Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-021-03459-4

ORIGINAL RESEARCH

# Privacy preserving authentication system based on non-interactive zero knowledge proof suitable for Internet of Things

Ashutosh Dhar Dwivedi[1] · Rajani Singh[2] · Uttam Ghosh[3] · Raghava Rao Mukkamala[2] · Amr Tolba[5,6] · Omar Said[4,6]

Received: 28 December 2020 / Accepted: 17 August 2021


**Abstract**
Nowadays, with the advancement of smart technologies, the Internet of Things (IoT) emerged as a booming technology that can provide better quality and facilities for the residents of smart cities. Smart cities can offer several services and have several applications in healthcare, transportation, education etc. Despite such a potential vision, the privacy of users on these IoT devices is a major concern. Most authentication schemes do not provide privacy and anonymity to legitimate users. to tackle this problem, we propose an efficient Zero Knowledge-based authentication scheme in the paper that authenticates devices on the network without knowing the information about user identity or revealing any other data entered by users. To explain our system framework at the micro-level, we apply our privacy-preserving scheme to IoT based healthcare applications, but it can easily be extended to the more general use cases where privacy-preserving authentication is required. This paper's second major contribution is designing the data encryption algorithm ZKNimble that is mainly suitable for lightweight devices. Once the user is authenticated using Zero Knowledge Proof, the ZKNimble cipher can be used for legitimate users' encryption and decryption processes.

**Keywords** Internet of Things · Smart cities · Healthcare · Authentication · Zero knowledge proof · Security · Privacy

## 1 Introduction

Nowadays, half of the world's population live in cities, and till 2050, almost 66 percent of the population will move towards urban areas (World urbanization 2020). Due to the rapid population growth in urban areas, it is important to optimize cities' resources. Therefore, adopting the concept of "smart city" that coordinates resources and technologies intelligently. Smart cities have several great applications such as smart parking applications, waste management, traffic congestion, air pollution etc (Fig. 1). The other areas, where IoT devices (Jeong et al. 2019; Kim et al. 2018) play an important role are healthcare (Chandrakar et al. 2020), telecommunication (Malik and Zatar 2020), e-commerce, e-governance etc. More and more devices are connected to the internet, including smart cameras, wifi-routers, sensors. All these devices are called the "Internet of Things", and

✉ Ashutosh Dhar Dwivedi
adhdw@dtu.dk; ashudhar7@gmail.com

Rajani Singh
rs.digi@cbs.dk

Uttam Ghosh
uttam.ghosh@vanderbilt.ed

Raghava Rao Mukkamala
rrm.digi@cbs.dk

Amr Tolba
atolba@ksu.edu.sa

Omar Said
o.saeed@tu.edu.sa

[1] Cyber Security Section, Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800, Lyngby, Denmark

[2] Centre for Business Data Analytics, Department of Digitalization, Copenhagen Business School, 2000 Frederiksberg, Denmark

[3] School of Computer Science and Engineering, Vanderbilt University, Nashville, TN 37235, USA

[4] Department of Information Technology, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia

[5] Department of Computer Science, Community College, King Saud University, Riyadh 11437, Saudi Arabia

[6] Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebin El-Kom 32511, Egypt

Published online: 08 September 2021

almost 50 billion devices are connected with the internet in 2020. Because of the huge amount of these small devices connected with the internet, several attacks reveal the secret information of users.

The most common device that is commonly used in the Internet of Things is RFID (Radio Frequency IDentification). Apart from various applications in different smart city applications, RFID's technology plays an important role in healthcare. The RFID is mostly used to track such as medical types of equipment, hospital supplies, patients and medications etc. However, data transit by RFID devices can easily be intercepted by an adversary and therefore, it poses a big threat to track the user. Similarly, several times patients use IoT devices to share privacy preserved data to cloud servers. To retrieve or store data over the cloud, users authenticate them using some private pieces of information. This private information could be the user name, password, or other information that helps cloud servers identify users and authenticate them to use the service. The authentication process can be mutual or one side. Alice and Bob both identify each other in mutual authentication, while in one side authentication, where only Bob wishes to verify Alice (such as server authenticate users). In this work, we only use one side authentication where the system deal with two parties, a prover and a verifier. The prover convinces its identity by using some secret key such as password and verifier has corresponding verification key that confirms the prover's claim. Several attacks are possible by an adversary (e.g. men in the middle) where they can hack this private information (such as password sent through an insecure channel) and can access server or cloud data. Similarly, wire sniffing (see Fig. 2) is another type of attack where an adversary on the network can have access to client and server connections. If users are working on public wifi or unsecured channels, then it is easy to access passwords. However, several solutions proposed to make a secured communication system against these attacks that we discuss later, but they could not provide strong security to the system.

Therefore, in this paper, we proposed a zero-knowledge proof based authentication scheme where Prover identifies itself to the verifier without revealing any information about itself.

