Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier -

Full-Resilient Memory-Optimum Multi-Party Non-Interactive Key Exchange

MAJID SALIMI[1], HAMID MALA[2], HONORIO MARTIN [3], AND PEDRO PERIS-LOPEZ. [4]
[1] Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran. (e-mail: M.Salimi@eng.ui.ac.ir)
[2] Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran. (e-mail: h.mala@eng.ui.ac.ir)
[3] Department of Electronic Technology, University Carlos III of Madrid, Avda. de la Universidad 30, 28911 Legan Spain (e-mail: hmartin@ing.uc3m.es)
[4] Department of Computer Science, University Carlos III of Madrid, Avda. de la Universidad 30, 28911 Legan Spain. (e-mail:pperis@inf.uc3m.es)

Corresponding author: Hamid Mala (e-mail:h.mala@eng.ui.ac.ir).

ABSTRACT Multi-Party Non-Interactive Key Exchange (MP-NIKE) is a fundamental cryptographic primitive in which users register into a key generation centre and receive a public/private key pair each. After that, any subset of these users can compute a shared key without any interaction. Nowadays, IoT devices suffer from a high number and large size of messages exchanged in the Key Management Protocol (KMP). To overcome this, an MP-NIKE scheme can eliminate the airtime and latency of messages transferred between IoT devices.

MP-NIKE schemes can be realized by using multilinear maps. There are several attempts for constructing multilinear maps based on indistinguishable obfuscation, lattices and the Chinese Remainder Theorem (CRT). Nevertheless, these schemes are inefficient in terms of computation cost and memory overhead. Besides, several attacks have been recently reported against CRT-based and lattice-based multilinear maps. There is only one modular exponentiation-based MP-NIKE scheme in the literature which has been claimed to be both secure and efficient. In this article, we present an attack on this scheme based on the Euclidean algorithm, in which two colluding users can obtain the shared key of any arbitrary subgroup of users. We also propose an efficient and secure MP-NIKE scheme. We show how our proposal is secure in the random oracle model assuming the hardness of the root extraction modulo a composite number.

INDEX TERMS Multi-Party Non-Interactive Key Exchange, Broadcast Encryption, Internet of Things, Random Oracle Model.

I. INTRODUCTION

In a key distribution scheme, an off-line Key Generation Center (KGC) distributes keying information through a secure channel to every node (user) in the network. Later, every pair of users in the system, by using the keying information they hold, will be able to determine a key known only to them. This operating mode enables them to have encrypted communications [1]. Suppose we have a set of n nodes. In its general form, called the multi-party scenario, the key distribution problem is not restricted to only pairs of users, but it must enable any arbitrary subset of these n nodes to determine a shared key [2]. A trivial solution to this problem is that a Trusted Authority (TA) generates $M = 2^n - n - 1$ symmetric keys, and assigns each to one of the M subsets with at least two members. Then, it gives the key for each group (subset of users) to the users who belong to this subset. Any node is a member of $G = 2^{n-1} - 1$ groups of at least two members. As a result, any node must store G distinct keys, which is impractical. Public-key cryptographic approaches can be employed to address this limitation [3]. When, instead of a pool of symmetric keys, any user receives only one public/private key pair from the KGC and employs its private key and other users' public keys to generate a shared symmetric key (without any interaction), the scheme is usually referred to as Non-Interactive Key Exchange (NIKE) [4].

In this article, we focus on multi-party solutions. Notably, in Multi-Party Non-Interactive Key Exchange (MP-NIKE) schemes, any user first registers into a KGC and receives a unique public/private key pair. Let W be a subset of registered users. Then any user $U_i \in W$ can compute a pre-shared key K_W using its private key and the public keys of other members of the group W.

We stress that in this article, we focus on key distribution not on the key agreement schemes and their associated features such as forward secrecy or authentication such as in [5], [6] and [7]. Nevertheless, in our scheme, by using the public key of each node as node identifier, nodes can be authenticated. Because no one else has access to the respective private key and without a valid private key, no one can compute the shared key, so we can be sure that nodes are authenticated. The proposed approach provides a long-term key, not a session key. The long-term key can then be employed as a symmetric key pre-shared among the nodes of the associated subgroup to run an authentication and/or session key agreement protocol.

A. APPLICATIONS OF MP-NIKE

As an underlying cryptographic protocol, NIKE has many applications including broadcast encryption, key management for wireless sensor networks (WSNs) [8], [9], and group communication for Internet of things (IoT) devices [10], [11].

Group Communication for IoT. One of the applications of MP-NIKE schemes is key management for group communication in Internet of Thing. Suppose a few smart objects in a smart home need to securely communicate, so first they need to securely establish a session key. The key agreement protocols for IoT, such as [12], [13], [14], [15], [16], [17] and [18] need to exchange a few messages, while message exchanging is costly and time-consuming. This is a serious challenge in resource-constrained devices employed in IoT systems; for example, the Maximum Transmission Unit (MTU) at the link layer of Industrial IoT (IIoT) technology, when IEEE 802.15.4 technology is adopted, is just equal to 127 bytes, so the Key Management Protocol (KMP) messages must be fragmented [19]. Therefore, even one message can cause too airtime latency. For example, two-party key agreement by running KMP protocol with implicit X.509 certificates takes 3.29 seconds (computation time plus air latency) in a single-hop network [19]. It gets even worse in multi-hop networks [19].

An MP-NIKE scheme can provide a distinct key between any subset of these objects and without needing to exchange any messages. Therefore by using MP-NIKE, the IoT nodes are not limited in two-way connections, and can securely establish a shared key between any arbitrary group of nodes. Furthermore, by employing MP-NIKE, nodes can securely broadcast messages for other nodes. The new IoT devices can natively and simply compute cryptographic primitives [19]. Furthermore, as we will compare in Section VII, the proposed MP-NIKE scheme is practical and efficient.

Broadcast encryption. A practical MP-NIKE scheme can be used to construct a broadcast encryption protocol [20]. Broadcast encryption is a way for broadcasting an encrypted message on a public channel, such that broadcasting server can be sure that just a group of authorized users can decrypt the message. The size and members of the group of authorized users is not constant, and it may change for any single message. Previous broadcast encryption schemes suffer from security faults like ciphertext size and public key size [21]. By using a practical MP-NIKE scheme, we can construct an efficient broadcast encryption scheme and solve the problem of long public key and ciphertexts [20].

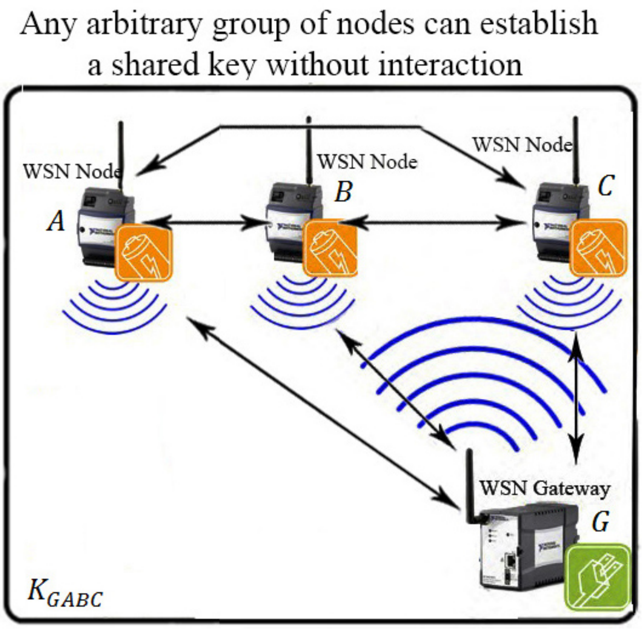
Any arbitrary group of nodes can establish a shared key without interaction
WSN Node
A
WSN Node
B
WSN Node
C
WSN Gateway
G
K_{GABC}

