Wireless Sensor Networks IKE Phase One Negotiation Approach based on Multivariate Quadratic Quasi-groups

Yassine Essadraoui and Mohamed Dafir Ech-cherif El Kettani
Information Security Research Team ISeRT, Mohammed V University in Rabat, Rabat, Morocco

Keywords: Internet Key Exchange, Wireless Sensor Network, Multivariate Quadratic Quasigroups, Internet of Things.

Abstract: Key management is one of the biggest problems in IoT security. The Internet Key Exchange (IKE) protocol is well known as a secure key exchange protocol to establish secure channels between two hosts. However, IKE uses RSA as public key cryptography algorithm that is based on Diffie-Hellman (DH) key exchange which is very heavy, in term of arithmetic operations, for very constrained resources devices such as the case for WSNs. In this paper, we propose to use Multivariate Quadratic Quasigroups (MQQ) to negotiate and share a secret key between two sensors. Phase 1 of the IKE protocol is supposed, by the proposal in this paper, to use MQQ instead of Diffie-Hellman (DH) key exchange.

1 INTRODUCTION

Internet is the biggest network used for different goals and offers different services. Each service requires some level of security (integrity, authorization, confidentiality, ...) and to achieve this goal different algorithms and protocols have been proposed and used. The most used protocol is IPsec. IPsec uses two protocols to provide traffic security services, Authentication Header (AH) and Encapsulating Security Payload (ESP)(R.~Atkinson, 1995). The Internet Key Exchange (IKE) protocol belongs to IPsec protocol suite and is used to establish secure channels between hosts implementing IPsec. IKE prepares what's called security associations (SA) that will be used, by IPsec protocol, for transmitting data securely between nodes involved in a communication (Hallqvist and Keromytis, 2000). In symmetric cryptography, two peers share a secret key and use it for encryption and decryption. This kind of cryptography is secure as well as the secret key is protected and kept secret between the two parties involved in the communication. As we can see peers need to exchange the secret key securely, with big systems where many communications should be established it's not easy to keep and exchange securely all shared keys. The Internet Key Exchange (IKE) protocol is used to produce, independently and dynamically, the same key in each communicating peer. IKE proceeds by authenticating both sides and negotiating encryption algorithms. The result of an IKE negotiation is a Security Association.

Internet Key Exchange relies on Diffie-Hellman (DH) algorithm to exchange the shared secret key between hosts. DH builds the shared secret key on each side without having to exchange the secret key between peers. In our context of wireless sensor networks (WSN) using IKE without modification won't be practical viewing WSN's constraints (limited calculation power, memory, energy, ...) also the Diffie-Hellman algorithm is a little heavy, in term of arithmetic operations needed to share the secret key between peers in communication, to be used in a sensor. Our approach is to use Multivariate Quadratic Quasi-groups to exchange the shared key and to do mutual authentication of communicating nodes. Our work will be based on a previous work (Essadraoui and Dafir Ech-cherif El Kettani, 2015) which proposed an authentication approach based on Multivariate Quadratic Quasi-groups (MQQ). In the proposed approach nodes do mutual authentication and share a secret key that could be used for symmetric encryption and decryption.

MQQ has emerged as an alternative to classical cryptography schemes and was seen that MQQ is faster in encryption and decryption. In (El-Hadedy et al., 2008) authors have done an implementation that is 10,000 times faster in decryption and 17,000 times faster in encryption than RSA. In (Quirino and Moreno, 2013) authors have found that in ARM

Essadraoui, Y. and Kettani, M.
Wireless Sensor Networks IKE Phase One Negotiation Approach based on Multivariate Quadratic Quasi-groups.
DOI: 10.5220/0006241902240231
In *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017)*, pages 224-231
ISBN: 978-989-758-245-5

platform MQQ is faster and uses less amount of memory compared to ECC (Zhu et al., 2008) and RSA(Breu et al., 2008).