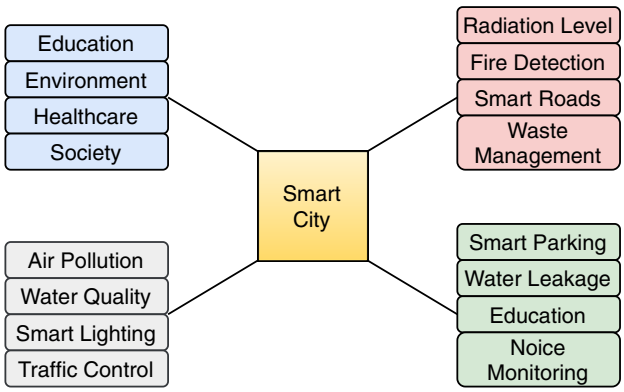


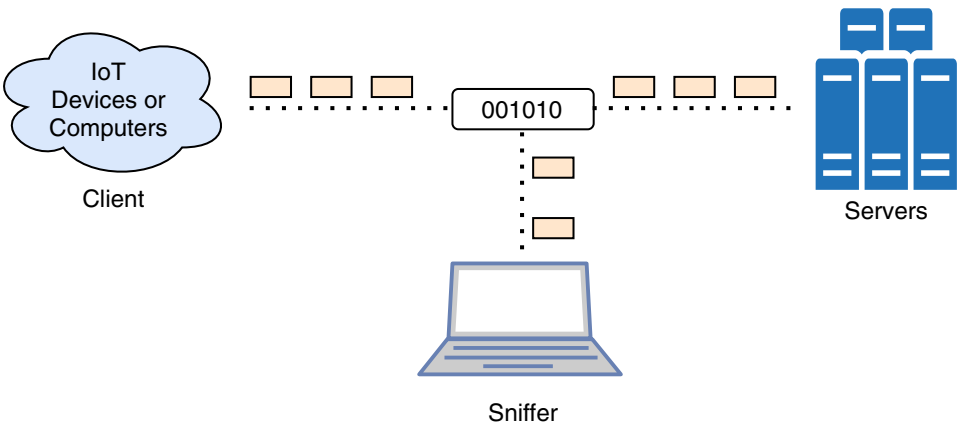

**Fig. 1** Overlay network

### 1.1 Our contribution

Our contribution is listed as follows:

- We proposed a non-interactive zero knowledge-based privacy-preserving authentication scheme that can be used in any Internet of Things based application for authentication purposes.
- To provide an enhanced level of security, we used a password-authenticated key exchange protocol to create each session.
- For data encryption, we designed a lightweight cipher ZKNimble that belongs to Feistel cipher's family. The cipher has a 64-bit block size that means it can encrypt the data block of size 64-bit at a time. The substitution and permutation layer design is chosen carefully and provides very tight security against the most popular linear and differential cryptanalysis attacks.

**Fig. 2** Packet sniffing



## 1.2 Background, motivation and related work

This section mainly gives readers motivation for this research work and discusses several possible attacks with traditional systems. The scenarios we discuss here involves two parties, a prover and a verifier. Party $A$ try to identify itself to another party $B$ to get the access of resources available at $B$. The protocols used for such scenarios are called Identification protocols. In such scenarios, the prover can have a secret key used to convince the verifier of its identity. The verifier also has a verification key that it uses for the confirmation of prover's claim. However, the following attacks are possible in such cases:

### 1.2.1 Direct attack

An adversary tries to gain Prover's or Verifier's login system's physical access in this attack. The attack happens when the attacker is in close physical proximity to the vulnerable login system. For example, an adversary can directly be able to copy the system data. An adversary can launch such an attack either by impersonating the verifier to the prover or prover to the verifier. To defend against such an attack, a normal password protocol is sufficient. For example, if one has a smart door with a digital lock, one cannot open it until he knows the password.

### 1.2.2 Active attack

In this attack, an adversary tries to learn and gain useful login information by actively participating in the prover interaction and later using it to impersonate the client user (prover) to the server (verifier). Consider a simple example where Alice wants to withdraw some money from the ATM. Alice did not know that she is interacting with a fake ATM. This fake ATM is designed to steal the customer's detail (ATM card details) instead of money. Alice cannot differentiate between the original and fake ATM as she inputted her card detail in the ATM, and it appeared to her as if it did not work. It actually happened in Manchester in 1993 when a criminal gang installed a fake ATM at a shopping mall. Later, the gang used those login credentials of customers who used their cards on fake ATMs to authenticate as the intended customer. To defend against such attack, a challenge-response technique between prover and verifier is required. Another version of an active attack is infecting client user's login system by using malware known as Trojan horse. In that case, the login system shows a fake login screen and fool the user by stealing his/her passwords. Later on, the stolen passwords can be used to impersonate login users to the client-server.

### 1.2.3 Denial-of-service attack

Denial-of-service (Salim et al. 2020) is also a type of Active attack, and in this attack, an adversary makes a system or network unavailable for the intended users (e.g., client user and server) by can denying the service to any individual user. An adversary can make several attempts to enter a wrong password so that the user's account gets locked. Although a firewall is required to defend against a DDOS attack by a single IP address, attacks made from several IP addresses are quite challenging to defend.

### 1.2.4 Passive attack (Eavesdropping)

In this attack, an adversary behaves as a passive observer rather than an active participant, so the system resources are not affected. Wiretapping is an example where an adversary eavesdrop. Consider another example of opening the car door using a wireless key hob. In this case, an adversary obtains the secret details of several interactions between the prover and verifier by eavesdropping on the radio channel. A simple password protocol cannot defend against such attacks; therefore, to defend against such attacks a more sophisticated protocol based on one-time passwords is required.

### 1.2.5 Brute force or exhaustive search attack

In this attack, an adversary simultaneously guesses the name and password in several attempts. This is a naive approach where an adversary exhaustively searches for the correct username and password combination. That means he/she has to check all the possible combinations until it is successful in cracking the correct one. This attack is the simplest one, but it is very time-consuming.

Brute Force attack has mainly three different variants described below:

1. *Targeted attack* This is the dictionary-based attack where an adversary guesses some popular words from the dictionary as a username. To guess the exact password, an adversary makes several login attempts.
2. *Trawling attack* This is opposite of the targeted attack. In this attack, an adversary first takes a password and then tries to guess the correct username corresponding to that password.
3. *Blind attack* In this attack, an adversary randomly guess the username and password.

