Towards fully collusion-resistant ID-based establishment of pairwise keys

Oscar Garcia-Morchon*, Ludo Tolhuizen*, Domingo Gomez†, Jaime Gutierrez†
*Philips Group Innovation, Research
High Tech Campus 34, Eindhoven, Netherlands
{oscar.garcia, ludo.tolhuizen}@philips.com
† University of Cantabria
Avda. de los Castros s/n, Santander, Spain
{domingo.gomez, jaime.gutierrez}@unican.es

***Abstract*—Usually a communication link is secured by means of a symmetric-key algorithm. For that, a method is required to securely establish a symmetric-key for that algorithm. This old key establishment problem is still relevant and of paramount importance both in existing computer networks and new large-scale ubiquitous systems comprising resource-constrained devices. Identity-based pairwise key agreement allows for the generation of a common key between two parties given a secret keying material owned by the first party and the identity of the second one. However, existing methods, e.g., based on polynomials, are prone to collusion attacks.**

In this paper we discuss a new key establishment scheme aiming at fully collusion-resistant identity-based symmetric-key agreement. Our scheme, the HIMMO algorithm, relies on two design concepts: Hiding Information and Mixing Modular Operations. Collusion attacks on schemes from literature cannot readily be applied to our scheme; our security analysis further shows that HIMMO's design principles prevent an attacker from performing a number of attacks. Also, the simple logic of the HIMMO algorithm allows for very efficient implementations in terms of both speed and memory. Finally, being an identity-based symmetric-key establishment scheme, HIMMO allows for efficient real-world key exchange protocols.

***Keywords*-Key distribution and establishment, polynomials, identity-based cryptography**

I. Introduction

This paper deals with the classical problem of key establishment. As in previous works [4],[8],[14], we focus on an *identity-based* (ID-based) scheme for symmetric-key agreement between pairs of devices in a network. That is, each node in the network has an identifier, and a Trusted Third Party (TTP) provides it with secret keying material - linked to the device identifier - in a secure way. A node that wishes to communicate with another node uses its own secret keying material and the identity of the other node to generate a common pairwise key.

Existing ID-based symmetric-key agreement schemes are prone to *collusion attacks*: secret keying material of various nodes can be combined in order to obtain information on the secret key generated by a pair of (other) nodes. This combining can be performed by colluding legitimate owner(s) of the nodes, or by an attacker who has compromised some nodes and obtained their secret keying material. Existing schemes [4],[8],[14] allow for efficient collusion attacks (see Section II), which implies that to prevent successful attacks with relatively few devices, much secret keying material must be stored in each node, which may be problematic in real-world applications since it increases CPU and storage needs.

This paper discusses a new ID-based key establishment scheme allowing for efficient operation – with respect to both the amount of stored keying material and the key computation time, which is especially relevant for resource-constrained devices – while it is based on mathematical problems for which the collusion attacks on the schemes from literature cannot readily be applied. We hope that our scheme, the HIMMO algorithm, and its underlying design principles can be a step towards fully collusion-resistant identity-based establishment of symmetric-keys.

***Definition** 1 (Fully collusion-resistant): An identity-based symmetric-key establishment scheme is* fully collusion-resistant *if for any set of colluding*

nodes no bit of a key shared by non-colluding nodes can be guessed with a probability higher than $1/2$ *in polynomial time.*

We believe that a crypto algorithm with this property can find many applications. In particular, HIMMO is attractive for scenarios with a trusted central party managing a network of devices such that the devices are resource-constrained and/or time plays a key role during key establishment. The reason is that the simplicity of HIMMO operation (based on polynomials) makes it highly efficient. Examples of applications with resource-constrained devices to which HIMMO can be applied are wireless sensor networks, RFID tags, or NFC. More specific examples are the 6LoWPAN and CoAP protocols that are being standardized by IETF and will allow for the so called IP-based Internet of Things in which thousands of resource-constrained devices will collect and exchange data with each other over the Internet [7]. Currently CoAP mandates the usage of DTLS that can be configured to work with a pre-shared key (PSK) as specified in RFC 4279. HIMMO might be easly integrated there so that the PSK is the one generated by HIMMO and the PSK-hint is the HIMMO identifier. Applications with strict time requirements include car-to-car communication and the setup of a secure call. Car-to-car communication has strong authentication needs, and moreover, authentication needs to be performed in a very short period of time. The same occurs for the setup of a secure call in which we do not want to increase delays or drain the smart phone batteries.

The rest of this paper is organized as follows. In Section II we give an overview of related work. Section III describes our HIMMO algorithm. In Section IV we discuss the design principles, underlying mathematical problems, and possible ways of configuring the algorithm. Section V presents the security analysis of HIMMO. We present our conclusions in Section VI. Appendix A contains a proof of Theorem 1, while Appendix B gives some background on the lattice-theoretic results used in Section V.

II. Previous identity-based symmetric-key distribution schemes

Matsumoto and Imai [8] give a nice description of the key distribution problem, and provide a solution that serves as a base for many other schemes from literature. They propose that a TTP chooses a secret function $f(x, y)$ that is *symmetric*, that is, $f(x, y) = f(y, x)$. The variables x and y are taken from a set of node identifiers $\mathcal{I}$, and the output from f is the key. The secret keying material for the node with identifier η is a function $KM_\eta(x) = f(x, \eta)$ which is such that $KM_\eta(\eta') = f(\eta, \eta')$ for all $\eta' \in \mathcal{I}$. As f is symmetric, the keys generated by two nodes for communicating with each other are equal.[1]

In [4], Blundo *et al.* choose the secret function $f(x, y)$ to be a symmetric bi-variate polynomial over a finite field of degree α in each variable; the identifiers are considered as field elements as well. Blundo *et al.* show that their scheme offers information-theoretic security as long as an attacker knows the secret keying material of α or less nodes. However, $\alpha+1$ colluding nodes can obtain the root keying material by simple Lagrange interpolation.

In order to avoid the simple interpolation attack, Zhang et al. [14] proposed a "noisy" version of the scheme of Blundo et al. [4]. Their basic idea is to provide node η with a polynomial $KM_\eta(x)$ that is "close" to, but not exactly the same as $f(x, \eta)$. Nodes η and η' can compute $KM_\eta(\eta')$ and $KM_{\eta'}(\eta)$ as before; these values are no longer equal, but because they are close they can be used to generate a shared key. We now describe the main steps:

- The TTP chooses a random symmetric, polynomial $f \in \mathbb{Z}_p[x, y]$ of degree α in each variable and a noise bound r with $r < p$. It also chooses at random univariate "noise" polynomials $g(y)$ and $h(y)$ of degree α over $\mathbb{Z}_p$. Next, it determines

$$\mathcal{N} := \{\eta \in \mathbb{Z}_p : g(\eta), h(\eta) \in [0, r]\}$$

- Each node is given an identifier from $\mathcal{N}$. For each node $\eta \in \mathcal{N}$, the TTP chooses a random bit $b_\eta \in \{0, 1\}$ and provides node η the univariate polynomial

$$KM_\eta(x) = f(x, \eta) + b_\eta g(x) + (1 - b_\eta)h(x).$$

- A node η wishing to communicate with node η' computes $KM_\eta(\eta')$ and takes its $\ell - r$ most significant bits as key (where ℓ is such that $2^{\ell-1} < p \leq 2^\ell$). It sends $\mathcal{H}(KM_\eta(\eta'))$ to node η', where $\mathcal{H}$ is a hash-function. Node η'

[1] Matsumoto and Imai in fact consider the more general situation that any group of t nodes must generate a common key; we restrict ourselves to the case $t = 2$.

computes three numbers, namely the $\ell - r$ most significant bits of $KM_{\eta'}(\eta)$, of $KM_{\eta'}(\eta)+2^r$, and of $KM_{\eta'}(\eta)-2^r$. Next it takes as key the number for which the hash-value agrees with the received hash-value $\mathcal{H}(KM_\eta(\eta'))$.