The remaining of this paper will be organized as follow, section 2 gives an overview of Internet exchange protocol, in section 3 we introduce multivariate quadratic quasi-groups, in section 4 an overview of wireless sensor networks is given, in section 5 we present the proposed lightweight version of IKE for WSNs, in section 6 we give a security analysis of the proposed protocol and in section 7 we give a conclusion.

2 OVERVIEW OF INTERNET KEY EXCHANGE (IKE)

Internet Key Exchange protocol (IKE) (Kivinen and Snyder, 2015) is used in conjunction with IPsec to dynamically and automatically create security associations (SA), especially for a large scale deployment IKE becomes very useful. IKE is based on the Internet Security Association and Key Management Protocol (ISAKMP), Oakley key management protocol and SKEME key management protocol. The IKE is two phases protocol. Phase One starts the process to establish SA between two peers by authenticating all peers, then creating a Diffie-Hellman key and agreeing on methods for phase two of IKE. The creation of Diffie-Hellman key means that peers will have a shared secret key. All exchanges and agreements preformed during phase one must be done securely. Figure 1 depicts IKE phase one process.

Phase Two consists of using all algorithms, methods and keys that have been chosen by agreements performed in phase one. During phase two a new security association is going to be built and prepared for IPsec, in fact IPsec security association is an agreement on methods and keys that will be used by IPsec. Figure 2 depicts IKE phase two process.

At the end we can say that IKE phase one prepares methods and keys for IKE phase two which itself prepares methods and keys for IPsec.

3 OVERVIEW OF MULTIVARIATE QUADRATIC QUASI-GROUPS (MQQ)

Multivariate Quadratic Quasi-groups (MQQ) is a system of m multivariate quadratic equations with n

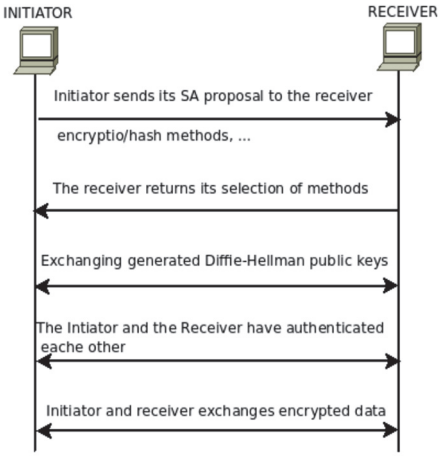

Figure 1: IKE phase one process.

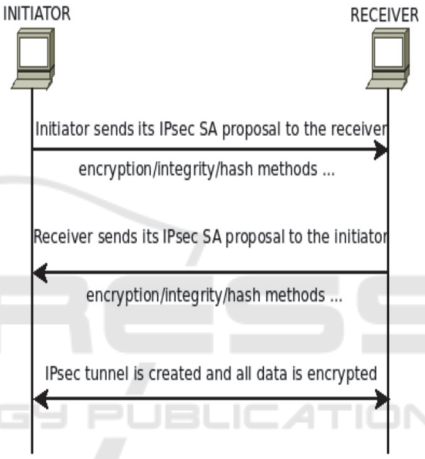

Figure 2: IKE phase two process.

variables over a finite field F. This system of multivariate quadratic equations is easy to evaluate for some given values from F but it's very difficult to find its inverse and if a brute force attack is launched against this system the time needed will be exponential depending on the number of variables. A MQQ system is considered secure because it doesn't rely on the difficulty of calculation of the problem of factorization and the discrete logarithms. STS (Shamir, 1994), TTM (Moh, 1999), HFE (Patarin, 1996), UOV (Kipnis et al., 1999) are examples of systems that have been proposed based on multivariate quadratic (MQ) problem but most of these schemes have been broken. In 2008 a new approach called Multivariate Quadratic Quasi-groups (MQQ)(Gligoroski et al., 2008a)(Gligoroski et al., 2008b) have been proposed by Gligoroski et al based on the theory of quasi-groups.