However, apart from these, it is possible that an adversary can use some other methods to make the exhaustive search attack. For example, he/she can pretend that she forget the username and password and then give the correct answer to the security question. So, guessing the security question can be another approach. There is much work proposed in response to that,

such as in Farash et al. (2016), Wong et al. proposed a robust authentication scheme in which the login and registration process is demonstrated over a secure channel. In Vaidya et al. (2016), the authors proposed a system based on session keys for the mutual authentication between user and object. But in most of the work, the overall system reveals some information for adversaries that can be used to attack the system. Therefore, to ameliorate these attacks, we implemented a Zero-Knowledge Proof-based authentication system in this work. The benefits of using these systems are: they do not transmit any information that can be used to recover the password.

### 1.3 Zero knowledge proofs and protocol

Zero-Knowledge protocol is a cryptographic mechanism where two parties, prover and verifier, are involved. One party (Alice) can prove to another party (Bob) that she knows the secret value $x$ without giving any information to Bob except the fact that Alice knows the truth. There are also some cases where Alice does not reveal full information but only limited information that helps Bob learn about Alice's secret value. To understand this, let's take an example of a Sudoku puzzle (see Fig. 3). Consider that Alice knows Sudoku's solution, and Bob ask for her help to solve the game in exchange for the money. But how will Bob be sure before paying the money to Alice that she knows Sudoku's solution? For this reason, Bob can ask Alice to show only a few rows of solutions. If Bob is still not convinced by her solution, he can ask again for another column of the solution. Such interaction where both parties exchange several messages by communicating with each other to build trust and get convinced by the proof of knowledge is called Interactive Zero-Knowledge Proof.

Zero-Knowledge proof can be used as an authentication protocol due to the following properties:

1. *Completeness* This property ensures that if the statement is correct, then the verifier will surely be convinced with the prover's statement. Prover can prove the statement any number of times and verifier can verify the statement in a similar way for any number of times.

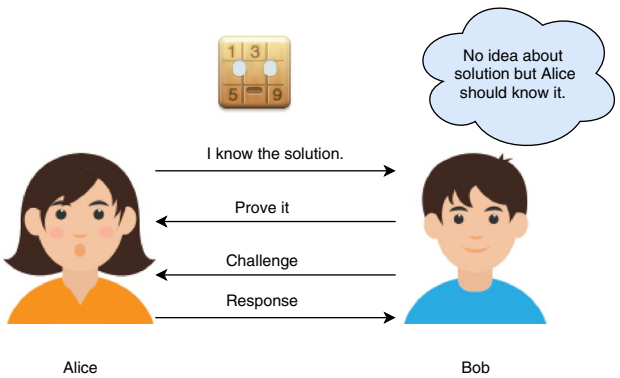



**Fig. 3** Interactive zero knowledge proof

2. *Soundness* This property ensures that if the statement is not true or false, the prover cannot cheat the verifier by stating and providing the false proof of that statement. Therefore, any false statement will not be accepted by the verifier and hence get rejected.
3. *Zero knowledge* This property ensures that if the statement is correct, the verifier is not able to learn any other information except the fact that statement is true. Thus, in the current scenario of this paper, where Alice wishes to identify herself, she is authenticated to use the server and can access information without revealing any identification details such as id or password.

Zero-knowledge protocol involves two parties: prover and verifier, and consists of three processes, namely: commitment, challenge, and verify explained below:

1. *Commitment* The first and most important phase of any Zero Knowledge Protocol construction is a commitment where the first party say Alice commits to her secret choice by choosing a random value or number from a finite set such that she cannot change it later. To do so, Alice generates a commitment by using any cryptographic commitment scheme, for example, Pedersen commitment scheme and sends it to another party, say Bob.
2. *Challenge* Once Bob receives the commitment, he sends a query or checkpoint called a challenge to Alice. By doing so, Bob wants to be sure that Alice really has a knowledge of the secret and is not cheating with him by giving some false commitment values. However, this computation process is time-consuming, and Bob has to wait for the response of Alice. Moreover, choosing the more complex zero-knowledge proofs can further increase the computation time.
3. *Verify* The third and the most important phase is verified where Bob validates Alice's solution of the challenges sent by him. If the solutions are correct, then Bob's accepts it with the overwhelming probability otherwise, he rejects it. So, the successful verification means that Bob gets convinced by Alice's solutions or proof. This further proves to Bob that Alice has real knowledge of the secret.

## 2 Related work

In the past, several researchers proposed interactive zero-knowledge proof for authentication. In Narwhal (2014), Cheu et al. presented a Zero-Knowledge based authentication scheme. The system proposed by them provided website authentication and saved the login systems against several vulnerabilities. Users computers can easily implement the Zero-Knowledge set-up provided by them without degrading the quality of communication experience. However, the system requires JavaScript

enabled computers, forcing the users to place some trust on the website. Similar work was done by Soewito et al. in Soewito and Marcellinus (2020) where authors presented a Zero Knowledge-based authentication system. In this paper, the authors used Zero-Knowledge Proof for authentication purpose and Advanced Encryption Standard (Rijndael) for data encryption.

However, the biggest issues with such interactive Zero Knowledge Proof is extra communication load to devices. In response to that, in Walshe et al. (2019), Walshe et al. proposed an authentication scheme for IoT and sensor devices that was based on Non-Interactive Zero-Knowledge Proof. Instead of using traditional ZKP, the authors replaced the ZKP NP-hard problem and used the Merkle tree to create the authentication challenge. The authors also performed some simulations to evaluate the performance of non-interactive Zero-Knowledge Proof against traditional Zero-Knowledge.

Zero-knowledge has already been used as a privacy preserving technique in different areas such as traffic management (Li et al. 2020), Crowdsourcing Internet of Things (Liu et al. 2020), identity management scheme in blockchain (Yang and Li 2020), and Vehicular Ad Hoc Networks (Rasheed et al. 2020; Gabay et al. 2019).