Albrecht et al. [1] designed an efficient collusion attack on the scheme of Zhang *et al.* based on error-correcting techniques, that works if the $4\alpha+1$ nodes collude. They also provide an attack that works with 3α colluding nodes, but has time complexity $\mathcal{O}(r)$. Then, they suggested a generalized scheme based on adding more noise:

- *The TTP also chooses a natural number u such that $4ur < p$ and, for each node $\eta \in \mathcal{N}$, integers a_η,b_η and c_η such that $a_\eta, b_\eta \in [-u, u]$ and $c_\eta \in [-ur, ur]$, and gives node η the univariate polynomial:*

$$KM_\eta(x) = f(x, \eta) + a_\eta g(x) + b_\eta h(x) + c_\eta.$$

Albrecht et al. also provided an attack on this new cryptography protocol of time complexity $O(\alpha^3 + 8\alpha u^3)$, and requiring only $\alpha + 3$ compromised nodes. Their attack consists of two steps. In the first step, by means of linear algebra methods, they recover the linear vector space generated by the univariate polynomials $g(x)$ and $h(x)$. In the second step, they use lattice reduction techniques to recover f, knowing the polynomials g and h.

III. The HIMMO algorithm

In this section, we describe our HIMMO algorithm for ID-based symmetric-key establishment that aims at achieving the desired full collusion resistance. It relies on two new design principles:

1) **Hiding of information**, e.g., by adding noise that is completely independent and random, for each node. This is similar to what is done by Zhang *et al.* [14], but they have only two possible noise contributions (the noise polynomials g and h, see previous section).
2) **Mixing of modular operations** by using m symmetric bi-variate polynomials with coefficients in the integers modulo q_i for generating the secret keying material.

A key difference with all previous schemes [1], [4], [14] is that the modules $q_1, \ldots, q_m$ are kept secret and are only known to the TTP, *not* to the nodes. The nodes do know, however, that each module differs a multiple of 2^b from a known constant N.

In our description, we use the following notation. For each real x, we denote by $\lfloor x \rfloor$ the value of x rounded downwards to the closest integer, that is,

$$\lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \leq x\}.$$

For integer a and integer $p \geq 2$, we denote by $\langle a \rangle_p$ the remainder of dividing a by p. Stated differently,

$$0 \leq \langle a \rangle_p \leq p - 1 \text{ and } a \equiv \langle a \rangle_p \bmod p.$$

Our ID-based symmetric-key establishment scheme comprises three phases:

1. System initialization

The TTP selects four public positive integers m, b, N and α satisfying:

$$2^{(\alpha+2)b-1} < N \leq 2^{(\alpha+2)b}.$$

The TTP also generates the following private material:

- m distinct positive integers $q_1, \ldots, q_m$ of the form $q_i = N - 2^b\beta_i$ where $1 \leq \beta_i \leq 2^b - 1$;
- m symmetric bi-variate polynomials $f_1(x, y), \ldots, f_m(x, y)$, all of degree at most α in each variable, such that for $i = 1, \ldots, m$, the polynomial f_i is in $\mathbb{Z}_{q_i}[x, y]$.

For $1 \leq i \leq m$, we write

$$f_i(x, y) = \sum_{j=0}^{\alpha} f_{i,j}(y)x^j \text{ with } f_{i,j}(y) \in \mathbb{Z}_{q_i}[y] \subset \mathbb{Z}_{q_i}[x, y].$$

2. Node registration

For each node $\eta \in \{1, \ldots, 2^b - 1\}$, that wants to register, the TTP selects $\alpha+1$ random integers $\epsilon_{\eta,j}$ (from now on called noise) satisfying the following equation:

$$|\epsilon_{\eta,j}| < 2^{(\alpha+1-j)b-2}, j = 0, \ldots, \alpha. \quad (1)$$

The TTP provides node η with the secret keying material coefficients $KM_{\eta,0}, KM_{\eta,1}, \ldots, KM_{\eta,\alpha}$, defined as:

$$KM_{\eta,j} = \left\langle \sum_{i=1}^{m} \langle f_{i,j}(\eta) \rangle_{q_i} + 2^b\epsilon_{\eta,j} \right\rangle_N. \quad (2)$$

3. Operational phase: key agreement

Node η generates its key with η' as:

$$K_{\eta,\eta'} = \left\langle \left\langle \sum_{j=0}^{\alpha} KM_{\eta,j}\eta'^{j} \right\rangle_{N} \right\rangle_{2^b} \quad (3)$$

With explicit examples, it can be shown that $K_{\eta,\eta'}$ and $K_{\eta',\eta}$ are not necessarily equal. It can be shown, however, that they are approximately equal, as described in the following theorem. The proof of Theorem 1 is deferred to Appendix A.

***Theorem** 1: Let $0 \leq \eta, \eta' \leq 2^b - 1$. Then we have that*

$$K_{\eta,\eta'} \in \{\langle K_{\eta',\eta} + jN \rangle_{2^b} \mid -\Delta \leq j \leq \Delta\},$$

where $\Delta = 3m + \alpha + 1$.

In order that devices η and η' agree on a common key, an additional step is performed. In this step, device η sends the value $\mathcal{H}(K_{\eta,\eta'})$ to device η', where the function $\mathcal{H}$ is such that $\mathcal{H}(i) \neq \mathcal{H}(K_{\eta,\eta'})$ for each potential key i (as indicated in Theorem 1) different from $K_{\eta,\eta'}$. In this way, η' finds the key $K_{\eta,\eta'}$ that is subsequently used to secure the communication link. An example of such a function $\mathcal{H}$ is a hash function like in [14].

IV. Design principles of HIMMO

As stated before, our HIMMO algorithm relies on two principles, namely (i) hiding of information and (ii) mixing of modular operations. Both principles further exhibit the feature that only partial knowledge on the used modules is available. This is described in more detail below.

A. Hiding of information

This design concept corresponds to the noisy polynomial interpolation problem of recovering an unknown polynomial $f(x) \in \mathbb{Z}_{q_1}[x]$ from approximate values of $f(\eta)$ at polynomially many points $\eta \in \mathbb{Z}_{q_1}$, see [11] and [12]. The case of a linear polynomial corresponds to the well known Hidden Number Problem (HNP). Among other applications, Boneh and Venkatesan in [5] found nice links between the HNP and the security of the Diffie-Hellman Key Exchange protocol.

