procedures associated with them. The user then has a client application on their device that allows them to verify their identity and communicate with other users in the system. Each user must use the server application to create an account. The server database stores their corporate email address and a facial image for verification. When a user makes an account, the system sends them a one-time authorisation code via email. This code associates the client program with a specific user's device. This is true for both single devices and those that are used in rotation at work. The images in the database must also be updated whenever the individual has a major change in their appearance, such as plastic surgery.

One of the most important aspects of the proposed system is that all authentication and key generation occur on the user's computer; no permanent internet connection or external servers are required. This allows each device to operate independently, consistent with the concept of pervasive computing. This method enhances transmission security, reduces the risk of connectivity failures,

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

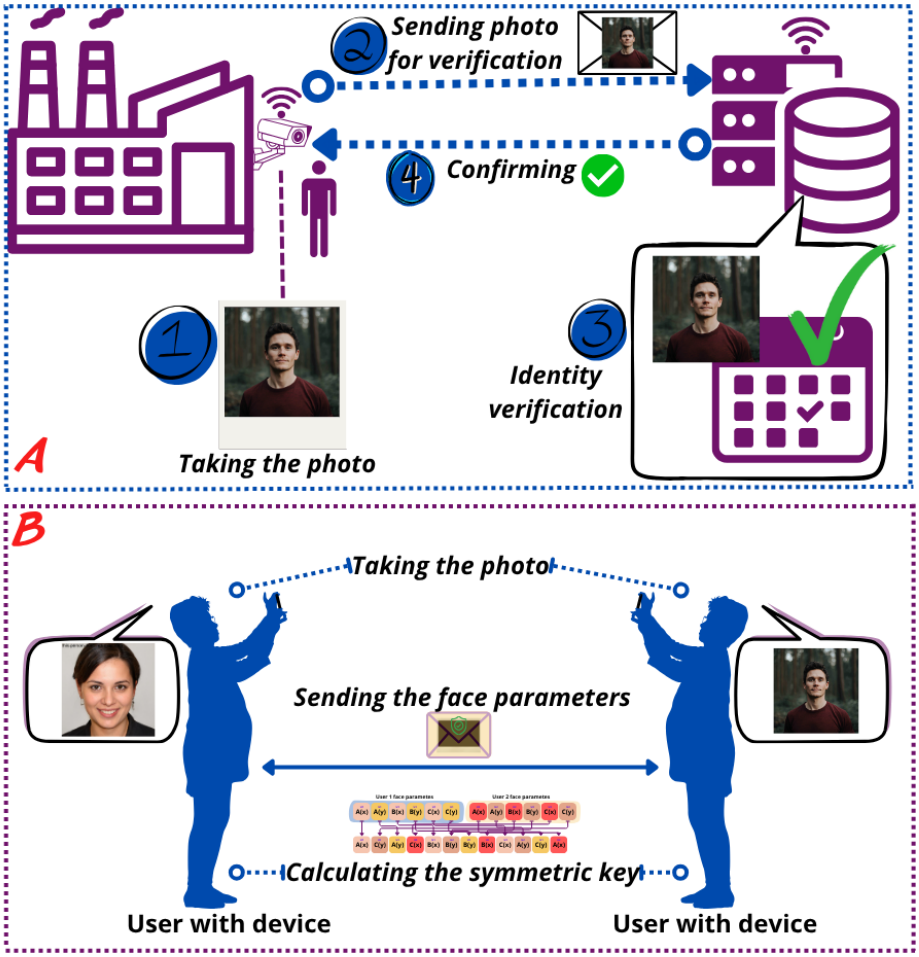


Fig. 1: Schematic diagram of the entire system.

and enables smooth operation in distributed settings in which multiple devices collaborate without a central control point.

To verify an individual's identity, the current camera image is compared with the one in the database. The frequency of verification depends on the company's needs. The user can perform additional tasks in the system after completing the verification process. One approach is to communicate with other users of the system using symmetric keys created specifically for them. This allows the user to send messages to another user in the system. The key is valid for only one communication session and is then erased from the devices. The system administrator sets the default length of the communication session.