FIGURE 1: MP-NIKE in WSN

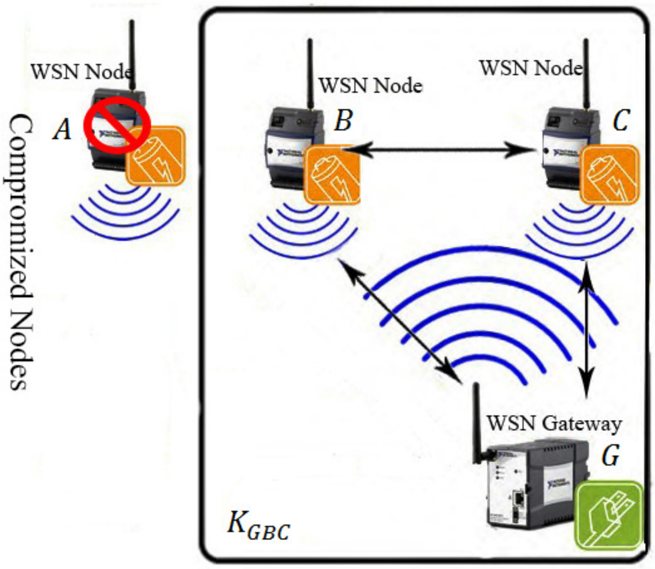

FIGURE 2: Removing the compromised node in WSN

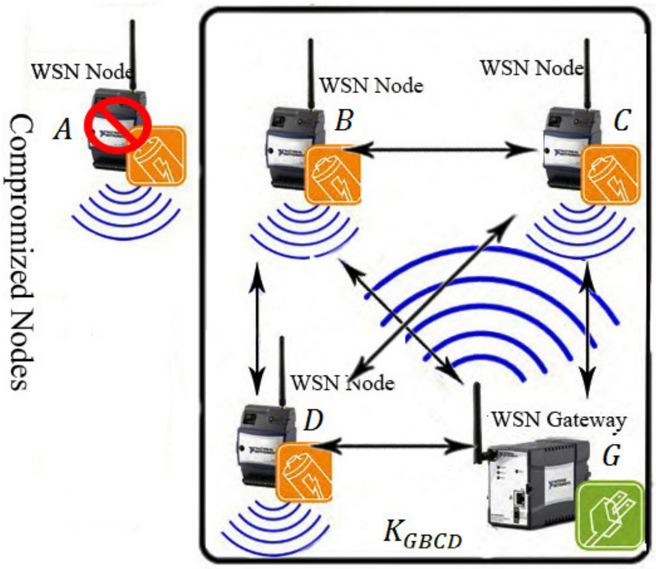

FIGURE 3: Adding a new node to the group

Key management for WSN. One of the other applications of the MP-NIKE is key management in Wireless Sensor Networks. Since message exchanging consumes battery power and, on the other hand, sensors have energy limitations, using interactive key exchange schemes, such as [22], [23] and [24], is not a proper solution to protect confidentiality in this sort of networks. On the one hand, using a single master key for all nodes has very low resilience, and if an adversary captures only one node, he will be able to compromise all nodes of the WSN. On the other hand, if we use a distinct pair-wise key for any two nodes of n nodes, then any node must store $n-1$ different keys. This solution creates a heavy storage burden on each node, also for adding a new node in the future, we would need to update the key chain of every single node.

Other pair-wise key distribution solutions, such as closest pair-wise key pre-distribution [25], random pair-wise key scheme [26], pair-wise key establishment protocol [27], combinatorial design-based pair-wise key pre-distribution scheme [28], etc. try to adjust the storage problem and yet provide key resilience, but none of them can guarantee key resilience with O(1) storage complexity. Furthermore, compromised nodes are a challenge in WSN, since they can behave arbitrarily and cooperate with others. Several solutions have been proposed in the literature to address this problem (e.g., [29] or [30]). Unfortunately, these solutions are highly demanding in terms of message exchange. Note that pairing-based two-party NIKE schemes impose massive computation cost and need powerful hardware, which is not considered to be affordable in WSN nodes, because of power and price limitations. The Simple Password-Based Encrypted Key Exchange (SPAKE2) protocol [5] provides forward secrecy, but, in addition to just being two-party, it requires agreement on a password between any two users and it is interactive. It means that any two users need to somehow agree on a shared password, so, it is not efficent at all.

Conversely, the proposed MP-NIKE scheme provides resilience and enables any subset of sensors to efficiently compute a shared key without any interactions; also, adding new nodes in future can be quickly addressed. Besides, removing a compromised node will be quickly done. Since the proposed scheme is based on modular exponentiation, its computation cost is lower than pairing-based two-party NIKE schemes. The overall scenario of employing MP-NIKE in WSN is shown in Fig. 1, Fig. 2 and Fig. 3. In Fig. 1, a group of four honest nodes A, B, C and G can easily and securely communicate by a shared key K_{ABCG} which is computed by each of them without any interaction with the others. Fig. 2 shows how they can easily remove a compromised node A just by using the shared key K_{BCG} computable only by B, C and G. The Fig. 3 illustrates that adding a new node can be quickly done.

B. OUR CONTRIBUTION

In 2014, Eskeland introduced a fully resilient and efficient MP-NIKE scheme based on modular exponentiation in RSA modulus [31]. Nevertheless, in this paper, we propose a Euclidean algorithm-based attack against this scheme, that invalidates Eskeland's claim.

Furthermore, we describe a new fully resilient MP-NIKE scheme. The computation cost of the proposed protocol for computing a shared key for a group W, with size $|W|$, is only $|W|-1$ modular exponentiations and every user needs to store just one public/private key pair of small size.

For the sake of comparison, obtaining a shared key among 19 users at 80-bit security level by using the 5Gen multilinear map (an extension of the CLT13) [32], takes 33 seconds. However, in our scheme, each user requires to compute 18 modular exponentiations in a 1024-bit modulus so that it takes about 281 milliseconds. We proved that the security of our scheme is equal to Fiat-Naor problem, (the security of Fiat-Naor scheme is based on the root extraction in RSA modulus, which is equal to RSA problem [33]). Also, since our proposal, compared with previous ones, is based on lighter cryptographic operations such as modular exponentiations, its computation cost is low.

C. PAPER ORGANIZATION

The rest of this paper is organised as follows. Section II introduces the related work. In Section III, the general model of MP-NIKE, as well as the required preliminaries and definitions, are described. In Section IV, we briefly review the Fiat-Naor and the Eskeland schemes and propose an attack on the Eskeland scheme. Our novel MP-NIKE scheme is introduced in Section V. The security of the proposed scheme is discussed in Section VI. In Section VII we compare computational and memory overhead of the proposed scheme against previous MP-NIKE schemes. Finally, we conclude the paper in Section VIII.

II. RELATED WORK

The first non-interactive key exchange scheme was introduced in the seminal work of Diffie and Hellman in 1976 [34]. Their proposal was secure and efficient, but it was bounded only to the two-party case. In 2008, Cash et al. introduced a new assumption, called the Twin Diffie-Hellman problem, and presented a new two-party NIKE scheme that is secure in the random oracle model [35]. In 2013, Freire et al. presented a new two-party NIKE scheme and proved its security using a game-based security model. However, they did not model the key registration process in their security model [4]. A year later, Freire et al. successfully modeled the key registration process in the security model of their new two-party NIKE scheme [36].

The first three-party non-interactive key exchange scheme was introduced by Joux in 2001 using bilinear pairings [37]. His protocol was identity-based, secure and efficient, but it could not be extended for more than three parties. Fiat and Naor first proposed the idea of MP-NIKE in 1992, where they also suggested the employment of this idea in broadcast encryption. Unfortunately, their scheme is only 1-resilient [33], i.e. the keys are protected only against one user; in other words, any two or more colluding members could compute the shared key of any group, whether they belong to this group or not. In 2003, Boneh and Silverberg showed that multilinear maps could be used to construct multi-party NIKE schemes (see Definition 4 and Lemma 2), but they also revealed that the bilinear Tate and Weil pairings could not be generalized to multilinear maps [20]. So, they could not propose any concrete multilinear map [20].

The design of a full-resilient, multi-party and non-interactive, key-exchange protocol remained an open problem until Garg, Gentry and Halevi [38] introduced the first multilinear map (GGH13) based on lattices. After that, Langlois in 2014 presented a more efficient GGH map called GGHLite [39], and then, Albrecht et al. proposed the first practical MP-NIKE scheme based on GGHLite [40]. This scheme, however, is inefficient in public parameters size and computational cost, as they declared in 80-bit security, computing a shared key between 7 users will take about 1.75 seconds on a 16-core CPU. In 2015, Hu and Jia showed that GGH and GGHLite are insecure [41]. In 2015, Gentry et al. proposed a graph-induced multilinear map from lattices [42], which for simplicity we call it GGH15. Recently, Coron et al. showed that GGH15 is insecure too [43]. Moreover, all of these multilinear maps are inefficient because they are based on using cryptographic operations with a high computational cost. Coron et al. in 2013, proposed another candidate construction of multilinear maps over integers [44], denoted by CLT13 for simplicity, which soon was broken by Cheon et al. [45]. Then, Coron et al. fixed their scheme [46], but this time Minaud and Fouque proposed an attack on this fixed scheme and downgraded it to the previous one [47]. Recently, Ma and Zhandry proposed another multilinear map based on CLT13 which is provably secure against previously known attacks, but its security its not proven in the standard security model [48]. Their scheme is a modified version of CLT13, so it is not efficient.

Multilinear maps and MP-NIKE schemes can be constructed by using $i\mathcal{O}$ [49]. Garg et al. proposed the first construction of $i\mathcal{O}$ for general boolean circuits by using multilinear maps [50], and then Rao [51], Yamakawa et al. [52], Boneh and Zhandry [49] and Khurana et al. [53] proposed some multilinear maps and MP-NIKE schemes based on $i\mathcal{O}$ and constrained Pseudo-Random Generator (PRG) [53]. Since multilinear maps are needed to construct $i\mathcal{O}$ [38], creating a multilinear map using $i\mathcal{O}$ seems impractical. To the best of our knowledge, so far no provably secure and practical $i\mathcal{O}$ and multilinear maps have been proposed in the literature. So, all of the $i\mathcal{O}$-based and multilinear map-based MP-NIKE schemes are either impractical or insecure [41], [43], [45], [47], [49].

In 2016, Chen et. al. proposed an identity-based MP-NIKE based on Witness Pseudo Random Function(WPRF) [54]. Constructing a WPRF needs asymmetric cryptographic multilinear map [55]. If we had access to an efficient multilinear map, we would use it to construct an MP-NIKE from the scratch [20].

MP-NIKE schemes are also called Non-Interactive Conference Key Distribution System (NICKDS) schemes. In 1998 Blundo et al. [56] proposed a k-secure t-conference NICKDS using multi-variable symmetric polynomials, which was secure against k colluding users. In Blundo scheme, no k colluding users do not obtain any information about any key of other users, but any $k+1$ colluding users can obtain key of all other users [56]. The bound of security parameter k in Blundo scheme with n users is equal to $n-t$, where t is the size of the conference, and each user needs to store a piece of information with the size of $\binom{k+t-1}{t-1}$ times the size of common key [56]. It means that for calculating a 128-bit key, in 50-secure 50-conference NICKDS system (for 100 users), each user needs to store about 2^{103}-bit data, which is infeasible.

