applied sciences

MDPI

Article

A Zero-Knowledge Proof System with Algebraic Geometry Techniques

Edgar González Fernández [1,*], Guillermo Morales-Luna [1] and Feliu Sagols [2]

[1] Department of Computer Science, CINVESTAV-IPN, Av. IPN 2508, Gustavo A. Madero, San Pedro Zacatenco, Mexico City 07360, Mexico; gmorales@cs.cinvestav.mx

[2] Department of Mathematics, CINVESTAV-IPN, Av. IPN 2508, Gustavo A. Madero, San Pedro Zacatenco, Mexico City 07360, Mexico; fsagols@math.cinvestav.edu.mx

* Correspondence: egonzalez@computacion.cs.cinvestav.mx; Tel.: +52-555-747-3756

Received: 4 November 2019; Accepted: 3 December 2019; Published: 8 January 2020

Abstract: Current requirements for ensuring data exchange over the internet to fight against security breaches have to consider new cryptographic attacks. The most recent advances in cryptanalysis are boosted by quantum computers, which are able to break common cryptographic primitives. This makes evident the need for developing further communication protocols to secure sensitive data. Zero-knowledge proof systems have been around for a while and have been considered for providing authentication and identification services, but it has only been in recent times that its popularity has risen due to novel applications in blockchain technology, Internet of Things, and cloud storage, among others. A new zero-knowledge proof system is presented, which bases its security in two main problems, known to be resistant, up to now, against quantum attacks: the graph isomorphism problem and the isomorphism of polynomials problem.

Keywords: graph isomorphism; isomorphism of polynomials; interactive proof system; multivariate cryptography; zero-knowledge proof

1. Introduction

The increasing use of powerful electronic devices and the availability of networks that provide ubiquitous and high-performance connectivity allow applications to transfer huge volumes of data in brief periods of time. Several transactions and secure connections are performed using reliable schemes of authentication and privacy based on complicated mathematical problems, which have remained unsolved up to now. The starting point of secure communications requires previous secret sharing or authentication, using for this purpose, public key cryptography (PKC). Though several cryptographic algorithms exist, only a few protocols are used in real-world applications due to their proven resistance and easy implementation: the well known procedure due to Rivest-Shamir-Adleman (RSA) [1], based on the factorization problem, and the Digital Signature Standard (DSS) [2] based on the discrete logarithm problem on finite groups. These algorithms are the base of several digital signature techniques, and authentication and identification protocols, which are commonly used for e-commerce, banking transactions, and government services, among others, and their applications have been increasing with the introduction of multifactor authentication and cryptocurrencies.

The rapid development of cryptanalysis techniques and quantum computers endanger these security measures, with the most alarming threat being the existence of an algorithm that can solve the factorization problem efficiently, provided a quantum computer can ever be built [3]. These issues make clear that new techniques must be studied and developed in preparation for possible realizations of these threats. Recently, zero-knowledge proofs (ZKP) have been considered as an alternative to design authentication and identification protocols. Protocols based on ZKP are built upon problems

which have not been solved yet by quantum computer algorithms; many of them originated from graph theory and NP-complete problems.

In addition to authentication and identification services, novel technologies (e.g., blockchain and cryptocoins [4]), which require anonymity services, have demonstrated in ZKP systems, a reliable technique to prove knowledge of specific data without disclosing details; say, whether an account has enough credit to buy an item. Current uses have also been reported in the direction of authentication in cloud storage [5] and Internet of Things (IoT) [6], encouraging the development of these sorts of protocols.

The method defined in this work produces key pairs from an associated isomorphism between a pair of graphs. The public key will be given by a system of equations. The private key will consist of a solution to the system. It will be shown that finding this solution is at least as difficult as finding an isomorphism between the associated graphs. At present, the fastest algorithm for solving the graph isomorphism (GI) problem runs in quasi-polynomial time [7]. However, an authentic prover will be ready to provide a solution efficiently.

2. Related Work

Interactive proof systems were presented by Goldwasser, Micali, and Rackoff [8] as a novel technique to demonstrate "knowledge" efficiently, in the sense that the verification of such knowledge should be performed easily. This method involves an exchange of information between two entities: the *prover*, which is determined to demonstrate the truthiness of a proposition to a second party, and the *verifier*,which in turn must be convinced of the assertion. The parties involved interact in a challenge-response process until the verifier is ready to decide that the prover's assertion is correct, or concludes that the claim is false. Interactive proof systems are said to be *zero-knowledge* if the verifier is not able to get any extra information from the interaction process, except the correctness of the statement. This kind of proof can be used by entities requiring authentication and identification services: access control or credit card validations, among many others.

One of the most typical examples of ZKP systems bases its security in the difficulty of solving the graph isomorphism problem (GI) [9]. The main components of this system are:

- The public key: two isomorphic graphs G and H.
- The private key: the pair (G, H) together with an isomorphism $\phi : G \to H$.
- The interaction algorithm between Peggy (the prover) and Victor (the verifier):

    1. Peggy starts the interaction by providing a random isomorphic graph K.
    2. Victor selects a random bit $b \in_R \{0, 1\}$ and sends it to Peggy.
    3. Considering $\psi_0 = \psi \circ \phi, \psi_1 = \psi$ Peggy must send ψ_b accordingly.
    4. Victor verifies that $\psi_0(G) = K$ or $\psi_1(H) = K$ depending on the choice of b.

The interaction procedure is based on the commutativity of the diagram shown in Figure 1 and the difficulty of constructing $\psi \circ \phi$ from ψ alone.

