A Quantum-Safe User Authentication Protocol for the Internet of Things

Kumar Sekhar Roy
and
Hemanta Kumar Kalita
North Eastern Hill University, Meghalaya, India

The forthcoming of the Internet of Things has opened the gates for numerous applications in several domains. Unfortunately, it also has brought along with it several security challenges. IoT devices, being compact in size, has several constraints. Therefore, it becomes a challenging task to define security protocols suitable for such constrained devices. Also, the significant strides made towards the development of Quantum computers pose a huge threat to traditionally used cryptosystems. It is known that a sufficiently large Quantum Computer running Shor's Algorithm can solve the integer factorization problem and the discrete logarithm problem. Thus, in our research, we provide an authentication protocol that utilizes Gate Way Node for high-end processing and suggest the usage of NTRU cryptosystem as the cipher suite. We suggest our authentication protocol in terms of cloud computing, as the number of IoT devices would be immense and cloud computing is better suited for processing and storing such large volumes of data. We also suggest the use of One Time Password, for adding another layer of security on top of the public-key cryptosystem. We analyze our authentication protocol and find that it is safe.

Keywords: Post Quantum Cryptography, Internet of Things, Authentication, One Time Password, Cloud Computing.

1. INTRODUCTION

Kevin Ashton in 1999 firstly used the term Internet of Things or IoT, Ashton et al. [2009]. Since then IoT has only grown in stature, popularity, and applicability. IoT devices mostly comprise small, constrained devices capable of sensing, actuating, etc. IoT devices have proved their applicability in several fields such as health care, logistics, smart grid, etc. To prove its applicability in such a wide range of fields, a wide range of IoT devices have been developed, thus creating a network of heterogeneous devices. These constrained devices have limited computational capabilities, storage, battery power, etc. Therefore, computationally intensive computations cannot be performed in such constrained devices. Thus, traditionally used cipher suites that were computationally intensive cannot be used in IoT. These feature provided researchers with new challenges as well as opportunities to design cipher suites that weren't computationally intensive and consumed minimal resources of the device. To aid such constrained devices from resource-consuming computations, researchers suggested the usage of Gateway Node (GWN) such as Zhu et al. [2010]. A GWN is relatively powerful then IoT devices, the GWN acts as an access point through which all data related to the IoT devices has to pass, these devices have better computational capabilities, more lifetime, better memory, etc. The GWN is placed in close proximity of the IoT devices or Edge devices. The IoT devices in the vicinity of GWN, sends the data to the GWN for processing and decision making, locally. These GWN, in turn, sends these data to the cloud server for further processing and storage. Therefore in our protocol, we consider three entities to be mutually authenticated, IoT devices, GWN, and the cloud server.

Cloud computing provides several benefits in term of IoT networks. As these networks would contain several million devices collectively, processing and storing such a huge repository of data would be a challenging task for a single centralized server. Therefore, cloud servers provide an efficient solution in handling such large amounts of data with its resource pool. In our protocol, we take the services provided by a cloud server to authenticate the GWN. The GWN would

Authentication type	Description	Example
Type 1	What you know	Password, passphrase, PIN, lock combination
Type 2	What you have	Smart card, token device
Type 3	What you are	Biometrics- fingerprint, palm print, retina/iris pattern, voice pattern

Table I: Types of Authentication

process the time-dependent computations and take decisions on its own and send that data to the cloud server, Time independent decisions would be taken by the cloud server itself.

Lamport [1981], firstly provided with the concept of using username and password for authentication. Since then, a large number of protocols and schemes have been developed that utilize username and password for authentication. For providing better security in our protocol we also utilize the services provided by One Time Password (OTP). OTP based security is also not a new concept and has been utilized exhaustively over the years.

Several research organizations around the globe have started to make giant strides towards the development of Quantum Computers recently. These computers differ from traditional electronic computers as they use qubits rather than bits. Bits have definitive value i.e. either 0 or 1, but qubits can be in a superposition of states at the same time. Therefore, computations that were not possible before, now seem possible. For decades, network security was based on cipher suites utilizing the integer factorization problem and the discrete logarithm problem. But, a sufficiently large Quantum computer running Shor's algorithm can easily solve these problems, Shor [1999]. Although the development of Quantum computers is still in its infancy, we have to be prepared for a day when such computers will arrive and render cryptosystems such as RSA and ECC vulnerable. Therefore in our protocol, we take the services of NTRU cryptosystem, which is a variant of Lattice-based Cryptography. Lattice-based cryptography is classified under Post-Quantum Cryptography (PQC), which are not vulnerable to algorithms running on Quantum Computers.

We organize our paper as follows, Section 2 provides the related work associated with our work. Section 3 addresses the system model taken into consideration. In Section 4, we provide the required preliminaries utilized in our protocol. In Section 5, we present our authentication protocol. We provide the attained results and analyze our protocol using various tools in Section 6. Finally, we conclude our paper in Section 7.