This principle applies at two different levels in HIMMO. First, in Equation 2 we see that for each keying material coefficient $KM_{\eta,j}$, parts of the sum of the polynomial evaluations are hidden by the noisy term $2^b\epsilon_{\eta,j}$. In the specific case of HIMMO for $m = 1$, this problem can be stated as follows:

***Problem** 1: Let q_1 be a positive integer such that $q_1 = N - \beta_1 2^b$ where $2^{(\alpha+2)b-1} < N \leq 2^{(\alpha+2)b}$ and $0 \leq \beta_1 < 2^b$. Suppose for many random values $\eta \in \{0, 1, \ldots, 2^b - 1\}$, the values $f(\eta) + \epsilon_\eta 2^b$ are given, where $f(x) \in \mathbb{Z}_{q_1}[x]$ is an unknown polynomial of known degree α, and ϵ_η is a jb-bits noisy term (for some $j \in \{1, \alpha + 1\}$). The problem consists in recovering the polynomial $f(x)$ in polynomial time.*

***Remark** 1: The problem is further enhanced by the fact that the attacker does not know q_1; all he knows is N and the form of q_1.*

Second, in Equation 3 only the b least significant bits of the polynomial evaluation are used as the key. The corresponding problem can be stated as follows:

***Problem** 2: Let N be a positive integer satisfying $2^{(\alpha+2)b-1} < N \leq 2^{(\alpha+2)b}$. Suppose for many random values $\eta \in \{0, 1, \ldots, 2^b - 1\}$, the values $\langle f(\eta) \rangle_{2^b}$ are given, where $f(x) \in \mathbb{Z}_N[x]$ is an unknown polynomial of known degree α. The problem consists in guessing any bit of $\langle \langle f(\eta') \rangle_N \rangle_{2^b}$ associated with another node η' with probability larger than 1/2.*

In both cases, the aim is to hide some information so that an attacker cannot recover the polynomial. The main security issue with this design principle is that the usage of a single polynomial does not remove the underlying ring structure because the generated key is approximately equal[2] to the one generated from the original polynomial:

$$K_{\eta,\eta'} \approx \langle \langle f_1(\eta, \eta') \rangle_{q_1} \rangle_{2^b} = \langle \langle f_1(\eta', \eta) \rangle_{q_1} \rangle_{2^b} \approx K_{\eta',\eta}$$

B. Mixing of modular operations ($m \geq 2$)

In Equation 2, we see (for $m \geq 2$) a mixing of modular operations in the sum $\sum_{i=1}^{m} \langle f_{i,j}(\eta) \rangle_{q_i}$. A natural computational mathematical problem arising from the above equation is the following.

[2]Equation 3 uses modulo N reductions, where $\beta_1 << N$ is missing, while here all reductions are modulo q_1.

For each $j = 1, \ldots, \alpha$, recover m polynomials $f_{i,j}(x) \in \mathbb{Z}_{q_i}[x], i = 1, \ldots, m$ of degree at most α from the values $\sum_{i=1}^{m} \langle f_{i,j}(\eta) \rangle_{q_i}$ at polynomially many points $\eta \in \mathbb{Z}_p$, where $p = \min(q_1, \ldots, q_m)$. We have not found any results in the literature related to this problem, and there are some indications that it is a hard computational problem as we will see later. A strategy to solve a weaker version of this problem, namely to compute the integer polynomial $\sum_{i=1}^{m} f_{i,j}(x)$ when the positive integers q_i are known, will be presented in Subsection V-C.

For HIMMO the mixing of modular operations problem can be written in an even more specific manner as follows:

***Problem** 3: Let $q_1, \ldots, q_m$ be m distinct positive integer numbers such that $q_i = N - \beta_i 2^b$, where $2^{(\alpha+2)b-1} < N \leq 2^{(\alpha+2)b}$ and $0 \leq \beta_i < 2^b$. Moreover, for $i = 1, \ldots, m$, let $f_{i,j}(x) \in \mathbb{Z}_{q_i}[x]$ have degree at most α. For η in $S = \{1, .., 2^b - 1\}$, we define $H(\eta) := \langle \sum_{i=1}^{m} \langle f_{i,j}(\eta) \rangle_{q_i} \rangle_N$. Given a number N_c of pairs $(\eta, H(\eta))$, the problem consists in guessing any bit of $H(\eta')$ associated to a known input value η' with a probability higher than 1/2.*

We observe that in Problem 3, each identifier η is much smaller than N due to the specific HIMMO construction.

***Remark** 2: Problem 3 is further enhanced by the fact that the attacker does not know the modules $q_1, \ldots, q_m$; all he knows is that each q_i differs a b bit unknown integer β_i multiple of 2^b from N.*

In order to explain the idea behind this second design principle, we consider a simple special case, *viz.* that for $1 \leq i \leq m$, we have that $f_i(x, y) = A_i x^\alpha y^\alpha$ for some $A_i \in \{1, \ldots, q_i - 1\}$. Moreover, we take $N = 2^{b(\alpha+2)} - 1$ and $\epsilon_{\eta,\alpha} = 0$. We write:

$$A_i \eta^\alpha = R_{i,\eta}^{(2)} 2^{b(\alpha+2)} + R_{i,\eta}^{(1)} 2^b + R_{i,\eta}^{(0)},$$

with $0 \leq R_{i,\eta}^{(0)} \leq 2^b - 1$, $0 \leq R_{i,\eta}^{(1)} \leq 2^{b(\alpha+1)} - 1$, and $0 \leq R_{i,\eta}^{(2)} \leq 2^{\alpha b} - 1$. As $q_i = 2^{b(\alpha+2)} - \beta_i 2^b - 1$, the single non-zero coefficient $KM_{\eta,\alpha}$ of node η is given by:

$$\left\langle \sum_{i=1}^{m} \langle f_{i,\alpha}(\eta) \rangle_{q_i} \right\rangle_N = \left\langle \sum_{i=1}^{m} \langle A_i \eta^\alpha \rangle_{q_i} \right\rangle_N =$$

Now, calling

$$\mathcal{Q}_i = \left\lfloor \frac{R_{i,\eta}^{(0)} + R_{i,\eta}^{(2)}}{2^b} \right\rfloor, \quad \mathcal{R}_i = \langle R_{i,\eta}^{(0)} + R_{i,\eta}^{(2)} \rangle_{2^b},$$

we have the following

$$\left\langle \sum_{i=1}^{m} \left\langle \left(R_{i,\eta}^{(1)} + \beta_i R_{i,\eta}^{(2)} \right) 2^b + \left(R_{i,\eta}^{(0)} + R_{i,\eta}^{(2)} \right) \right\rangle_{q_i} \right\rangle_N =$$

$$\left\langle \sum_{i=1}^{m} \left\langle \left(R_{i,\eta}^{(1)} + \beta_i R_{i,\eta}^{(2)} + \mathcal{Q}_i \right) 2^b + \mathcal{R}_i \right\rangle_{q_i} \right\rangle_N \approx [3]$$

$$\left\langle \sum_{i=1}^{m} \left(R_{i,\eta}^{(1)} + \beta_i R_{i,\eta}^{(2)} + \mathcal{Q}_i \right) 2^b + \mathcal{R}_i \right\rangle_N \quad (4)$$

In this example, we observe that the modulo computations affect the $b(\alpha + 1)$ most significant bits of the keying material in a way that is dependent on β_i. By adding over i, these β_i-dependencies are mixed. We also see mixing in the b least significant bits of the keying material, as they depend on the sum of the most and least significant bits of $A_i \eta^i$. The nice aspect of the design is that the components originating from different polynomials $f_i(x, y)$ hide each other so that an attacker can only observe the sum modulo N, learning nothing about the individual components.