Malina et al. (2018) proposed an efficient two-factor zero knowledge based authentication protocol for fast access control systems and user-things identification schemes. Their protocol is secure against common attacks. Partala et al. (2020) provided a comprehensive survey on the applicability of non-interactive zero knowledge proofs into Blockchain. Wei et al. (2021) proposed a Privacy-preserving message authentication scheme for Internet of Things. Other important work in the same direction are (Park et al. 2019b; Aboushosha et al. 2020; Park et al. 2019a; Choi and Ahn 2019; Lee et al. 2020).

## 3 Proposed system

There are several applications where Zero-Knowledge proofs can be used as a solution to the privacy issue. However, we propose a Zero-Knowledge based authentication protocol to identify the real and legitimate users in this work. Once the user is identified as an authenticated user, the key exchange session is started. The required or queried data is being encrypted and decrypted by using the session keys. The whole process is divided into three steps.

- Sender Authentication: If Alice is sending a message to Bob, Alice will be authenticated using a Zero-Knowledge Proof. Alice does not need to send any user id or password to authenticate herself.
- Data can be encrypted by using our proposed encryption algorithm that is also suitable for lightweight devices.

### 3.1 Authentication (identification) and login protocol

Throughout the paper, we assume two fictional characters, Alice and Bob. There are two parties in the identification problem: prover (Alice) and a verifier (Bob). Alice is a client user or prover, while Bob is a server or verifier. A cryptographic method by which one party (Prover) wishes to identify itself to another party (Verifier) to access the resource available to that party is known as identification protocol. There are several scenarios where such identification protocol can be used for example, opening a smart digital door lock, Unlocking a car using a wireless key fob, Login at bank's ATM or online bank account etc. Formally an Identification protocol can be defined as follows:

**Definition 1** An identification protocol consists of three algorithms, namely Key Generation ($\mathcal{G}$), Prover ($\mathcal{P}$) and Verifier ($\mathcal{V}$) described as below. We denote the protocol by $\mathcal{I} = \{\mathcal{G}, \mathcal{P}, \mathcal{V}\}$.

**Algorithm 1:** Identification Protocol $\mathcal{I} = \{\mathcal{G}, \mathcal{P}, \mathcal{V}\}$

1. **Key Generation ($\mathcal{G}$):**
   It is a probabilistic algorithm that generates two cryptographic keys $(s_k, v_k)$. Here $s_k$ is a secret or private key while $v_k$ is a verification key. Depending on the situation, $v_k$ can be either private or public. Note that in this algorithm, there is no input but provides a pair of keys as output.
2. **Prover ($\mathcal{P}$):**
   It is an interactive protocol algorithm that takes previously generated secret key $s_k$ as input and provides no output. So, here only one of the generated key is being assigned to one party.
3. **Verifier ($\mathcal{V}$):**
   It is an interactive protocol algorithm which takes previously generated verification key $v_k$ as input and provides **accept** or **reject** as output. So, here one of the generated keys is being assigned to another party, and it either accepts or rejects the prover's request.

We consider the interactive protocol, which means Alice having a secret key $s_k$ interacts several times by exchanging messages with Bob, who has a verification key $v_k$. At the end of interaction between Alice and Bob, If Alice has $s_k$ and Bob has $v_k$ then for all possible output $(s_k, v_k)$ of Key generation algorithm Bob should output **accept** the request with probability 1. This is a necessary requirement for a successful identification protocol.

### 3.2 Schnorr's identification protocol (SIP)

This protocol is secure against direct attacks and eavesdropping attacks, assuming that the discrete logarithmic problem is hard.

Let $\mathbb{Z}_q$ be a subgroup of a cyclic group $\mathbb{G}$ of prime order $q$. The generator of $\mathbb{G}$ is $g$, also called primitive root. Let $\mathcal{C}$ be a subset of a subgroup $Z_q$. We use some simple modular arithmetic.

Schnorr's protocol $\mathrm{ID}_{\mathrm{Sch}} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ uses the argument of knowledge of secret key. Assume that Alice (prover) has a secret key say $s_k = k$ where $k$ is chosen from the key-space $\mathbb{Z}_q$. Corresponding public key or verification key of Bob(verifier) is $v_k = g^k$. To prove her identity, Alice tries to convince Victor that she knows the secret key $k$. For this, both interact with each other and involve in a challenge-response procedure as described below.

**Algorithm 2:** Key Generation Algorithm ($\mathcal{G}$) in SIP

Prover chooses a number $a$ uniformly at random from the subgroup $\mathbb{Z}_q$.
A pair of secret or private and verification key are generated as follows:

$s_k = a$, **and** $v_k = g^a$.

Generated key pair $(a, g^a)$ is the output of this probabilistic algorithm:

**Algorithm 3:** Zero knowledge proofs in SIP

This probabilistic Prover Verifier Algorithm $(\mathcal{P} - \mathcal{V})$ takes the pair of keys $s_k = \delta$, $v_k = g^\delta$ as input. The interaction between prover and verifier works as follows:

1. **Commitment:**

   Prover chooses a number $a$ uniformly at random from the subgroup $\mathbb{Z}_q$ and computes a commitment $U$ as follows:

   $A = g^a$.

   Prover then sends the commitment value $A$ to the verifier.
2. **Challenge:**

   Verifier $\mathcal{V}$ selects a number $c$ uniformly at random from space $\mathcal{C}$ and sends this value to the prover. Value $c$ is called the challenge sent by the verifier to prover.
3. By choosing two random numbers, prover can draw a straight line $a + \delta \cdot m$ with slope $m$ and intercept $a$. When he received a challenge value $c$ he puts $m = c$ into the equation of the straight line and gets a unique point on this line that is

   $z = a + \delta \cdot c$.