The system uses convolutional neural networks, notably the newest YOLOv11 and VGGFace models. These models are trained on specific datasets so they

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

can reliably detect faces and link them to the correct users. The system also includes basic liveness verification features, such as motion- or texture-based checks, which help distinguish between a live person in front of the camera and attempts to spoof the system with photos or printed images. These checks are performed locally on the edge device, without relying on external servers. The process of authentication involves capturing a photograph, analysing it, comparing it to a reference model in the database, and finally confirming the user's identity. If any doubt arises—for example, low image quality, low detection confidence, or suspicious liveness signals—the system will not accept the verification. The entire procedure operates over the IoT network, and biometric data is transmitted securely using a session key. The technology protects data throughout its lifecycle, integrates with existing platforms, and enables real-time communication. Future work will extend the current liveness checks with more advanced anti-spoofing techniques, including 3D mask and deepfake detection, as part of further security evaluation.

The new method for generating session keys begins with randomly selected biometric facial traits. The process takes place on the user's device. After the user selects someone to talk to, the software prompts them to create a key and captures a selfie of both devices. The coordinates of three selected facial points, such as the pupils, chin, and corners of the eyes, are determined from the images. Then, the devices send the coordinates to each other in an encrypted way (asymmetrically). The time stamp of each message (the number of seconds since the Unix Epoch) is also used. The locations and timestamps from both devices are used to create the final session key. This ensures that both devices generate the same key without transmitting the key itself.

The process of mixing parameter values during key calculation is performed analogously to [22], i.e., we move the last fragment of the numerical sequence to the second position, the penultimate to the fourth, etc. However, the data processing process for generating a session symmetric key is carried out in the following steps:

1: Convert all face coordinates of both users into a binary value.
2: Combine them according to [22] into one binary number BN.
3: Convert both timestamps into a binary value and divide them into 6 blocks.
4: Combine them according to [22] into one binary number TS.
5: Combine BN and TS into a new $BNTS$ value.
6: $i = 0$
7: **while** $i < X$ **do**:
8: Divide $BNTS$ into blocks of four bits.
9: Rearrange the blocks of BN (similarly as in [22]).
10: Swap the odd blocks, for example, 1st with 3rd, 5th with 7th, etc.
11: Move the first $BNTS$'s block to the end of the block sequence.
12: Combine blocks into new $BNTS$ value.
13: **end while**
14: Divide the processed number into blocks of eight bits.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

15: Replace each block with its decimal representation.
16: Convert decimal values to ASCII characters.
17: Return the symmetric key.

The number of rounds in which binary values are processed is X. During implementation, the system administrator sets this value. A good number of rounds should keep the keys safe while also keeping the system running smoothly. More rounds mean more key distribution and more complexity. However, this could also entail higher computational costs, depending on the end devices' computational power.

Example

To demonstrate the operation of the symmetric session key generation method, we will use the following simplified facial coordinates: for the first User: $A(1,2)$, $B(1,2)$, $C(1,2)$, and for the second User: $A(3,4)$, $B(3,4)$, $C(3,4)$.

We also used simplified timestamp values of 123 and 321. We used simple numerical values to illustrate the proposed method and describe the steps performed in a single round of biometric parameter processing. If more rounds are used (step 6), the remaining rounds should be processed in the same manner. Generating a symmetric session key is as follows:

1. *Conversion of all face coordinates of both users into a 4-bit binary value*:
   $A_1(x) = 1, A_1(y) = 2, B_1(x) = 1, B_1(y) = 2, C_1(x) = 1, C_1(y) = 2$
   0001 0010 0001 0010 0001 0010 0001 0010 0001 0010 0001 0010
   $A_2(x) = 3, A_2(y) = 4, B_2(x) = 3, B_2(y) = 4, C_2(x) = 3, C_2(y) = 4$
   0011 0100 0011 0100 0011 0100 0011 0100 0011 0100 0011 0100
2. *Combine them according to the scheme from [22] into one binary number* BN.
   0001 0100 0010 0011 0001 0100 0010 0011 0001 0100 0010 0011 0001 0100 0010 0011 0001 0100 0010 0100 0001 0011 0010 0100 0011 0000
3. *Convert both timestamps into a binary value and divide them into six blocks.*
   $TS_{U1} = 123$
   1111011
   $TS_{U2} = 321$
   101000001
   1111011101000001
   000 111 1011 101 000 001
4. *Combine them according to the scheme from [22] into one binary number* TS.
   000 001 111 000 1011 101
5. *Combine* BN *and* TS *into a new* $BNTS$ *value.*
   0001 0100 0010 0011 0001 0100 0010 0011 0001 0100 0010 0011 0001 0100 0010 0011 0001 0100 0010 0100 0001 0011 0010 0100 0011 0000 001 111 000 1011 101
6. *Perform* $BNTS$ *in 1 round:*