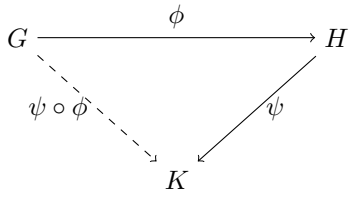


Figure 1. Composition of graphs.

The GI problem can be easily solved for the average case with state-of-the-art solvers, such as nauty, Traces [10], saucy [11], and bliss [12], among others. In addition to these results, Babai [7] has proposed a novel technique reducing the complexity of GI to quasi-polynomial time (with a running time of $2^{O((\log n)^c)}$). Nevertheless, efforts to construct difficult instances have been made. Contrary to

what is expected, these cases might not provide suitable cases for cryptographic purposes but lower complexity bounds by solving particular cases with tuned algorithms.

Grigoriev [13] generalizes the aforementioned construction by studying other mathematical objects possessing the commutativity property shown in Figure 1. This allows considering transformations with similar behaviour, such as homomorphisms and endomorphisms in group and ring theory. The required characteristics to obtain a resistant protocol are:

- that transformations ϕ and ψ are difficult to invert.
- The possibility of obtaining $\psi \circ \phi$ easily from ϕ and ψ.

The ZKP system based on GI is compliant with these restrictions, but further problems are introduced with similar characteristics, mainly related to graph theory, such as the subgraph problem or the colorability problem, and problems concerning group and ring endomorphisms, among others. Some of these problems are known to be NP-hard, which provides an advantage over the GI problem, whose membership to the NP-complete group is currently unknown, but expected to be false.

Later, Patarin [14] introduced the Isomorphism of Polynomials Problem (IP), which relates affine spaces by means of affine transformations. Given two sets of polynomials of the same size, we say that both sets are isomorphic if there are affine transformations that define a bijection from one set into the other. Formally, the IP problem is stated as follows:

Definition 1. *Consider two vector spaces $\mathbb{F}^m$ and $\mathbb{F}^n$ of dimensions m and n, respectively, over a finite field $\mathbb{F}$ and two quadratic transformations $F = (f_1, \dots f_m)$, $\overline{F} = (g_1, \dots, g_m)$. Each f_i, g_i is a quadratic polynomial. F and $\overline{F}$ are isomorphic if there are $S : \mathbb{F}^m \to \mathbb{F}^m$ and $T : \mathbb{F}^n \to \mathbb{F}^n$ such that $\overline{F} = S \circ F \circ T$.*

The composition of affine transformations is itself an affine transformation. Thus, the composition of isomorphisms can be defined as straightforward. The original scheme considers two affine transformations S, T, but a simplification which consists of discarding one of them (or equivalently, setting a transformation as the identity) leads to defining the IP on one or two secrets (IP1s and IP2s correspondingly). The proposed authentication scheme is very similar to that defined for GI.

Both IP1s and IP2s have been considered for a new brand of cryptographic primitives known as multivariate cryptography [15,16]. Theses primitives are based on the $\mathcal{MQ}$ problem, which consists of finding a common solution of a set of polynomials in several variables in a given vector space (commonly, over a finite field $\mathbb{F}_q$). A traditional procedure for key generation in multivariate public key cryptography (MPKC) involves two major phases:

- Private key generation. A set of polynomials $F = \{f_1, \dots, f_m\}$ is generated in such a way that the problem of finding a common root for every f_i is easy.
- Public keys derivation. From the private key polynomial set F, we generate a new polynomial set $\overline{F} = \{\overline{f}_1, \dots, \overline{f}_n\}$. For this set, the problem of finding a common root must be computationally difficult. Otherwise, a malicious entity would be able to perform sensitive operations, like deciphering and digital signing.

The most common construction techniques base their security in the intractability of IP; for this, the affine transformations S and T must be kept in secrecy since the recovery of the private polynomial set with knowledge of the affine transforms is a computationally easy task. Further methods for private key generation can be found in [17].

The origins of MPKC can be traced back to the scheme proposed by Matsumoto and Imai in [15,16]. The proposed cryptosystem (known as the Matsumoto–Imai (IM) cryptosystem) was broken a few years later [18]. Since then, many other families of schemes have been proposed, including the unbalanced oil-vinegar (UOV) [19], the hidden field equations (HFE) [14], and the Rainbow [20] schemes. Currently, the National Institute of Standards and Technology is working on the development of quantum-resistant cryptographic standards, many of them based on MPKC [21].

The rapid development of MPKC has also caused advances in algorithms for solving multivariate systems. These provide very useful cryptanalytic attacks that, according to the target, can be classified into two main groups:

- Ciphertext decryption. In this case the primary goal is to get the original plaintext from the captured ciphertext. These attacks make use of polynomial system solvers such as the Buchberger algorithm [22] to compute Groebner bases. On each new ciphertext obtained, the algorithm must be executed.
- Private key recovery. The private key consists of the private set F and the transformations S_1, S_2. If this information is disclosed, every ciphertext ciphered with the disclosed key is vulnerable. Examples of these algorithms are: high rank, MinRank, and separation of oil and vinegar [23], VI.5.4.

Up till now, the most reliable algorithms for solving general polynomial systems have been those based on the Buchberger algorithm, which has an exponential running time [22], even for the average case. Additional aspects regarding asymptotic studies on graphs and Groebner bases are provided in [24] and [25].

3. Mathematical Background

In this section, we provide a brief introduction to the basic concepts used throughout this work.