2. BACKGROUND AND RELATED WORK

Authentication can be classified into different types, as shown in Table I. It is always desirable to use at least two different types of authentication while authenticating an entity, also known as Two-factor authentication. Two-factor authentication is not a new concept and has been utilized by several researchers over the years such as Selvarajan [2007]. Therefore, for the purpose of our protocol we utilize password-based authentication using NTRU cryptosystem and to add an additional layer of security we add OTP based authentication on the user end. Initiated by Haller [1995], One Time Password has been exhaustively utilized for authentication, most of the times, on top of another authentication method and thus providing two-factor and relatively stronger authentication. Several different types of OTP generation methods have shown up since then, a noticeable few are Open Authentication (OATH) Challenge Response Algorithm (OCRA) by M'Raihi et al. [2011], Time based One Time Password (TOTP) by M'Raihi et al. [2011], Hash-based Message Authentication (HMAC) based One Time Password (HOTP) by M'Raihi et al. [2005], etc. Several other OTP based authentication systems exist based on RSA by Eldefrawy et al. [2010], ECC by Shivraj et al. [2015], Linear Secret Sharing (LSS) by Miceli [2011], etc. which provide strong cryptographic primitives. Although RSA and ECC are not included in Post-Quantum Cryptography, they provide strong security in traditional electronic computers.

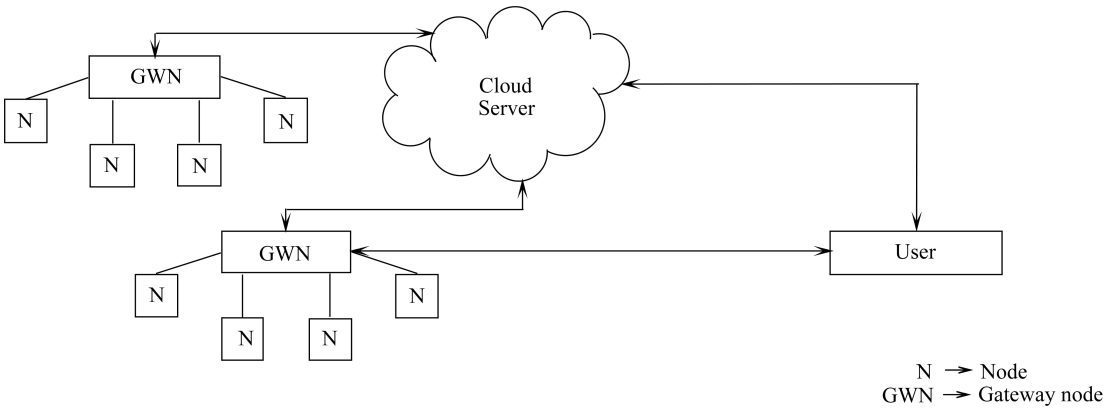

Figure 1. User-Cloud-IoT Architecture

The use of SMS (Short Messaging Service) for delivering OTP is not appreciated by NIST, as there were several reports of security breaches when SMS was used for delivering OTP [1]. The service of delivering OTP can be given to a trusted third-party application such as Auty, DUO, Google Authenticator, etc. Although, an intensive discussion of OTP is beyond the scope of this paper.

Over the recent past, several authentication protocols have been proposed by several researchers. Most of them claim to be lightweight by utilizing simple hash-based authentication like Chuang and Chen [2014], Yang and Yang [2010], Yoon and Yoo [2013], Amin et al. [2018]. Although, most of them were proven to be insecure against several attacks, such as user impersonation attack, anonymity, etc. by Chuang and Chen [2014] and Yang and Yang [2010]. A detailed analysis of resistance against several attacks is provided in Section 6.

3. SYSTEM MODEL

The architecture, taken into consideration to authenticate a user is depicted in Figure 1. We assume, for the purpose of our scheme that IoT devices are already authenticated with the GWN. There are several ways through which IoT devices can be authenticated with the GWN such as bootstrapping, physically unclonable functions, etc. We also assume that the user device is powerful enough to perform computations regarding public-key cryptography encryption/decryption. When a user wants to access an IoT device or network, it should register itself with the concerned cloud, which in turn will use the services provided by OTP to authenticate the user. Similarly, a GWN also would have to register itself to the cloud (without using OTP). The cloud server would act as a trusted third-party amidst the user and the GWN. Once a user or a GWN is registered in the cloud server, they only have to provide the username and password to access the services provided by the cloud. The GWN would consistently provide data gathered from the IoT devices to the cloud for processing and storage. An authenticated user would be able to access this data based upon its access rights. We also provide a user-GWN login phase so that, if required, the user can directly access time-sensitive data from the GWN.

4. PRELIMINARIES

4.1 Lattices

Lattices are geometric objects that have a strong association with patterns and have evolved to be a prime candidate in the post-quantum realm. Almost two decades ago lattices were suggested as the basis of cryptography due to hard mathematical problems associated with it.

[1] https://www.infosecurity-magazine.com/news/databasemisconfiguration-leaks/

Joseph Louis Lagrange and Carl Friedrich Gauss were among the first mathematicians to study lattices and the mathematical prospects associated with it. But, Ajtai [1996] showed the use of lattices as a cryptosystem in a seminal result. Lattice-based schemes have come to be proven as highly resistant to sub-exponential and quantum attacks. There are two problems with which cryptography associated with lattices are based on, the Shortest Vector Problem(SVP) and the Closest Vector Problem(CVP), both of which are proven to be NP-hard problems. A lattice L is a set of points in the n-dimensional Euclidean space Rn in real analysis, It has a strong periodicity property. Any basis of L can be defined as a set of vectors arranged in such a way that any element of L is uniquely represented as their linear grouping with integer coefficients. Each lattice has infinitely many different bases when the value of n is at least 2. All lattices over Rn have infinitely many elements, whereas in cryptography entities such as the ciphertext, public-key, and private key must be chosen from a finite space (bit strings of some fixed length). The algorithms concerning Lattice-based Cryptography are mentioned in the following section.