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

(a) Divide *BNTS* into blocks of four bits.
0001 0100 0010 0011 0001 0100 0010 0011 0001 0100 0010 0011 0001 0100 0010 0011 0001 0100 0010 0100 0001 0011 0010 0100 0011 0000 0001 0111 0000 1011 0101
(b) *Rearrange the blocks of BNTS (similarly as in [22]).*
0001 0101 0100 1011 0010 0000 0011 0111 0001 0001 0100 0000 0010 0011 0011 0001 0100 0100 0010 0010 0011 0011 0001 0001 0100 0100 0010 0010 0011 0001
(c) *Swap the odd blocks, for example, 1st with 3rd, 5th with 7th, etc.*
0100 0101 0001 1011 0011 0000 0010 0001 0001 0111 0100 0011 0010 0000 0100 0001 0011 0010 0010 0100 0001 0011 0011 0100 0100 0001 0011 0010 0010 0001
(d) *Move the first BNTS's block to the end of the block sequence.*
0001 0100 0101 0001 1011 0011 0000 0010 0001 0001 0111 0100 0011 0010 0000 0100 0001 0011 0010 0010 0100 0001 0011 0011 0100 0100 0001 0011 0010 0010

7. *Divide the processed number into blocks of eight bits.*
00101000 10100011 01100110 00000100 00100010 11101000 00110010 00000100 00010011 00100010 01000001 00110011 01000100 00010011 00100010
8. *Replace each block with its decimal representation.*
40 163 102 4 34 232 50 4 19 34 65 51 68 34
9. *Convert decimal values to ASCII characters.*
(£ f k " è 2 k ! " A 3 D "
10. *Return the symmetric key.*
(£fk"è2k!"A3D"

Use case: edge authentication in smart manufacturing

A practical use case for the proposed system can be found in a smart manufacturing environment, where access to production zones and communication with industrial devices must remain both secure and fast. In such a setting, workers move between different sections of the facility, each protected by local IoT terminals equipped with cameras and processing capabilities. Before entering a restricted area, the system verifies the worker's identity directly on the edge device and immediately generates a temporary session key for secure communication with the local control unit.

This approach is particularly useful in industrial environments because it removes the need to contact a central server each time authentication is required. In a factory, even a short delay can disrupt workflow or affect machine coordination, so local processing becomes a real advantage. If the internet connection is unstable or unavailable, the system can still operate normally, which makes it suitable for distributed and time-sensitive production settings. The temporary key can then be used to protect short exchanges between the worker's device and the machine terminal, for example, when confirming access rights, sending maintenance data, or authorising a control action. Since the key exists only for a limited period and is discarded after the session ends, the risk of long-term

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

key exposure is reduced. This makes the system not only efficient but also well aligned with the security demands of modern industrial IoT.

What makes this scenario especially relevant is that it demonstrates the method's broader value beyond biometric verification itself. The system supports local decision-making, short-lived trust relationships, and device-level autonomy — all of which are important in pervasive computing. In that sense, it is not just an authentication mechanism, but a lightweight security layer that can fit naturally into edge-based industrial infrastructure.

4 Experimental Results

We conducted several tests using both real and synthetic facial images to ensure that the proposed method performed well and efficiently. The investigations encompassed the validation of the biometric recognition method and the assessment of symmetric-key generation utilising image pairs. We conducted all tests on a machine running Kali Linux, an Intel Core i7 processor, and 32 GB of RAM. We prepared a set of 30,000 pictures from [9] and [16] for testing. All photographs were set to a resolution of 512x512 pixels, which facilitated processing and reduced hardware-dependent differences. The images were in colour and depicted people of diverse ages and races. The generated photos varied in light intensity, facial angle relative to the camera, and resolution. We chose this method because it best illustrates how IoT systems are used in practice. We also sought to make the exam more representative by including images of people of diverse ages, races, and genders.

Additionally, we evaluated the system's user identification and verification using a curated set of tagged photographs. The established verification method accurately authenticated the images, aligning them with the identifiers retained in the database. Nonetheless, users external to the database were not authenticated. The results obtained are as follows: a true positive rate of 99.99%, a false positive rate of 0.01%, a precision of 99.99%, a recall rate of 99.99%, and an accuracy of 99.99%. The results validated the system's superior performance.