3.1. Graphs

A *graph* is a pair (V, E), where $V = \{v_1, \ldots, v_n\}$ is a set of n elements—the *vertices*; and E is a subset of $\binom{V}{2} = \{e \subset V | \, \#e = 2\}$, the *edges*. The *order* and *size* of G are the cardinalities of the sets V and E, respectively. Two different vertices $u_1, u_2 \in V$ are *adjacent* if they are connected by an edge. Analogously, two different edges $e_1, e_2 \in E$ are *adjacent* if they share one and only one vertex. The graph $\overline{G} = (V, \overline{E})$ defined by $\overline{E} = \{v_i v_j \in \binom{V}{2} \mid v_i v_j \notin E\}$ is the *complementary graph* of G. This consists of pairs of non-adjacent vertices.

If two disjoint subsets $V_1, V_2 \subset V$ exist such that $V_1 \bigcup V_2 = V$ and such that every edge has vertices in both sets V_1 and V_2, then the graph is said to be *bipartite*. Furthermore, G is *complete bipartite* provided that every vertex in V_1 is connected to every vertex in V_2 and vice versa.

Now, consider two graphs $G = (U, D)$ and $H = (V, E)$. Consider a bijections of sets $\phi : U \to V$ that preserves edges; i.e., if $\{u, v\} \in D$ implies $\{\phi(u), \phi(v)\} \in E$. The ϕ is an *isomorphism* between G and H, and G and H are said to be *isomorphic*, denoted $G \approx H$. The graph isomorphism problem is defined as the task of finding an isomorphism between G and H, or deciding that they are not isomorphic. Formally, GI can be defined as follows.

DECISION PROBLEM
Instance: Two graphs $G = (U, D), H = (V, E)$.
Solution: $\begin{cases} 1 & \text{If there is an isomorphism } \phi : G \to H \\ 0 & \text{Otherwise.} \end{cases}$

SEARCH PROBLEM
Instance: Two graphs $G = (U, D), H = (V, E)$.
Solution: Either a proof that H and G are not isomorphic or the isomorphism $\phi : G \to H$.

Finally, a *matching* in a graph G is a subset $M \subseteq E$ with the property that no to edges $e_1, e_2 \in M$ are adjacent. The matching is *perfect* if, in addition, every vertex of G is an paired by an edge of M.

3.2. Polynomial Ideals and Algebraic Sets

Consider the finite field of q elements $\mathbb{F}_q$ and the ring of polynomials in n variables over $\mathbb{F}_q$, denoted $R = \mathbb{F}_q[X_1, \ldots, X_n]$. A subset $I \subset R$ is an *ideal* if

- For every $f, g \in I$, $f + g \in I$;
- For every $f \in I$, $h \in R$ the product $hf \in I$.

Then, considering a finite set of polynomials $F = \{f_1, \ldots, f_m\} \subset R$, we can define the *ideal generated by F* as follows

$$(F) = \{h_1 f_1 + \ldots + h_m f_m | h_i \in R, i = 1, \ldots, m\}.$$

A common root for the polynomials f_i for $i = 1, \ldots, m$ is also a root for any $f \in (F)$. The *zero-set* for the ideal I, denoted V_I, consists of all the points $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$ such that $f(x_1, \ldots, x_n) = 0$ for every $f \in I$. By considering an algebraic extension of the base field $\mathbb{F}_q$, the zero-set is known as the *algebraic set* of I.

We can now formalize $\mathcal{MQ}$ as a decision problem. Additionally, we state the related search problem.

DECISION PROBLEM.
Instance: An ideal $I \subset \mathbb{F}_q[X_1, \ldots, X_n]$.
Solution: $\begin{cases} 1 & \text{If } V_I \neq \emptyset; \\ 0 & \text{Otherwise.} \end{cases}$

SEARCH PROBLEM
Instance: An ideal $I \subset \mathbb{F}_q[X_1, \ldots, X_n]$.
Solution: Either a proof that $V_I = \emptyset$ or a point $x \in \mathbb{F}_q^n$ such that $x \in V_I$.

A solution to the search problem provides a solution to the decision problem immediately. If we are able to find a solution for the polynomial system $f_1 = \ldots = f_m = 0$ we conclude that $V_I \neq \emptyset$. This means that solving the search problem is at least as difficult as solving the decision problem, which is known to be NP-complete.

As mentioned before, any solution for a set of polynomials is also a solution for the ideal generated by that set. Most of the system solvers work based on this fact, by finding a set of "representatives" with better properties, making the resolution task easier. Finding these representatives has been already explored by Buchberger, who proposed the construction of the so-called Groebner bases. We can mention improved versions of the Buchberger algorithm, such as F4 and F5. They have been successful in attacking cryptographic schemes, such as the HFE and the Matsumoto–Imai [26], and some variations of UOV [27]. Despite these efforts, the complexity of these algorithms, even in average instances of $\mathcal{MQ}$, is fully exponential [28].

3.3. Zero-Knowledge Proof Systems

Some handy cryptographic tools used for authentication and identification services are zero-knowledge proofs. A basic description of such systems consists of two parts: the *verifier* performs a series of questions to the *prover*, who must answer correctly in each round to convince the verifier. The prover will be capable of answering correctly on each round only if he has legitimate information.

For this process to be securely implemented, some characteristics regarding the interaction of the involved parties are desirable. The whole verification process should be computationally efficient for an authentic verifier, whereas it must be infeasible for an unauthentic prover to impersonate the authentic one. Furthermore, no information that allows a malicious verifier to reveal the prover's secret can be gathered, though this is commonly relaxed to "no statistically significant information." The following points summarize the desirable characteristics of a ZKP system:

- *Completeness*. An authentic prover will always be accepted by an honest verifier.
- *Soundness*. Upon interacting with a non-authentic prover, the verifier will reject it with a very high probability.