Multivariate Quadratic cryptography consists of solving systems of multivariate quadratic equations

over a finite field.

Let's consider a system P of m multivariate quadratic polynomial equations with n variables over a finite field F.

$$P = \begin{cases} y_1 = p_1(x_1, \ldots, x_n) \\ y_2 = p_2(x_1, \ldots, x_n) \\ \cdot \\ \cdot \\ y_m = p_m(x_1, \ldots, x_n) \end{cases} \quad (1)$$

where y_i belongs to F. Each p_k is a polynomial of degree two over F of the form:

$$p_k(x_1, \ldots, x_n) = \sum_{1 \le i \le j \le n} a_{ij}^{(k)} x_i x_j + \sum_{i=1}^{n} b_i^{(k)} x_i + c^{(k)} \quad (2)$$

where $a_{ij}^{(k)}$, $b_i^{(k)}$, $c^{(k)}$ belong to F and $1 \le k \le m$. The coefficients a^k, b^k and c^k are called the quadratic, linear and constant parts of the polynomial p_k, respectively. The challenge is to find a solution x to the system P. The MQ-problem is a NP-complete problem (Patarin et al., 1997).

The m multivariate quadratic equations of P system are composing the public key, as we mentioned before the MQ problem is NP-complete so we need an easily invertible (Ding and Yang, 2009) trapdoor function to be defined into polynomial equations.

The construction of P, the public key, is done through a composition of three functions S, P' and T such as $P = T \circ P' \circ S$ where $S: F^n \rightarrow F^n$ and $T: F^m \rightarrow F^m$ are linear or affine transformations and are easily invertible and they are used to hide the function P' which is a quadratic function such as $P': F^n \rightarrow F^m$. P' should be easily invertible. The private key is the triple (S, P', T). Figure 3 illustrates how to use the MQ public and private key in cryptography.

In this paper we are interested in MQ-schemes that are based on quasi-groups which means that the central map P' will be constructed using quasi-groups. Any multiplication group is a quasi-group. In particular, an abelian group is where multiplication is commutative and associative. However, quasi-group multiplications are not required to be associative. It is in this sense that quasi-groups are considered to be "non associative groups". Finite quasi-groups are characterized as having bordered Latin squares for their multiplication tables. (Smith, 2007)

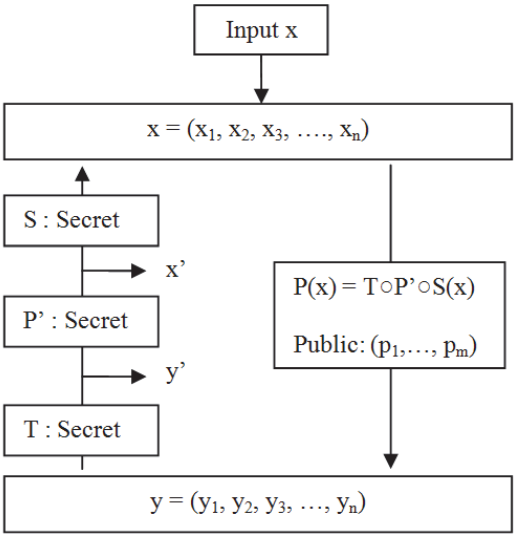

Figure 3: Public key encryption based on MQ system.

As an example of Quasi-groups we mention Latin squares. A Latin square is an n × n square filled with n different symbols, in such a way that each symbol occurs exactly once in each column and in each row. Each Latin square is a multiplication table of a quasi-group. Tables in Figure 4 list a Latin square and a corresponding finite quasi-group.

1	3	4	2	5
3	2	1	5	4
4	1	5	3	2
5	4	2	1	3
2	5	3	4	1

*	1	2	3	4	5
1	1	3	4	2	5
2	3	2	1	5	4
3	4	1	5	3	2
4	5	4	2	1	3
5	2	5	3	4	1

Figure 4: A Latin square and its corresponding Quasi-group.