We also evaluate the symmetric-key generation technique utilising the assembled collection of photos. We standardised the parameter values obtained from the facial images to the range $< -1000, 1000 >$. Exemplary symmetric keys, computed using our methodology, are as follows:

- $ciWuqu252deVb7EgU3naaVad$ for coordinates [(278, 51), (1, 51), (101, 129), (57, 227), (289, 220), (99, 85)],
- $bw26p125asMWcVYqmcBGghq2$ for coordinates [(138, 162), (31, 224), (142, 290), (58, 86), (110, 283), (297, 242)],
- $imeqKZ2caHK2aJS9VaaKeta1$ for coordinates [(8, 254), (71, 243), (46, 158), (284, 185), (188, 19), (174, 281)].

Subsequently, we investigated entropy and collisions for the computed keys. The entropy study employed the conventional Shannon entropy measure [10], derived from the distribution of symbol frequencies in the keys. This facilitated a

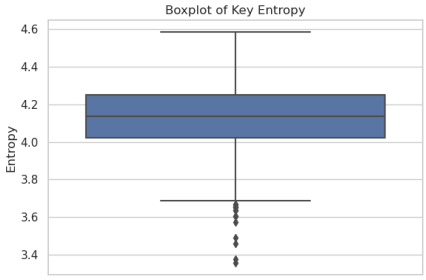
ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

quantitative evaluation of the degree of unpredictability and variation of the produced data. In this phase of the tests, the key generation algorithm was provided with sets of photo pairs as input. Biometric processing and key creation were conducted for each pair. The entropy of the generated keys, normalised to the interval, ranges from 4.1 to 4.2, corresponding to approximately 0.9–0.92 bits per character in a 24-character, 192-bit representation. This indicates that the keys are not fully random in the strict cryptographic sense, but achieve a relatively high level of unpredictability, sufficient for the system's use as short-lived session keys. The majority of keys have comparable entropy, indicating the uniformity and reproducibility of the creation process. Keys exhibiting low entropy (below 3.6 bits) should be regarded as anomalies, likely due to limited variability in the input data (for example, consistent facial positioning within the frame). Overall, the entropy distribution indicates that the proposed method yields highly unpredictable keys, substantially enhancing their resistance to brute-force attacks. Figure 2a illustrates a box plot of the entropy values for the produced keys. The observed pattern indicates the system's commendable scalability: as additional devices run parallel sessions, overall system performance improves. This implies that users experience negligible latency, even in a substantial, active network.

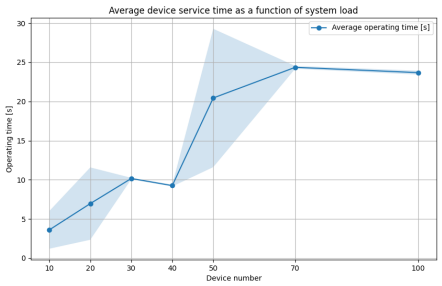
The analysis of the generated keys' uniqueness demonstrated the absence of collisions in the tested sample, confirming that each key was separate from the others. This outcome illustrates the system's robust capacity to produce clear, distinct values, even following numerous processing iterations. Absolute uniqueness is crucial for security, as it mitigates the risk of assigning identical keys to different devices or users. In practice, this indicates that the proposed strategy provides sufficient diversity to significantly reduce the probability of repetition in an IoT setting. In light of the entropy and uniqueness findings, we additionally evaluated the keys' resilience against brute-force attacks. We employed the Hashcat tool [11] to assess the viability of such attacks. For each generated key, we created a series of ciphertexts containing one, ten, and one hundred words. We utilised the AES-256-ECB NOKD technique to encrypt communications with the generated keys, and thereafter attempted to decrypt the resulting ciphertexts using Hashcat. All tested ciphertexts remained unbroken. Elevated entropy signifies a greater unpredictability of keys, hence complicating successful attacks. Collision analysis verified that no produced key was replicated, which is essential for communication security. Efforts to decipher the keys using techniques such as Hashcat were futile, thereby confirming the system's robustness. Ultimately, we assessed the system's scalability and efficiency within a distributed IoT framework. We modelled a virtual network of IoT devices. Each device concurrently managed key-generation sessions utilising pairs of photos extracted from the complete dataset. We executed the simulations in a multi-threaded format, with each thread representing an individual device. We quantified the total runtime of each device and the overall simulation duration across varying numbers of devices and sessions per device in the test series.