   Value $z$ is called the response to the challenge sent by the prover to the verifier.
4. **Verify:**

   Now, the verifier checks if $g^z = A \cdot \delta^c$ hold or not. If the equality holds then the verifier outputs **accept** otherwise outputs **reject**.

For the non-interactive version of the protocol, the verifier's challenge is replaced by the hash of the commitment calculated by the prover. So, now prover publishes the value $A$ along with the challenge $c = Hash(A)$ and the verifier only have to check if the hash value is calculated correctly or not.

To prove the security of this protocol, we need to assume that $|\mathcal{C}|$ is super-poly which means an adversary is computationally bounded. Therefore, it is computationally infeasible for the adversary to break the discrete log problem.

### 3.3 Password-authenticated key exchange (PAKE)

Password-authenticated key agreement is a cryptographic mechanism used to generate the cryptographic keys of one or more parties. It is an interactive method that means that both parties exchange the message for establishing such cryptographic keys. Moreover, it is built on the assumption that both parties know the password. PAKE provides strong security as it is safe against the man in middle attacks and eavesdroppers. It also provides strong user privacy as it utilizes the concept of Zero-Knowledge proofs so that leaking of user's login data to the unauthorized party is almost impossible.

To generate the session key, both PAKE protocol and Diffie-Hellman use the shared password. In fact, PAKE protocols are the extensions or variants of a Diffie-Hellman key exchange. This makes the key exchange authenticated and prevents man-in-the-middle attacks, as the key depends on a password that is not sent but is only with both parties as their secret.

Since the PAKE protocol guarantees strong security to weak passwords, it has an important application in the Internet of Things areas. Suppose you want to connect an IoT device or smart devices such as smart light, smart-toothbrush, smart-speaker, etc. to your smartphone and use a smart device's app. In such a case, PAKE is used as the users, it is much preferable to use a small for digit pin instead of a long, highly secure password (Fig. 4).

#### 3.3.1 Registration protocol

The registration protocol works as follows:

- The client user chooses either a username or email address as his/her unique identifier, let's call it UID. Client user then uses a browser to send UID to the server.
- The server sends back a QR code that contains both the client user's identity UID and server's domain name, which we refer as the server's identity and denote by

SID. Along with these identities, the server also sends the URL needed by a mobile device to send the initialization data. The user must use only this URL.

- The client user scans the QR code and secretly chooses a master password to say $m_{psw}$. Master secret (traditional password), can be of arbitrary length and of any nature. Since our protocol has a Zero-Knowledge property, no unauthorized party or adversary can learn anything about the secret from the password creator.
- Now the app on mobile device computes the hash function and the public parameter $v$ as shown below:

  $x = \text{Hash}(\text{UID}, \text{SID}, m_{psw})$ and $v = g^x$.

  The user sends his/her identity UID and the public parameter $v$ to the previously received URL.
- The server checks if the client identifier is not already used and valid. Also, if the UID is an email address, the server should first verify the email, ensuring that the client user has access to the provided email address. In the affirmative case in both checks, the server permanently stores this UID along with his public value $v$.

### 3.3.2 Authentication protocol

The authentication protocol works as follows:

- The client user opens the browser and starts the login process by sending his/her UID to the server.
- After receiving the UID, the server search for the corresponding parameter $v$ to generate a pair of private and public cryptographic keys. Private or secret key $b$ is randomly chosen by the server while the public key is calculated as

  $B = kv + g^b$.

- The server sends back own identity SID generated public key $B$ and user identity UID to both the browser and a mobile device. Now consider the two possible scenarios.
  - If the mobile device is connected to the internet and receives the server's public key $B$. Then a fingerprint of $B$ is displayed on both the mobile device app and the browser. Now the client user is asked to make sure if both displayed fingerprints match.
  - If the mobile device is not connected to the internet, the client user must have to select an alternative method such as Webcam or Bluetooth for transferring the public key $B$ from the browser to the mobile device app.
- The client user can either use a Password Pswd or mobile device's sensor to accept the authentication request on his/her mobile device. Now by using the master password $m_{psw}$, the session key $K$ and its proof $M$ is computed as shown below:
- In the verification process, the proof $M$ is being verified. To do so, the server first computes its own session key $S = (Av^u)^b$ and $K = \text{Hash}(S)$. If it matches the received data, the server authenticates the user successfully and automatically redirects the client to the requested page.


**Fig. 4** PAKE application

**Algorithm 4:** Password Key Exchange Protocol

- Compute the Hash of client-server's identifiers along with user's master password

  $x = \text{Hash}(\text{UID}, \text{SID}, m_{psw})$.

- Generate the Client user's private key $a$ by choosing a value uniformly at random.
- Compute the Client user's public key $A$

  $A = g^a$.

- Compute the Hash of the pair of public-private key

  $u = \text{Hash}(A, B)$.

- Compute the Session key

  $S = (B - kg^x)^{(a+ux)}$.

- Compute the Hash of Session key

  $K = \text{Hash}(S)$

  .
- Zero knowledge proof of session key

  $M = \text{Hash}_K(l, \text{UID}, \text{SID}, A, B, d)$.

### 3.4 Data encryption

Once the sender is authenticated as a real user, encryption and decryption of data is another goal. The major challenge is to provide an efficient, secure and lightweight cipher responsible for encrypting and decrypting data. The biggest challenge for the designer of ciphers is to make balance between *cost*, *security* and *performance*. However, getting all these properties creates a trilemma, and only two of them is possible out of three. In this work, we proposed a new encryption algorithm (named ZKNimble) suitable for lightweight devices. Following matrices are considered to provide an efficient encryption algorithm.