III. PRELIMINARIES

In this section, we introduce the notations used in this paper, the necessary definitions, lemmas and the general model of Non-Interactive Key Exchange (NIKE).

A. NOTATIONS

The Notations and abbreviations used in this paper are outlined in Table 1.

B. NIKE GENERAL MODEL

The general model of MP-NIKE consists of the following four algorithms.

TABLE 1: Notations and abbreviations

Symbol	**Description**
p, q	Two large primes
N	Modulus of computation
$\mathbb{G}$	A multiplicative subgroup of $\mathbb{Z}_N^*$
g	A generator of $\mathbb{G}$
$\mathsf{O}()$	Random Oracle
$H()$	A one-way hash function
W	A group of users aiming to compute a shared key
K_W	Shared key of the group W
U_i	User i
PP	Public parameters
msk	The KGC's master secret key
e_i	Public key of user i
d_i	Private key of user i
γ	Security parameter
A	The adversary
B	The challenger algorithm
W_c	Challenge set
s	$\lvert W_c \rvert$

1) ***Setup*(γ).** Given a security parameter γ as input, the Key Generation Center (KGC) runs setup(γ) to generate a master secret key msk and a set of public parameters PP. Then, the KGC announces PP to all users.
2) ***KeyGen*(PP, msk, i).** In this phase, the user U_i is authenticated by the KGC, and the KGC generates a valid public/private key pair (e_i, d_i), saves them in its database and finally gives them to user U_i.
3) ***SharedKey*(PP, d_i, $\{e_j : U_j \in W, j \neq i\}$).** Each member U_i of group W runs this algorithm to obtain a valid shared key K_W. All members of W are able to compute this shared key, while other users which do not belong to W cannot compute it.
4) ***Join*(PP, K_W, e_s).** Suppose there exists a group W with a key K_W shared among its members. When a new user U_s wants to join this group to form a new group $W' = W \cup \{U_s\}$, it must run *SharedKey*(PP, d_s, $\{e_j : U_j \in W', j \neq s\}$), while the other members of W can update the shared key K_W to $K_{W'}$ with a lower computational overhead. The output of this algorithm is a new shared key $K_{W'}$, where $W' = W \cup \{U_s\}$.

C. DEFINITIONS AND LEMMAS

To prove the security of our scheme, we need the following definitions and lemmas.

Definition 1. (Safe modulus): *Let $N = p' \cdot q'$ be the product of two large primes p' and q'. Then, N is called a safe RSA modulus if $p' = 2p + 1$ and $q' = 2q + 1$, where p and q are also prime numbers.*

Lemma 1. *(Generators of a subgroup): Let $N = p' \cdot q' = (2p + 1)(2q + 1)$ be a safe RSA modulus. Besides, assume that $\mathbb{G}$ is a subgroup of order $p \cdot q$ in $\mathbb{Z}_N^*$ and $\hat{\mathbb{G}}$ is a subgroup of order p from $\mathbb{G}$. Under these conditions, if g is a generator of $\mathbb{G}$ then $g_1 = g^q \bmod N$ is a generator of $\hat{\mathbb{G}}$ [60].*

Definition 2. (q-th residue): *Let $N = p' \cdot q' = (2p+1)(2q+1)$ be a safe RSA modulus and $\mathbb{G}$ be a subgroup of order $p \cdot q$ in $\mathbb{Z}_N^*$. Then, an element g_1 is a q-th residue in $\mathbb{Z}_N^*$, if there exists at least one element $\alpha \in \mathbb{G}$ such that $\alpha^q \bmod N = g_1$.*

Definition 3. (The Fiat-Naor problem): *Let $N = p'q' = (2p+1)(2q + 1)$ be a safe RSA modulus. Besides, assume that g is a private generator of the subgroup $\mathbb{G}$ of $\mathbb{Z}_N^*$ of order q. Under these conditions, and given $(y, g^y \bmod N, c)$, where c is coprime to y, compute $g^c \bmod N$ [33]. It is assumed that the Fiat-Naor problem is intractable and equivalent to the root extraction in RSA modulus and RSA problem [33].*

Definition 4. (Multilinear map (mmap)): *Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative groups of the same prime order. The map $e : \mathbb{G}_1^n \rightarrow \mathbb{G}_2$ is an n-multilinear map, if it satisfies the following two properties [20]:*

1) *Multilinearity. If $a_1, \dots, a_n \in \mathbb{Z}$ and $x_1, \dots, x_n \in \mathbb{G}_1$ then*

$$e({x_1}^{a_1}, \dots, {x_n}^{a_n}) = e(x_1, \dots, x_n)^{a_1 \cdots a_n} \quad (1)$$

2) *Non-degeneracy. Let $g \in \mathbb{G}_1$ be a generator of $\mathbb{G}_1$, then $e(g, \dots, g)$ must be a generator of $\mathbb{G}_2$.*

Lemma 2. *(Realization of MP-NIKE by using multilinear map (mmap)): An MP-NIKE scheme can be performed by using multilinear maps. Let the map $e : \mathbb{G}_1^n \rightarrow \mathbb{G}_2$ be an n-multilinear map, g represent a generator of $\mathbb{G}_1$ and $g_t = e(g, \dots, g)$ be a generator of $\mathbb{G}_2$. Suppose that members of group $W = \{U_1, \dots, U_{n+1}\}$ with $n + 1$ users want to compute a shared key K_W. Any user $U_i \in W$ chooses a random integer a_i and publishes g^{a_i} as its public key. Now any user $U_i \in W$ can compute the shared key K_W as below.*

$$K_W = e(g^{a_1}, \dots, g^{a_{i-1}}, g^{a_{i+1}}, \dots, g^{a_{n+1}})^{a_i} = g_t^{a_1 \dots a_{n+1}} \quad (2)$$

D. MP-NIKE SECURITY MODEL

We adopt the security model of Non-Interactive Conference Key Distribution (NICKD) of [61], which is an extension of the Bellare-Rogaway security model [62] and [63]. In this security model, the adversary is allowed to use three types of oracles: Test, Reveal and Corrupt. The adversary can adaptively corrupt users of his choice and obtain corrupted users' keys by using Corrupt oracle. By using the Reveal oracle, the adversary can obtain the shared key of any arbitrary group as below. The adversary gives an arbitrary group W to the Reveal oracle as input, and then this oracle sends the repective shared key as output to the adversary. A Corrupt query can also obtain the information leaked by a Reveal query [61], since the adversary can compute the shared key of any arbitrary group W just by corrupting one user $U_i \in W$. So, we omit the Reveal oracle in our model.

To prove the security of an MP-NIKE scheme, by contradiction, we suppose there exists an adversary A that can distinguish between a random bit string and the shared key of a specific group W^* of arbitrary size s, which it does not access to private keys of its members. Then, we show that there exists an algorithm B that can solve a hard problem by invoking algorithm A. In this model, at the first step, the adversary A must commit to s, by announcing it to algorithm B. Now the algorithm B as a challenger plays the following game with adversary A.

For the sake of formalisation, we adapt the model of [61] to five phases as below.

1) *Commit.* At the first step, the adversary A must choose s and then commit to s by sending it to the algorithm B.
2) *KeyGen.* The algorithm B generates q_c sets $\langle W_1, \ldots, W_{q_c} \rangle$, where each of them contains exactly s valid public keys. It also generates the associated private keys and keeps them private. Finally, it gives these q_c sets $\langle W_1, \ldots, W_{q_c} \rangle$ in a random order to the adversary.
3) *Phase 1.* In this phase, the adversary is allowed to ask q_c Corrupt queries and one Test query from the algorithm B. The formal definition of these oracles is as follows.
   - *Corrupt(U_i)* (or equivalently *Corrupt(e_i)*). The adversary can corrupt any user U_i adaptively by using Corrupt oracle. The adversary gives U_i as input to Corrupt oracle and then this oracle sends the respective private key d_i as output to the adversary.
   - *Test(W^*).* When the adversary decides to terminate Phase 1, it chooses one of the q_c input sets of public keys, say W^*, such that for all $U_i \in W^*$, U_i should not have appeared in none of the Corrupt(U_i) queries, and then sends Test(W^*) to the challenger. After receiving this query, the challenger generates a random bit $b \in \{0, 1\}$: if $b = 0$ then the challenger sends K_{W^*} to the adversary, otherwise, it generates a random string $rand \leftarrow \{0, 1\}^\lambda$ and sends it back to the adversary, where λ is the bit length of K_{W^*}.
4) *Phase 2.* This phase is the same as Phase 1, except that the adversary does not have access to the Test oracle and it is not allowed to ask Corrupt(U_i), where $U_i \in W^*$.
5) *Guess.* In this phase, the adversary guesses b by a bit $b' \in \{0, 1\}$ and sends it to the challenger. If $b' = b$ the adversary wins the game.

Finally, we give the following definition concerning the security offered by the MP-NIKE scheme.