- *Zero-knowledge*. A malicious verifier is not capable of getting any extra information from the challenge-response procedure, other than the correctness of the assertion.

This means that a verifier will always accept an authentic prover. However, a malicious prover has a chance to impersonate an authentic one, but with very small probability.

4. Construction of the Polynomial System

We proceed by developing the construction of the polynomial set based on an isomorphism between graphs.

Consider two isomorphic graphs $G = (U, D)$ and $H = (V, E)$ of order n and size e. Denote by $K_{U,V}$ the complete bipartite graph on the vertex set $U \bigcup V$. It is possible to obtain a perfect matching M in the graph $K_{U,V}$ by choosing edges $u_i v_k, u_j v_l$ if and only if both $u_i u_j$ and $v_k v_l$ are edges in their respective graphs. In other words:

(i) If $u_i u_j \in D$ and $v_k v_l \notin E$, edges $u_i v_k$ and $u_j v_l$ cannot lie in M simultaneously.
(ii) If $v_k v_l \in E$ and $u_i u_j \notin D$, edges $u_i v_k$ and $u_j v_l$ cannot lie in M simultaneously.

A perfect matching M gathered in this fashion can also be regarded as a bijection ϕ of the vertices of U and V, defining an isomorphism between their corresponding graphs. The aforementioned conditions are an equivalent way to assert:

$$u_i u_j \in D \iff \phi(u_i)\phi(u_j) \in E.$$

What has been explained can be observed in Figure 2.

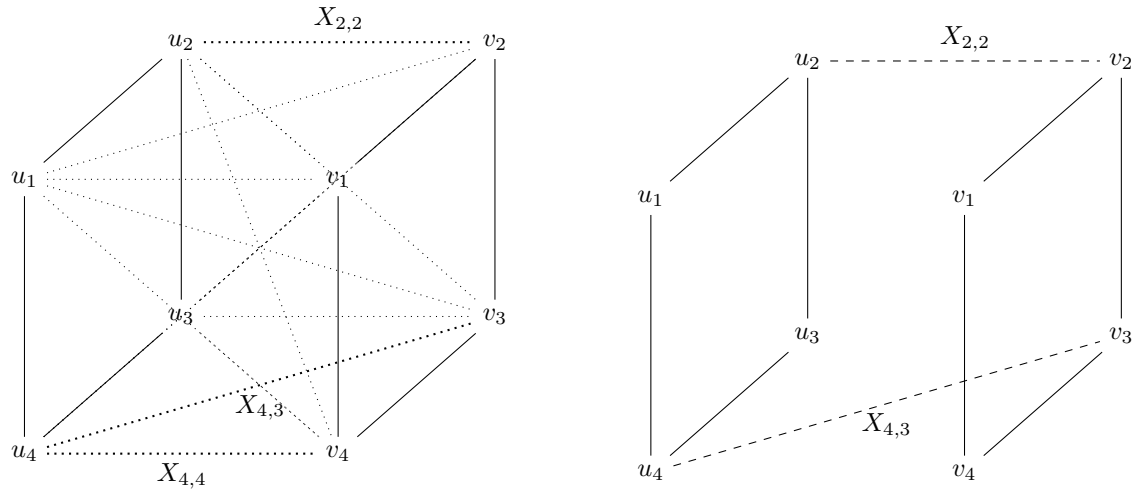


Figure 2. Process of generating the polynomial set associated to graph isomorphism. (**a**) An isomorphism between G and H can be seen as a perfect matching in the graph $K_{U,V}$, preserving adjacencies between G and H. (**b**) The edges $u_2 v_2$ and $u_3 v_4$ cannot belong simultaneously to M because $u_2 u_3 \in D$, but $v_2 v_4 \notin E$. The polynomial $X_{2,2} X_{3,4}$ is added to the ideal I.

Now, we translate the notion of isomorphism between graphs to a strictly algebraic language. The idea is to perform a proper reduction from GI to $\mathcal{MQ}$ motivated by conventional reductions of several problems in graphs to Boolean quadratic polynomials [29,30]. For this, we need to consider a set of n^2 variables, denoted $\{X_{i,k}\}$ for $i, k = 1, \ldots, n$. The first set of polynomials to append, restrict any possible solution to values in the set $\{0, 1\}$. The polynomials are defined as follows:

$$X_{i,k}^2 - X_{i,k} \text{ for } i, k \in \{1, \ldots, n\}. \tag{1}$$

These could be discarded if the restriction is made clear by considering only solutions over the binary vector space $\mathbb{F}_2^n$. The next batch of polynomials restricts the zero-set to solutions that represent a perfect matching; i.e., exactly one vertex u_i from U is connected to one vertex of V and vice versa. This associates the solutions to the existence of a perfect matching M.

$$\begin{aligned} \sum_{k=1}^{n} X_{i,k} - 1 \qquad & \text{for } i = 1, \dots n \\ \sum_{i=1}^{n} X_{i,k} - 1 \qquad & \text{for } k = 1, \dots n. \end{aligned} \tag{2}$$

The last set of polynomials guarantee that the solution is related exclusively to the isomorphism arising from the perfect matching:

$$\begin{aligned} & X_{i,k}X_{j,l} \text{ for every } i, j, k, l \text{ which satisfy} \\ & \quad \left(u_i u_j \notin D \wedge v_k v_l \in E\right) \vee \\ & \quad \left(u_i u_j \in D \wedge v_k v_l \notin E\right). \end{aligned} \tag{3}$$

The construction of the polynomial set is now complete.