Thus, our HIMMO algorithm applies the second design concept by using q_i with such a form that they introduce non-linear operations when the TTP generates the secret keying material for node η from the secret bi-variate polynomials. However, the public modulus N and the secret moduli $q_1, \ldots q_m$ share a structure that allows for the generation of a b-bit key by means of Equation 3. Thus, the smart part of the cryptoblock happens in the step in which the TTP generates the keying material shares from the secret root keying material, creating a non-linear keying material structure in the most significant bits of the secret keying material coefficients as shown in the specific example in Equation 4. Later, during key establishment only the common terms of q_i, namely N, are used so that a common key can be generated mod N, i.e., without requiring knowledge of the secret terms β_i. Thus, the resulting b-bit key combines the contributions from all polynomials over different rings:

[3]The effect of the reduction module q_i due to carry propagation is limited due to the form of q_i.

TABLE I
USAGE OF THE DESIGN PRINCIPLES IN HIMMO

Protection of	Hidding Information	Mixing Modular Operations	Secret Moduli?
Root keying material	Yes	Yes	Yes, β_j can be secret
Keying material	Yes	No	No, N is public.

$$K_{\eta,\eta'} \approx \left\langle \sum_{i=1}^{m} \langle f_i(\eta,\eta')\rangle_{q_i} \right\rangle_{2^b} = \left\langle \sum_{i=1}^{m} \langle f_i(\eta',\eta)\rangle_{q_i} \right\rangle_{2^b} \approx K_{\eta',\eta}$$

C. Usage of these design principles in HIMMO

The two above design principles are applied to protect the root keying material and keying material shares. Table I summarizes where the design principles are applied and whether the module can be kept secret or not.

V. SECURITY ANALYSIS

HIMMO aims at achieving the fully collusion-resistant property by relying on Problem 1, Problem 2 and 3. In order to provide a clearer security analysis, we first introduce the threat model and classification of cryptanalysis methods on collusion-resistant systems in subsection V-A. Specific approaches for the cryptanalysis of HIMMO – summarized in Table II according to the definitions presented in subsection V-A– are described in the four subsequent subsections. In particular, *counting bounds* (Section V-B) provides arguments related to the minimum number of devices needed to attack the root keying material and a keying material share. Attacking the root keying material (Section V-C) refers to a strategy suggested by I. Shparlinski and M. Albrecht for theoretically attacking the root keying material even for $m \geq 2$. *Attacking a keying material share* (Section V-D) presents an approach in which an attacker can try to attack the keying material share of a non-compromised node. This approach is directly related to the Noisy Polynomial Interpolation Problem and existing results can be applied to assess a secure configuration for HIMMO. Section V-E discusses the feasibility of attacking the root keying material for a very specific configuration ($m = 1$) depending on the secrecy of q_1. Section V-F summarizes our findings.

A. Threat model and classification of cryptanalysis methods

Throughout our analysis on collusion-resistant systems, the TTP is considered to be secured. We also consider the communication links between pairs of devices to be secure so that an attacker cannot derive the pairwise keys shared between devices. We assume that an attacker can only compromise the keying material shares associated to nodes in a set S_c. In our classification, we consider two criteria, namely (i) the goal of the attacker and (ii) the impact level of a given cryptanalysis approach.

We distinguish between three possible **attack goals**:

1) **Attacking the root keying material:** In this type of attack, the attacker aims to recover the root keying material (or an equivalent structure). If successful, this attack gives the attacker full access to all keying material shares and keys in the system as well.
2) **Attacking a keying material:** In this type of attack, the attacker aims at recovering the keying material share $KM_\eta(x)$ (or equivalent structure) of node η. If successful, this attack gives the attacker full access to all keys associated to device η.
3) **Attacking a key:** In this type of attack vector, the attacker aims at recovering the key $K_{\eta,\eta'}$ shared between a pair of non-compromised devices.

When trying to perform the above actions, the attacker might be more or less successful.

Cryptanalysis impact: refers to the actual impact of a specific cryptanalysis method:

1) **Theoretical cryptanalysis:** we define a theoretical cryptanalysis method as an approach that – theoretically - would allow an attacker to gain some information related to the root keying material, or keying material shares, or keys shared between non-compromised devices, but that either requires extra information that is not available to the attacker or

does not specify how the method would be performed.

2) **Low-severity cryptanalysis:** we define a low-severity practical cryptanalysis method as an approach that can allow an attacker to gain some information about the root keying material, keying material shares, or pairwise-keys shared between non-compromised nodes from a set $|S_c|$ of compromised nodes η. The method is low-severity if HIMMO can be configured with practical parameters (m, α, b, etc) that prevent the attack from performing the attack and still allow for an efficient implementation of HIMMO.
3) **High-severity cryptanalysis:** we define a high-severity practical attack as an approach that given information from a set $|S_c|$ of compromised nodes η, leads to a degradation of the security level of the root keying material, keying material shares, or pairwise-keys shared between non-compromised nodes. The method has a high-severity impact if it fully breaks HIMMO or potentially secure parameters (m, α, b, etc) would be too high to allow for an efficient implementation of HIMMO.

B. Bounds on the number of compromised nodes to retrieve the root keying material and the keying material shares

In this short subsection we consider the minimum amount of information required to recover the set of m bi-variate polynomials that comprise HIMMO's root keying material or the keying material share of a node.

In the scheme [4] defined by the bi-variate symmetric polynomial $f(x, y)$ of degree α in each variable, an attacker who compromises at most α learns no information about any key that is shared between non-compromised nodes. However, an attacker who compromised $\alpha + 1$ nodes can use interpolation to recover the polynomial $f(x, y)$ and thus recover all the keys in the system. This scheme is information theoretic secure. In the case of HIMMO a trivial lower bound of colluding nodes required to retrieve the root keying material is $m(\alpha + 1)$, because an arbitrary univariate polynomial $g(X)$ of degree α with coefficient in an integral domain is determined if and only if we know $\alpha + 1$ values $g(\eta)$. A non-trivial bound is more complicated because among other reasons (i) HIMMO outputs only a part of the bits, (ii) the moduli p_j have some structure (only differ the term β_j, and (iii) the identifiers are only b bits long.

Similarly, for retrieving the keying material share of an uncompromised node, at least $\alpha + 1$ nodes need to collide if the whole output is known. Since only b bits of the polynomial are used as key and N is $(\alpha + 2)b$ bits long, one may also argue that approximately $(\alpha + 1)(\alpha + 2)$ nodes need to be compromised to recover a keying material share.