4.2 NTRU

The first version of NTRU was developed by mathematicians Hoffstein et al. [1998], which was called NTRU. In our survey, we come across the latest variant of NTRU i.e. NTRU Prime, proposed by Bernstein et al. [2017]. In their work, they prove that their algorithm is stronger than the original NTRU by creating stronger algebraic structure. The algorithm is as follows.

4.2.1 *Key Generation.* The receiver generates a public-key as follows:

(1) Generate a uniform random small element $g \in R$. Repeat this step until g is invertible in R=3.
(2) Generate a uniform random t-small element $f \in R$. (Note that f is nonzero and hence invertible in R/q, since t 1.)
(3) Compute $h = g/(3f)$ in R/q. (By assumption q is a prime larger than 3, so 3 is invertible in R/q, so $3f$ is invertible in R/q.)
(4) Encode h as a string h'. The public-key is h'.
(5) Save the following secrets: f in R; and $1/g$ in $R/3$.

4.2.2 *Encryption.* The sender generates a ciphertext as follows:

(1) Decode the public-key h', obtaining $h \in R$=q.
(2) Generate a uniform random t-small element $r \in R$.
(3) Compute $hr \in R$=q.
(4) Round each coefficient of hr, viewed as an integer between -(q - 1)/2 and (q - 1)/2, to the nearest multiple of 3, producing $c \in R$. (If $q \in 1 + 3Z$, as in the case study $q = 9829$, then each coefficient of c is in {-(q - 1)/2, . . . , -6, -3, 0, 3, 6, . . . , (q 1)/2}. If $q \in 2 + 3Z$ then each coefficient of c is in {-(q + 1)/2, . . . , -6, -3, 0, 3, 6, . . . , (q + 1)/2}.)
(5) Encode c as a string c'.
(6) Hash r, obtaining a left half C ("key confirmation") and a right half K.
(7) The cipher-text is the concatenation Cc'. The session key is K.

4.2.3 *Decryption:.* The receiver decapsulates a cipher-text Cc' as follows:

(1) Decode c', obtaining $c \in R$.
(2) Multiply by $3f$ in R/q.
(3) View each coefficient of $3fc$ in R/q as an integer between -(q - 1)=2 and (q - 1)/2, and then reduce modulo 3, obtaining a polynomial e in $R/3$.
(4) Multiply by $1/g$ in $R/3$.

(5) Lift e/g in $R/3$ to a small polynomial $r' \in R$.
(6) Compute c', C', K' from r' as in encryption.
(7) If r' is t-small, $c' = c$, and $C' = C$, then output K'. Otherwise, output False.

If Cc' is a legitimate cipher-text then c is obtained by rounding the coefficients of hr to the nearest multiples of 3; i.e., $c = m + hr$ in $R=q$, where m is small.
Howgrave-Graham [2007] in his research titled "A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU" performed an attack which included lattice reduction at first and then performed meet in the middle attack. The attack methodology performed faster than odlyzko's attack, Howgrave-Graham et al. [2003]. The author assumed this attack as an improved result of an attack performed byCoppersmith and Shamir [1997] which used only lattice reduction presented in 1996. The author suggests that for NTRU to be secure the private vector needed to be thickened or use a trinary vector which would make the meet in the middle attack substantially harder to perform without increasing the parameter N by much. The NTRU prime algorithm although was published much later (2016) then these proposed attacks, not enough evidence exists that the proposed attacks would work on NTRU prime as well.

5. PROPOSED PROTOCOL

Our proposed protocol would include a total of five phases ie. user Registration Phase, Gateway Node Registration Phase, user Login Phase, GWN accessing phase, and user-GWN login phase. For the purpose of our protocol we assume that the GWN is a powerful device which is capable of performing complex operations such as public-key encryption, we also assume that IoT devices are pre-authenticated with the GWN. We also assume that the user device is also capable of performing public-key encryption/decryption. The five phases of our protocol are as follows.

Symbol	Description
S_i	Cloud Server
U_{id}	Users Identity
nonce	A unique Pseudo random number
ACK	Acknowledgment
PUB_{S_i}	S_i's Public Key
pwd_i	i_{th} users password
Ph no.	users Phone Number
OTP	One Time Password
FSP	Fail Safe Password
GWN	Gateway Node
PUB_{GWN_i}	i_{th} GWN's Public key
C	Password between GWN_i and S_i
PRN	Pseudo Random Number

Table II: Symbols and Description

5.1 User Registration Phase

This is the first phase of our authentication scheme, where a user tries to register himself on the network. The user initiates this phase by sending a registration request involving its' identity along with a nonce and a time-stamp, encrypted by the cloud servers public-key pub_{S_i}. The nonce and time-stamp is essential to avoid any kind of replay attack. The cloud server upon receiving this message extracts the User ID, time-stamp and nonce. The cloud server then records the nonce and time-stamp. In the next step, the cloud server sends back an acknowledgment (ACK = nonce + 1) along with a request to provide the users' password and a phone number, encrypted by users' public-key. The User then verifies the acknowledgment, hence achieving mutual authentication. The User then replies to the cloud server with its' password and phone number, encrypted by pub_{s_i}. The cloud server extracts the password and stores it in it's encrypted