Definition 5. (Fully resilient MP-NIKE scheme): *MP-NIKE scheme $\mathcal{E}$ is fully resilient (q_c, T, ϵ)-secure, if in the above described game any adversary A, which is allowed to ask q_c queries from Corrupt oracle, cannot distinguish a random string (i.e., $rand$) from the true shared key K_{W^*} with an advantage greater than ϵ in time T.*

$$Adv_A^{\mathcal{E}} = \left| Pr[b' = b] - \frac{1}{2} \right| \leq \epsilon. \quad (3)$$

IV. THE FIAT-NAOR AND THE ESKELAND NIKE SCHEMES

In this section, we briefly introduce the Eskeland NIKE scheme, but first we need to review the Fiat-Naor scheme. Then, we present a coalition attack against the Eskeland's NIKE scheme in which any two colluding users can compute the shared key of any other group of users.

A. THE FIAT-NAOR SCHEME

This scheme is based on the intractability of the factorisation of RSA moduli and is secure against any adversary that has access to at most one public/private key pair. This scheme works as follows.

The KGC generates an RSA modulus $N = pq$, where p and q are large primes, and then selects at random a generator g from $\mathbb{Z}_N^*$ and keeps it as a master secret key msk for itself. To generate a valid public/private key for user U_i the KGC selects at random a prime y_i and then computes the private key of the user U_i as $d_i = g^{y_i} \bmod N$. Finally, the KGC gives $e_i = y_i$ and d_i as public key and private key to user U_i. Let W be a group of users aiming to compute a shared secret key. User U_i, where $U_i \in W$, computes the shared key as follows.

$$K_W = d_i^{\prod\limits_{j:U_j \in W, j \neq i} e_j} \bmod N = g^{\prod\limits_{j:U_j \in W} y_j} \bmod N. \quad (4)$$

Suppose that a new user U_s wants to join W to form a new group $W' = W \cup \{U_s\}$. It must run *SharedKey*$(PP, d_s, \{e_j : U_j \in W', j \neq s\})$ to compute a new shared key as below.

$$K_{W'} = d_s^{\prod\limits_{j:U_j \in W} e_j}. \quad (5)$$

Any other user U_j who has already joined W simply computes

$$K_{W'} = K_W{}^{e_s} \bmod N. \quad (6)$$

The Fiat-Naor NIKE scheme is 1-resilient but insecure against collaboration of two or more adversaries. In other words, any adversary accessing at least two public/private key pairs can compute g by using the Euclidean algorithm and can break the Fiat-Naor scheme [33]. Stated differently, given $(a, g^a \bmod N)$ and $(b, g^b \bmod N)$, the adversary can compute $g^{gcd(a,b)} \bmod N$, by using the Euclidean algorithm and performing a sequence of modular exponentiations on $g^b \bmod N$ and $g^a \bmod N$. Note that a and b are prime, so the result equals g.

B. THE ESKELAND SCHEME

Let $N = pq$ be an RSA modulus, g be a public generator of $\mathbb{Z}_N^*$ and $H()$ be a secure one-way hash function. The KGC selects at random a secret u. The public key of user U_i is computed as $e_i = H(\text{identity of user } U_i)$ and its private key is generated by the KGC as $d_i = z_i u + v_i\varphi(N)$, where $z_i = e_i \bmod \varphi(N)$ and v_i is a unique random element of $\mathbb{Z}_N^*$. Then, any user $U_i \in W$ computes the pre-shared key of group W as below.

$$K_W = g^{d_i \prod\limits_{j:U_j \in W, j\neq i} e_j} \bmod N = g^{u \prod\limits_{j:U_j \in W} z_j} \bmod N. \tag{7}$$

C. ATTACK ON THE ESKELAND SCHEME

Eskeland claimed that his scheme is fully resilient against any number of colluding adversaries [31]. The colluding adversaries have access to each other's private keys and also the shared key of the groups to which they belong. Nevertheless, they do not have access to the master secret key of the KGC, nor to the private key of honest users (i.e. secrecy of private keys – Security Requirement-1 in [31]). However, we perform an attack by removing the effect of multipliers of $\varphi(N)$ and show how the secrecy of groups keys (Security Requirement-2 in [31]) is not guaranteed. Mathematically, we show this below:

Let e_i and e_j be two public keys in the Eskeland scheme and $gcd(e_i, e_j) = 1$. By using the extended Euclidean algorithm we can efficiently compute the integers a and b such that $ae_i - be_j = 1$ [64].

Lemma 3. *Let e_i and e_j be two (coprime) public keys in the Eskeland scheme. Besides, we assume that a and b are two integers such that $ae_i - be_j = 1$. Then, it is satisfied that $(az_i - bz_j) \bmod \varphi(N) = 1$.*

Proof. $e_i = z_i + k_i\varphi(N)$ for some integer k_i, and $e_j = z_j + k_j\varphi(N)$ for some integer k_j. If $ae_i - be_j = 1$, then we have that

$$\begin{aligned} ae_i - be_j &= (az_i - bz_j) + (ak_i - bk_j)\varphi(N) \\ &= (az_i - bz_j) \bmod \varphi(N) = 1. \end{aligned} \tag{8}$$

■

Based on Lemma 3, if $ae_i - be_j = 1$, any two colluding users U_i and U_j can compute $u' = ad_i - bd_j$ which is equal to $u \bmod \varphi(N)$ as below.

$$\begin{aligned} u' = ad_i - bd_j &= (az_i - bz_j)u + (av_i - bv_j)\varphi(N) \\ &= (az_i - bz_j)u \bmod \varphi(N) = u \bmod \varphi(N). \end{aligned} \tag{9}$$

Then, given u', the colluding users U_i and U_j can compute $g^u = g^{u'} \bmod N$. They can subsequently compute $g^{u \prod\limits_{U_i \in W'} e_i} \bmod N$ as the shared key of any arbitrary group W'.

V. NOVEL MP-NIKE SCHEME

In this section, we present a new MP-NIKE scheme, which is secure against any coalition of malicious users.

The proposed scheme consists of four phases as below.

1) *Setup(γ).* For a given security parameter γ, the KGC produces a safe RSA modulus $N = p'q' = (2p+1)(2q+1)$, where p, q, p' and q' are large primes. Let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_N^*$ of order pq and $m = \lceil \log_2 pq \rceil$ be the bit length of pq. The KGC selects at random a generator $g \in_R \mathbb{G}$ and presents a hash function $H : \mathbb{Z}_N^* \longrightarrow \{0,1\}^\lambda$, i.e. the output of $H(\cdot)$ is a λ-bit string.
   Finally, the KGC publishes $\{N, H(\cdot), g^p\}$ as the public parameters PP and keeps the master secret key $msk = (p, q)$ for itself.
2) *KeyGen(PP, msk).* The KGC selects two uniformly random $m/2$-bit odd integers y_i and k_i, and computes $e_i = (py_i + qk_i), d_i = g^{py_i} \bmod N$ and sends (e_i, d_i) as a valid public/private key pair to user U_i.
3) *SharedKey(PP, d_i, $\{e_j : U_j \in W, j \neq i\}$).* Suppose W is a subset of users of size $|W|$, aiming to obtain their preshared key K_W. Given the public key of other users of W, each user $U_i \in W$ computes K_W as below.

$$F_W = d_i^{\prod\limits_{j:U_j \in W, j\neq i} e_j} = g^{\left(p^{|W|} \prod\limits_{j:U_j \in W} y_j\right)} \bmod N, \tag{10}$$

$$K_W = H(F_W). \tag{11}$$

4) *Join(PP, F_W, e_s).* Suppose a user U_s wants to join a group W to extend it to $W' = W \cup \{U_s\}$. It must run *SharedKey*$(PP$**,** d_s**,** $\{e_j : U_j \in W\})$, while other users in W can simply compute the shared key of group W' as below.

$$F_{W'} = F_W^{e_s} \bmod N, \tag{12}$$

$$K_{W'} = H(F_{W'}). \tag{13}$$

Note that if (e_i, d_i) and (e_j, d_j) are valid public/private key pairs, then for any pair of integers α and β, $(\alpha e_i + \beta e_j, d_i^\alpha d_j^\beta \bmod N)$ are also valid public/private key pairs. In the proposed scheme, like many other cryptographic schemes (e.g. Boneh and Franklin's identity-based encryption scheme [65]), the adversary can generate some random public/private key pairs from other valid public/private key pairs, but it cannot find the private key for a given public key.

A. THE PROPOSED MP-NIKE AS A BROADCAST ENCRYPTION SCHEME

In this section, we propose a broadcast encryption scheme based on the proposed MP-NIKE protocol. The proposed MP-NIKE protocol can be used to construct practical broadcasting encryption with low computation cost and low message overhead. The proposed scheme consists of three phases as below [21].

Brod_Setup(η, γ)**.** The broadcasting server runs *Setup(γ)* to obtain the public parameters PP and msk. Then, it generates η public/private key pairs by runing *KeyGen(PP, msk)* procedure and saves them in its database. Finally the broadcasting server gives public/private key pairs to η users.

Brod_Encrypt(W, PP, M)**.** Suppose the broadcasting server wants to encrypt a message M for a group W of authorised users. At the first step, it obtains a private key d_i of one user U_i where $U_i \in W$ from its database. Now the broadcasting server runs *SharedKey(*PP*,* d_i*,* $\{e_j : U_j \in W, j \neq i\}$*)* to obtain K_W. Then it encrypts a message M, for example by the AES algorithm, under the group key K_W and sends the resulted ciphertext $CT = Enc_{K_W}(M)$ along with the list of public keys of authorised users W over the broadcasting channel.