C. Attacking the root keying material with $m \geq 2$

This attack is strongly related to Problem 3 introduced in Subsection IV-B, and works as follows. Compromise $|S_c|$ nodes with identifiers $\eta_1, \ldots, \eta_{S_c}$ to obtain the integers $KM_{\eta_k,j}$ for $k = 1, \ldots, |S_c|$ and $j = 1, \ldots, \alpha$. The attacker aims to recover the polynomials $f_i(x, y)$ for $i = 1, \ldots, m$. This is equivalent to recover the $m(\alpha + 1)$ univariate polynomials $f_{i,j}(x) \in \mathbb{Z}_{q_i}[x]$. According to the previous subsection, $|S_c|$ should be greater than $m(\alpha + 1)$. Now, for fixed j and k, there exists an integer γ such that

$$KM_{\eta_k,j} = \sum_{i=1}^{m} \langle f_{i,j}(\eta_k) \rangle_{q_i} + 2^b \epsilon_{\eta_k,j} - \gamma N$$

Since $N \geq q_i$ and $|\epsilon_{\eta,j}| < 2^{b\alpha-2}$, we have that $0 < \gamma \leq m$. So, for fixed m the attacker could remove γN, multiplying the time complexity with $(m+1)^{|S_c|}$ complexity by trying all possible values for the γ's. Then, for fixed j, the attacker wants to recover the univariate polynomials $f_{i,j}(x) \in \mathbb{Z}_{q_i}[x], i = 1, \ldots, m$ from the $|S_c|$ integers:

$$\sum_{i=1}^{m} \langle f_{i,j}(\eta_k) \rangle_{q_i} + 2^b \epsilon_{\eta_k,j}, \quad k = 1, \ldots, |S_c|. \quad (5)$$

We do not have any clue how to solve this problem, even if the positive integers q_i are known.

For the weaker task of computing the integer polynomial $\sum_{i=1}^{m} f_{i,j}(x) \in \mathbb{Z}[x]$ from the above equation (5) and assuming that q_i are known, a strategy was suggested by I. Shparlinski and M. Albrecht during the Workshop on Mathematical Cryptology WMC 2012 where a preliminary version of HIMMO was presented. The basic idea is based on the following remark. For every η_k, $k = 1, \ldots, |S_c|$ we have

$$\sum_{i=1}^{m} \langle f_{i,j}(\eta_k) \rangle_{q_i} = \sum_{i=1}^{m} f_{i,j}(\eta_k) + 2^b \epsilon_{\eta_k,j} + \sum_{i=1}^{m} \lambda_{j,k}^{i} q_i,$$

TABLE II
CLASSIFICATION OF IDENTIFIED CRYPTANALYSIS METHODS TO HIMMO

HIMMO Threat	Cryptanalysis impact	Attack goal
Subsection V-B	Theoretical	Attack the root keying material Attack a keying material
Subsection V-C	Theoretical	Attack the root keying material
Subsection V-D	Low-severity	Attack a keying material
Subsection V-E	Low-severity	Attack the root keying material

where $\lambda_{i,j}$ are integer numbers whose absolute value can be bounded. Thanks to the identifiers η_k belong to a small set $\{1,\ldots,2^b-1\}$ we can easily get at the following bound:

$$|\lambda_{i,j}| \leq 2^{b\alpha+1}.$$

Then, we are in the typical conditions to apply lattice reduction techniques. For simplicity's sake and without loss generality we suppose that $m = 2$. Consider the $(2|S_c|+\alpha)$-dimensional lattice $\mathcal{L}$ spanned by the rows of the following matrix:

$$\begin{pmatrix} q_1 & \ldots & 0 & 0 & \ldots & 0 & 2^b & \ldots & 0 \\ 0 & \ddots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & \ldots & q_1 & 0 & \ldots & 0 & 0 & \ldots & 2^b \\ q_2 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & \ddots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & \ldots & q_2 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \eta_1 & \ldots & \eta_{|S_c|} & 2^{-b} & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\ \eta_1^\alpha & \ldots & \eta_{|S_c|}^\alpha & 0 & \ldots & 2^{-b} & 0 & \ldots & 0 \end{pmatrix} \tag{6}$$

and the vectors $\vec{t}, \vec{h} \in \mathbb{R}^{|S_c|+\alpha+|S_c|} = \mathbb{R}^{2|S_c|+\alpha}$ as follows,

$$\vec{t} = (\langle f_{1,j}(\eta_1)\rangle_{q_1} + \langle f_{2,j}(\eta_1)\rangle_{q_2} + 2^b\epsilon_{\eta_1,j}, \ldots, \langle f_{1,j}(\eta_{|S_c|})\rangle_{q_1} + \langle f_{2,j}(\eta_{|S_c|})\rangle_{q_2} + 2^b\epsilon_{\eta_{|S_c|},j}, 0, \ldots, 0, 0, \ldots, 0),$$

$$\vec{h} = (\langle f_{1,j}(\eta_1)\rangle_{q_1} + \langle f_{2,j}(\eta_1)\rangle_{q_2}, \ldots, \langle f_{1,j}(\eta_{|S_c|})\rangle_{q_1} + \langle f_{2,j}(\eta_{|S_c|})\rangle_{q_2}, 2^{-b}c_1, \ldots, 2^{-b}c_\alpha, 2^b\lambda^1_{j,k}, \ldots, 2^b\lambda^1_{j,|S_c|})$$

where $f_{1,j}(x) + f_{2,j}(x) = \sum_{i=1}^{\alpha} c_i x^i$, for $k = 1,\ldots,|S_c|$.

It is expected that when $|S_c|$ is large enough, then with a high probability the solution of the approximating Closest Vector Problem with known target vector $\vec{t}$ and lattice $\mathcal{L}$ will be the unknown vector $\vec{h}$. However, this seems not to be the case because the classical Minkowski's theorem shows the there is a small vector in the lattice, since the volume of the lattice $\mathcal{L}$ is $O(2^{(\alpha+3)b|S_c|-b\alpha)})$ and $\|\vec{t}-\vec{h}\| = O(|S_c|2^{(\alpha+1)b})$. If the noise is small enough, $|S_c|$ and b are big enough implementation shows that this kind of attack works. We refer to Appendix B for further details.

***Remark** 3: If an attacker could solve this problem, he would get some information that is not present in the system. However, notice that it is not clear that this would allow breaking the cryptosystem because it only provides the added value of all bi-variate polynomials, and no the individual polynomial coefficients $f_{i,j}$.*

D. Attacking a keying material share

At the beginning of this section we have proposed a classification of attacks depending on the severity and the goal of the attacker. One of the possible goals of the attacker was to attack a share. The attacker is only interesting in getting the share of a specific node η. This is related to Problem 2. This subsection describes how this could be done in HIMMO and discusses security parameters.
The attack works as follows: the attacker compromises $|S_c|$ nodes with identifiers $\eta_1,\ldots,\eta_{|S_c|}$ together with their shares, that is, he knows $|S_c|$ univariate polynomials $F_{\eta_k}(x) \in \mathbb{Z}_N[x]$, and aims to compute the unknown univariate polynomial $F_\eta(x) = \sum_{i=1}^{\alpha} a_i x^i \in \mathbb{Z}_N[x]$ for any node $\eta \neq \eta_k, k = 1,\ldots,|S_c|$. Then the attacker realizes that the keys between the nodes can be easily converted in the least significant bits of the evaluation of the polynomial which correspond to the share in node η.