A quasigroup (Q, *) of order 2^d is called Multivariate Quadratic Quasigroup (MQQ) of type $Quad_{d-k}Lin_k$ if exactly d - k of the polynomials f_i are of degree 2 (i.e., are quadratic) and k of them are of degree 1 (i.e., are linear), where $0 \le k < d$. (Gligoroski et al., 2008b)

Q is a quasi-group such that a1,a2, a3,... an belong to it, then the encryption operation, which is defined over the defined elements, maps those elements to another vector b_1, b_2, b_3,...b_n such that the elements of the resultant vector also belong to the same quasi-group. Markovski and Dimitrova (Dimitrova and Markovski, 2004) show that mapping of an incoming stream of data depends on the initial multiplier element. The mathematical equation used for encryption (Dimitrova and Markovski, 2004) is defined by:

$$E_a(a_1, a_2, a_3, \ldots, a_n) = b_1, b_2, b_3, \ldots, b_n \quad (3)$$

where the output sequence is defined by:

$$b_1 = a * a_1 \quad (4)$$

where i increments from 2 to the number of elements that have to be encrypted, and a is the hidden key.

A MQQ system looks like the following system:

$$\begin{aligned} y_1 &= x_{12} + x_1x_3 + x_2x_3 + x_2x_4 + x_{32} + x_3x_4 + 1 \\ y_2 &= x_{12} + x_1x_2 + x_1x_3 + x_{22} + x_2x_4 + x_{32} + x_{42} + 1 \\ y_3 &= x_1x_2 + x_1x_4 + x_2x_3 + x_2x_4 + x_{32} + x_3x_4 + x_{42} \\ y_4 &= x_1x_2 + x_1x_3 + x_{22} + x_2x_3 + x_3x_4 \end{aligned} \quad (5)$$

with x, y belong to $(F_q)^n$ and F is a finite field. Solving this system is NP-complete over any field.

The system of equations must include a hidden trapdoor function P' and two other functions S and T. P' is a quadratic form, that is easily invertible [12]. T and S are affine forms that are very hard to guess or calculate. The public key P is the composition of T, P' and S, we hide the trapdoor function P' by composition with T and S as follows:

$$P = T \circ P' \circ S \quad (6)$$

Public key must be a one-way function. The private key is the secret matrices T, P', S.

4 OVERVIEW OF WIRELESS SENSOR NETWORKS (WSN)

A Wireless Sensor Network (WSN) is a network of nodes called sensors. Sensors are resource constrained devices which have as main role sensing data and send it to other sensors or a more powerful Base Station (BS). The BS will gather all sent data from sensors and process it. All sensors in a WSN communicate through a wireless connectivity, these sensors have, in general, as mission to sense data and send it to a Base Station (BS), which is, in general, a static node see Figure 5.

Sensor nodes wait for a specific event (smoke, fire, sound …) to happen. Once the event of interest occurs, the sensor node (one or many) that have detected the event gathers all relevant data to the event and sends it to the Base Station. The sensor node can send data directly to the BS (if it was near to it) or by hope-by-hope until the BS. It may happen that the BS receives multiple reports on the same event by the event's surrounding nodes.

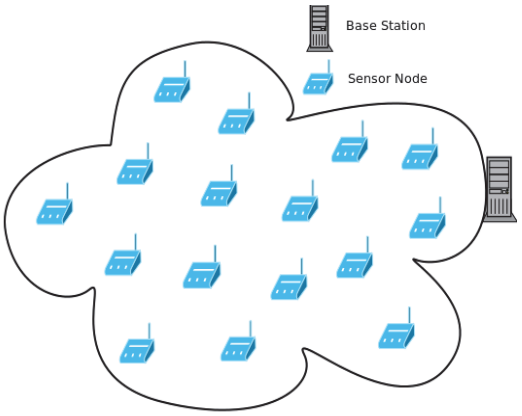

Figure 5: Wireless sensor network architecture.