The results demonstrated stable system operation, with an average runtime of 13 seconds per device (for the whole simulation), without any overloads or

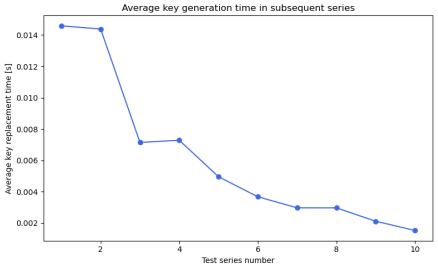
ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22



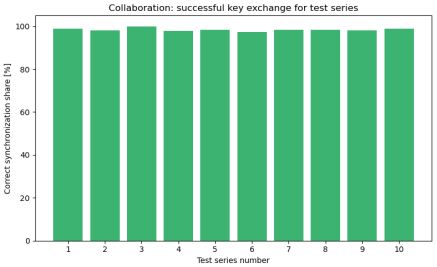
(a) Box plot of the entropy values for the generated keys.



(b) Average device service time as a function of system load.



(c) Average key generation time in subsequent series.



(d) Summary of successful key exchange for test series.

Fig. 2: Summary of performed tests.

errors. Table 3 shows statistics of performed simulations. In the rows, we assigned the following parameters (column *Par.*):

- Count - Number of observations (number of devices / sessions per device),
- Mean - Arithmetic mean of the runtime,
- Std - Standard deviation – how much the times differ from the mean,
- Min - Shortest measured runtime,
- 25% - (Q1) 1st quartile – the time below which 25% of the observations lie,
- 50% - Median – the middle value, divides the samples in half,
- 75% - (Q3) 3rd quartile – the time below which 75% of the data lie,
- Max - Longest measured runtime.

The average runtime reflects the typical session duration, while the standard deviation indicates the system's reliability; lower values suggest consistent results and a lack of atypical delays. Quartiles enable the analysis of time distribution and the identification of outliers.

Figure 2b illustrates the average service time of devices as a function of system load and the number of devices utilised. The experiments undertaken validated the substantial effectiveness and efficiency of the proposed system in

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

Table 3: Statistics of the 10 test series conducted

Par.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Count	10 / 10	10 / 50	20 / 10	20 / 50	30 / 30	40 / 20	50 / 20	50 / 50	70 / 30	100 / 20
Mean	5.39	25.93	10.31	53.86	47.98	44.51	56.58	147.51	123.58	116.40
Std	0.10	0.26	0.28	0.45	0.81	0.54	0.76	0.93	1.18	2.39
Min	4.83	25.40	9.70	52.78	44.57	43.33	54.04	144.97	119.53	106.77
25%	4.98	25.83	10.14	53.66	47.87	44.11	56.25	147.01	123.02	115.85
50%	5.02	25.98	10.36	53.96	48.12	44.54	56.70	147.56	123.75	116.88
75%	5.10	26.11	10.52	54.15	48.51	44.95	56.97	148.24	124.29	117.83
Max	5.13	26.28	10.68	54.54	48.83	45.55	57.91	148.96	125.54	119.84

both laboratory settings and distributed-environment simulations. In subsequent phases, we intend to broaden the research to include assessments of resilience to spoofing attacks, evaluations of the effects of variations in illumination and demographic settings, and pilot implementations in real-world IoT contexts.

The latter phase of the research assessed the interoperability and efficacy of key generation in a distributed network of IoT devices, employing simulated test series with varying numbers of nodes and key-exchange sessions. A distinct configuration (quantity of devices and sessions per device) was established for each test series, facilitating a thorough examination of both scalability and system reliability. Ten test scenarios were employed, aligned with the test series table, ranging from modest networks, such as 10 devices with 10 sessions (100 total attempts), to a big setup of 100 devices accommodating 20 sessions (2,000 key exchange attempts). In each test, two designated devices were tasked with generating and synchronising a key; the duration of the exchange and the process outcome (i.e., if the key was synchronised correctly) were recorded. The success rate of the key exchange and the average execution time were documented.