So, this is the noisy polynomial interpolation problem introduced in Subsection IV-A of recovering the polynomial $F_\eta(x)$ from the $|S_c|$ values $\langle F_\eta(\eta_k)\rangle_{2^b}$ for $k = 1,\ldots,|S_c|$. Note that Problem 2 is tailored to Equation 2 while here we deal

with the key generation step (Equation 3) in which only b bits are outputted from the evaluation of an α degree polynomial with $b(\alpha+2)$ bits coefficients in a b bits identifier. The ratio of outputted bits and coefficient size is $\alpha+2$. Still, the conceptual problem is basically the same, the attacker only sees a part of the evaluation.

To analyze the severity of this attack we use Theorem 8 from [11] and/or Theorem 5 from [12] based also on lattice reduction techniques. According to those papers, the proposed algorithm has five main requirements to work:

- N is a prime number, although this can be avoided using heuristic arguments,
- the identifiers are uniformly distributed in the set $\{0,\ldots,N-1\}$,
- the number of colluding nodes $|S_c|$ is greater than $2\alpha\sqrt{b}$,
- the quality of the approximation should be good enough; this is true if α is $O(\sqrt{b})$, and
- we have to assume that $\langle F_\eta(\eta_k)\rangle_{2^b} = \langle F_{\eta_k}(\eta)\rangle_{2^b}$, for $k=1,\ldots,|S_c|$.

Computer experiments confirm that the attack does indeed work if $|S_c|$ is large enough and $\alpha < \sqrt{b}$ even if the identifiers are only uniformly distributed in the much smaller range $\{1,\ldots,2^b-1\}$. However, if these conditions are not fulfilled, the lattice reduction does not provide a correct answer. This is a positive result since it allows us to derive the expected order of magnitude for HIMMO parameters so that the above attacks do not work. It is an open question if the fact that the HIMMO identifiers are small affects the requirements on the HIMMO parameters or can even allow for a better attack. A more general open question about the noisy polynomial reconstruction is posed in paper [3].

Independently of the above algorithm, and even assuming that an attacker could find the perfect algorithm to efficiently attack the system, there exists still a reason why it is not going to work in HIMMO. The reason is that the common key generated between two nodes (the fifth condition in the above list) can differ a value Δ specified in Theorem 1, that is, a priori, we can not assume that:

$$\langle F_\eta(\eta_k)\rangle_{2^b} = \langle F_{\eta_k}(\eta)\rangle_{2^b}$$

Thus, the input for an algorithm as the one above is noisy and makes that the attacker has to do a search over all possible values for a generated key. This would theoretically increase the number of calls to the algorithm in an exponential way being $O(\Delta^{|S_c|})$. However, for small m, the actual is not that bad for the attacker: experimental results show that the probability distribution of $(K_{\eta,\eta'}-K_{\eta',\eta})$, where the pair (η,η') is taken at random, is very concentrated around 0 for those small values of m.

E. Attacking the root keying material with $m=1$ and secret q_1

Section V-C shows that it is not trivial to attack the root keying material if $m \geq 2$. Section V-D gives a first positive indication about the security of HIMMO when dealing with an attack to a keying material share.

This section shortly deals with an attack against the root keying material in the case $m=1$, that is, Problem 1 in which the noise hides part of the keying material coefficients. In this situation, there is no mixing and a single bi-variate root keying material is used to generate a keying material share (a noisy polynomial) stored on a node. An attacker can observe all bits of the polynomial except those that are affected by the noise. It is fundamental to remark that these are many more bits than in the situation in Section V-D in which just b bits where exposed, and thus, if q_1 is known, we believe that it is rather difficult to prevent an attacker from performing an attack based on [11] and/or [12] against the root keying material if enough devices have been compromised. If q_1 is secret, the situation becomes interesting again. The results in [13] and [6] can be applied for the case of a linear polynomial ($\alpha=1$), even if q_1 is secret. For non-linear polynomials, existing attacks cannot directly be applied to our scheme.

F. HIMMO security and design principles

The two design principles introduced in Section IV are applied to HIMMO in such a way that they can prevent an attacker from recovering the root keying material or keying material shares as described in the last subsections. It is worth reviewing how this is done. HIMMO as any of the predecessors includes the root keying material, keying material shares, and keys. An attacker can compromise several nodes and the associated keying material shares. Given this keying material shares there is a risk of the attacker recovering the root keying material. Given keys there is a risk of

TABLE III
HOW HIMMO PROTECTS AGAINST DIFFERENT TYPES OF ATTACK VECTORS

Protection against	Relies on	Requires at least	Further improved by
Attacking a keying material share	Hiding information	$\alpha > \sqrt{b}$ *or* $\lvert S_c \rvert < 2\alpha\sqrt{b}$	taking large m
Attacking the root keying material	Hiding Information and Mixing Modular Operations	$m \geq 2$	making β_i secret

the attacker recovering keying material shares, or even the root keying material.

HIMMO relies on the first design principle (see Subsection IV-A) to prevent the attacker from recovering non-compromised keying material shares from keys derived from compromised keying material shares. An attacker can derive many b-bits keys, and from them, he should be able to recover a polynomial of degree α with coefficients of size $(\alpha+2)b$ bits according to HIMMO's specification. Thus, if we increase α we do not only increase the polynomial degree but also the ratio between the number of bits that an attacker can see and the size of the coefficients, namely $b(\alpha+2)$. This ratio is, thus, $\alpha+2$ As discussed in Subsection V-D, if $\alpha > \sqrt{b}$, it seems to be difficult to attack HIMMO even if an attacker captures $|S_c| > 2\alpha\sqrt{b}$ nodes.

The reader might ask himself why this first principle is not sufficient to prevent an attacker from recovering the root keying material from the keying material shares (see Section V-E). This is easy to understand if we observe the size of the noise in Eq. 1, or alternatively the amount of bits known by an attacker. Basically, the amount of information available to an attacker is much higher in this case. In other words, the ratio between the number of bits that an attacker can see (the keying material share) and the number of bits to recover (the root keying material) is big and actually close to $1/2$ in average. Although keeping q_1 secret can indeed help, to solve this issue (see Section V-E), we apply the second design principle (see Subsection IV-B) to prevent an attacker from attacking the root keying material from the keying material shares. This corresponds to the analysis in Subsections V-B and V-C.