Surrounding nodes may collaborate to send one single report to the BS. The BS after receiving the report can process it and decide whether to send orders to concerned nodes in view to have more details or to express its need of a specific information or to forward the result of its processing to the external world. As you can see the BS is acting as a gateway between the WSN and the external world.

WSN permit observing places that were unobservable or were difficult to observe and give a way to monitor specific event for long duration. WSN also offers a variety of potential applications to industry, science, transportation, security, ...

Flexibility of deployment that offers WSN has, Conversely, many challenges such as Responsiveness, Self-Configuration and Adaptation, Scalability, Privacy and Security, Energy Efficiency, Heterogeneity, ...

Before starting the conception of a security protocol that satisfy requirements above, we should emphasize that sensor nodes are resource constrained devices which is a limiting factor for designing efficient security procedures. We focus on two main constraints: Memory and Energy.

Memory: Sensor nodes are not designed to store a huge amount of data, this is because the very little amount of available memory. Little memory size is a limiting factor for designing sophisticated security algorithms or using existing security algorithms (RSA, ...). In order to build an efficient security scheme, the code size of the security algorithm must be small. A typical sensor node processor is of 4-8 MHz, having 4KB of RAM, 128KB flash and ideally 916 MHz of radio frequency (Kavitha and Sridharan, 2010). For example, TelosB has a 16-bit, 8 MHz RISC CPU with only 10K RAM, 48K program memory, and 1024K flash storage.

Energy: Sensor nodes, besides having small amount of memory, have also limited amount of energy, in fact each sensor node has a small battery power. Sensor nodes haven't to do complicated processing that consumes energy. Energy is a very constraining factor for wireless sensor nodes. Once a sensor node has been deployed in WSN it's difficult or very expensive to try to recharge it or to replace it. Hence, sensor node's battery life must be extended as longer as possible and for any conception or implementation of a security algorithm in a sensor node the energy consumption must be well studied and measured. Adding a security protocol to a sensor node without considering the energy factor is not a wise strategy and this will affect and make troubles for the whole network as well as for the sensor node itself. When we talk about energy consumed by a security algorithm, mainly we talk about the processing needed for security procedures like encryption, decryption, signing and verifying data. Several WSN's Operating systems provide features to save and economize energy (Healy et al., 2007)

Sensor nodes are widely used in many fields like health, military, environmental conditions, … viewing their critical mission sensor nodes need to be secured and viewing their limited memory and energy need efficient and optimized security algorithms. Wireless communication between sensor nodes facilitates intruders mission where the need to guarantee integrity and confidentiality of exchanged data and authenticity of all sensor nodes in a way that only legal nodes that will participate in communications. (Shi and Perrig, 2004)

5 PROPOSED LIGHTWEIGHT IKE FOR WSN

A wireless sensor network is a set of sensors. Sensors are resource constrained devices (small amount of memory, low computational power, ...) so for any protocol conception those constraints should be taken into account. In this work we propose a light version of IKE protocol suitable for sensors, more precisely we suggest an approach of authentication and common secret key sharing between peers and using IKE. Peers authentication and sharing secret key are performed during IKE phase one. We consider the scheme in Figure 5.

At first the Base Station (BS) generates its Public ($BS_{PK} = T_{BS} \circ P'_{BS} \circ S_{BS}$) and Private ($T_{BS}$, P'_{BS}, S_{BS}) MQQ-based keys and publishes the public key to all nodes. Any node who desires to join the network should first be registered by the BS. At the end of registration phase, the joining sensor's MQQ-based public key could be published as a registered sensor node. The BS serves as a certification authority, it guarantees that a specific public key belongs to the node who claims it belongs to.