Brod_Decrypt(W, i, d_i, PP, CT). Any user U_i from the authorized group W computes *SharedKey(*PP*,* d_i*,* $\{e_j : U_j \in W, j \neq i\}$*)* and decrypts the ciphertext CT by using K_W.

In addition to broadcast encryption, the proposed scheme can be used in various applications such as IoT or WSN, where a set of smart devices (e.g. in the home or deployed in the countryside) want to communicate securely. In this scenario, the IoT devices may have different access permissions, and some messages must not be decryptable to some elements. Besides, when a node is compromised, it must not be able to decrypt any message. In this case, the other nodes of the network can compute a shared key for a group in which the compromised node is not a member.

VI. SECURITY ANALYSIS

In this section, we prove the security of our proposal in two steps. First, we show that our scheme is 1-resilient. Second, we demonstrate that if the proposed MP-NIKE scheme is 1-resilient then it is also full-resilient. This proof uses the random oracle model based on the adopted security model of Non-Interactive Conference Key Distribution (NICKD) of [61], which is an extension to the Bellare-Rogaway security model [62] and [63].

A. STEP 1: THE PROPOSED SCHEME IS 1-RESILIENT

Theorem 1 shows that the proposed scheme is 1-resilient and proves that given PP and one public/private key pair, the adversary is not able to obtain private key of any other user.

Theorem 1. *(The proposed scheme is 1-resilient): Let* $N = p\acute{}q\acute{}$ *be a safe RSA modulus. In addition, assume that* $\mathbb{G}$ *is a subgroup of* $\mathbb{Z}_N^*$ *of order pq, and g is a generator of* $\mathbb{G}$*. Also,* $\mathbb{G}_{\mathbb{1}}$ *is a subgroup of* $\mathbb{Z}_N^*$ *of order q and* g^p *is a generator of* $\mathbb{G}_{\mathbb{1}}$*. Under these assumptions, if the Fiat-Naor problem is intractable, then the proposed scheme is 1-resilient.*

Proof. Suppose there is an adversary A that given the public parameters PP and one public/private key pair ($e_\alpha = y_\alpha p + k_\alpha q$, $d_\alpha = g^{py_\alpha} \bmod N$), can obtain the private key of a challenge public key $e_c = y_c p + k_c q \in \mathbb{Z}_N^*$, which is equal to $d_c = g^{py_c} \bmod N$. Then, we show that there is an algorithm B that can solve an instance of the Fiat-Naor problem over the group $\mathbb{G}_{\mathbb{1}}$ with generator $g^p \bmod N$.

Phase 1. Let $e_r = (N-1)/2 = 2pq + p + q = p + (2p+1)q = p + k_r q$. Algorithm B is given the modulus of computation N as well as an instance of the Fiat-Naor problem $(y_\alpha, (g^p)^{y_\alpha})$ as input, and is asked to output the Fiat-Naor private key for a specific public key y_c (which is supposed to be $g^{py_c} \bmod N$).

Algorithm B first transforms the given Fiat-Naor key pair to a NIKE key pair. To do this, it computes $e_\alpha = e_r y_\alpha = p(y_\alpha) + q(k_r y_\alpha)$. Note that $g^{py_\alpha} \bmod N$ is the corresponding NIKE private key for the public key e_α. Moreover, algorithm B changes the given challenge y_c to a valid public key of the NIKE scheme $e_c = e_r y_c = p(y_c) + q(y_c(k_r))$. Finally it sends e_α, g^{py_α} and e_c as a valid key pair and a challenge public key in the proposed NIKE scheme to the adversary A.
Phase 2. Suppose the adversary A is can compute the private key $d_c = g^{py_c} \bmod N$ and sends it back to B. It allows B to solve the given instance of the Fiat-Naor problem. In detail, B outputs $g^{py_c} \bmod N$, which is the answer to the given Fiat-Naor problem. ■

In Theorem 2, we prove that in the proposed scheme an adversary that has access to several public/private key pairs does not have any advantage over the adversary that has access to only one public/private key pair, public parameters and the generator g^p. Note that since the adversary can compute $g^p \bmod N$, including $g^p \bmod N$ in the public parameters PP does not cause any security flaw to the scheme.

B. STEP 2: THE PROPOSED SCHEME IS FULL-RESILIENT

In Theorem 2, we prove that the proposed scheme is full-resilient in the random oracle model. In other words, it formally shows that in the proposed solution no coalition of adversaries can obtain other users' private keys or compute the shared key of some group $W\acute{}$ of which they are not valid members. Note that if the adversary can compute the private key of any member of the group, it will be able to calculate the shared key of the group too. Thus, Theorem 2 also proves that the adversary is not able to compute the private key for another specific public key.

Theorem 2. *In the proposed MP-NIKE scheme, suppose that the hash function* H *is modelled as a random oracle* O*. Let* A *be an adversary who is allowed to ask* q_c *queries from the* Corrupt *oracle. Besides s/he has advantage* ϵ *to distinguish a random string from the shared key of a group* W_c *of size s and has no access to the private keys of the group members. Then there is an algorithm* B *that has at least an advantage of* $\frac{\epsilon}{e(q_c+1)}$ *against the proposed scheme given only one public/private key pair.*

Proof. Suppose there is an adversary A that given the public parameters PP and several public/private key pairs can break the proposed scheme. Then we construct an algorithm B that can break the security of the proposed scheme given only one public/private key pair (which contradicts Theorem 1). Suppose there is an algorithm C which gives the public parameters PP and a public/private key pair to algorithm B as input and asks this algorithm to compute the shared key for a specific group W_c. Then, the algorithm B as the challenger for A gives PP to A and tries to respond to Corrupt queries issued by A. Then the algorithm B tries to find the shared key for the group W_c from the messages of adversary A. In other words, the algorithm B is challenged by a set W_c of s public keys and has to output the shared key K_{W_c} as the response to this challenge (Note that K_{W_c} is the shared key which the owners of these public keys can compute in the proposed MP-NIKE scheme). Algorithm B plays the next game with adversary A who can calculate the shared key of some group W^* if it is allowed to receive q_c private keys corresponding to q_c public keys chosen by itself. In this game algorithm B, with some probability, responds to the Corrupt queries issued by A and attempting to provide the q_c private keys requested by A. Moreover, adversary A is forced to obtain its required hash values only through sending requests to the random oracle which is controlled by algorithm B. Thus, by storing and using the values that the adversary A has asked for their hashes, the algorithm B would find the key corresponding to group W_c.

For the sake of simplicity, we suppose that before starting the game the adversary A must commit to the size of the group W^*, which it wants to attack. For computing the shared key for this specific group, the algorithm B as the challenger plays the following game with the adversary A:

1) *Commit.* At the first step, the adversary chooses s, which is the size of the group W^*, and then commits to s.

2) *Initialization.* Let $N = \acute{p}\acute{q}$ be a safe RSA modulus, $\mathbb{G}$ be a subgroup of $\mathbb{Z}_N^*$ of order pq. The algorithm B sends s to the algorithm C and then C gives the public parameters $PP = \{N, g^p \bmod N, H\}$, a challenge set $W_c = \{e_{c_1}, e_{c_2}, \ldots, e_{c_s}\}$, where $e_{c_i} = y_{c_i}p + k_{c_i}q$ for $i = 1, 2, \ldots, s$ are valid public keys, as well as a public/private key pair ($e_\alpha = y_\alpha p + k_\alpha q, d_\alpha = g^{py_\alpha} \bmod N$) to the algorithm B as input.
Let $e_r = (N-1)/2 = 2pq + p + q = p + (2p+1)q = p + k_r q$, so $(e_r, g^p \bmod N)$ is a valid public/private key pair. Finally, the algorithm B sends the public parameters PP to the adversary A.
Throughout the next phase, we use a random oracle $\mathsf{O}(\cdot)$ to simulate the hash function H. The challenger controls this random oracle, and the adversary can obtain the hash value for any arbitrary string by asking it from this random oracle. In other words, the adversary A gives F_W to the random oracle and then the random oracle generates a unique random string, saves it in its database and finally sends this string to the adversary.
Random Oracle $\mathsf{O}(F_W)$. The value of F_W is given to the random oracle $\mathsf{O}(\cdot)$ as input to generate K_W which is the hash value of F_W. Upon receiving a new query, the challenger first searches its database, if it finds a tuple matching that value, it will return the respective K_W to the adversary. If, however, there is no matching tuple in its database, it will choose a random λ-bit string K_W and returns it to the adversary. Then it saves the tuple $\langle F_W, K_W \rangle$ into its database.

3) *KeyGen.* The challenger generates a list $list$, which is empty at bigining, and generates q_c sets of public keys $W_1, \ldots, W_{q_c}$, of size s each, as below.
For generating each set W_j, the challenger chooses $2s$ m-bit random integers $b_{i,j}$ and $r_{i,j}$, for $i \in \{1, \ldots, s\}$. Then, it generates tuples $\langle e_{1,j}, d_{1,j} \rangle, \ldots, \langle e_{s,j}, d_{s,j} \rangle$ as below, and saves them in its list.

$$\begin{aligned} e_{i,j} &= r_{i,j}e_r + b_{i,j}e_\alpha \\ &= p(r_{i,j} + y_\alpha b_{i,j}) + q(k_\alpha b_{i,j} + k_r r_{i,j}) \\ &= \acute{y}_{i,j}p + \acute{k}_{i,j}q, \end{aligned} \tag{14}$$

$$\begin{aligned} d_{i,j} &= (g^p)^{r_{i,j}}(g^{py_\alpha})^{b_{i,j}} \bmod N \\ &= g^{p(b_{i,j}y_\alpha + r_{i,j})} \bmod N. \end{aligned} \tag{15}$$

where, $\acute{k}_{i,j}$ and $\acute{y}_{i,j}$ are some unknown random integers.
Finally, the challenger includes the public keys $e_{1,j}, \ldots, e_{s,j}$ in W_j and sends the $q_c + 1$ sets $\langle W_1, \ldots, W_{q_c} \rangle$ and W_c in a random order to the adversary. Note that the adversary is not able to distinguish W_c from other sets.