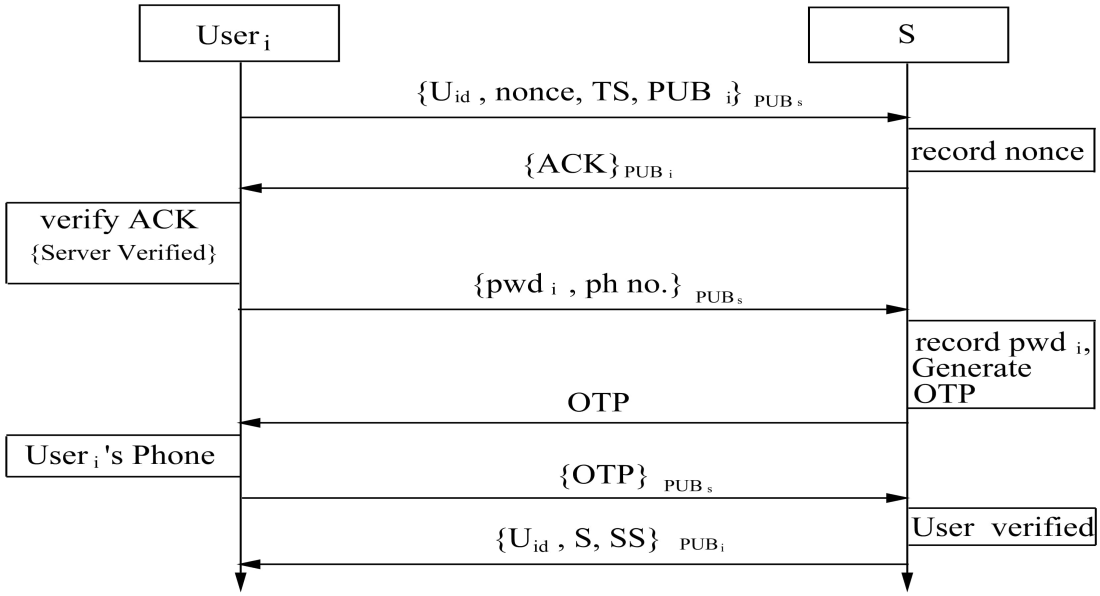

Figure 2. User Registration Phase

form to avoid privileged insider attack. The cloud server then sends the OTP over to the user through a trusted third-party application (and not through SMS). The user then replies with the same OTP in its' encrypted form to the cloud server, hence achieving two factor and mutual authentication. The cloud server then replies with a shared secret, which would be common for all the entities connected to the cloud. The shared secret would be utilized for authentication among entities. The shared secret is assumed to be a pseudo random number. The whole user registration process is shown in Figure 2.

5.2 Gateway Node Registration Phase

This phase marks the registration of a gateway node into the cloud server. The GWN initializes its' registration by sending its' identity, time-stamp, the cloud servers identity, a nonce and its' public-key, encrypted by the server's public-key. The server then records the nonce, time-stamp and GWN's identity in its' encrypted form, to avoid privileged insider attack. The cloud server then replies with an acknowledgment along with an assigned password, shared secret, time-stamp and a nonce, encrypted by the GWN's public-key. Nonces and time-stamps are used in both instances to avoid any kind of replay attack. The GWN, in turn, records the password for future use and stores the nonce. The GWN then replies the cloud server with the password and an acknowledgment. Thus, achieving mutual authentication for both parties, by verifying the password and recording the nonce.

5.3 User Login Phase

This phase marks the actual logging in and authentication of the user into the cloud server. We assume that the user has already been through the registration phase and is now registered in the cloud server. The user sends the cloud server U_{id},pwd_{id} time-stamp and nonce as a login request, encrypted by the cloud servers public-key. The cloud server upon receiving this encrypted message decrypts the message to extract U_{id} and pwd_{id} and hence verify them with the record in its database. The record is stored in the cloud servers database in it's encrypted form and therefore has to be decrypted for comparison. Once the user's credentials have been verified the cloud server sends an OTP through a trusted third-party application. The user upon receiving this OTP, resends the OTP in it's encrypted form. Thus establishing mutual authentication. The