5. Zero-Knowledge Protocol

Our next goal is to employ the theory developed in Section 4 to established the announced ZKP.

Let us start by generating a graph G and a random isomorphism ϕ, which can be obtained as a random bijection of its vertex set. In this way, we create a second graph H which is isomorphic to G with isomorphism ϕ. Now, let F_0 be the polynomial system resulting from the process of construction shown in Section 4. A solution $\mathbf{x_0}$ for the system F_0 is found by setting $X_{i,k} = 1$ if $u_i v_k \in M$, and $X_{i,k} = 0$ otherwise. The polynomial set F_0 will be public and is used as the public key. The private key will be the pair $(F_0, \mathbf{x_1})$.

The interaction process starts by generating a second isomorphic graph K, which can be performed by applying a random bijection ψ on the vertex set of H. Knowing the graph H and the applied permutation allows one to obtain a second polynomial set F_1 and a its corresponding solution $\mathbf{x_1}$. The following diagram (Figure 3) allows visualization of the operation performed.

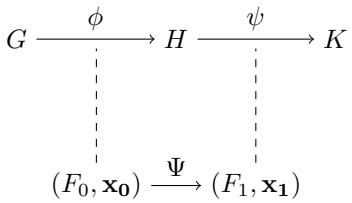


Figure 3. Graph composition and resulting systems.

Though the pair $(F_1, \mathbf{x_1})$ can be obtained in the same fashion as the pair $(F_0, \mathbf{x_0})$, i.e., by computing the polynomial set related to the corresponding graph isomorphism, a more direct approach consists of directly applying suitable permutations to the subindices k and l for the variables obtained from the edges of H and $\overline{H}$. In fact, let us define the permutation σ_ϕ by $\sigma_\phi(i) = k$ if $\phi(u_i) = v_l$. Then, the edge $u_i u_j \in D$ transforms into edge

$$\phi(u_i)\phi(u_j) = v_{\sigma_\phi(i)} v_{\sigma_\phi(j)}.$$

A similar permutation σ_ψ, dependent on the action ψ, is obtained by relating edges of graph H and edges of graph K. The set of polynomials fulfilling condition (3) leads to a direct definition of the set of polynomials corresponding to H and K obtained from the public polynomial set as

$$X_{\sigma_\phi(i),\sigma_\psi(k)} X_{\sigma_\phi(j),\sigma_\psi(l)}. \tag{4}$$

A solution for the system F_1 is provided by applying permutations σ_ϕ, σ_ψ to reorder the entries of the vector $\mathbf{x_1}$ in a similar fashion.

Observe that applying the permutation σ_ψ to the subindices of $X_{i,k}$ is equivalent to applying an affine transformation T, which might be represented by a matrix with one and only one element with value 1 on each column and each row (a permutation matrix) defined by

$$T(i,j) = \begin{cases} 1 & \text{if } j = \sigma_\psi(i) \\ 0 & \text{otherwise.} \end{cases}$$

A similar transformation S is related to ϕ; this time, it is applied on the right side.

$$S(i,j) = \begin{cases} 1 & \text{If } j = \sigma_\phi(i) \\ 0 & \text{Otherwise.} \end{cases}$$

Indeed, S, T can be used to compute the new polynomial (see $\Psi(F_1) = S \circ F_0 \circ T$) and the new solution to such a system by $\mathbf{x_1} = \Psi(\mathbf{x_0}) = S \cdot \mathbf{x_0} \cdot T$, which consists of matrix multiplications.

Finally, if instead of using the isomorphism $\psi : H \to K$ to obtain the second polynomial system, the composition $\gamma = \psi \circ \phi$ is used, we get a third system, constructed by computing the new set $X_{i,\sigma_\gamma(k)} X_{j,\sigma_\gamma(l)}$, which requires a single permutation, and in matrix notation, only the inner affine transformation T. Since both systems rely on the difficulty of computing a graph isomorphism, theoretically, any one of them could be used without losing security in the defined protocol.

5.1. Authentication Protocol

The complete authentication protocol is outlined by the following steps, which are performed between Peggy (the prover) and Victor (the verifier):

Key Generation:

1. Peggy picks a graph G and randomly generates a permutation of the set $\{1, \ldots, n\}$. This permutation is used to create the isomorphic graph H together with its isomorphism ϕ, and then, the public key F_0 using the technique aforementioned. The private key is the pair $(F_0, \mathbf{x_0})$, which consists of the public polynomial system together with a solution to the system.

Authentication:

1. Peggy generates a permutation σ for the set $\{1, \ldots, n\}$ at random and computes the polynomial system F_1, which is sent to Victor as a *compromise*.
2. Victor creates a challenge by selecting at random $b \in \{0, 1\}$. Victor sends b to Peggy.
3. Once Peggy has received b she must answer accordingly:
   - If $b = 0$, she sends the transformation Ψ to Victor.
   - If $b = 1$, then she sends the solution $\mathbf{x_1}$ of F_1.
4. According to the value of b Victor performs the following to authenticate Peggy:
   - If $b = 0$, he computes the system $F_1' = \Psi(F_0)$ and verifies whether he $F_1' = F_1$.
   - If $b = 1$, he checks whether $F_1(\mathbf{x_1}) = 0$ or not.

5.2. Verification of the Protocol

In order to admit the proposed ZKP system as valid, it must fulfill the defining requirements: completeness, soundness, and zero knowledge.

Completeness. Consider Peggy and Victor as authentic entities. On each iteration, Peggy generates a pair $(F_i, \mathbf{x_i})$ from a random permutation σ of the variables. Both can be computed efficiently by her, since she already has knowledge of the original solution $(F_0, \mathbf{x_0})$, and subsequently, can provide a correct answer to the challenge.