4) *Phase 1.* The adversary is allowed to ask q_c queries from Corrupt oracle as well as one query from the Test oracle. These oracles are controlled by the challenger B.
   - *Corrupt*(e_i). Upon receiving the Corrupt(e_i) query, where $e_i \in W_1 \cup W_2 \ldots, W_{q_c} \cup W_c$, the challenger searches its list $list$ to find the respective set W_j, where $e_i \in W_j$. If $W_j = W_c$ rejects the query and the game is finished (because in this case, B is unable to compute the corresponding private key). Otherwise, the challenger retrieves $\langle e_i, d_i \rangle$ from $list$ and sends d_i back to the adversary.
   - *Test*(W^*). When the adversary decides to end Phase 1, it chooses one of $q_c + 1$ sets of public keys, such that for all $e_i \in W^*$, e_i must appear in none of the Corrupt(e_i) queries. Then the adversary sends Test(W^*) to the challenger. After receiving this query, the challenger searches its list $list$ to find respective set W_j. If $W_j \neq W_c$ it sets $b = 0$, reject the query and terminates the game. Since it means that challenger can compute the shared key K_{W^*} by itself.

TABLE 2: Comparison of NIKE schemes in terms of security and efficiency

Scheme	Security	Computational complexity	Based on	Memory complexity
Interactive Two-Party key agreement [19]	Secure	3.29 s in single-hop network	ECC-implicit X.509 certificates^{+}	$O(1)$
Interactive Two-Party key agreement [19]	Secure	10.41 s in single-hop network	ECDSA-explicit X.509 certificates, DER format^{++}	$O(1)$
Interactive Two-Party key agreement [19]	Secure	14.05 s in single-hop network	ECDSA-explicit X.509 certificates, PEM format $^{++}$	$O(1)$
GGH [38]	Not proven & broken by [41]	More than 33 s for 19 parties***	Lattice	$O(\lambda^5 \log(\lambda))$ †
GGHLite [39], [40]	Not proven & broken by [41]	1.75 s for 7 parties on 16-core CPU	Lattice	$O(\lambda \log^2(\lambda))$
GGH15 [42]	Not proven & broken by [43]	More than 33 s for 19 parties***	Lattice	$\Omega(d^5\lambda^2 \log^4(d\lambda))$ (d is diameter of graph)
CLT13 [44]	Not proven & broken by [45]	134 s for 19 parties**	CRT	$O(\lambda^2 \log(\lambda))$
5Gen extension of CLT13 [32]	Not proven	33 s for 19 parties**	CRT	$O(\lambda^2 \log(\lambda))$
CLT15 [46]	Not proven & broken by [47]	More than 33 s for 19 parties***	CRT	$O(\lambda^2 \log(\lambda))$
Rao [51]	Secure in the standard model	More than 33 s for 19 parties*	$i\mathcal{O}$	$O(poly(\lambda))$
Yamakawa et al. [52]	Secure in the standard model	More than 33 s for 19 parties*	$i\mathcal{O}$	$O(poly(\lambda))$
Boneh and Zhandry [49]	Secure in the standard model	More than 33 s for 19 parties*	$i\mathcal{O}$	$O(poly(\lambda))$
Khurana et al. [53]	Secure in the standard model	More than 33 s for 19 parties*	$i\mathcal{O}$	$O(poly(\lambda))$
Ma and Zhandry [48]	Not proven & not broken	More than 33 s for 19 parties***	CRT	$O(\lambda^3)$
Blundo et al. [56]	k-secure in the standard model	More than 33 s for 19 parties***	Symmetric polynomials	$\binom{k+t-1}{t-1}^{+++}$
Eskeland [31]	Not proven & **broken in this paper**	281 ms for 19 parties on CC2538 Chip	Root extraction in RSA modulus	$O(1)$
Proposed scheme	**Secure in the random oracle model**	281 ms for 19 parties on CC2538 Chip	Root extraction in RSA modulus	$O(1)$

* The $i\mathcal{O}$-based schemes seem to be imperactical because multilinear maps are needed to construct $i\mathcal{O}$ [38], while multilinear maps can be used to construct MP-NIKE scheme by itself [20].

** Google Compute Engine servers with a 32-core CPU at 2.5 GHz, 208 GB RAM, and 100 GB disk storage.

*** The 5Gen extension of CLT13 is the most efficient multilinear map [32], [58], so other multilinear maps take more than 33 seconds.

$^{+}$ ECDH key exchange certified using the implicit Elliptic Curve Qu-Vanstone (ECQV) certificates [59] (Using CC2538 chip and IEEE 802.15.4e technology).

$^{++}$ ECDH key exchange with public coefficients certified using the ECC Digital Signature Algorithm (ECDSA) encoded through the standard Privacy Enhanced Mail (PEM) format and the binary Distinguished Encoding Rules (DER) format (Using CC2538 chip and IEEE 802.15.4e technology).

$^{+++}$ t is the size of conference and d is the degree of resiliency.

† λ is the security parameter.

Based on the security model of Section III-D, it must send a shared key K_{W^*} to the adversary, but since by playing the rest of the game the challenger will not learn any useful information from the adversary, the challenger finishes the game. Otherwise, if $W_j = W_c$, then the challenger sets $b = 1$, which means it cannot compute K_{W^*} by itself and it must send a random string to the adversary to employ the response of the adversary for computing K_{W^*}. So, the challenger generates a random string $rand \leftarrow \{0,1\}^{\lambda}$ and sends it back to the adversary.

5) *Phase 2.* This phase is the same as Phase 1, except that the adversary does not access the Test oracle and it is not allowed to send $\mathsf{Corrupt}(e_i)$, where $e_i \in W^*$. Moreover, it is allowed to send q_H queries, to the random oracle O, aggregately in Phase 1 and 2.
6) *Guess.* In this phase, the adversary must guess a bit $b\' \in \{0,1\}$ and send it to the algorithm B. If $b\' = b$ it means that the adversary has previously obtained K_{W^*} by sending $\mathsf{O}(F_{W^*})$ to the random oracle. So, the tuple $\langle F_{W^*}, K_{W^*} \rangle$ is available in the random oracle database. Now the challenger must determine it. The challenger checks that if ${F_i}^{e_\alpha} = {d_\alpha}^{\prod_{k:U_{c_k} \in W_c} e_{c_k}}$, for all tuples $\langle F_i, K_i \rangle$ of database of random oracle O. Then it outputs $H(F_{W_i})$ as the shared key of the group W_c.

Analysis. The game will be successfully terminated if the challenger rejects none of the adversary queries. The probability of rejecting none of the q_c $\mathsf{Corrupt}$ queries is δ^{q_c}, where $\delta = \frac{sq_c}{(q_c+1)s} = \frac{q_c}{q_c+1}$ and the probability of not rejecting the Test query is at least equal to $(\frac{1}{q_c+1}) = (1-\delta)$. On the other hand, the probability of responding to all the q_H random oracle queries is 1. Hence, the game will be successfully terminated with probability $(\frac{q_c}{q_c+1})^{q_c}(1-\frac{q_c}{q_c+1})$.
By definition 5 the advantage of the adversary A to break our scheme is $Adv_A^{\mathcal{E}} = \left| Pr[b\' = b] - \frac{1}{2} \right| \leq \epsilon$. So if the game is finished successfully, the algorithm B will be able to obtain K_{W^*} from the records of the random oracle database. It means that if the adversary is able to break our scheme with advantage ϵ, then the challenger can break the security of the proposed scheme with only one public/private key pair with advantage $\epsilon\,(\frac{q_c}{q_c+1})^{q_c}(1-\frac{q_c}{q_c+1}) \approx \frac{\epsilon}{e(q_c+1)}$ and with an additional time of about q_H for retrieving K_{W^*} among the q_H records of the random oracle database.

■

The technique used in the proof of Theorem 2 is the same as the technique of analysis of Boneh and Franklin IBE scheme [65].

Note that we have assumed that the bit length of the e_i's in the real scheme and the above game are equal. The adversary is not able to distinguish between the public/private key pairs generated in the proposed scheme and the public/private key pairs generated in the above game, because algorithm B selects the $r_{i,j}$'s and $b_{i,j}$'s at random. So in $e_{i,j} = p(r_{i,j} + y_\alpha b_{i,j}) + q(k_\alpha b_{i,j} + r_{i,j}k_r)$ the values of $(r_{i,j} + y_\alpha b_{i,j})$ and $(k_\alpha b_{i,j} + r_{i,j}k_r)$ are random integers. We can, therefore, be sure that the $(e_{i,j}, d_{i,j})$'s look like uniformly distributed random numbers and the adversary cannot distinguish those from random integers.