In the registration phase, a Sensor Node (SN) who desires to join the network sends its ID to the BS. The SN generates its MQQ-based Public/Private keys and uses BS public key ($BS_{PK}=T_{BS} \circ P'_{BS} \circ S_{BS}$) to resend its public key ($SN_{PK} = T_{SN} \circ P'_{SN} \circ S_{SN}$) plus the signature of its MAC address encrypted using the known BS public key (BS_{PK}): $T_{BS} \circ P'_{BS} \circ S_{BS}$ (Sig(ID) + Sig(MAC) + SN_{PK}). BS is the only node able to decrypt the message. The BS gets the SN's public key then declares the SN as a registered node and publishes SN's MQQ-based public key.

After registration, SNs, in view to communicate, must authenticate each other and share a secret key. The Initiator sends its list of cryptographic proposals (SA_{prop}) to the receiver. The receiver selects from the proposals and responds by sending its selected list of cryptographic algorithms (SA_{select}). At this point we have started the IKE phase one process and all exchanged messages have the ISAKMP (Maughan D, Schertler M, Schneider M, 1998) header (HDR). HDR contains:

Initiator's Cookie (8 octets)
Receiver's Cookie (8 octets)
Next Payload (1 octet)
Major Version (4 bits)
Minor Version (4 bits)
Exchange Type (1 octet)
Flags (1 octet)
Message ID (4 octets)
Length (4 octets)

In view to share a secret key and do mutual authentication peers do as follows:

1) The Initiator generates its cookie C_I, prepares its proposals list SA_{prop} and sends them to the receiver with signature of its ID_I plus timestamp T_I using its MQQ-based private key (T_{SNI}, P'_{SNI}, S_{SNI}). The Initiator encrypts the signature of its ID_I and T_I using the receiver's published public key ($SN_{PKR} = T_{SNR} \circ P'_{SNR} \circ S_{SNR}$) to be sure that only the Receiver that could decrypt the message.

2) The receiver responds with its selected proposal SA_{select}, its generated cookie C_R and the signature of its ID_R plus a timestamp T_R using

its MQQ-based private key (T_{SNR}, P'_{SNR}, S_{SNR}). The receiver encrypts the signature of its ID_R and T_R using the Initiator's MQQ-based public key ($SN_{PKI} = T_{SNI} \circ P'_{SNI} \circ S_{SNI}$) to be sure that only the Initiator that could decrypt the message.

The Initiator decrypts the received message from the receiver, then gets the Receiver's ID_R, the timestamp T_R.

The receiver decrypts the received message from the Initiator, then gets the Initiator's ID_I, the timestamp T_I.

3) At this stage both peers have exchanged their IDs and identified each other's public keys, then the Initiator calculates a hash of a generated nonce N_I $HASH(N_I)$ and encrypts the hash with the Receiver's public key then sends it to the Receiver.

4) The receiver extracts the nonce N_I and also do the same by calculating a hash of a generated nonce N_R $HASH(N_R)$ and encrypts the hash with the initiator's public key and sends it to the Initiator.

Now both peers have succeeded to exchange their generated nonce and they could construct their shared secret key by doing a composition of both nonces $N_I * N_R$. We summarize this process of mutual authentication and sharing secret key in Figure 6.

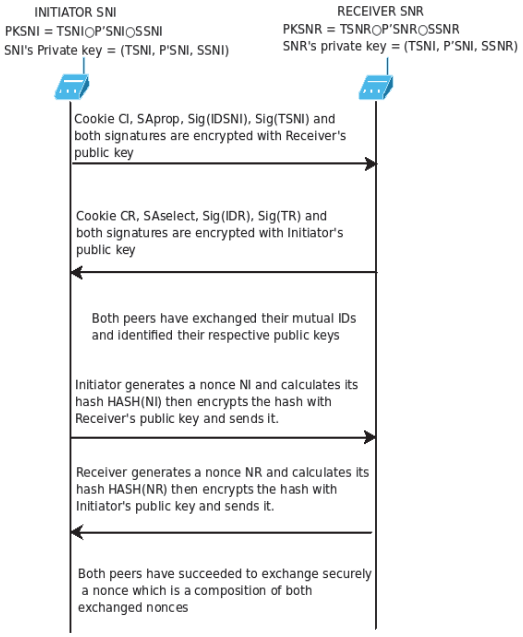

Figure 6: Authentication and share of secret nonce.