Soundness. Consider a rogue prover Robert, who wants to deceive Victor by claiming knowledge of the solution $\mathbf{x_1}$. He might proceed in two different ways:

1. He creates a new system from F_0 by using any random permutation σ to the variable subindices. If Victor sends $b = 0$ Robert will be able to provide $\Psi : F_0 \to F_1$; however, if $b = 1$ he will not be able of compute the solution $\mathbf{x_1} = \Psi(\mathbf{x_0})$.
2. From a made-up solution $\mathbf{x'_0}$, Robert can compute set of polynomials F'_0 having $\mathbf{x'_0}$ as solution. Then if Victor sends $b = 1$, Robert can deceive Victor; on the other hand, if Victor send $b = 1$, Robert must provide the transformation $\Psi : F_0 \to F_1$ which is computed from a valid σ. Since the problem is strongly related to GI, this will be a difficult task, and for this reason, infeasible.

In any case, the chance of succeeding is $\frac{1}{2}$ at each round. After n rounds, the probability is $\frac{1}{2^n}$, which becomes insignificant as n grows.

Zero-Knowledge. Finally, zero-knowledge is provided for the following reasons: having knowledge of the systems F_0 and F_1, it is infeasible to compute Ψ or its solution $\mathbf{x_1}$ in polynomial time, since we have built these objects based on difficult tasks: solving the GI problem or the **MQ** problem. At every iteration a piece of information is provided. If Ψ is disclosed, it is not possible to compute $\mathbf{x_0}$ without knowledge of the solution $\mathbf{x_1}$. For the second case, if $\mathbf{x_1}$ is exposed, then, unknowing Ψ, it is not possible to recover $\mathbf{x_0}$.

5.3. Possible Attacks

We will consider that a malicious entity, a rogue prover (Robert), wants to play the role of Peggy. He can try the following strategy.

Robert can flip a coin to obtain a random value r to decide how to proceed. If $r = 0$, Robert randomly generates a system F'_1 with a given solution that he knows. If Victor challenges with $b = 1$, Robert is able to provide the solution, but if $b = 0$, he will not have the corresponding transformation $\Psi : F_0 \to F_1$. Alternatively, if Robert obtains $r = 1$, he computes a random permutation to obtain a transformation of the system F_0. If Victor challenges with $b = 0$, Robert will be able to provide the required transformation, but, on the contrary, if Victor chooses to send $b = 1$, he will fail to compute a suitable solution. It has been noted that the probability of cheating with this strategy is insignificant after n rounds for an n big enough.

Now we suppose that Robert attacks as a malicious verifier, who wants to obtain information about the secret key, so he plays the role of Victor. He can try asking several times until he can gets the same set of polynomials twice. This would give hem access to the private key. The first time he challenges Peggy with $b = 0$ so he can get the permutation. In subsequent times, he sends $b = 1$ and gets the solution to the corresponding system. If the first random permutation is repeated at some time, Robert can compute the solution to the public system by applying σ^{-1} to the subindices of the solution. There are $n!$ different ways of permuting n elements. This makes the strategy infeasible, since he will have to perform an exponential number of challenges.

Finally, it is possible to solve these problems by breaking the protocol with more sophisticated tools:

- Solving $\mathcal{MQ}$. Using a polynomial system solver to find a solution for the polynomial system F_1 would extract the private key (or another suitable private key $\mathbf{x'_1}$).

- Solving IP. This is done by computing the affine transformations T and S, that make two quadratic transformations $\overline{F}$ and F isomorphic; i.e., $\overline{F} = S \circ F \circ T$. In our construction, the permutation applied to subindices can be regarded as a special case of IP where S and T are permutation matrices.
- Addressing GI. We need to retrieve the initial isomorphic graphs from the polynomial set and find an isomorphism, which leads to forge a private key.

At present-day, authors are not aware of quantum algorithms solving, efficiently, any of the forenamed problems.

6. Computational Complexity

An analysis of computational cost of the transformation of the GI instance is performed next. Observe that, for conditions (1) and (2) every pair (i,k) for $i,k \in \{1,\ldots,n\}$ must be considered. This can be done in $O(n^2)$.

The next step consists of including the polynomials required to comply with condition (3). The following verifications are made:

1. For every $u_i u_j \in D$, look for the edges $v_k v_l \in \overline{E}$. The corresponding polynomials $X_{i,k}X_{j,l}$ are added to the system.
2. For every $v_k v_l \in E$, look for the edges $u_i u_j \in \overline{D}$ and append the corresponding polynomials $X_{i,k}X_{j,l}$ to the system.

To show that the complexity of such transformation is performed in polynomial time, a very rough upper bound for the size of D can be set to $\frac{n(n-1)}{2}$, corresponding to a complete graph. A similar upper bound can be established for $\overline{E}$. The set of polynomials appended in 1 is computed with two nested loops, the outer one traveling over every edge in D, while the inner loop must visit every edge in $\overline{E}$. Then, the number of steps for this operation is bounded by $\frac{n^2(n-1)^2}{4}$. The second set of polynomials gathered from E and $\overline{D}$ can be obtained following analogous arguments. Then, the time complexity of such an operation is $O(n^4)$, which is polynomial on the order of G. Of course, this upper bound is not reached due to the relation between of the sizes of a graph and its complement, but this is enough to argue why the construction takes a polynomial number of steps; thus, the reduction of GI to $\mathcal{MQ}$ can performed efficiently.