VII. PERFORMANCE EVALUATION

In this section, we compare the proposed scheme to the previous MP-NIKE proposals. The proposed scheme is efficient in computation time and also in terms of memory complexity. It is remarkable how it can work on microcontrollers like CC2538. The CC2538 is a wireless microcontroller System-on-Chip (SoC) with 32KB on-chip RAM and up to 512KB on-chip flash. Based on the data-sheet of the Texas Instruments CC2538 [57] ch. 22, p.503, an RSA-CRT-1024 (modular exponentiation by using CRT algorithm) takes around 15.6 ms on this chip [57]. We can state that in our proposed scheme for obtaining a shared key among nineteen users at 80-bit security, each user requires to compute 18 modular exponentiations in a 1024-bit modulus which takes about $18 \times 15.6 = 281$ milliseconds. For comparison, in the 5Gen multilinear map (extension of CLT13), which is the most efficient multilinear map [32], [58], the 19-party key agreement in 80-bit security takes about 33 seconds and, as shown in Table 2, its memory complexity is polynomial in security parameter [32]. By using the GGHLite multilinear map, computing the shared key among seven users takes 1.75 seconds in a 16-core CPU [39]. The CLT13 and GGHLite are the only multilinear maps which have been implemented so far. Table 2 compares the previous proposals against the proposed scheme in terms of security, computational overhead and memory complexity. In our proposed scheme, each entity stores only one public/private key and a public parameter, which is supposed to be 4096 bits aggregately in 80-bit security. As illustrated in Table 2, among the non-interactive key exchange schemes, only the Eskeland's scheme and our proposed scheme have $O(1)$ of memory overhead.

VIII. CONCLUSION

The key distribution problem is linked to symmetric cryptography approaches and is critical in environments with a large number of users or where users are changing [66], [67]. The MP-NIKE schemes aim to face with this issue efficiently. In this paper, we present an attack against the Eskeland's MP-NIKE scheme and then we proposed a new MP-NIKE scheme which is secure against any number of colluding users.

In terms of performance, our proposal is efficient in its various phases, including setup, key generation, deriving a shared key and updating the shared key after adding a new user. Besides, the practical applicability of our proposal is clear. In particular, we can use the proposed scheme in various cryptographic applications like broadcast encryption and secure group communication for IoT and WSNs.

As showed and proven in Section VI, our proposed scheme bases its security on the intractability of the Fiat-Naor problem, which is equal to the root extraction modulo a composite number. While the security of the proposed MP-NIKE scheme relies on the random oracle model, an attractive future work will be to introduce a new MP-NIKE proposal which is secure in the standard model.

Finally, we would like to highlight that cybersecurity issues from the full plethora of IoT devices may be the next big nightmare for information and communications technology security administrators. We hope this contribution will increase the security of these devices and solve some of their problems.

IX. APPENDIX

Based on Corollary **??** computing y_c from $e_c = y_c p + k_c q$ is interactible.

Corollary. *In the proposed NIKE scheme, given a public/private key pair* $(e_\alpha = y_\alpha p + k_\alpha q, d_\alpha = g^{py_\alpha} \bmod N)$ *the adversary cannot compute the secret* y_{c_i} *corresponding to challenged public key* $e_{c_i} = y_{c_i} p + k_{c_i} q$

Proof. We prove the Lemma by contradiction. Suppose the adversary can compute y_{c_i} for some public key $e_{c_i} = y_{c_i} p + k_{c_i} q$. Then, since g^p is known to the adversary, s/he can compute $d_{c_i} = g^{py_{c_i}} \bmod N$. The above contradicts the 1-resiliency of the proposed NIKE scheme proved in Theorem 1. ■

In Theorem 1 we proved that the proposed MP-NIKE is 1-resilient, In Lemma **??** we show that adding $g^p \bmod N$ as public parameter does not violate to security of scheme. Based on Corollary **??** computing y_c from $e_c = y_c p + k_c q$ is interactible. So, given e_c and g^p computing $g^{py_c} \bmod N$ is intractible too.

Lemma 4. *Let* $g^p \bmod N$ *be a generator of subgroup* $\mathbb{G}$ *of* $\mathbb{Z}_N^*$ *of order* q*. Given* $g^p \bmod N$, $e_\alpha = y_\alpha p + k_\alpha q, d_\alpha = g^{py_\alpha} \bmod N$ *and* $e_c = y_c p + k_c q$ *as input computing* $d_c = g^{py_c} \bmod N$ *is interactible.*

Proof. Suppose the adversary A can compute $d_c = g^{py_c} \bmod N$ then we show that there is an algorithm B which can compute $d_c = g^{py_c} \bmod N$ just by having $g^p \bmod N$ and $e_c = y_c p + k_c q$ as input. Let $e_r = (N-1)/2 = 2pq + p + q = p + (2p+1)q = p + k_r q$ and $x \in_R \mathbb{G}$. Given $g^p \bmod N$ we can compute a valid public/private key pair as bellow

$$e_x = xe_r = xp + xk_r q \tag{16}$$

$$d_x = (g^p)^x = g^{xp} \bmod N \tag{17}$$

Suppose there is a adversary A which can compute $g^{py_c} \bmod N$. So we give $g^p \bmod N$, $e_x = y_x p + k_x q, d_x = g^{py_x} \bmod N$ and $e_c = y_c p + k_c q$ to the adversary A. If adversary A can compute $g^{py_c} \bmod N$ the Corollary **??** will be negative. ■

REFERENCES

[1] D. R. Stinson, M. B. Paterson, Cryptography: Theory and Practice, 4th ed., CRC Press, 2019.

[2] Y. Zhang, Y. Xiang, T. Wang, W. Wu, J. Shen. An over-the-air key establishment protocol using keyless cryptography. Future Generation Computer Systems (79-1), 2018, pp. 284-294.

[3] Y. Li, Y. Yu, B. Yang, G. Min, H. Wu. Privacy preserving cloud data auditing with efficient key update. Future Generation Computer Systems (78-2), 2018, pp. 789-798.

[4] E. S.V. Freire, D. Hofheinz, E. Kiltz and K. G. Paterson. Non-Interactive Key Exchange. Proc. Int. Conf. Public Key Cryptography-PKC, LNCS 7778, 2013, pp. 254-271.

[5] J. Becerra, D. Ostrev, and M. Skrobot. Forward Secrecy of SPAKE2. IACR Cryptology ePrint Archive, Report 2015/1037, 2015, https://eprint.iacr.org/2019/351.pdf.

[6] D. Wang, P. Wang, W. C. Chenyu. Efficient Multi-Factor User Authentication Protocol with Forward Secrecy for Real-Time Data Access in WSNs. ACM Transactions on Cyber-Physical Systems, 3(1), 2019, pp. 1-25.

[7] D. Wang, w. Li and P. Wang. Measuring Two-Factor Authentication Schemes for Real-Time Data Access in Industrial Wireless Sensor Networks. IEEE Transactions on Industrial Informatics, 14(9), 2018, pp. 4081-4092.

[8] S. Athmani, A. Bilami, D. E. Boubiche. EDAK: An Efficient Dynamic Authentication and Key Management Mechanism for heterogeneous WSNs. Future Generation Computer Systems (92), 2019, pp. 198-210.

[9] R.l Amin, SK H. Islam, G. P. Biswas, M. S. Obaidat. A robust mutual authentication protocol for WSN with multiple base-stations. Ad Hoc Networks (75-76), 2018, pp. 1-18.

[10] H. Guo, Y. Zheng, X. Li, Z. Li, C. Xia. Self-healing group key distribution protocol in wireless sensor networks for secure IoT communications. Future Generation Computer Systems (89), 2018, pp. 713-721.

[11] K. Mahmood, X. Li, S. A. Chaudhry, H. Naqvi, S. Kumari, A. K. Sangaiah, J. J.P.C. Rodrigues. Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure. Future Generation Computer Systems (88), 2018, pp. 491-500.

[12] J. Shen, M. Sangman and I. Chung. A Novel Key Management Protocol in Body Area Networks, ICNS: The 7th International Conference on Networking and Services, 2011, pp. 246-251.

[13] Y. Li. Design of a Key Establishment Protocol for Smart Home Energy Management System, 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), 2013, pp. 88-93.

[14] L. Veltri, S. Cirani, S. Busanelli and G. Ferrari. A novel batch-based group key management protocol applied to the Internet of Things. Ad Hoc Networks, 11(8), 2013, pp. 2724-2737.

[15] A. K. Das, M. Wazid, A. R. Yannam, J.J.P.C. Rodrigues, Y. Park. Provably Secure ECC-Based Device Access Control and Key Agreement Protocol for IoT Environment. IEEE Access, vol. 7, 2019, pp. 55382-55397.

[16] Z. Xu, C. Xu, W. Liang, J. Xu and H. Chen. A Lightweight Mutual Authentication and Key Agreement Scheme for Medical Internet of Thing. IEEE Access, vol. 7, 2019, pp. 53922-53931.

[17] Z. Li and D. Wang. Achieving One-Round Password-Based 2 Authenticated Key Exchange over Lattices. IEEE Transactions on Services Computing, 2019(8), pp.1-19.

[18] S. Paliwal. Hash-Based Conditional Privacy Preserving Authentication and Key Echange Protocol Suitable for Industrial Internet of Thing. IEEE Access, vol. 7, 2019, pp. 136073-136093.

[19] S. Sciancalepore, G. Piro, G. Boggia, G. Bianchi. Public Key Authentication and Key Agreement in IoT Devices With Minimal Airtime Consumption. IEEE Embedded Systems Letters. 9(1), 2017. pp. 1-4.