6 SECURITY CONSIDERATIONS

- Man-in-the-middle Attack

Man-in-the-middle attack is where two peers believe that they are communicating directly to each other, but in reality they are communicating through an intruder node that have succeeded to impersonate both peers and have gained access to their information and their exchanged messages.

Let's consider this situation where 2 sensor nodes SNA and SNB are initiating an IKE session and a malicious node SNC tries to enter in between SNA and SNB to do a Man-in-the-middle attack.

Our proposal is free from this attack viewing that any joining sensor node is first registered by the Base Station (BS) that stores each sensor node's MQQ-based public key, ID and MAC address. In the process of exchanging nonces sensor nodes exchange their mutual IDs encrypted using the public key, which have been already published by the BS, of the other sensor node. A malicious node can't do a man-in-the-middle attack because all public keys are published by the BS and sensor nodes have exchanged their IDs encrypted using each other's public key.

- Replay Attack

A replay attack is where a malicious node intercepts a message and tries later to reuse it by resending the intercepted message to a receiver in view to trick it or to gain some privileges illegally.

Our proposal is free from this attack viewing that any exchanged message between 2 sensor nodes contains the cookie (ISAKMP header) of the Initiator or the Receiver and among information included in a peer's cookie is the timestamp that the sender estimated for the moment when the message was sent. Any sensor node compares any received message's timestamp with the last received timestamp and it cancels the communication if there was any inconsistency between timestamps.

- Denial-of-service (DoS) Attack

A denial-of-service happens when a network or a node is targeted in view to make it unavailable or unresponsive temporally or permanently.

Our proposal is free from this attack viewing that every exchanged message contains ISAKMP header which contains Initiator's and Receiver's cookie (C_I, C_R)

- Impersonation Attack

An impersonation attack is where a malicious node presents itself to other network nodes by impersonating a legitimate node in view to exchange messages on behalf the legitimate node.

Our proposal is free from this attack viewing that an attacker couldn't read a message encrypted by the legitimate node's MQQ-based public key (which has been already published by the BS).

7 CONCLUSION

A lightweight version of the IKE protocol based on Multivariate Quadratic Quasi-groups (MQQ) is discussed in this paper. MQQ is not based, like RSA or ECC, on the difficulty of solving factorization problem or discrete logarithm and is for this reason MQQ is considered as a post-quantum algorithm. MQQ has emerged as an alternative to classical cryptography schemes and was seen, as mentioned before, that MQQ is faster, than RSA and ECC, in encryption and decryption so it's a promising public key cryptography alternative, especially to adapt existing protocols, like IKE in this paper, to wireless sensor network's context. A security analysis of the proposed protocol against some known attacks is done and is shown that the proposed protocol is secure. We are in preparation of some implementation for this approach to endorse it with experimental results.

REFERENCES

Breu, F., Guggenbichler, S., Wollmann, J., 2008. PKCS #1: RSA Encryption Version 1.5. Vasa, Request for Comments 1–19.

Dimitrova, V., Markovski, J., 2004. On Quasigroup Pseudo Random Sequence Generators. Proc. 1-st Balk. Conf. Informatics, Y. Manolopoulos P. Spirakis eds 21–23.

Ding, J., Yang, B.-Y., 2009. Multivariate public key cryptography, in: Bernstein, D.J., Buchmann, J., Dahmen, E. (Eds.), Post-Quantum Cryptography. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 193–241. doi:10.1007/978-3-540-88702-7_6

El-Hadedy, M., Gligoroski, D., Knapskog, S.J., 2008. High performance implementation of a public key block cipher-mqq, for fpga platforms, in: Reconfigurable Computing and FPGAs, 2008. ReConFig'08. International Conference on. pp. 427–432.