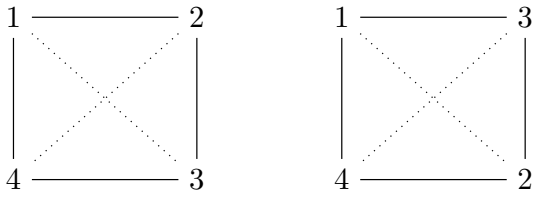
Toy Example

In this section, the construction of public and private key, together with the transformations required during the authentication procedure, are shown providing a small example.

We start by showing the construction of a polynomial set. Let us consider the graph $G = (U, D)$, where $U = \{1,2,3,4\}$ and $D = \{(1,2),(1,4),(2,3),(3,4)\}$. Consider the permutation

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 2 & 4 \end{pmatrix}.$$

After applying σ to the set U, we get the graph $H = \{V, E\}$ defined by $V = U$ and $E = \{(1,3),(1,4),(2,3),(2,4)\}$. The complementary graphs $\overline{G}$ and $\overline{H}$ are determined by the edge sets $\overline{D} = \{(1,3),(2,4)\}$ and $\overline{E} = \{(1,2),(3,4)\}$ respectively. Graphs G, H and their complements (shown by dashed lines) are shown in Figure 4.



(**a**) Graph G (**b**) Graph H

Figure 4. Isomorphic graphs G, H and complements indicated by dashed lines.

We start by building the polynomial set by fulfilling condition (1), which appends 16 polynomials:

$$X_{i,j}^2 - 1 \qquad \text{for } i, j \in \{1, 2, 3, 4\}.$$

As already mentioned, these could replaced by considering solutions over a binary vector space, something useful when the amount of data to be exchanged faces restrictions. Subsequently, condition (2) is addressed by considering the polynomials

$$X_{i,1} + X_{i,2} + X_{i,3} + X_{i,4} - 1 \qquad \text{for } i = 1, 2, 3, 4$$

$$X_{1,j} + X_{2,j} + X_{3,j} + X_{4,j} - 1 \qquad \text{for } j = 1, 2, 3, 4.$$

Finally, the polynomials obtained from condition (3) are added to the polynomial set. To understand the process, let us consider an edge in D; say, (1,2). The edges not contained in H are (1,2) and (3,4), as seen in Figure 4. These edges introduce the polynomials $X_{1,1}X_{2,2}$ and $X_{1,3}X_{2,4}$. The set of polynomials obtain by considering $\{u_i u_j \in D \wedge v_k v_l \notin E\}$ is shown next

$$X_{1,1}X_{2,2},\ X_{1,1}X_{4,2},\ X_{2,1}X_{3,2},\ X_{3,1}X_{4,2}$$
$$X_{1,3}X_{2,4},\ X_{1,3}X_{4,4},\ X_{2,3}X_{3,4},\ X_{3,3}X_{4,4}.$$

Finally, by considering the edges in $\overline{G}$ and H, we get another set of eight polynomials:

$$X_{1,1}X_{3,3},\ X_{1,2}X_{3,3},\ X_{2,1}X_{4,3},\ X_{3,1}X_{4,2}$$
$$X_{1,1}X_{3,4},\ X_{1,2}X_{3,4},\ X_{2,3}X_{3,4},\ X_{3,3}X_{4,4}.$$

A root of these polynomials related to the isomorphism between these graphs can be computed by letting $x_{i,\sigma(i)} = 1$ for $i = 1, 2, 3, 4$ and zero in other case. Explicitly,

$$x_{i,j} = \begin{cases} 1 \text{ if } (i,j) \in \{(1,1),(2,3),(3,2),(4,4)\} \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

The polynomial system created with the polynomials here described together with the solution defined in (5) conform to the public key F_0 and the private key $(F_0, \mathbf{x_0})$.

Proceeding with the iterative procedure between prover and verifier to perform the authentication step, a new polynomial system and its solution is computed using either a new graph isomorphism or directly a random permutation σ on the subindices, as shown in Section 5.2. The construction is similar to what we have done above.

7. Conclusions and Future Work

A novel, alternative zero-knowledge authentication protocol whose security relies in the difficulty of solving $\mathcal{MQ}$ and GI has been proposed. A set of polynomials was built in such a way that a solution is related to an isomorphism between graphs. That way, it is guaranteed that the protocol is at least as secure as the classical ZKP based uniquely in GI. It has also been shown that the implementation is computationally feasible. Also, the transformation applied on the polynomial set depends on a permutation, which makes the computation lightweight. Since most of the information interchanged at every challenge-response round consists of a set of polynomials, which is a bit string in the order of $O(n^4)$, further research on the possibility of reducing the number of polynomials in the system without weakening the proof system is desirable to provide a complete implementation of the authentication protocol. Additionally, it is expected that future research will be done in the direction of providing difficult instances of GI to be employed in the protocol presented in the current work.

Supplementary operations could be considered to improve the presented system, which would consist of using general affine transformation S, T instead of permutations alone, as has been remarked in the authentication protocol presented in Section 5.2. In this case, the systems constructed can be additionally hardened by performing a more general isomorphism form $\Psi(F_0) = S \circ F_1 \circ T$, where S and T are random affine transformations. Observe that the amount of information transferred in each authentication round grows by using two transformations and non-sparse matrices. A more detailed study on the hardness of such instances is needed to decide if these modifications are useful.

Author Contributions: All authors contributed equally to the development and writing of this work. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the partial support of Mexican CONACYT. The first author has a grant from Conacyt's Scholarship Program. The last two authors have been partially supported by Conacyt's National System of Researchers.