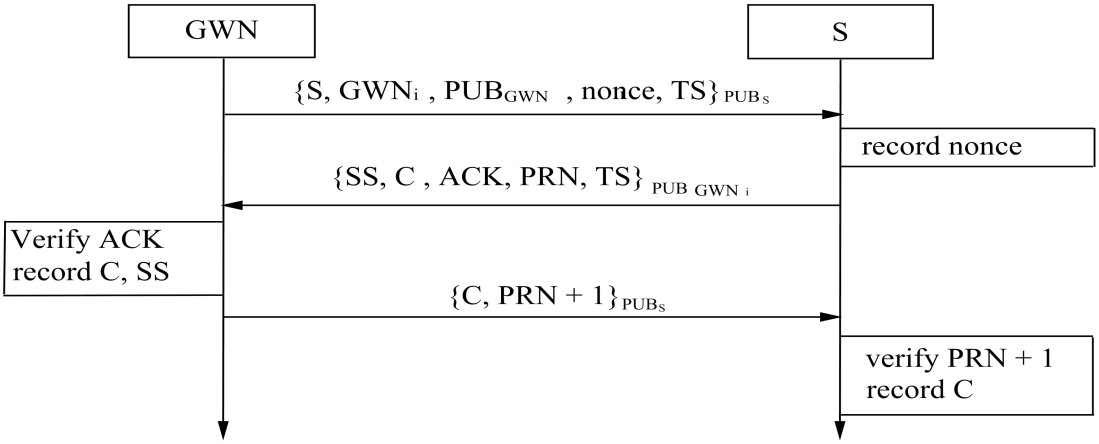

Figure 3. Gateway Node Registration Phase

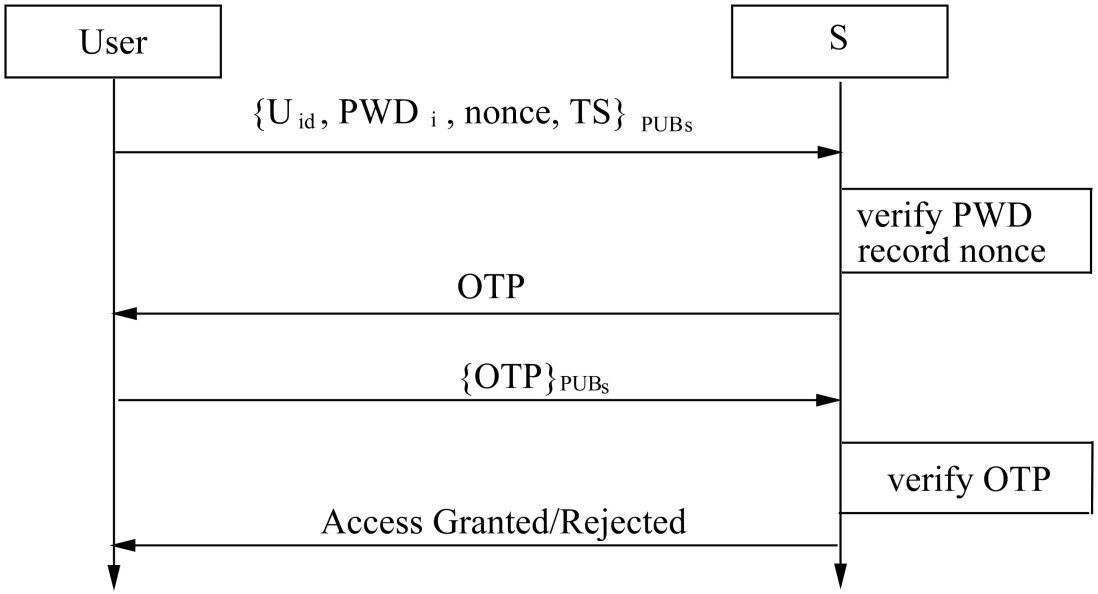

Figure 4. User login

cloud server then grants/rejects the access request. It is questionable that, an adversary might try to raise a resource exhaustion attack. To raise a resource exhaustion attack, an attacker will try to send the same message over and over in quick succession. Therefore we suggest that an incoming message be checked for the same ciphertext received in the last few seconds. If found same, the IP address should be blocked from the server.

5.4 GWN accessing phase

The cloud server, whenever required to access the data from the GWN in real time (for the user) has to establish mutual authentication and therefore provides an access request by sending C, time-stamp and a nonce to the GWN. The GWN verifies password C, which is stored in its' encrypted form to avoid privileged insider attack. If C is correct, the access is granted to the cloud server by sending an acknowledgment.

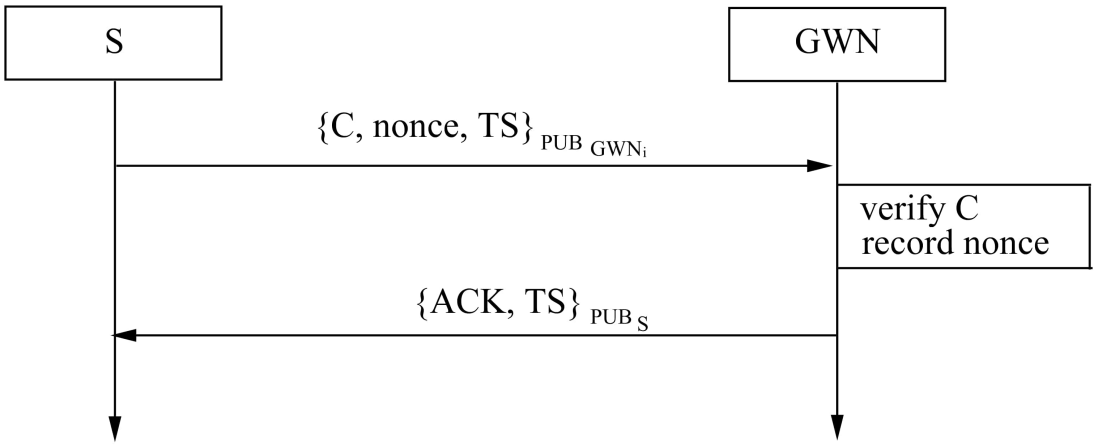

Figure 5. Accessing GWN

5.5 User-GWN login phase

This phase marks the logging in of a user to the GWN directly. A situation might arise when a centralized server might fail or a user needs time-sensitive data, in such situations, it is always better for the user to access the GWN directly. Before a user can access the GWN, the user has to be registered in the GWN as well. Therefore the user simply sends its credentials along with a nonce and a time-stamp encrypted by the cloud servers public-key to the GWN, the GWN, in turn, send the exact message to the cloud server, encrypted by its own public. We assume here that the GWN is already authenticated with the cloud server. The cloud server then verifies the users' identity and sends back an acknowledgment along with the users' credentials including the relevant password. The GWN records these credentials and sends a reply to the user containing the concerned acknowledgment and the shared secret. The user, upon receiving this message, verifies the shared secret to confirm the GWN's identity. The user then replies the GWN with its password and identification for registration as well as verification. The GWN verifies the password provided by the user along with the one provided by the cloud server. The GWN then asks the cloud server to provide an OTP, to implement two-factor authentication. The cloud server sends an OTP to the GWN encrypted with PUB_{GWN_i}. The GWN simply forwards the OTP, along with an acknowledgment encrypted with PUB_i. The user then replies with the OTP, hence creating mutual authentication.