- *Throughput* The rate at which output can be produced with respect to time is called throughput. The measurement of throughput is expressed in bits per second [*bps*]. Mathematically it can be expressed as:

$$Throughput = \frac{number\_of\_output\_bits}{time}$$

- *Area* The area of a cipher is generally calculated by adding individual gates used and called gate equivalents (GE). It can be expressed or measured in $\mu m^2$.
- *Efficiency* The ratio of throughput and area is expressed as the efficiency of any cipher. Mathematically it can be expressed as:

$$efficiency = \frac{throughput}{area}$$

- *Power* Power consumption is another major component that plays an important role when designing the cipher. The measurement of power is micro Watt [$\mu W$] and mathematically, it can be expressed as:

$$P = \left(\frac{1}{2} \cdot C \cdot V_{dd}^2 + Q_{sc} \cdot V_{dd}\right) \cdot f \cdot N + I_{leak} \cdot V_{dd}$$

  In the above equation, $Q_{sc}$ is short circuit charge, $C$ is circuit capacitance, $V_{dd}$ is the supply voltage, $I_{leak}$ is leakage current, $N$ is switching activity, and $f$ is operating frequency.

### 3.5 Specifications of ZKNimble

The cipher is designed to follow the family of Feistel Ciphers (Type 1). ZKNimble is a block cipher of size 64-bit represented by $4n$, and this block is divided into four parts of word size $n = 16$. ZKNimble takes plaintext and divide it into four words of size 16-bit each and produce a ciphertext as an output with the same size after 32 rounds of encryption. The cipher uses permutation and substitution layer operations to provide the property of confusion and diffusion. Initially, the 32-bit main key is used that is divided into 2 part, key(LSB) and key(MSB) and passed through the round function of the key generation algorithm that produces keys for each round (Fig. 5).

#### 3.5.1 Round function

The cipher has mainly three component: encryption, decryption and key expansion. The operations used for cipher has following operations:

- Permutation-Box $P$
- Substitution-Box $S$
- bitwise XOR, $\oplus$

where, $k \in GF(2)^n$. For the decryption process, cipher uses the inverse permutation box and inverse substitution box. S-Box used as a substitution layer plays an important role in the construction of any cipher. Major important attacks such as linear or differential attacks mostly depend on the design of S-Box. Therefore it is essential to choose the design of S-Box so that they are secured against these two attacks. S-box components are non-linear by nature and provide confusion property to the cipher. The S-Box design is shown in Table 2. The second layer used to provide diffusion property is the permutation layer shown in Table 1.