The second and third columns in Table III summarize the way HIMMO protects against different types of attack vectors. Additionally, the fourth column indicates secondary protections. The difficulty of attacking the keying material shares increases for large values of m since this value (i) affects the size of Δ in Theorem 1 and (ii) experimental results show that the probability distribution of $(K_{\eta,\eta'} - K_{\eta',\eta})$ where the pair (η, η') is taken at random is more uniform in the range $[-\Delta, \Delta]$ for large m. Attacking the root keying material becomes more difficult if the β_i terms in the q_i are kept secret since this prevents even the construction of the lattice used in Section V-C. See also Section V-E

VI. CONCLUSIONS

Our HIMMO algorithm addresses the old key establishment problem in a different way bringing many advantages. Operationally, it allows for direct ID-based pairwise key establishment simplifying protocol operation. Computationally, the design concepts relying on polynomials allow for very fast operation with minimal memory and energy needs. These features make HIMMO very suitable for new applications such , e.g., the Internet of Things. From a security point of view, the design concepts seem to be sound and existing attacks do not apply; however, as the design concepts are fairly new, further analysis is required. To this end, we have introduced a framework for the security analysis so that attacks on a collusion-resistant system such as HIMMO can be classified according to their goal and impact level. The first design concept in HIMMO presents links to the noisy polynomial interpolation problem, and thus, it might make possible partial security analysis of our scheme by reusing existing literature. This paper has done a first step in this direction with positive results. To the best of our knowledge, our second design concept, mixing of the evaluation of polynomials using different modules, has not been explored in literature so far. Although suggested attacks do not seem to work even applying lattice techniques, more analysis is needed as well. The task of an attacker with regard to both design concepts is further complicated by the fact that he only has partial knowledge on which modules have been used.

APPENDIX A: PROOF OF THEOREM 1

In this appendix, we provide a proof of Theorem 1. In the proof, we rely on the fact that for each integer a and each positive integer N we have that

$$a = \lfloor \frac{a}{N} \rfloor N + \langle a \rangle_N. \tag{7}$$

***Lemma** 1: Let a, b, N be integers, with N positive, then we have*

$$\langle a+b \rangle_N = \langle a \rangle_N + \langle b \rangle_N - \lambda N,$$

for some integer $\lambda \in \{0, 1\}$.

Proof: Using (7), we have that

$$\langle a+b \rangle_N = \langle \langle a \rangle_N + \langle b \rangle_N \rangle_N = \langle a \rangle_N + \langle b \rangle_N - \lambda N,$$
$$\text{where } \lambda = \left\lfloor \frac{\langle a \rangle_N + \langle b \rangle_N}{N} \right\rfloor.$$

As $0 \leq \lambda < 2$ and λ is integer, the lemma follows. □

■

***Lemma** 2: Let α, b, p, N and β be positive integers such that $p = N - \beta 2^b$, $\beta < 2^b$ and $2^{(\alpha+2)b-1} < N \leq 2^{(\alpha+2)b}$. Let $h(X) = \sum_{i=0}^{\alpha} h_i X^i$ be a polynomial with integer coefficients such that $0 \leq h_i \leq p-1$. If $\eta \in \{1, \ldots, 2^b - 1\}$, then,*

$$\langle h(\eta) \rangle_p = \langle h(\eta) \rangle_N + \mu 2^b - \lambda N,$$

for some integers μ, λ and with $\lambda \in \{0, 1, 2\}$.

Proof: It is clear that

$$h(\eta) = \lambda_1 p + \langle h(\eta) \rangle_p = \lambda_2 N + \langle h(\eta) \rangle_N$$

with $\lambda_1 = \left\lfloor \frac{h(\eta)}{p} \right\rfloor$ and $\lambda_2 = \left\lfloor \frac{h(\eta)}{N} \right\rfloor$. Hence,

$$\begin{aligned}\langle h(\eta) \rangle_p &= \langle h(\eta) \rangle_N + \lambda_2 N - \lambda_1 p \\ &= \langle h(\eta) \rangle_N + \lambda_1 (N-p) + (\lambda_2 - \lambda_1) N \\ &= \langle h(\eta) \rangle_N + 2^b \beta \lambda_1 + (\lambda_2 - \lambda_1) N \\ &= \langle h(\eta) \rangle_N + \mu 2^b - \lambda N.\end{aligned}$$

where $\mu = \beta \lambda_1$ and $\lambda = \lambda_1 - \lambda_2$. Since $N > p$, $\lambda \geq 0$.
Moreover,we have that

$$\lambda = \lambda_1 - \lambda_2 \leq \frac{h(\eta)}{p} - \lambda_2 \leq \frac{h(\eta)}{p} - \frac{h(\eta)}{N} + 1$$

and thus

$$\lambda = h(\eta) \frac{2^b \beta}{Np} + 1.$$

We clearly have that

$$h(\eta) = \sum_{i=0}^{\alpha} h_i \eta^i < \sum_{i=0}^{\alpha} p 2^{ib} < p \frac{2^{(\alpha+1)b}}{2^b - 1}.$$

As a consequence, we have that $\lambda < \frac{2^{(\alpha+2)b}}{N(2^b-1)}$. From the bounds on β and N, we infer that $\lambda < 3$. As λ is an integer, it follows that $\lambda \leq 2$.

■

***Remark** 4: Note that if $\beta_i < \frac{N}{2^{b(\alpha+2)}}(2^b - 1)$, then in Lemma 2 we have that $\lambda \in \{0, 1\}$.*

We are now in a position to prove Theorem 1. By definition,

$$K_{\eta,\eta'} = \langle \langle \sum_{j=0}^{\alpha} KM_{\eta,j} (\eta')^j \rangle_N \rangle_{2^b} \text{ with}$$
$$KM_{\eta,j} = \sum_{i=1}^{m} \langle f_{i,j}(\eta) \rangle_{q_i} \quad + 2^b \epsilon_{\eta,j}.$$

We write

$$\phi_{i,\eta}(x) = \sum_{j=0}^{\alpha} \langle f_{i,j}(\eta) \rangle_{q_i} x^j \text{ and}$$
$$F_\eta(x) = \sum_{i=1}^{m} \phi_{i,\eta}(x) + 2^b \sum_{j=0}^{\alpha} \epsilon_{\eta,j} x^j.$$

Then

$$K_{\eta,\eta'} = \langle \langle F_\eta(\eta') \rangle_N \rangle_{2^b}.$$

We introduce the following notation for convenience,

$$\phi_{i,\eta}(X) = \langle f_i(X, \eta) \rangle_{q_i} = \sum_{j=0}^{\alpha} \langle g_{i,j}(\eta) \rangle_{q_i} X^j,$$

where $f_i(X, Y) = \sum_{j=0}^{\alpha} g_{i,j}(Y) X^j$, where $g_{i,j}(Y) \in \mathbb{F}_{q_i}[Y]$. From this, it is clear that,

$$F_\eta(X) = \sum_{i=1}^{m} \phi_{i,\eta}(X) + 2^b \sum_{j=0}^{\alpha} \epsilon_j X^j.$$