6. RESULT AND ANALYSIS

Symbol	Description
–	no computational cost
n	no. of users
C_f	cost of executing the fuzzy extractor
C_h	cost of executing one way hash function
C_s	cost of executing symmetric encrypt/decrypt
C_a	cost of executing asymmetric encrypt/decrypt
C_{exp}	cost of executing an exponential operation
C_{OTP}	cost of calculating the OTP

Table III: Symbols

We use two-factor authentication to ensure that the authentication process between a user and the cloud server is secure. We utilize Public-key encryption based on Lattice-based Cryptography

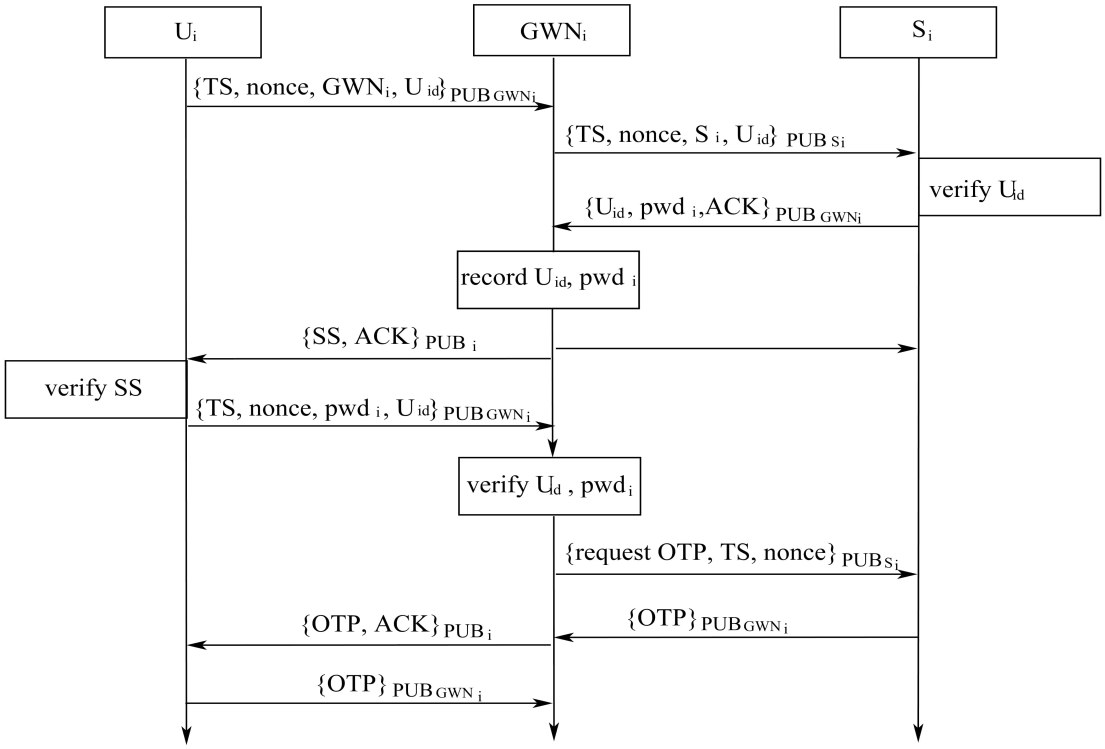

Figure 6. User-GWN Login Phase

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/ureg.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 38 nodes
  depth: 4 plies
```

(a) OFMC

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/myproto1.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 10 states
  Reachable  : 4 states
  Translation: 0.01 seconds
  Computation: 0.00 seconds
```

(b) Cl-AtSe

Figure 7. Verification of User Registration Phase

as suggested by NIST to ensure that the Cipher suite is unbreakable as of now through traditional or quantum computers. We also use Time based OTP (TOTP) based on RFC 4226 and RFC 6238 using HMAC SHA512 to ensure that the OTP generation is safe. Although as per NIST Grassi et al. [2017] we suggest the use of internet based apps rather than GSM to transfer OTP. We use nonce and time-stamp in most of the instances in our protocol to avoid a replay attack. We also ensure in our protocol that there is mutual authentication between the two parties involved. We ensure that the password is stored in the cloud server in its encrypted form so that even if

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/GWNreg.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.02s
  visitedNodes: 38 nodes
  depth: 4 plies
```

(a) OFMC

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/GWNreg.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 2 states
  Reachable  : 0 states
  Translation: 0.01 seconds
  Computation: 0.00 seconds
```

(b) Cl-AtSe

Figure 8. Verification of GWN Registration Phase

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/userlogin.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.02s
  visitedNodes: 18 nodes
  depth: 4 plies
```

(a) OFMC

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/userlogin.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 2 states
  Reachable  : 0 states
  Translation: 0.01 seconds
  Computation: 0.00 seconds
```

(b) Cl-AtSe

Figure 9. Verification of User Login Phase

		Chuang et al.	Yang et al.	Yoon et al	Amin et al.	Proposed
Registration	user	C_h	-	C_h	$2C_h$	$3C_a$
	Server	-	-	-	-	-
	RC	n(2 C_h)	n($3C_h + C_a + C_f$)	(n+m)C_h	$3C_h$	C_{OTP}+$3C_a$
Login	user	$4C_h$	$4C_h + C_a + C_f$	$2C_h + C_a$	$6C_h$	C_a
	Server	-	-	-	-	$C_{OTP} + C_a$
Authentication	user	$5C_h$	$C_h + C_a$	$3C_h + C_a$	$3C_h$	C_a
	Server	$8C_h$	$3C_h$+$2C_a$	$5C_h$+$2C_a$	C_h+$3C_h$	$C_{OTP} + C_a$
	RC	-	-	$7C_h$	$10C_h$	-