[20] D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. Contemporary Mathematics, 2002, 324, pp. 71-90.

[21] D. Boneh, C. Gentry, B. Waters. Collusion Resistant Broadcast Encryption With Short Ciphertexts and Private Keys. Proc. Int. Conf. Advances in Cryptology-Crypto '05, LNCS 3621, 2005, pp. 258-275.

[22] G.N. Purohit and A.S. Rawat. Revocation and Self-Healing of keys in Hierarchical Wireless Sensor Network. Int. J. Comput. Sci. Inf. Technol, 2(6), 2011, pp. 2909-2914.

[23] Q. Jiang, S. Zeadally, J. Ma and D. He. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. IEEE Access, vol. 5, 2017, pp. 3376-3392.

[24] I. Mansour, G. Chalhoub and P. Lafourcade. Key Management in Wireless Sensor Networks. Journal of Sensor and Actuator Networks, 4(3), 2015, pp. 251-273.

[25] D. Liu, and P. Ning. Location-based pairwise key establishment for static sensor networks. In 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003, pp.52-61.

[26] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In IEEE Symposium on Research in Security and Privacy, 2003, pp.1-17.

[27] S. Zhu, S. Xu, S. Setia, and S. Jajodia. Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach. In 11th IEEE International Conferenceon Network Protocols (ICNP03). 2003, pp. 326-333.

[28] S. Camtepe, and B. Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. IEEE/ACM Transactions on Networking, 15(2), 2007, pp. 346-358.

[29] Z. Lin, Y. Qu, L. Jing, B. Zhao. Compromised Nodes in Wireless Sensor Network. Advanced Web and Network Technologies, and Applications, 2006, pp. 224-230.

[30] M. Kim and H. Cho. Energy-Efficient Detection of Compromised Nodes in Wireless Sensor Networks. International Journal of Applied Engineering Research (13), 2018, pp. 5589-5599.

[31] S. Eskeland. Non-interactive secure multi-party key establishment. Tatra mountains, 2014, 60, (1), pp. 47-55.

[32] K. Lewi, A. J. Malozemoff, D. Apon, B. Carmer, A. Foltzer, D. Wagner, D. W. Archer, D. Boneh, J. Katz and M. Raykova. 5Gen: A Framework for Prototyping Applications Using Multilinear Maps and Matrix Branching Programs. In Cryptology ePrint Archive, Report 2016/619, 2016.

[33] A. Fiat and M. Naor. Broadcast encryption. Proc. Int. Conf. Advances in Cryptology-CRYPTO, LNCS 773, 1993, pp. 480-491.

[34] W. Diffie and M. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, 1976, 22(6), pp. 644-654.

[35] D. Cash, K. Kiltz and V. Shoup. The Twin Diffie-Hellman Problem and Applications. Proc. Int. Conf. Advances in Cryptology-EUROCRYPT, LNCS 4965, 2008, pp. 127-145.

[36] E. S.V. Freire, J. Hesse and D. Hofheinz. Universally Composable Non-Interactive Key Exchange. Proc. Int. Conf. Security and Cryptography for Network, LNCS 8642, 2014, pp. 1-20.

[37] A. Joux. A one round protocol for tripartite Diffie-Hellman. Proc. Int. Conf. Algorithmic Number Theory Symposium, LNCS 1838, 2000, pp. 385-394.

[38] S. Garg, C. Gentry and S. Halevi. Candidate multilinear maps from ideal lattices. Proc. Int. Conf. Advances in Cryptology-EUROCRYPT, LNCS 7881, 2013, pp. 1-17.

[39] A. Langlois, D. Stehl´e and R. Steinfeld. GGHLite More Efficient Multilinear Maps from Ideal Lattices. Proc. Int. Conf. Advances in Cryptology-EUROCRYPT, LNCS 8441, 2014, pp. 239-256.

[40] M. R. Albrecht, C. Cocis, F. Laguillaumie and A. Langlois. The whole alphabet (and then some) agree on a key in one round: making multilinear maps practical. In IACR Cryptology ePrint Archive, Report 2014/928, 2014, http://eprint.iacr.org/2014/928.

[41] Y. Hu and H. Jia. Cryptanalysis of GGH map. Technical report, IACR Cryptology ePrint Archive, Report 2015/301, 2015, http://eprint.iacr.org/2015/301.

[42] C. Gentry, S. Gorbunov and S. Halevi. Graph-induced multilinear maps from lattices. Proc. Int. Conf. Theory of Cryptography(TCC), LNCS 9015, 2015, pp. 498-527.

[43] J. S. Coron, M. S. Lee, T. Lepoint and M. Tibouchi. Cryptanalysis of GGH15 Multilinear Maps, IACR Cryptology ePrint Archive, Report 2015/1037, 2015, http://eprint.iacr.org/2015/1037.

[44] J. S. Coron, T. Lepoint and M. Tibouchi. Practical multilinear maps over the integers. Proc. Int. Conf. Advances in Cryptology-CRYPTO, LNCS 8042, 2013, pp. 476-493.

[45] J. H. Cheon, K. Han, C. Lee, H. Ryu and D. Stehl'e. Cryptanalysis of the multilinear map over the integers. Proc. Int. Conf. Advances in Cryptology-EUROCRYPT, LNCS 9056, 2015, pp. 3-12.

[46] J. S. Coron, T. Lepoint and M. Tibouchi. New Multilinear Maps over the Integers. Proc. Int. Conf. Advances in Cryptology - CRYPTO 2015, LNCS 9215, 2015, pp. 267-286.

[47] B. Minaud and P. A. Fouque. Cryptanalysis of the New Multilinear Map over the Integers. Cryptology ePrint Archive, Report 2015/941, 2015.

[48] F. Ma, and M. Zhandry. The MMap strikes back: obfuscation and new multilinear maps immune to CLT13 zeroizing attacks. In Theory of Cryptography (TCC18), 2018, pp. 513-543.

[49] D. Boneh and M. Zhandry. Multi-party key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. Proc. Int. Conf. Advances in Cryptology-CRYPTO, LNCS 8616, 2014, pp. 480-499.

[50] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. Proc. Int. Conf. Foundations of Computer Science(FOCS), 2013, pp. 40-49.

[51] V. Rao. Adaptive multi-party non-interactive key exchange without setup in the standard model. IACR Cryptology ePrint Archive, Report 2014/910, 2014, http://eprint.iacr.org/2014/910.

[52] T. Yamakawa, S. Yamada, G. Hanaoka and N. Kunihiro. Self-bilinear Map on Unknown Order Groups from Indistinguishability Obfuscation and Its Applications. Proc. Int. Conf. Advances in Cryptology-CRYPTO, LNCS 8617, 2014, pp. 90-107.

[53] D. Khurana, V. Rao and A. Sahai. Multi-Party Key Exchange for Unbounded Parties from Indistinguishability Obfuscation. Proc. Int. Conf. Advances in Cryptology-ASIACRYPT, LNCS 9452, 2015, pp. 52-57.

[54] Y. Chen, Q. Huang and Z. Zhang. Sakai-Ohgishi-Kasahara Identity-Based Non-Interactive Key Exchange Revisited and More. In International Journal of Information Security. 15(1), 2016, pp. 15-33.

[55] M. Zhandry. How to Avoid Obfuscation Using Witness PRFs. In Theory of Cryptography, 2015, pp 421-448.

[56] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and Moti Yung. Perfectly Secure Key Distribution for Dynamic Conferences. In Inf. Comput. 146(1), 1998, pp. 1-23.

[57] CC2538 System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee/ZigBee IP. https://www.ti.com/lit/ug/swru319c/swru319c.pdf, April 2012.

[58] F. Ma, and M. Zhandry. New Multilinear Maps from CLT13 with Provable Security Against Zeroizing Attacks. In Cryptology ePrint Archive, Report 2017/946, 2014.

[59] Explaining Implicit Certificates. Technical report, Certicom, Available at:https://www.certicom.com/content/certicom/en/code-and-cipher/explaining-implicit-certificate.html, 2004.

[60] A. Menezes, P. Van Oorschot and S. Vanstone. Handbook of applied cryptography, CRC Press, Boca Raton, Florida, 1996.

[61] R. Safavi-Naini and SH. Jiang. Non-Interactive Conference Key Distribution and Its Applications. Proc. Int. Conf. ASIACCS '08 Proceedings of the 2008 ACM symposium on Information, computer and communications security, 2008, pp. 271-282.

[62] M. Bellare and P. Rogaway. Entity authentication and key distribution. In Advances in Cryptology-CRYPTO, 1993, pp. 232-249.

[63] E. Bresson, O. Chevassut, D. Pointcheval and J. Quisquater. Provably Authenticated Group Di±e-Hellman Key Exchange. In ACM CCS'01. 2001, pp.255-264.

[64] A. Menezes, P. V. Oorschot and S. Vanstone. Handbook of applied cryptography. 1996, pp. 65-66.

[65] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. Proc. Int. Conf. Advances in Cryptology-CRYPTO, LNCS 2139, 2001, pp. 213-229.

[66] P. Vijayakumar, Victor Chang, L. Jegatha Deborah, Bharat S. Rawal Kshatriya. Key management and key distribution for secure group communication in mobile and cloud network. Future Generation Computer Systems (84), 2018, pp. 123-125.

[67] A. S. Reegan, E. Babura. Polynomial and multivariate mapping-based triple-key approach for secure key distribution in wireless sensor networks. Computers & Electrical Engineering (59), 2017, pp. 274-290.