Figure 2c illustrates the correlation between the average key generation duration and the quantity of devices and sessions in each test iteration. In the initial two cases, the average generation time was noted to be the highest (0.014 [s]). With the increase in devices and sessions, the average processing time has consistently decreased, falling below 0.002 seconds in the most recent test execution. The findings validate the algorithm's significant scalability and optimisation. As the network size increases or the load distribution becomes more even, the efficiency of the key exchange process improves. In smaller networks, as observed in preliminary testing, key generation takes longer. Conversely, as the number of devices and sessions increases, the average duration remains extremely short, approximately 0.002 seconds. This pattern illustrates the system's scalability: as additional devices run parallel sessions, the total system performance improves. This implies that customers experience negligible delays, even in a large, congested network.

Figure 2d illustrates the proportion of accurate key synchronisation in each iteration. An exceptionally high success rate—nearly or over 98%—was attained

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

in all assessments. This verifies the system's exceptional reliability across different device quantities and workloads. The proportion of accurate key exchanges remains consistent across network scales, a key characteristic for distributed systems used in practical IoT settings. The analyses demonstrate that the proposed symmetric biometric authentication approach is effective in large-scale, distributed device networks, ensuring a rapid, reliable, and secure key-exchange process regardless of implementation scale. The accuracy of exchanges remains exceptionally high, approaching 98–100%, irrespective of the number of devices and sessions involved. This demonstrates that the system consistently synchronises keys as the network expands. The elevated percentage of successful exchanges is essential for effective operations in distributed IoT, as it indicates that users can rely on consistent communication without concerns about session loss or key misallocation.

The experiments were conducted using multiple repetitions of the same test scenarios, and the results show high reproducibility across runs. For each configuration, the observed entropy values and key generation times differed by less than 5% between runs, indicating that the system produces consistent behavior under identical conditions. This confirms that the reported performance characteristics are not outliers, but representative of the system's stable operation.

5 Discussion

The proposed method addresses a major challenge in modern IoT systems: ensuring fast, secure user identification and confidential communication. Using biometric data, particularly facial features, to generate session keys eliminates the need for centralised key management, storage, or exchange. This makes the system more resilient to key-theft attacks, especially in dynamic, distributed IoT environments. The system's strength lies in its minimal hardware requirements. Key generation is computationally lightweight, enabling deployment on resource-constrained devices such as security cameras, sensors, or mobile phones. The architecture is simple and does not require complex cryptographic methods, making it suitable for organizations with limited technical resources. Test results demonstrate high effectiveness and stability, with facial recognition performance remaining strong and generated keys exhibiting high randomness and resistance to brute-force attacks. Performance analysis confirms that the system can handle multiple users simultaneously with efficient resource utilization.

However, some limitations exist. System performance depends on facial image quality. Biometric detection and extraction may be less accurate under poor lighting, incorrect head pose, or low resolution. While alignment aids can help, environmental changes remain a challenge. Increasing user loads require advanced traffic management, like multi-channel queuing and distributed processing, to maintain reliability. Legal considerations are also important. Although biometric data is not permanently stored, processing photos for authentication still requires compliance with data protection regulations, including obtaining user consent and ensuring transparency.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

Basic liveness detection makes spoofing attempts with images or masks more difficult. However, as sophisticated spoofing techniques like deepfakes and 3D masks evolve, additional anti-spoofing measures are necessary. Overall, the system provides a strong foundation for developing biometric solutions in IoT environments, balancing security, efficiency, and practicality. Further research will focus on enhancing security against advanced threats and evaluating performance under real-world conditions.

6 Conclusions

This research demonstrates that secure biometric authentication and dynamic session key generation can be effectively implemented in distributed IoT systems using facial features. The solution provides reliable user verification and strong cryptographic properties, without the need for permanent key storage or centralised servers. Tests confirmed high recognition accuracy, high entropy of the generated keys, and effective resistance to brute-force attacks.

The architecture is lightweight and robust, making it suitable for resource-limited devices as well as diverse, real-world environments. Performance evaluations show that the system is both reliable and scalable, capable of handling high loads without significant degradation. Local processing aligns well with current requirements for privacy and flexibility in pervasive computing, though several technological and regulatory aspects still need to be addressed.

Current tests do not yet cover the impact of extreme environmental conditions or advanced spoofing techniques. These aspects will be included in the next phase of evaluations to improve the system's real-world applicability. Future work will extend the biometric foundations, enhance liveness detection, and address practical deployment issues in actual IoT settings. The proposed technique offers a viable alternative to traditional cryptographic schemes and supports the growing demand for secure, decentralised authentication in modern IoT systems.