Table IV: Comparison of Computation cost

the server is accessed the password is not disclosed. A brief comparison of the execution time of different cryptosystems is given in Figure 11. For the analysis of the computational cost, we use the symbols provided in Table III. A brief comparison of computational cost is given in Table IV.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/GWNlogin.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.02s
  visitedNodes: 18 nodes
  depth: 4 plies
```

(a) OFMC

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/GWNlogin.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.00 seconds
  Computation: 0.00 seconds
```

(b) Cl-AtSe

Figure 10. Verification of GWN Login Phase

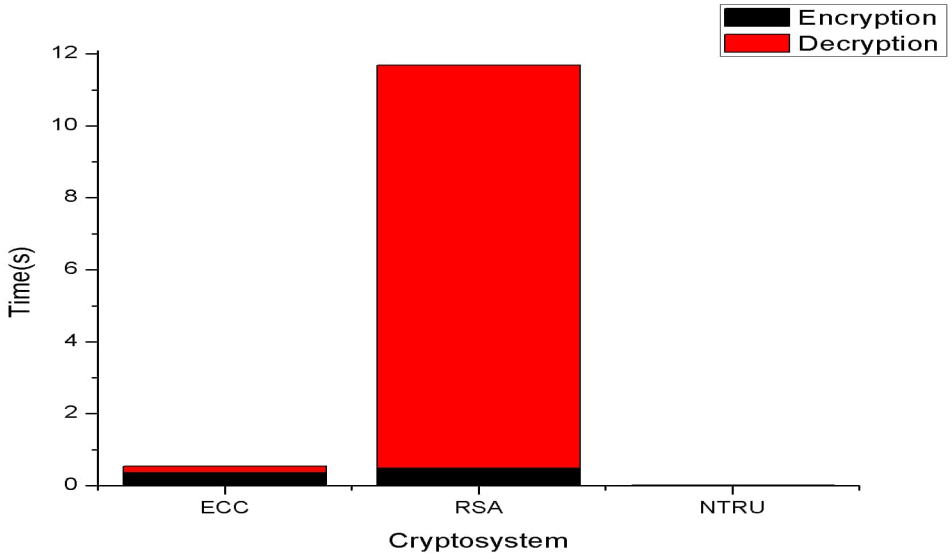

Figure 11. Execution Time comparison

Attacks	Xue.	Wang.	He.	Chuang.	Amin.	Proposed
Offline password guessing	N	N	Y	Y	Y	Y
user impersonation	N	N	Y	N	Y	Y
Resist replay	Y	N	N	Y	Y	Y
Mutual Authentication	N	N	Y	Y	Y	Y
Session Key disclosure	N	N	N	Y	Y	Y
Fail safe	N	N	N	N	N	Y
Quantum Computer	-	-	-	-	-	Y

Table V: Attack Comparison

Roughly the time complexity of different operations follows the order $C_{exp}>>C_a>C_s>C_h$.

6.1 Security Analysis

We compare our authentication protocol with several other protocols in terms of resistance against several attacks, a brief comparison is made in Table V.

(1) **Offline password guessing:** To avoid any sort of brute force attack, we suggest the users to provide strong passwords including numbers and punctuation marks. We use OTP to provide an additional layer of security, so that even though an adversary guesses the password correctly, he/she cannot access the network.
(2) **User impersonation:** By gaining some sensitive data about a registered user, an attacker can impersonate that user. The authentication scheme cannot allow such an attack therefore we use OTP to provide a second layer of security to avoid such an attack.
(3) **Resist replay:** An adversary might try to resend a previously sent message to impersonate a registered user to gain access to the network which the attacker had previously captured. To avoid such an attack we utilize nonces and time-stamps in most of the instances.
(4) **Mutual Authentication:** It is when all the communicating parties can authenticate each other. Each party involved in the communication process authenticates the other and makes sure that the other communicating party is legitimate. Therefore, we make sure that all the parties involved in the communication process are mutually authenticated.
(5) **Privileged insider attack:** When a user with enough privileges in the network collects sensitive data about a legitimate user and tries to impersonate that user in another network, it is known as privileged insider attack. To avoid such attack, we suggest that the password and user ID to be stored in its encrypted form.
(6) **Man in the Middle attack:** It is a classical attack where an eavesdropper acts as a sender to the receiver and vice versa. To avoid such an attack we verify the resistance of our protocol against this attack by using AVISPA tool and hence prove that our protocol is safe.
(7) **Resource Exhaustion attack:** It is when an adversary sends multiple requests to the server demanding a server response. The adversary tries to send the same message repeatedly in quick succession. The server would check for the ciphertext with previously received ciphertext within a few seconds. If found same, the user IP address will be blocked.

We analyze our protocol with Automated Validation of Internet Security Protocols and Applications (AVISPA) by Armando et al. [2005], through High-Level Protocol Specification Language (HLPSL) by Von Oheimb [2005]. AVISPA tool is recognized to be an effective tool for verification of security protocols, we show that our protocol is safe in both OFMC and Cl-AtSe modes of the tool. Figure 7 to Figure 10 shows that our protocol is safe in the AVISPA tool.