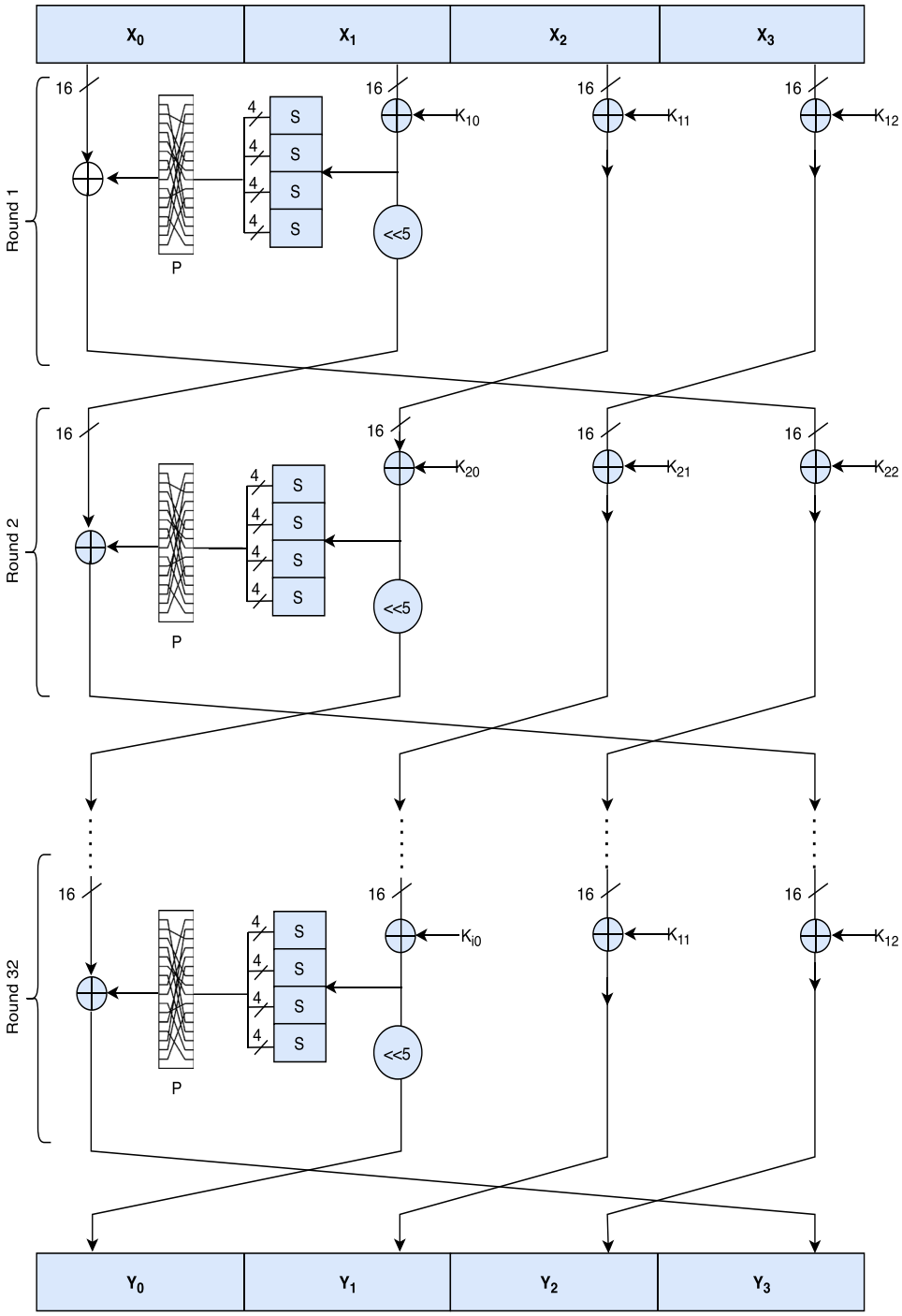



**Fig. 5** The round function of ZKNimble

**Table 1** Permutation box of ZKNimble

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P(x) | 10 | 15 | 0 | 5 | 14 | 3 | 4 | 9 | 7 | 2 | 8 | 13 | 1 | 12 | 11 | 6 |

**Table 2** Substitution box of ZKNimble

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S(x) | C | 5 | B | 9 | 6 | 0 | D | A | E | 3 | 8 | F | 4 | 1 | 7 | 2 |

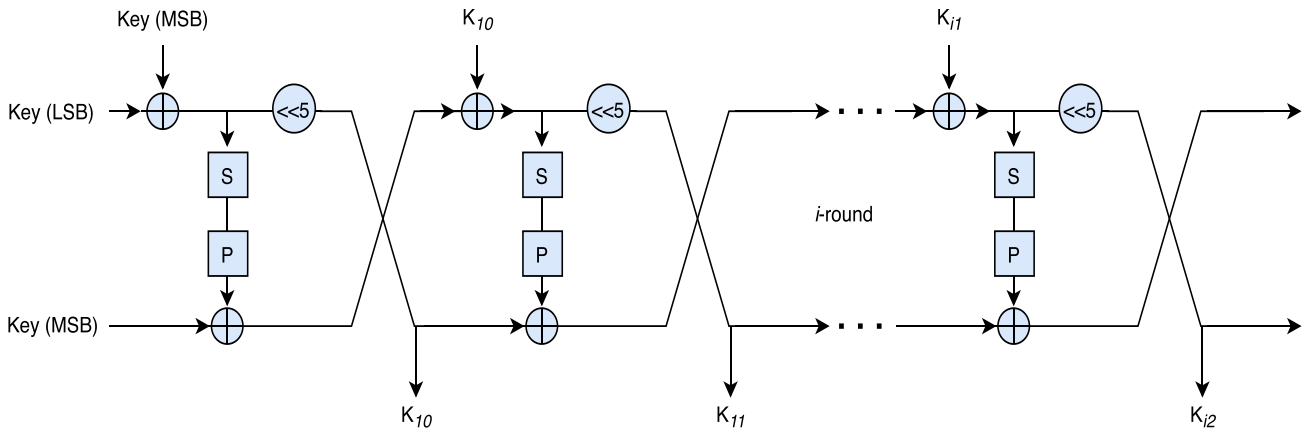



**Fig. 6** Key generation diagram of ZKNimble

### 3.6 Key schedule

The initial key $K$ is of size 32-bit and divided into two half key(LSB) and key(MSB) and passed through key generation function shown in the Fig. 6. The key generation function has 32 steps, and each step consists of 3 rounds that generate $K_{i0}, K_{i1}, K_{i2}$ keys for *i th* round of the cipher.

## 4 Security analysis of proposed system

- *Direct attack* In the proposed scheme, we use zero-knowledge proofs, which require the knowledge of the password. So, our proposed system is secure against such attacks.
- *Active attack* Our non-interactive zero-knowledge proofs do not leak any extra information, and thus, an adversary learns nothing about the login detail. So, our proposed system is secure against such active attacks.
- *Denial-of-service attack* Our proposed system is secure against such attacks. In case of resource unavailability, the authentication system will be temporarily suspended. Also, one-time password authentication key exchange will prevent the system from the DoS attack.
- *Passive attack (Eavesdropping)* Since our proposed system relies on non-interactive zero-knowledge proofs. Therefore, a passive observer can learn nothing about the login credential of any user. Moreover, zero-knowledge proofs are built in such a way that a passive observer or adversary does not have any advantage of eavesdropping in the proposed authentication system.
- *Brute-Force or exhaustive search attack* We assume that our adversary is computationally bounded in polynomial time. Moreover, the Completeness and soundness property of our zero-knowledge proofs guarantee that our system is secure against any kind of Brute-Force attack.

## 5 Conclusion and future work

Application of privacy-preserving authentication system is not only restricted to healthcare domain but it such authentication system has several applications in other areas such as Vehicular Ad Hoc Networks, Wireless sensor Networks, Smart home, Smart city etc.

Privacy of the user is one of the important problems in the contemporaneity of the Internet of things based applications such as smart city, smart devices, cctv cameras etc. The paper proposed a privacy-preserving authentication-encryption system that uses Non-Interactive Zero-Knowledge Proof for authentication and ZKNimble cipher to encrypt the data. Due to the non-interactive behaviour of the Zero Knowledge Proof, the whole system is very suitable for lightweight devices. Once a user is identified as authentic, the ZKNimble cipher can be used to encrypt or decrypt the data. The cipher is designed to perform well on lightweight devices, and we designed the substitution layer to make it safe against linear or differential attacks.