References

1. Abhishek, K., et al.: On random number generation for kernel applications. Fundamenta Informaticae **185** (2022)
2. Adams, C.: Dictionary attack. In: Encyclopedia of Cryptography, Security and Privacy, pp. 637–638. Springer (2025)
3. AlQahtani, A.A.S.: Key derivation: A dynamic pbkdf2 model for modern cryptographic systems. Cryptography **9**(2), 39 (2025)
4. Ang, K.W., Chekole, E.G., Zhou, J.: Unveiling the covert vulnerabilities in multi-factor authentication protocols: A systematic review and security analysis. ACM Computing Surveys (2025)
5. Bhatti, D.S., Saleem, S., Lee, H.N., Kim, K.I.: A dynamic symmetric key generation at wireless link layer: information-theoretic perspectives. EURASIP Journal on Wireless Communications and Networking **2024**(1), 66 (2024)
6. Choudhary, V., Guha, P., Pau, G., Mishra, S.: An overview of smart agriculture using internet of things (iot) and web services. Environmental and Sustainability Indicators p. 100607 (2025)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22

7. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory **22**(6), 644–654 (1976)
8. Dutta, M., Gupta, D., Tharewal, S., et al.: Internet of things-based smart precision farming in soilless agriculture: Opportunities and challenges for global food security. IEEE Access (2025)
9. Generator, R.F.: Random face generator. this-person-does-not-exist.com
10. Grzywacz, N.M.: Perceptual complexity as normalized shannon entropy. Entropy **27**(2), 166 (2025)
11. Hashcat: Hashcat - advanced password recovery. https://hashcat.net/hashcat/
12. He, Y., Li, W., Zhang, M., et al.: Automated electric heating and curing system for concrete in actual cold environments based on the internet of things. Construction and Building Materials **489**, 142344 (2025)
13. Heron, S.: Advanced encryption standard (aes). Network Security **2009**(12), 8–12
14. Kumar, A., Mishra, A.: Evaluation of cryptographically secure pseudo random number generators for post quantum era. In: 2022 IEEE 7th International conference for Convergence in Technology (I2CT). pp. 1–5. IEEE (2022)
15. Mathkor, D.M., Mathkor, N., Bassfar, Z., Bantun, F., Slama, P., Ahmad, F., Haque, S.: Multirole of the internet of medical things (iomt) in biomedical systems for managing smart healthcare systems: An overview of current and future innovative trends. Journal of Infection and Public Health (2024)
16. Midjourney, I.: Midjourney. https://www.midjourney.com
17. Pradeep, S., Muthurajkumar, S., Ganapathy, S., Kannan, A.: Symmetric key and polynomial-based key generation mechanism for secured data communications in 5g networks. Soft Computing pp. 1–16 (2024)
18. Rahul, B., Kuppusamy, K., Senthilrajan, A.: Dynamic dna cryptography-based image encryption scheme using multiple chaotic maps and sha-256 hash function. Optik **289**, 171253 (2023)
19. Srivastava, M., Siddiqui, A.T., Srivastava, V.: Application of artificial intelligence of medical things in remote healthcare delivery. In: Handbook of Security and Privacy of AI-Enabled Healthcare Systems and Internet of Medical Things, pp. 169–190. CRC Press (2024)
20. Stevens, M.: Cryptanalysis of sha-1. Symmetric Cryptography, Volume 2: Cryptanalysis and Future Directions p. 181 (2024)
21. Szymoniak, S.: Secure system with security protocol for interactions in healthcare internet of things. Bulletin of the Polish Academy of Sciences Technical Sciences **72**(5), e151049 (2024)
22. Szymoniak, S., Kubanek, M.: Biometry-based verification system with symmetric key generation method for internet of things environments. Scientific Reports **15**(1), 5464 (2025)
23. Tkacik, T.E.: A hardware random number generator. In: International Workshop on Cryptographic hardware and embedded systems. pp. 450–453. Springer (2002)
24. Touil, H., El Akkad, N., Satori, K., Soliman, N.F., El-Shafai, W.: Efficient braille transformation for secure password hashing. IEEE Access (2024)
25. Tsai, M.Y., Cho, H.H.: A high security symmetric key generation by using genetic algorithm based on a novel similarity model. Mobile Networks and Applications **26**(3), 1386–1396 (2021)
26. Umamaheswari, S., Vishal, N., Pragadesh, N., Lavanya, S.: Secure data transmission using hybrid crypto processor based on aes and hmac algorithms. In: 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA). pp. 1–6. IEEE (2023)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_22