7. CONCLUSION

Authentication of IoT devices poses a challenge as IoT devices have numerous constraints based on computation capabilities, lifetime, power consumption, etc. Therefore, in our research, we propose the usage of Gateway Nodes which have higher computational capabilities and thus can perform complex operations. We propose the use of cloud server as the central entity responsible for looking into the proceedings in the network, including Authentication. We also provide a user-GWN login phase so that the user can access any time-sensitive data directly from the GWN, if such a situation ever arises. For the purpose of proper and secure authentication, we use NTRU-prime. This algorithm falls under Post-Quantum Cryptography and therefore are secure against attacks through Quantum Computers. To provide better security we employ two-factor authentication by using One Time Password. We verify our protocol using AVISPA tool to ensure that our protocol is secure. Future endeavors would include more focus on designing and implementing an efficient Fail Safe mechanism so that a legitimate user has access to a device at all time.

References

Ajtai, M. 1996. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM, 99–108.

Amin, R., Kumar, N., Biswas, G., Iqbal, R., and Chang, V. 2018. A light weight authentication protocol for iot-enabled devices in distributed cloud computing environment. *Future Generation Computer Systems 78*, 1005–1019.

Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuéllar, J., Drielsma, P. H., Héam, P.-C., Kouchnarenko, O., Mantovani, J., et al. 2005. The avispa tool for the automated validation of internet security protocols and applications. In *International conference on computer aided verification*. Springer, 281–285.

Ashton, K. et al. 2009. That internet of things thing. *RFID journal 22,* 7, 97–114.

Bernstein, D. J., Chuengsatiansup, C., Lange, T., and van Vredendaal, C. 2017. Ntru prime: reducing attack surface at low cost. In *International Conference on Selected Areas in Cryptography*. Springer, 235–260.

Chuang, M.-C. and Chen, M. C. 2014. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. *Expert Systems with Applications 41,* 4, 1411–1418.

Coppersmith, D. and Shamir, A. 1997. Lattice attacks on ntru. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 52–61.

Eldefrawy, M. H., Khan, M. K., and Alghathbar, K. 2010. One-time password system with infinite nested hash chains. In *Security Technology, Disaster Recovery and Business Continuity*. Springer, 161–170.

Grassi, P. A., Perlner, R. A., Newton, E. M., Regenscheid, A. R., Burr, W. E., Richer, J. P., Lefkovitz, N. B., Danker, J. M., and Theofanos, M. F. 2017. Digital identity guidelines: Authentication and lifecycle management [including updates as of 12-01-2017]. Tech. rep.

Haller, N. 1995. The s/key one-time password system.

Hoffstein, J., Pipher, J., and Silverman, J. H. 1998. Ntru: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium*. Springer, 267–288.

Howgrave-Graham, N. 2007. A hybrid lattice-reduction and meet-in-the-middle attack against ntru. In *Annual International Cryptology Conference*. Springer, 150–169.

Howgrave-Graham, N., Silverman, J. H., and Whyte, W. 2003. A meet-in-the-middle attack on an ntru private key. Tech. rep., Technical report, NTRU Cryptosystems, June 2003. Report.

Lamport, L. 1981. Password authentication with insecure communication. *Communications of the ACM 24,* 11, 770–772.

Miceli, C. 2011. One time password scheme via secret sharing techniques.

M'Raihi, D., Bellare, M., Hoornaert, F., Naccache, D., and Ranen, O. 2005. Hotp: An hmac-based one-time password algorithm. Tech. rep.

M'Raihi, D., Machani, S., Pei, M., and Rydell, J. 2011. Totp: Time-based one-time password algorithm. Tech. rep.

M'Raihi, D., Rydell, J., Bajaj, S., Machani, S., and Naccache, D. 2011. Ocra: Oath challenge-response algorithm. Tech. rep.

Selvarajan, B. 2007. Simple two-factor authentication. US Patent App. 11/267,148.

Shivraj, V., Rajan, M., Singh, M., and Balamuralidhar, P. 2015. One time password authentication scheme based on elliptic curves for internet of things (iot). In *2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*. IEEE, 1–6.

Shor, P. W. 1999. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review 41,* 2, 303–332.

Von Oheimb, D. 2005. The high-level protocol specification language hlpsl developed in the eu project avispa. In *Proceedings of APPSEM 2005 workshop*. 1–17.

Yang, D. and Yang, B. 2010. A biometric password-based multi-server authentication scheme with smart card. In *2010 International Conference On Computer Design and Applications*. Vol. 5. IEEE, V5–554.

Yoon, E.-J. and Yoo, K.-Y. 2013. Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. *The Journal of supercomputing 63,* 1, 235–255.

Zhu, Q., Wang, R., Chen, Q., Liu, Y., and Qin, W. 2010. Iot gateway: Bridgingwireless sensor networks into internet of things. In *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*. Ieee, 347–352.

Dr. Hemanta Kumar Kalita received his B.E. degree in 1997 in Computer Science and Engineering from Dibrugarh University, India and his M.Tech degree in Information Technology from Tezpur university, India in 2002. He recieved his Ph.D degree from Jadavpur University in 2013. He is currently working as an associate Professor in Department of Information Technology, North Eastern Hill University. His research interest includes Big Data Analysis, Adhoc Network Security, Performance Engineering, Spatial Data Mining, and Artificial Intelligence. He was the recipient of FOSS India award in 2008.

Mr. Kumar Sekhar Roy received B.Tech and M.tech degree on Information Technology from North Eastern Hill University, India, on July 2013 and 2015 respectively. He is currently a Ph.D. scholar in North Eastern Hill University in India. His research focus includes wireless channel security, steganography, post quantum cryptography and network security for constrained devices.