Acknowledgments: The support from ABACUS-CINVESTAV (Conacyt, EDOMEX-2011-C01-165873) is gratefully acknowledged as well.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rivest, R.; Shamir, A.; Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
2. National Institute of Standards and Technology. *Digital Signature Standard (DSS)*; Federal Information Processing Standards Publication 186-4: Gaithersburg, MD, USA, July 2013.
3. Shor, P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Comput.* **1997**, *26*, 1484–1509. [CrossRef]
4. Hopwood, D.; Bowe, S.; Hornby, T.; Wilcox, N. *Zcash Protocol Specification*; Technical Report; Zerocoin Electric Coin Company: Denver, CO, USA, 2016.
5. Yu, Y.; Au, M.H.; Ateniese, G.; Huang, X.; Susilo, W.; Dai, Y.; Min, G. Identity-Based Remote Data Integrity Checking With Perfect Data Privacy Preserving for Cloud Storage. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 767–778. [CrossRef]
6. Beydemir, A.; Sogukpinar, I. Lightweight zero knowledge authentication for Internet of things. In Proceedings of the 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 5–8 October 2017, pp. 360–365.
7. Babai, L. Graph Isomorphism in Quasipolynomial Time. In Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing, Cambridge, MA, USA, 18–21 June 2016; ACM: New York, NY, USA, 2016; pp. 684–697.

8. Goldwasser, S.; Micali, S.; Rackoff, C. The Knowledge Complexity of Interactive Proof-systems. *SIAM J. Comput.* **1989**, *18*, 186–208. [CrossRef]
9. Bellare, M.; Micali, S.; Ostrovsky, R. Perfect Zero-knowledge in Constant Rounds. In Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 13–17 May 1990; ACM: New York, NY, USA, pp. 482–493.
10. McKay, B.D.; Piperno, A. Practical graph isomorphism, II. *J. Symb. Comput.* **2014**, *60*, 94–112. [CrossRef]
11. Codenotti, P.; Katebi, H.; Sakallah, K.A.; Markov, I.L. Conflict Analysis and Branching Heuristics in the Search for Graph Automorphisms. In Proceedings of the International Conference on Tools with Artificial Intelligence of the IEEE, Herndon, VA, USA, 4–6 November 2013; pp. 907–914.
12. Junttila, T.; Kaski, P. Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs. In Proceedings of the Meeting on Algorithm Engineering & Experiments, New Orleans, LA, USA, 6 January 2007; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2007; pp. 135–149.
13. Grigoriev, D.; Shpilrain, V. Authentication schemes from actions on graphs, groups, or rings. *Ann. Pure Appl. Log.* **2010**, *162*, 194–200. [CrossRef]
14. Patarin, J. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, 12–16 May 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 33–48.
15. Imai, H.; Matsumoto, T. Algebraic methods for constructing asymmetric cryptosystems. In Proceedings of the 3rd International Conference on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, Grenoble, France, 15–19 July 1985; Springer: Berlin/Heidelberg, Germany, 1985; pp. 108–119.
16. Matsumoto, T.; Imai, H. Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption. In Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Davos, Switzerland, 25–27 May 1988; Springer: Berlin/Heidelberg, Germany, 1988; pp. 419–453.
17. Ding, J.; Gower, J.E.; Schmidt, D.S. *Multivariate Public Key Cryptosystems*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 25, pp. 1–61.
18. Patarin, J. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In Proceedings of the 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, 27–31 August 1995; Springer: Berlin/Heidelberg, Germany, 1995; pp. 248–261.
19. Kipnis, A.; Patarin, J.; Goubin, L. Unbalanced Oil and Vinegar Signature Schemes. In Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 206–222.
20. Ding, J.; Schmidt, D. Rainbow, a New Multivariable Polynomial Signature Scheme. In Proceedings of the Third International Conference on Applied Cryptography and Network Security, New York, NY, USA, 7–10 June 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 164–175.
21. National Institute of Standards and Technology. Candidate Quantum-Resistant Cryptographic Algorithms Publicly Available. Available online: https://www.nist.gov/news-events/news/2017/12/candidate-quantum-resistant-cryptographic-algorithms-publicly-available (accessed on 4 November 2019).
22. Buchberger, B. An Algorithmic Criterion for the Solvability of a System of Algebraic Equations. In *Gröbner Bases and Applications*; Number 251 in Lond Math S; Cambridge University Press: Cambridge, UK, 1998; pp. 535–545.
23. Bernstein, D.J.; Buchmann, J.; Dahmen, E. *Post-Quantum Cryptography*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2009.
24. Belov, A.Y. Linear Recurrence Equations on a Tree. *Math. Notes* **2005**, *78*, 603–609. [CrossRef]
25. Ufnarovskii, V.A. Combinatorial and asymptotic methods in algebra. In *Itogi Nauki i Tekhniki. Sovremennye Problemy Matematiki. Fundamental'nye Napravleniya*; VINITI: Moscow, Russia, 1990; Volume 57, p. 5–177.
26. Faugère, J.C.; Joux, A. Algebraic cryptanalysis of Hidden Field Equations (HFE) Using Gröbner Bases. In Proceedings of the 23rd Annual International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 2003; pp. 44–60.
27. Braeken, A.; Wolf, C.; Preneel, B. A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes. In Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, 14–18 February 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 29–43.

28. Bard, G. *Algebraic Cryptanalysis*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2009.
29. Goldreich, O. *Computational Complexity: A Conceptual Perspective*; Cambridge University Press: Cambridge, UK, 2008.
30. Nemhauser, G.L.; Wolsey, L.A. *Integer and Combinatorial Optimization*; Wiley-Interscience: New York, NY, USA, 1988.