Essadraoui, Y., Dafir Ech-cherif El Kettani, M., 2015. Wireless sensor node's authentication scheme based on Multivariate Quadratic Quasi-groups. 2015 Third World Conf. Complex Syst. 1–6. doi:10.1109/ICoCS.2015.7483320

Gligoroski, D., Markovski, S., Knapskog, S.J., 2008a. A Public Key Block Cipher Based on Multivariate Quadratic Quasigroups. Proc. Am. Conf. Appl. Math. abs/0808.0, 44–49.

Gligoroski, D., Markovski, S., Knapskog, S.J., 2008b. Multivariate Quadratic Trapdoor Functions Based on Multivariate Quadratic Quasigroups, in: Proceedings of the American Conference on Applied Mathematics, MATH'08. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, pp. 44–49.

Hallqvist, N., Keromytis, A. ~D., 2000. Implementing Internet Key Exchange (IKE), in: Proceedings of the Annual USENIX Technical Conference, Freenix Track, ATEC '00. USENIX Association, Berkeley, CA, USA, pp. 201–214.

Healy, M., Newe, T., Lewis, E., 2007. Power Management in Operating Systems for Wireless Sensor Nodes, in: Sensors Applications Symposium, 2007. SAS '07. IEEE. pp. 1–6. doi:10.1109/SAS.2007.374366

Kavitha, T., Sridharan, D., 2010. Security vulnerabilities in wireless sensor networks: A survey. J. Inf. Assur. Secur. 5, 31–44.

Kipnis, A., Patarin, J., Goubin, L., 1999. Unbalanced oil and vinegar signature schemes, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), EUROCRYPT'99. Springer-Verlag, Berlin, Heidelberg, pp. 206–222. doi:10.1007/3-540-48910-X_15

Kivinen, T., Snyder, J., 2015. Signature Authentication in the Internet Key Exchange Version 2 (IKEv2). Request for Comments 1–18. doi:10.17487/rfc7427

Maughan D, Schertler M, Schneider M, T.J., 1998. Internet Security Association and Key Management Protocol (ISAKMP), Rfc 2408, Request for Comments. IETF. doi:10.17487/rfc2408

Moh, T., 1999. A public key system with signature and master key functions. Comm. Algebr. 27, 2207–2222.

Patarin, J., 1996. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. Adv. Cryptology-EUROCRYPT, EUROCRYPT'96 1–40. doi:10.1007/3-540-68339-9_4

Patarin, J., Patarin, J., Goubin, L., Goubin, L., 1997. Trapdoor one-way permutations and multivariate polynominals. Inf. Commun. Secur. First Int. Conf.

ICICS'97, Beijing, China, Novemb. 11-14, 1997, Proc. 1334, 356–368. doi:10.1007/BFb0028491

Quirino, G.S., Moreno, E.D., 2013. Architectural Evaluation of Asymmetric Algorithms in ARM Processors. Int. J. Electron. Electr. Eng. 1, 39–43. doi:10.12720/ijeee.1.1.39-43

R.~Atkinson, 1995. {S}ecurity {A}rchitecture for the {I}nternet {P}rotocol, Request for Comments. IETF.

Shamir, A., 1994. Efficient Signature Schemes Based on Birational Permutations, in: \ifnum\shortbib=1 {CRYPTO}\else{Advances in Cryptology -- {CRYPTO}}\fi'93. Springer-Verlag, pp. 1–12. doi:10.1007/3-540-48329-2_1

Shi, E., Perrig, A., 2004. Designing secure sensor networks. IEEE Wirel. Commun. 11, 38–43. doi:10.1109/MWC.2004.1368895

Smith, J.D.H., 2007. An Introduction to Quasigroups and Their Representations. CRC Press.

Zhu, L., Jaganathan, K., Lauter, K., 2008. Elliptic Curve Cryptography (ECC) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT). Request for Comments.