**Funding** The work of Ashutosh Dhar Dwivedi is supported by the Independent Research Fund Denmark for Technology and Production under Grant 8022-00348A. The work of Rajani Singh is funded by the Danish Ministry of Education and Science, Digital Pilot Hub and Skylab Digital. The work of Omar Said is funded by Taif University Researchers Supporting Project number (TURSP-2020/60), Taif University, Taif, Saudi Arabia.

**Data availability** Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

## References

Aboushosha B, Ramadan RA, Dwivedi AD, El-Sayed A, Dessouky MM (2020) SLIM: a lightweight block cipher for internet of health

things. IEEE Access 8:203747–203757. https://doi.org/10.1109/ACCESS.2020.3036589
Chandrakar P, Sinha S, Ali R (2020) Cloud-based authenticated protocol for healthcare monitoring system. J Ambient Intell Humaniz Comput 11(8):3431–3447. https://doi.org/10.1007/s12652-019-01537-2
Choi J, Ahn S (2019) Scalable service placement in the fog computing environment for the iot-based smart city. J Inf Process Syst 15(2):440–448
Farash MS, Turkanović M, Kumari S, Hölbl M (2016) An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment. Ad Hoc Netw 36:152–176. https://doi.org/10.1016/j.adhoc.2015.05.014 (**ISSN 1570-8705**)
Gabay D, Akkaya K, Cebe M (2019) A privacy framework for charging connected electric vehicles using blockchain and zero knowledge proofs, pp 66–73. https://doi.org/10.1109/LCNSymposium47956.2019.9000682
Jeong Y-S, Park JH (2019) Iot and smart city technology: challenges, opportunities, and solutions. J Inf Process Syst 15(2):233–238
Kim NY, Rathore S, Ryu JH, Park JH, Park JH (2018) A survey on cyber physical system security for iot: issues, challenges, threats, solutions. J Inf Process Syst 14(6):1361–1384
Lee Y, Rathore S, Park JH, Park JH (2020) A blockchain-based smart home gateway architecture for preventing data forgery. Hum Centric Comput Inf Sci 10:9. https://doi.org/10.1186/s13673-020-0214-5
Li W, Guo H, Nejad M, Shen CC (2020) Privacy-preserving traffic management: a blockchain and zero-knowledge proof inspired approach. IEEE Access 8:181733–181743. https://doi.org/10.1109/ACCESS.2020.3028189
Liu W, Wang X, Peng W (2020) Secure remote multi-factor authentication scheme based on chaotic map zero-knowledge proof for crowdsourcing internet of things. IEEE Access 8:8754–8767. https://doi.org/10.1109/ACCESS.2019.2962912
Malik H, Zatar W (2020) Agent based routing approach to support structural health monitoring-informed, intelligent transportation system. J Ambient Intell Humaniz Comput 11(3):1031–1043. https://doi.org/10.1007/s12652-019-01202-8
Malina L, Dzurenda P, Hajny J, Martinasek Z (2018) Secure and efficient two-factor zero-knowledge authentication solution for access control systems. Comput Secur 77:500–513. https://doi.org/10.1016/j.cose.2018.05.006 (**ISSN 0167-4048**)
Narwhal (2014) https://courses.csail.mit.edu/6.857/2014/files/15-cheu-jaffe-lin-yang-zkp-authentication.pdf. Accessed 3 Sept 2021
Park D-M, Kim S-K, Seo Y-S (2019a) S-mote: SMART home framework for common household appliances in iot network. J Inf Process Syst 15(2):449–456
Park JH, Salim MM, Jo JH, Sicato JCS, Rathore S, Park JH (2019b) Ciot-net: a scalable cognitive iot based smart city network architecture. Hum Centric Comput Inf Sci 9:29. https://doi.org/10.1186/s13673-019-0190-9
Partala J, Nguyen TH, Pirttikangas S (2020) Non-interactive zero-knowledge for blockchain: a survey. IEEE Access 8:227945–227961. https://doi.org/10.1109/ACCESS.2020.3046025
Rasheed AA, Mahapatra RN, Hamza-Lup FG (2020) Adaptive group-based zero knowledge proof-authentication protocol in vehicular ad hoc networks. IEEE Trans Intell Transp Syst 21(2):867–881. https://doi.org/10.1109/TITS.2019.2899321
Salim MM, Rathore S, Park JH (2020) Distributed denial of service attacks and its defenses in iot: a survey. J Supercomput 76(7):5320–5363. https://doi.org/10.1007/s11227-019-02945-z
Soewito B, Marcellinus Y (2020) Iot security system with modified zero knowledge proof algorithm for authentication. Egypt Inform J. https://doi.org/10.1016/j.eij.2020.10.001 (**ISSN 1110-8665**)
Vaidya B, Makrakis D, Mouftah HT (2016) Two-factor mutual authentication with key agreement in wireless sensor networks. Secur Commun Netw 9(2):171–183. https://doi.org/10.1002/sec.517
Walshe M, Epiphaniou G, Al-Khateeb H, Hammoudeh M, Katos V, Dehghantanha A (2019) Non-interactive zero knowledge proofs for the authentication of iot devices in reduced connectivity environments. Ad Hoc Netw 95:101988. https://doi.org/10.1016/j.adhoc.2019.101988 (**ISSN 1570-8705**)
Wei J, Phuong TVX, Yang G (2021) An efficient privacy preserving message authentication scheme for internet-of-things. IEEE Trans Ind Inform 17(1):617–626. https://doi.org/10.1109/TII.2020.2972623
World urbanization (2020) https://ourworldindata.org/urbanization#how-many-people-will-live-in-urban-areas-in-the-future. Accessed 3 Sept 2021
Yang X, Li W (2020) A zero-knowledge-proof-based digital identity management scheme in blockchain. Comput Secur 99:102050. https://doi.org/10.1016/j.cose.2020.102050 (**ISSN 0167-4048**)