PROOF OF THEOREM 1: First, we compute the following

$$\langle F_\eta(\eta') \rangle_N = \langle \sum_{i=1}^{m} \phi_{i,\eta}(\eta') + 2^b \sum_{j=0}^{\alpha} \epsilon_j (\eta')^j \rangle_N =$$
$$= \sum_{i=1}^{m} \langle \phi_{i,\eta}(\eta') \rangle_N + \langle \sum_{j=0}^{\alpha} 2^b \epsilon_j (\eta')^j \rangle_N + \lambda_1 N$$

for some integer λ with $-m \leq \lambda_1 \leq 0$. Using the upper bound on the ϵ's we find

$$\langle F_\eta(\eta')\rangle_N = \sum_{i=1}^{m}\langle\phi_{i,\eta}(\eta')\rangle_N + 2^b\sum_{j=0}^{\alpha}\epsilon_j(\eta')^j + \lambda_2 N,$$
$$\text{with } -m-\frac{1}{2}(\alpha+1) \leq \lambda_2 \leq \frac{1}{2}(\alpha+1).$$

By applying Lemma 2 we obtain that

$$\sum_{i=1}^{m}\langle\phi_{i,\eta}(\eta')\rangle_N = \sum_{i=1}^{m}\langle\phi_{i,\eta}(\eta')\rangle_{q_i} + \lambda_3 N + \mu 2^b,$$

for some integers μ and λ_3 with $-2m \leq \lambda_3 \leq 0$. We thus have that

$$K_{\eta,\eta'} = \langle\langle F_\eta(\eta')\rangle_N\rangle_{2^b} = \langle\sum_{i=1}^{m}\langle f_i(\eta',\eta)\rangle_{q_i} + \lambda N\rangle_{2^b}.$$

for $\lambda = \lambda_2 + \lambda_3$ and so $-\frac{1}{2}(\alpha+1) - 3m \leq \lambda \leq \frac{1}{2}(\alpha+1)$. Similarly,

$$K_{\eta',\eta} = \langle\langle F_{\eta'}(\eta)\rangle_N\rangle_{2^b} = \langle\sum_{i=1}^{m}\langle f_i(\eta,\eta')\rangle_{q_i} + \lambda' N\rangle_{2^b},$$

for some integer λ', with $-\frac{1}{2}(\alpha+1) - 3m \leq \lambda' \leq \frac{1}{2}(\alpha+1)$.
As the f_i's are symmetric, we have proven Theorem 1. □ ■

APPENDIX B: MINKOWSKI'S THEOREM

We cite the next result from [9, Theorem 4].

***Lemma** 3: Suppose that $\mathcal{L}$ is a lattice of determinant d in an n-dimensional vector space $\mathbb{R}^n$ and S is a convex subset of $\mathbb{R}^n$, symmetric with respect the origin and with a volume greater than $2^n d$, then S contains at least one lattice point other than the origin.*

For our case, it is just necessary to take,

$$S = \{\vec{x} \in \mathbb{R}^n \mid \|\vec{x}\| \leq 2n\sqrt[n]{d}\},$$

So, the lattice contains at least one nonzero element $\vec{v}$. Indeed, the problem of studying the minimum radius such that any lattice has a vector inside a ball has been studied as the Hermite constant. This has application to the following problem that it is solved by Babai's algorithm [2] .

***Problem** 4 (Approximating CVP): The approximating Closet Vector Problem (CVP) is defined as follows. Given a target $\vec{t}$ and a lattice $\mathcal{L}$ of dimension s, find a vector $\vec{h} \in \mathcal{L}$ such that*

$$\|\vec{t} - \vec{h}\| \leq 2^{s/2}\min_{\vec{u}\in\mathcal{L}}\{\|\vec{t} - \vec{u}\|\}.$$

If the approximating CVP has a unique solution, then no non-zero vector $\vec{x} \in \mathcal{L}$ satisfies

$$\|\vec{x}\| \leq \min_{\vec{u}\in\mathcal{L}}\{\|\vec{t} - \vec{u}\|\}.$$

Indeed, if $\vec{x} \in \mathcal{L}$ satisfies the above equation and $\vec{h}$ satisfies

$$\|\vec{t} - \vec{h}\| = \min_{\vec{u}\in\mathcal{L}}\{\|\vec{t} - \vec{u}\|\},$$

then

$$\|\vec{t} - \vec{h} - \vec{x}\| \leq \|\vec{t} - \vec{h}\| + \|\vec{x}\| \leq 2\min_{\vec{u}\in\mathcal{L}}\{\|\vec{t} - \vec{u}\|\}.$$

REFERENCES

[1] M. Albrecht, C. Gentry, S. Halevi, and J. Katz. "Attacking cryptographic schemes based on perturbation polynomials". In E. Al-Shaer, S. Jha, and A.D. Keromytis, editors, ACM Conference on Computer and Communications Security, pages 1-10. ACM, 2009.

[2] L. Babai. "On lovász' lattice reduction and the nearest lattice point problem". Combinatorica, 6(1):1–13, 1986.

[3] S. Blackburn, D. Gomez, J. Gutierrez, and I. Shparlinski. "Reconstructing noisy polynomial evaluation in residue rings". J. Algorithms. 61(2):47-59, 2006.

[4] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. "Perfectly-secure key distribution for dynamic conferences" . In E.F. Brickell, editor, CRYPTO '92, volume 740 of Lecture Notes in Computer Science, pp. 471-486. Springer, 1992.

[5] D. Boneh and R. Venkatesan. "Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes". In N. Koblitz, editor, CRYPTO, volume 1109 of Lecture Notes in Computer Science, pp. 129-142. Springer, 1996.

[6] S. Contini and I. Shparlinksi. "On Stern's Attack Against Secret Truncated Linear Congruential Generators". 10th Australasian conference on Information Security and Privacy, pp. 52-60. 2005.

[7] T. Heer, O. Garcia-Morchon, R. Hummen, S.L. Keoh, S. Kumar, and K. Wehrle. "Security challenges in the IP-based Internet of Things". Wireless Personal Communications, 61(3): 527-542, 2011.

[8] T. Matsumoto and H. Imai, "On the key predistribution system: a practical solution to the key distribution system", in C. Pomerance (Ed): Advances in Cryptology, CRYPTO'87, volume 293 of Lecture Notes in Computer Science, pp. 185-193, Springer, 1988.

[9] P.Q. Nguyen, "Hermite's constant and lattice algorithms". In P.Q. Nguyen and B. Vallée, (Eds.), *The LLL Algorithm: Survey and Applications*, Springer, 2009.

[10] D. Micciancio and S. Goldwasser, "Complexity of lattice problems" , The KluwerInternational Series in Engineering and Computer Science, 671, Kluwer AcademicPublishers, Boston, MA, 2002. A cryptographic perspective.

[11] I. E. Shparlinski, "Sparse polynomial approximation in finite fields" , Proc. 33rd ACM Symp. on Theory of Comput., Crete, Greece, July 6-8, 2001, 209-215.

[12] I.E. Shparlinski and A. Winterhof. ”Noisy interpolation of sparse polynomials in finite fields”. Appl. Algebra Eng. Commun. Comput., 16(5):307-317, 2005.
[13] J. Stern. ”Secret linear congruential generators are not cryptographically secure”. 28th IEEE Symp. on Foundations of Computer Science, pp. 201-224, 1987.
[14] W. Zhang, M. Tran, S. Zhu and G. Cao, ”A random perturbation-based scheme for pairwise key establishment in sensor networks”, 8th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc), 2007.